Mechanical design of structures -Optimization of structures under fatigue life criterion

Christian Fourcade

UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES
université PARIS-SACLAY

Mechanical design of structures - Optimization of structures under fatigue life criterion -

Christian Fourcade

abstract

These lectures are devoted to the presentation of a new computational procedure for fatigue analysis of structures. This method, which is based on the theory of hysteresis operators, consists to reduce computation of the damage $\mathscr{D}$ caused by a time varying stress $t \in [0,T] \mapsto \Sigma_e(t)$ to the energy dissipated in the hysteresis loops of the image $\mathscr{H}_\mu(\Sigma_e)$ of Σ_e by an appropriately calibrated Preisach operator $\mathscr{H}_\mu$.

We then see that this formalism allows to reduce the structure optimization problem, which consists to seek design parameters u minimizing the damage in some given parts of a structure, to the minimization of the mapping

$$u \mapsto \mathscr{D}(u) = \int_0^T \left| \mathscr{H}'_\mu(\Sigma_e, t) \right| dt$$

where $\Sigma_e(x_u)$ is a numerical mapping governed by a system of second order differential equations

$$M_u \ddot{x} + W_u \dot{x} + K_u x = F(t)$$

describing the dynamical behavior of the considered structure.

Furthermore, we provide and validate a series of algorithms allowing to solve the optimization problem by a steepest descent method tailored to manage large dynamical problems derived from finite element models. At last, the theoretical results obtained in this course are illustrated with the help of numerous examples, intended for supporting the relevancy of the approach and providing implementation templates in design engineering software.

Keywords: Optimal design, Structure analysis, Beam theory, Fatigue analysis, Hysteresis, Dynamical systems.

INTRODUCTION

Version française.

Le dimensionnement à la fatigue est une question récurrente qui est posée dans les processus de conception des structures soumises à des chargements variables dans le temps. Il consiste par exemple à déterminer les paramètres géométriques tels que “galbes, épaisseurs, répartitions de masses, points d’entrées des efforts, etc.”, qui permettent de garantir une durée de vie satisfaisante à la structure lorsque celle-ci est sollicitée par un chargement répété suffisamment longtemps, qui est une donnée d’entrée du problème d’ingénierie, appelée profil de mission. Pour répondre à cette question, on utilise souvent le retour d’expérience acquis sur des projets antérieurs pour simplifier la spécification technique de durée de vie en la déployant, par exemple, en objectifs de raideurs (locales et globales) ou de fréquences propres. Ceux-ci sont ensuite complétés par des cahiers des charges géométriques tels que rayons de raccordements, épaisseurs minimales de tôles, etc., afin de prendre en compte dans le dimensionnement d’un savoir faire d’ingénierie. L’approche est souvent enrichie par un calcul de durée de vie qui permet d’affiner la détection des zones de faiblesses et de vérifier, sur le cas d’espèce, l’adéquation entre l’objectif initial et les objectifs déployés. Cette méthode de conception, qui semble naturelle et facile à mettre en œuvre du point de vue de la simulation numérique, présente toutefois les inconvénients de conduire à des structures sur-dimensionnées, par exemple en masse, et de ne concerner que des concepts déjà bien connus. Dans un paysage où l’optimisation fait de plus en plus partie intégrante de la panoplie des outils numériques d’aide conception mis à la disposition des ingénieurs pour concevoir, dans des délais de plus en plus restreints, des structures qui sont requises être à la fois plus légères et plus résistantes, il semblait opportun de développer une méthode d’optimisation de structure sur critère de fatigue. Cela permet en effet à l’ingénieur d’exploiter au mieux, par une approche purement simulation, le potentiel d’un espace de conception, fixé par la définition du produit, et donc de limiter le nombre des itérations calculs-essais (parfois tardives) nécessaires à la mise point. Dans ce contexte, l’objet de ce cours est de présenter les principes fondamentaux

d'une méthode d'optimisation de structure sous critère d'endommagement. Dans la mesure où nous avons choisi de disposer cette méthode aux algorithmes de plus forte descente[1], la question principale consiste à définir une procédure de calcul de la dérivée de l'endommagement par rapport aux variables de conception. Plus précisément, en supposant que les matrices de masse M_u, de raideur K_u et d'amortissement W_u de la structure dépendent de façon régulière des paramètres de conception, que nous notons génériquement $u \in \mathbb{R}^m$, l'endommagement de certaines zones de la structure est un nombre $\mathscr{D}(\Sigma_e) \leq 1$ qui est calculé en post-traitant en fatigue le résultat d'un calcul dynamique de structure défini par l'équation d'état

$$M_u \ddot{x} + W_u \dot{x} + K_u x = F(t)$$

où $t \in [0,T] \mapsto F(t)$ est un terme de chargement qui est défini par le profil de mission. Le post-traitement en fatigue aux standards de l'industrie consiste à calculer l'endommagement de certaines zones "à risque" de la structure en trois étapes, qui consistent à

- décomposer le cycle de chargement[2] $\Sigma_e(u)$ en cycles élémentaires par la méthode du "rain-flow counting",
- évaluer à l'aide des courbes de Wöhler du matériau l'endommagement engendré par chaque cycle élémentaire,
- additionner les endommagements élémentaires en appliquant la règle de cumul de fatigue de Palmgren-Miner, pour obtenir l'endommagement total engendré par le cycle de chargement considéré.

Nous verrons que cette procédure (qui est implémentée dans les codes de calcul de fatigue) ne permet pas d'exploiter le calcul des variations pour exprimer la dérivée de l'endommagement par rapport aux variables de conception d'une structure et qu'il nous faut reformuler le calcul de dommage en termes d'opérations fonctionnelles portant sur des signaux définis en temps continus (ici les fonctions a variations bornées) pour l'exprimer comme la variation totale de l'image du cycle de chargement par un opérateur d'hysteresis approprié. Cela pose le cadre formel qui permet

1^o/ d'établir que sous certaines conditions de régularité de la solution de l'équation d'état, le dommage est une fonction dérivable des variables de conception

2^o/ et de fournir, via l'intégration d'une équation ajointe, un moyen de calcul de cette dérivée.

Ce cours est rédigé de la façon suivante, en quatre chapitres:

1^o/ On rappelle dans le premier chapitre les éléments classiques de calcul de dommage et on formalise de façon précise les principes du calcul d'endommagement tel qu'ils sont présentés dans les traités classiques d'analyse de la fatigue.

[1] Qui consistent à explorer l'espace de conception en se dirigeant à l'opposé de la dérivée du critère de dimensionnement par rapport aux variables de conception.

[2] Qui est un signal temporel obtenu à partir d'un invariant des contraintes dans la zone de structure considérée.

2^o/ Sur la base des résultats introduits dans le premier chapitre, on reformule dans le second chapitre le calcul d'endommagement pour l'adapter aux signaux définis en temps continus. On montre en particulier qu'il est possible de calibrer la mesure μ d'un opérateur de Preisach[3] $\mathscr{H}_\mu$ pour exprimer l'endommagement $\mathscr{D}$ par l'intégrale

$$\mathscr{D} = \int_0^T \left| \mathscr{H}'_\mu(\Sigma_e, t) \right| dt$$

où Σ_e est une fonction numérique qui dépend des variables d'état x_u et $\dot{x}_u$.

3^o/ Cela nous permet, dans le quatrième chapitre, d'expliciter une équation adjointe pour calculer la dérivée de l'endommagement, considéré comme une fonction des variables de conception u. Ce chapitre est précédé d'un "kit de survie en optimisation" qui a pour vocation à rappeler au non spécialiste quelques principes des algorithmes d'optimisation disponibles aujourd'hui sur le marché. Nous verrons que ces algorithmes sont relativement complexes mais que, faisant confiance à leurs robustesses numérique, l'utilisateur "standard" a pour tâche principale de leur fournir une procédure de calcul du critère et sa dérivée ; la vitesse de convergence étant bien entendu proportionnelle à la précision du calcul de dérivée, qu'il convient donc de vérifier avec la plus grande rigueur.

4^o/ Comme l'équation adjointe obtenue dans le quatrième chapitre est une équation différentielle posée en temps rétrograde et excitée par les variables d'états x_u et $\dot{x}_u$ son intégration numérique est, dans le cas général, une opération très coûteuse en mémoire. Nous nous proposons donc, dans le troisième chapitre, d'exploiter la caractère linéaire à la fois de l'équation d'état et de l'équation adjointe pour définir un processus de réduction de modèle qui permet de réduire de façon significative le volume des données nécessaire à l'intégration simultanée de ces équations.

Ce cours étant situé à la frontière entre la mécanique, les mathématiques et l'informatique, j'ai souhaité formuler les principaux résultats "théoriques" en termes d'algorithmes, eux mêmes traduits en programmes "MATLAB" pour permettre à la lectrice ou au lecteur de vérifier par la pratique les résultats énoncés dans les Propositions et Théorèmes qui synthétisent les principales étapes du cours. Dans la mesure nous n'aborderons ni les méthodes de paramétrages géométriques d'une structure ni le calcul des dérivées des matrices élémentaires par rapport aux variables de conception, les résultats présentés ici doivent être considérés comme les principes fondamentaux d'un programme d'optimisation de structure sous critère de fatigue. Nous verrons dans une prochaine étape comment implémenter les algorithmes proposés ici dans les logiciels de calcul de structure. Nous verrons que cela nécessite une reconception de ces logiciels afin de les prédisposer aux calculs des gradients et de les interfacer avec les logiciels de CAO[4] pour obtenir un outil d'aide à la décision pleinement efficace.

[3] Il s'agit d'un opérateur à mémoire interne, qui permet d'identifier et de compter les cycles de chargement élémentaires présents dans le signal Σ_e.

[4] Les méthodes iso-géométriques proposées par P. de NAZELLE [**31**] et S. JULISSON [**18**] fournissent à cet égard quelques éléments de réponse à cette question dans le cadre de l'optimisation de forme des structures surfaciques.

English version.

FATIGUE dimensioning is a recurrent question posed in the design processes of structures submitted to time variable loadings. It consists, for instance, to determine the geometrical parameters such as "curvatures, thicknesses, mass distributions, entry points of forces, etc.", allowing to guarantee a given service life for the structure when this one is submitted to a long repeated loading (which is an input data of the engineering problem, reffered to as mission profile). To answer this question, experience return aquired on previous projects is often used to deploy the technical specification of service life into simpler requirements on stiffness (local and global) or eigenfrequencies. These simplified targets are furthermore supplemented by geometrical specifications such as connecting radii, minimum thicknesses of sheets, etc., in order to account for empiric engineering know-how. Nowaday the approach is enriched by a lifetime or damage computation allowing to refine the detection of the weaknesses zones and to verify on the particular case adequacy between initial deployed objectives. This designing method, which seems quite natural and easy to implement from the point of view of numerical simulations has, however,the drawbacks of leading to oversized structures, for example in weight, and only concerns concepts already well known. In a CAE landscape where optimization is becoming more and more an integral part of the panoply of numerical tools helping the engineers to design, in more and more limited time, structures which are required both lighter and resistant, it seemed timely to develop a method of structural optimization under fatigue damage criteria. This allows engineers to make the best use, through a purely simulation approach, of the potential of a design space (induced by the product definition) and thus to limit the number of "tests/computations" iterations, often belately performed in the project planning. In this context, purpose of this course is to present the basic principles of a structural optimization method under fatigue damage criterion. Since we have chosen to tailor the method for gradient based optimization algorithms[5], the main question is to define a procedure to compute the derivative of the damage with respect to the design variables. More precisely, assuming that the mass M_u, the damping W_u, and the stiffness K_u matrices of the structure depend smoothly on the design parameters, generically denoted by $u \in \mathbb{R}^m$, the fatigue damage in some (inserting) areas of the structure is a number $\mathscr{D}(\Sigma_e) \leq 1$ which is calculated by post-processing in fatigue the results of a dynamical simulation defined by the state equation

$$M_u \ddot{x} + W_u \dot{x} + K_u x = F(t)$$

where $t \in [0,T] \mapsto F(t)$ is a loading term which is defined by the mission profile. Industry-standard fatigue post-processing consists to calculate fatigue damage in some "risky zones " of structure within three steps , which consist to

- split up the loading cycle[6] $\Sigma_e(u)$ into elementary cycles by “rain-flow counting” method,

[5] Which consists to explore the design space by moving in the opposite direction to the derivative of criterion with respect to the design variables.

[6] Which is a temporal signal obtained from an invariant of the stress tensor computed in the considerd area of the structure.

- evaluate with the help of Wöhler's curves of the material the damage caused by each elementary cycle,
- add the elementary damages by applying the Palmgren-Miner fatigue accumulation rule, to obtain the total damage.

We will see that this procedure (which is implemented in the fatigue software) doesn't allow to use the "calculus of variations" to compute the derivative of the damage with respect to the design variables and that we must reformulate the damage computation process in terms of functional operations applied to time continuous signals (here the functions with bounded variations) to write it down as the total variation of the image of the loading cycle by an appropriate hysteresis operator. This set up the formal framework which permits to

1^o/ establish that, under some regularity hypothesis about the solution of the state equation, the damage is a differentiable function of the design variables,

2^o/ and to provide, via the integration of an adjoint equation, the means to calculate this derivative.

Theses notes are written as follows, in four chapters:

1^o/ In the first chapter we remind and formalise precisely the elements of damage calculus such as they are presented in the classical treatises of fatigue analysis.

2^o/ On the basis of the results introduced in the first chapter, we reformulate, in the second chapter, the damage computation process to adapt it to time continuous signals. We particularly show that we can calibrate the measure μ of a Preisach operator[7] $\mathscr{H}_\mu$ to write down the damage $\mathscr{D}$ as the integral

$$\mathscr{D} = \int_0^T \left| \mathscr{H}'_\mu(\Sigma_e, t) \right| dt$$

where Σ_e is a numerical mapping depending on the sate variables x_u and $\dot{x}_u$.

3^o/ This allows us, in the fourth chapter, to write down an adjoint equation to calculate the derivative of the damage, considered as a function of the design variables u. This chapter is preceded by an "optimization survival kit" aiming to remind the non-specialist some principles of optimization algorithms available today on the market. We will see that these algorithms are relatively complex but that, being confident on their numerical robustness, the main task of the lambda user is to provide them with a procedure for the computation of the criterion and its derivative; the speed of convergence being of course proportional to the precision of the calculated derivative, which must therefore be verified with the greatest rigor.

4^o/ As the adjoint equation obtained in the fourth chapter is a differential equation posed backward in time and excited by the state variables x_u and $\dot{x}_u$ its numerical integration is usually a very expensive operation in terms of memory storage. We therefore propose, in the third chapter, to exploit the linear character of both the state and the adjoint equation to define a model reduction procedure aming to

[7] It is an internal memory operator, which identifies and counts the elementary loading cycles present in the signal Σ_e.

significantly reduce the amount of data necessary for the simultaneous integration of these equations.

This course beeing located on the border between mechanics, mathematics and computer science, I have reformulated the theoretical key findings in terms of algorithms translated in turn into "MATLAB" programs in order to allow the reader to verify by practice the results stated in the Propositions and Theorems which summarize the main steps of the course. As we don't discuss the geometrical parameterization methods of a structure or the ways to compute the derivatives of the elementary matrices with respect to the design variables, the results presented here must be considered as the fundamental principles of a structural optimization program under fatigue live criterion. We will see in a next step how to implement the proposed algorithms in structural analysis software. We will particularly point out that to obtain an effective decision-making tool, we will have to predispose the CAE software to gradient calculations and to interface them with the CAD software[8].

[8] Iso-geometric methods proposed by de NAZELLE [**31**] and JULISSON [**18**] provide some parts of answer to this question in the framework of shape optimization of the surface structures.

CONTENTS

CHAPTER 1

BASIC PRINCIPLES OF FATIGUE ANALYSIS

THE basic principles of fatigue analysis of a structure are made up of the four following building blocks:

E1/ *Excitation forces*, measured or computed from mission-profiles, are given under the form of temporal signals –two typical examples are depicted in the figures (Fig. 1.2 and (Fig. 1.3)– and are played long enough, roughly speaking 3 or 4 weeks, on a test bench –see figure (Fig. 1.1)– to achieve 10^7 loading cycles or to get failure.

E2/ *A numerical model of a flexible structure*, for instance a fully equipped car-body, whose mechanical behavior is "more or less linear"; there are never large displacements but non-linear models can be needed to reproduce local plastification due to the loading levels: compare to this purpose the forces introduced in the car for "3 bumps crossing" depicted in figure (Fig. 1.2), to those which are plotted in figure (Fig. 1.3) for the "Cobbled runway" mission profile.

E3/ *Experimental data* are given under the form of fatigue tests results, which are carried out on specimens submitted to uni-axial loading: tension/compression, bending, twisting, etc. such as the Wöhler or $S-N$ curves depicted in figure (Fig. 1.4) to define the *alternating stress* $\sigma_a(N)$ *leading to failure after* N *loading cycles*, at a given stress average σ_m.

E4/ The damage caused on a sample, pre-loaded at a given stress average σ_m, by n alternating cycles of amplitudes σ_a is assumed to be defined as the following positive number lower than 1:

$$\mathscr{D}(\sigma_a, \sigma_m, n) = \frac{n}{N_r(\sigma_a, \sigma_m)}$$

Fig. 1.1. **Example of test bench for fatigue analysis**. The mission profiles are measured on an endurance runway (on the left) and reproduced on a test bench until failure. The most damaging mission profiles are plotted in figures (Fig. 1.2) and (Fig. 1.3)

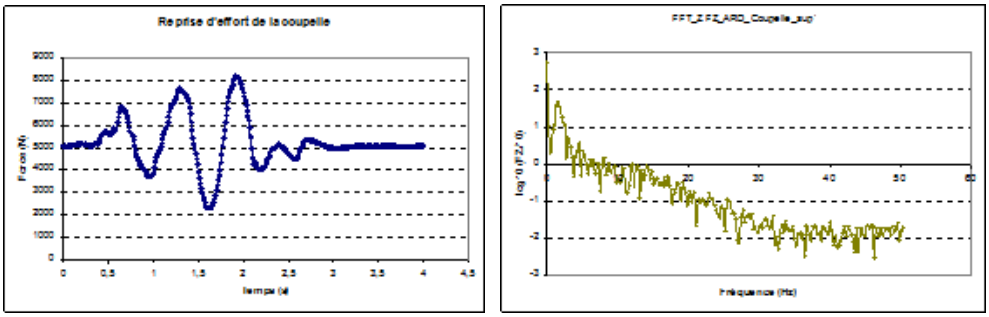


Fig. 1.2. **Example of mission profile "3 bumps crossing"**. The spectrum of the signal (on right) doesn't contain high frequencies and the response of the structure can be assumed to be quasi-static. Loading (on left) can, however, cause permanent plastic deformations at the connecting points between body and axles to justify non-linear quasi-static simulations.

where $N_r(\sigma_a,\sigma_m)$ is the number of cycles to failure of the pre-loaded sample submitted to the alternating stress σ_a — it the reciprocal function of the previously introduced Wöhler mapping $N \mapsto \sigma_a(N)$, obtained a average stress σ_m.

We adopt the convention $\mathscr{D}(\sigma_a,\sigma_m,n) = 0$ if σ_a doesn't reach the asymptotic value of the $S-N$ curve.

The damage caused by non-symmetric loading containing p sections of alternating amplitude σ_{a_i} at average σ_{m_i} (for $1 \le i \le p$) *is a number, lower than* 1, defined as follows with the help of *the Palmgren-Miner's accumulation law*

$$\mathscr{D}(\sigma) = \sum_i \mathscr{D}(\sigma_{a_i},\sigma_{m_i},n_i) \left(= \sum_{i=1}^{p} \frac{n_i}{N_r(\sigma_{a_i},\sigma_{m_i})} \right) \tag{1.1}$$

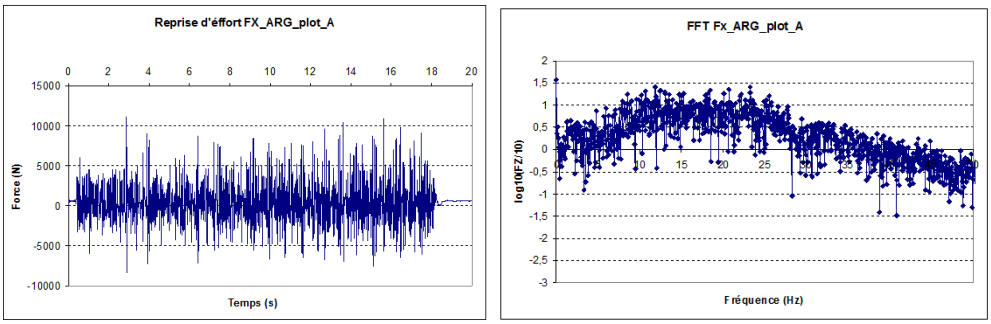


Fig. 1.3. **Example of mission profile "Cobbled runway"**. While remaining at a fairly low level, the excitation spectrum (on the right) can be energetic at high frequencies (frequency range $10-40Hz$) which may justify of linear dynamic simulations to reproduce the over-stresses resulting from excitability of the body natural modes such as global torsion or bending modes.

where, see figure (Fig. 1.5), n_i is the number of alternating cycles of amplitudes σ_{a_i} at averages σ_{m_i} and of number of cycles to failure $N_r(\sigma_{a_i},\sigma_{m_i})$ occurring in the loading signal[1].

We say that a loading cycle leads to failure when the so calculated damage $\mathscr{D}$ is greater than 1.

The rest of this Chapter, organized as follows

Contents

aims at formalizing and justifying (in Definition 1.1 page 25) the damage computation procedure introduced in the steps E_3)-E_4) and summarized on the diagram in figure (Fig. 1.6). *We would point out that Definition 1.1 is the starting point of the developments of Chapter 2 aiming at establishing the mathematical properties of the damage calculation process, which are used in the Chapter 4 for structural optimization purposes.*

[1]The difficulty is to identify in an arbitrary signal the alternating cycles and their averages; the rain-flow method, introduced page 20, is a way to do the task.

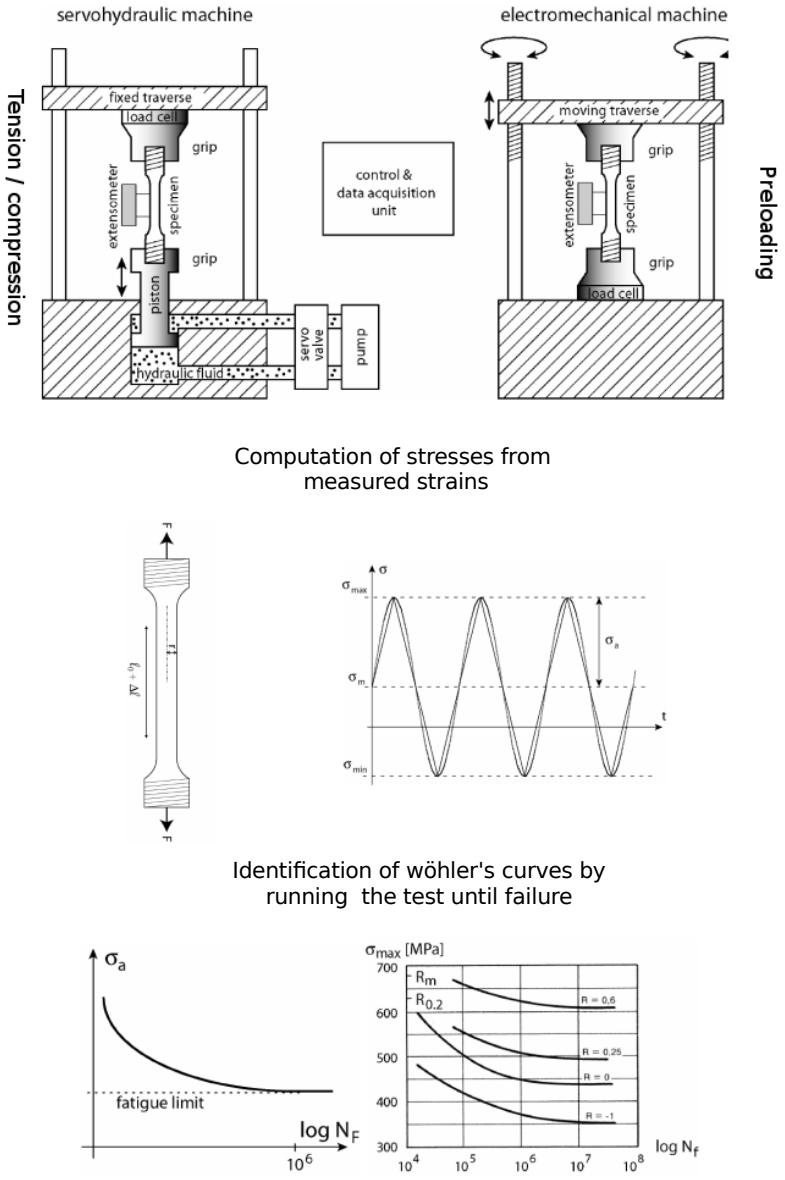


Fig. 1.4. **Experimental devices to identify Wöhler abacuses.** A Wöhler mapping is a family of experimental curves allowing to deduce, from a given alternating load, the number of loading cycles a sample can support until failure. The sample is pre-loaded in tension or compression in order to achieve a given mean stress σ_m, it is then submitted to a sinusoidal stress of half-amplitude σ_a until failure. The obtained number N_r of cycles defines a point on the so called Wöhler's curve (plotted in semi-ln scale) on left. By varying σ_m in the previous tests, we obtain an abacus (on the right) allowing to define the number $N_r(\sigma_a, \sigma_m)$ of alternating cycles the sample can support before failure. *This relationship is assumed to depend only on the constitutive material.*

Fig.a

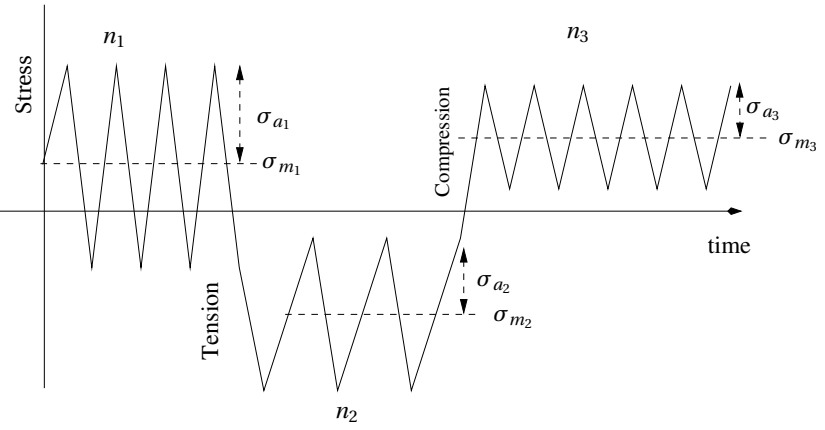


Fig.b

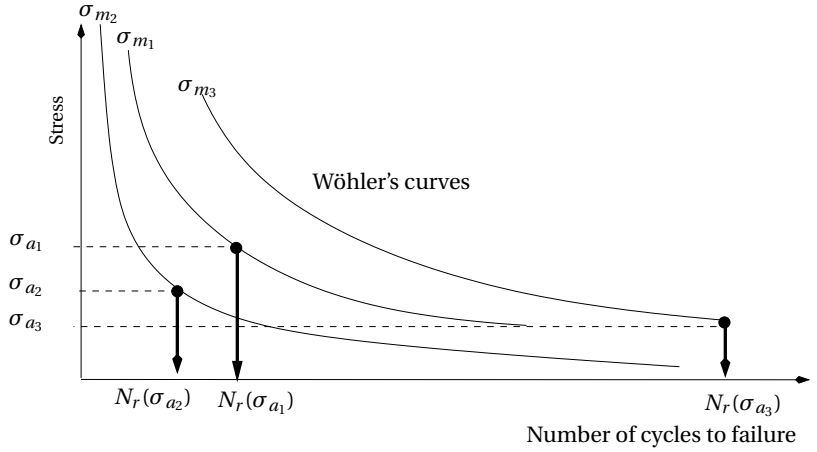


Fig. 1.5. **Identification of the number of cycles to failure via Wöhler's curves**. After having identified in the signal in figure (Fig.a) the alternating cycles and their averages, the damage defined by the formula (1.1) is the sum $\mathscr{D} = \sum_{i=1}^{3} \frac{n_i}{N_r(\sigma_{a_i},\sigma_{m_i})}$: the signal plotted on this figure is made up of n_i alternating cycles σ_{a_i} at average σ_{m_i}. Note that the order of appearance of the loading sequences has no impact on the total damage.

1.1. Basic principles

This Section is subdivided into five sub-sections, which are intending for

1^o/ introducing the approach of fatigue analysis based on the Wöhler's mappings to compute the number of cycles to failure of a structure according to the applied solicitations. *We particularly point out the limitations of this approach to predict lifetime of " weakly loaded " structures* where, see figure (Fig. 1.7), *the asymptote of a Wöhler's curve is quite difficult to identify*, as well analytically as experimentally;

2^o/ detailing the damage computation method for complex loadings;

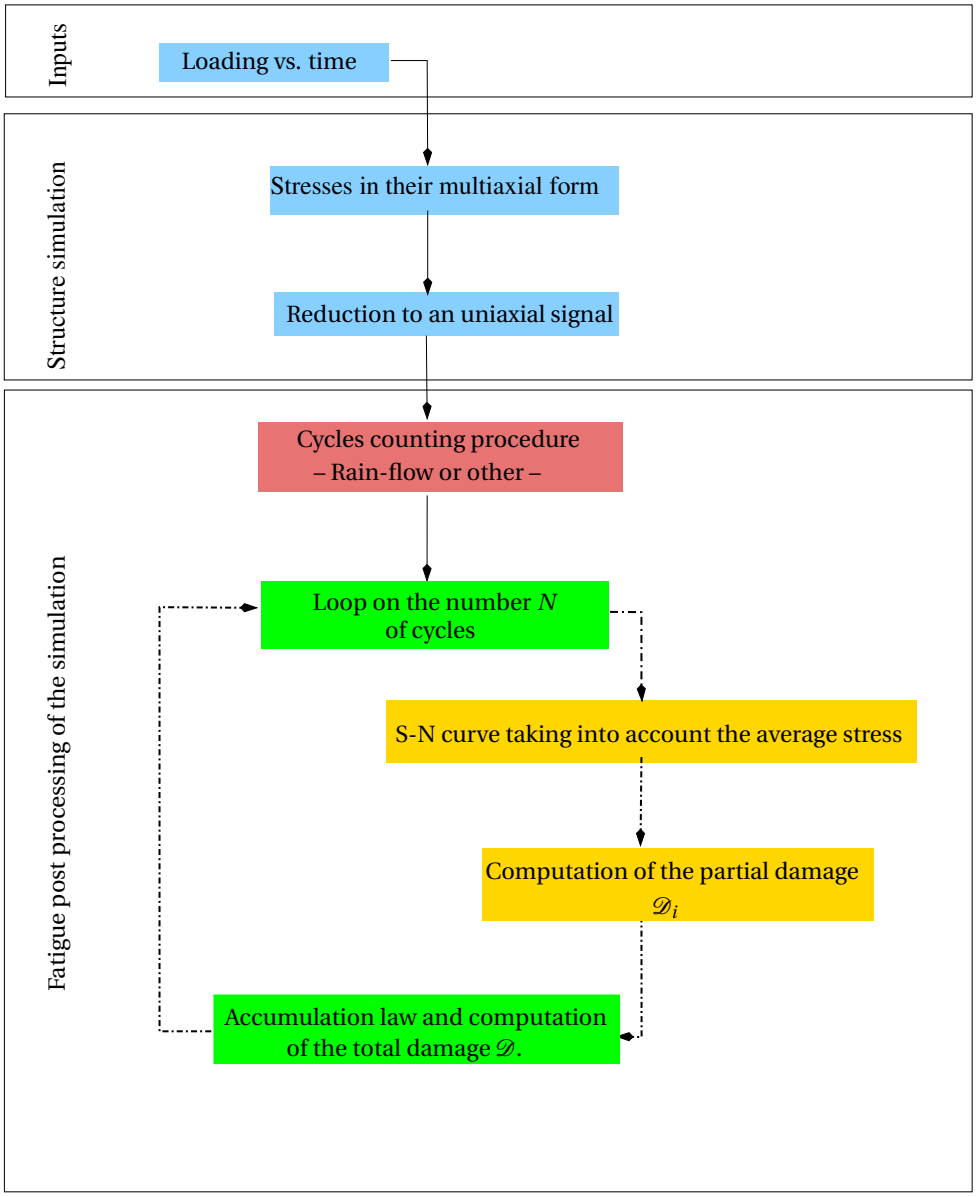


Fig. 1.6. **Fatigue analysis as post-processing of a structure simulation.** It is carried out within three steps:

F_1) the first (in red) aims to encode the signal in elementary cycles;
F_2) the next step (in yellow) consists to compute the impact of each elementary cycle on the structure's lifetime;
F_3) to deduce, in the "green step" the value of the damage $\mathscr{D}$ with the help of a fatigue accumulation law.

3^o/ and at last, formalizing the "damage computation procedure" when the cycle counting is performed by the rain-flow counting algorithm and the damage accumulation is calculated with the help of the Palmgren-Miner's rule.

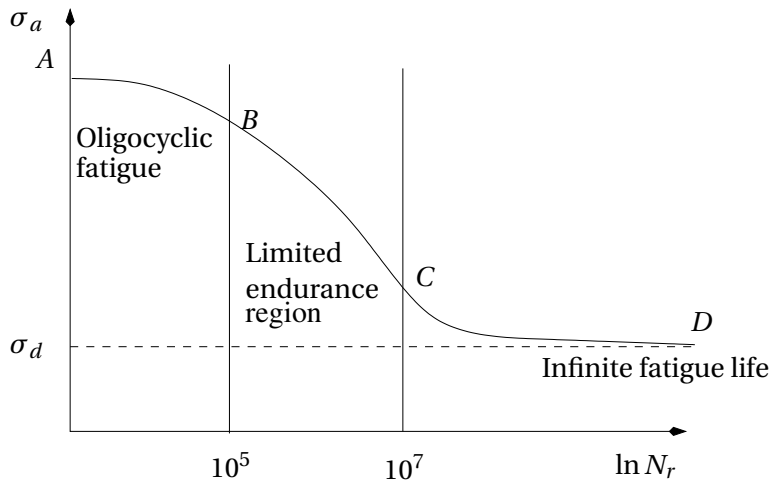


Fig. 1.7. **Generic form of a Wöhler's curve**. We are mainly interested by the part BCD of this curve, in other words, we will assume that the applied loading is low enough to ensure lifetime greater than 10^5 cycles. In this case, failure is not accompanied with overall plastic deformation. However, permanent plastic deformation due to the first loading cycles can occur.

Wöhler's curves. Several analytical representations of Wöhler's curves are proposed in the literature, they reproduce more or less accurately parts BC or CD of a generic Wöhler's curve defined in figure (Fig. 1.7), where σ_d, referred to as fatigue limit, is the stress level below which specimen's lifetime remains unmodified[2].

Among all the relationships between number of cycles to failure N_r and alternating stress σ_a, let's mention the following which are used daily in engineering:

- *Wöhler's formula (linear in* $\ln N_r$*)*

$$\sigma_a = a_w - b_w \ln(N_r) \quad \text{or} \quad \frac{1}{N_r} = \exp\left(\frac{\sigma_a - a_w}{b_w}\right)$$

  where a_w and b_w are two positive constants. As

$$\lim_{N_r \to \infty} \sigma_a(N_r) = -\infty$$

  this formula only approaches the part BC of the generic Wöhler's curve;
- *Basquin's formula (linear in* ln *scales)*

$$\ln(\sigma_a) = a_b - b_b \ln(N_r) \quad \text{or} \quad \sigma_a^{\frac{1}{b_b}} = \frac{C_b}{N_r}$$

  In this case $\lim_{N_r \to \infty} \sigma_a(N_r) = 0$, but this formula is a straight line in the logarithmic axes and not in the semi-logarithmic scales.

[2]For material such as aluminum the fatigue limit is $\sigma_d = 0$; this means that any loading, even at very low level, is damaging.

- *Stromeyer's formula (Basquin's formula shifted by σ_d)*

$$\ln(\sigma_a - \sigma_d) = a_s - b_s \ln(N_r)$$
$$(\sigma_a - \sigma_d)^{\frac{1}{b_s}} = \frac{C_s}{N_r} \text{ extended by 0 for } \sigma_a < \sigma_d \tag{1.2}$$

  In this case we have $\lim_{N_r \to \infty} \sigma_a(N_r) = \sigma_d$ and this formula can be used to interpolate sections CD of the Wöhler's curves;
- *Bastenaire's formula* is a four parameters interpolation formula, defined as follows:

$$N_r + B = \frac{A e^{-C(\sigma_a - \sigma_d)}}{\sigma_a - \sigma_d} \tag{1.3}$$

  where A, B and C are experimental constants[3]. This curve has a point of inflection and the straight line $\sigma_a = \sigma_d$ is its asymptote, in the semi-logarithmic scale; it allows thus, see figure (Fig. 1.8), to interpolate the part BD of a generic Wöhler's curve.

These parametric curves can be identified with a few number of tests, they *are thus used to minimize the building-cost of the Wöhler's curves.* The reader interested by this kind of representation of the Wöhler's curve may see SURESH [**34**], but a lot of other references are available.

REMARK 1.1 (Effect of dispersions) *There may be a significant dispersion in obtaining the Wöhler's curve of a given material*, especially for low loading cycles or, in other words, for large lifetime tests. For a given stress level, the ratio between the maximal and the minimal number of cycles to failure can exceed 10; this dispersion can result from heterogeneity, surface defects associated with machining or stamping or metallurgical factors etc. So, see figure (Fig. 1.9), *we associate a Wöhler's curve to a probability level of failure*, which is usually defined at probability 50% of failure.

Use of number of cycles to failure given by a Wöhler's curve in a fatigue simulation only indicates that there is as much chance of getting failure as not.

REMARK 1.2 As the coefficients b_s of the Stromeyer or Basquin's formulas define the slopes of Wöhler's curves in logarithmic scales, these formulas allow to piecewise interpolate an experimental Wöhler's curve: Basquin's formula permits for instance to interpolate its steep-sloped parts while Strohmeyer's formula is intended for interpolating the asymptote. However the Wöhler's curve in blue in figure (Fig. 1.10) can be identified with a reasonable accuracy level by a Strohmeyer's formula with the following numerical values

$$b_s = 0.42 \quad C_s = 2.9E+09 \text{ and } \sigma_d = 220\,MPa \tag{1.4}$$

[3]Note that, denoting σ_0 the solution of the equation

$$\sigma_0 = \frac{A}{B} e^{-C\sigma_0}$$

the formula (1.3) makes sense only for $\sigma_a - \sigma_d \geq \sigma_0$.

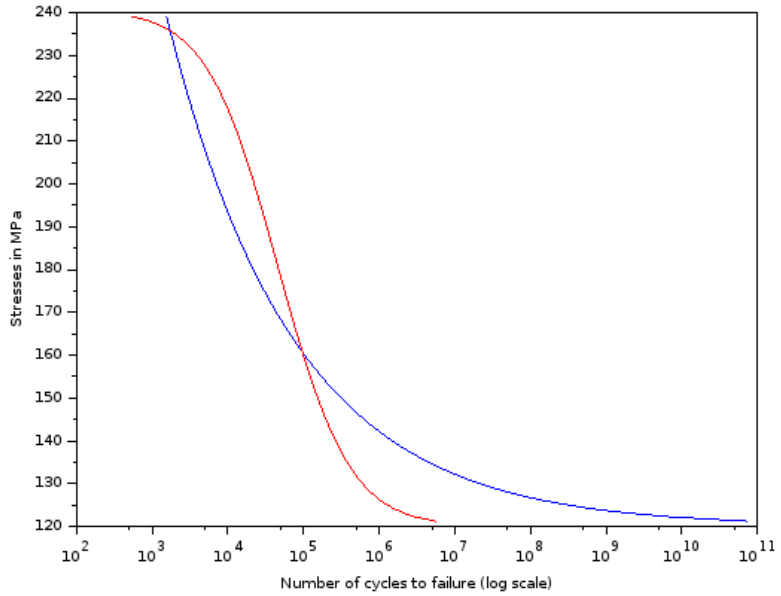


Fig. 1.8. **Comparison between Bastenaire (in red) and Stromeyer (in blue) parametrizations of a Wöhler's curve**. In both cases we suppose that $\sigma_d = 120MPa$ and $N_r = 1.E+05$ for $\sigma_a = 160MPa$. The identification leads the following numerical values: $C_s = 1.5E+11$, $b_s = 0.26$ and $A = 6.8E+06$, $B = 1.9E+04$, $C = 0.009$. This example shows that Bastenaire interpolation has a point of inflection and converges faster to the fatigue limit than the Strohmeyer's formula, this is due to the fact that the identification requires an additional point: in this example, we have introduced the additional point $N_r = 1.5E+02$ for $\sigma_a = 240MPa$.

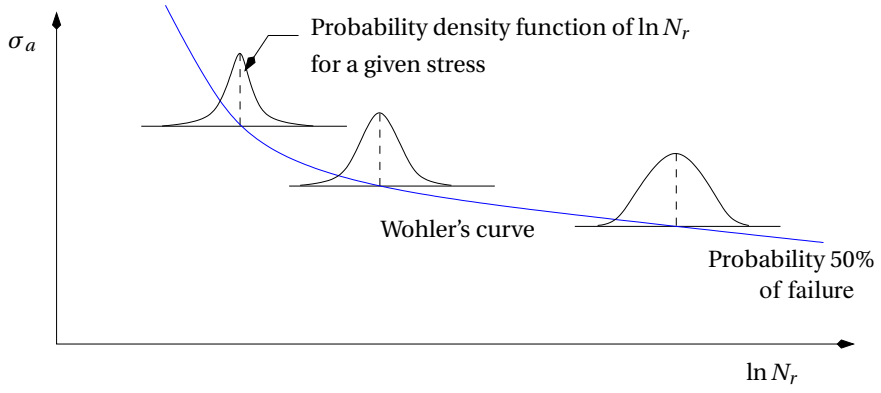


Fig. 1.9. **Probabilistic version of a Wöhler's curve**. It is carried out on several samples of the test specimen. To define a probabilistic version of Wöhler's curves we have to identify the equiprobability curves which associate to each number of cycles a probability of failure p, for which the Wöhler's curve is the middle curve 50%; it is usually assumed that the distribution of the logarithm $\ln N_r$ of the number of cycles to failure satisfies a normal distribution for a given stress σ_a.

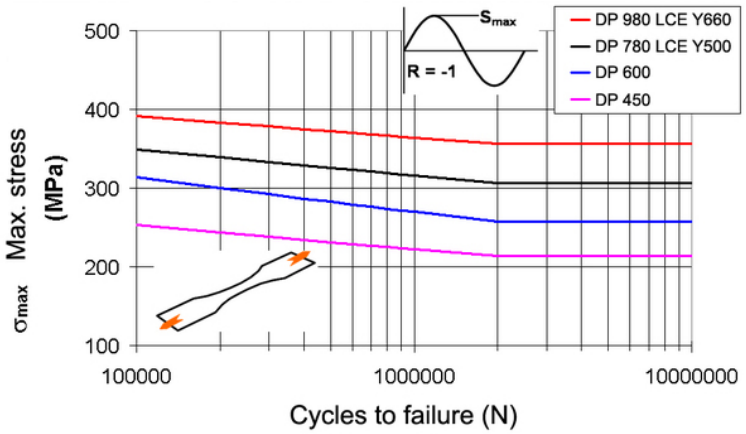


Fig. 1.10. **Example of tension-compression Wöhler curves (Arcelor-Mittal).** These curves are measured at mean stress $\sigma_m = 0$, for three different materials. This figure shows poorness of the tests performed to characterize the material fatigue behavior: *We will see in the following that lifetime simulations carried out on a complex structure with this kind testing results don't lead to relevant results.*

Average effect. Usually Wöhler's curves are obtained from fatigue tests carried out on samples which are submitted to symmetric alternating loads, but when these tests are performed at non-zero stress average σ_m the specimen's lifetime is significantly modified, especially when σ_m is large compared with σ_a: *a tensile mean stress decreases the lifetime, while mean compressive stress increases it.* To avoid carrying out fatigue tests at non-zero mean stresses, we define a corrective formula of specimen's lifetime according to the applied mean stress. Basically, the method consists to *estimate a symmetric alternating stress σ'_a which generates the same number of cycles to failure as the one which would be caused by an alternating stress σ_a at average $\sigma_m \neq 0$.* This "equivalent" stress σ'_a is usually assumed to be of the form

$$\sigma'_a = \frac{\sigma_a}{f(\sigma_m)}$$

where f is a positive numerical mapping such that

$$\lim_{\sigma_m \to 0} f(\sigma_m) = 1 \quad \text{and} \quad \lim_{\sigma_m \to \Sigma_0} f(\sigma_m) = 0$$

Note that the first condition is obvious, while the second one means that there is a tensile mean stress Σ_0 which causes immediate failure.

The following corrective functions are commonly used:

- *Goodman's formula*:

(1.5) $$f_1(\sigma_m) = 1 - \frac{\sigma_m}{R_m}$$

  where R_m is the tensile strength limit.
- *Soderberg*:

(1.6) $$f_2(\sigma_m) = 1 - \frac{\sigma_m}{R_e}$$

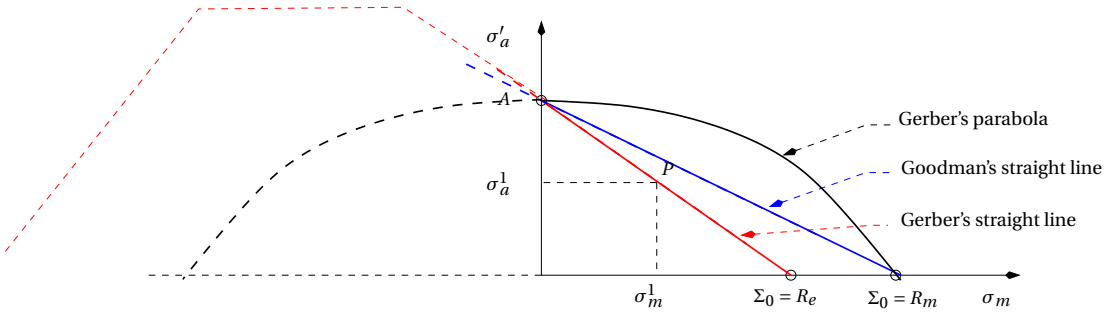


Fig. 1.11. **Haigh's diagram to account for mean stress effect on fatigue lifetime**. If $\sigma_m > 0$ (resp. $\sigma_m < 0$), it is a tensile stress (resp. a compressive stress). Let a number N be given and σ'_a be the alternating stress such that $N = N_r(\sigma'_a, 0)$ obtained from the Wöhler's curve at zero mean stress. If Σ_0 is a given stress which causes "immediate failure", then the couples $P = (\sigma^1_a, \sigma^1_m)$ are assumed to generate the same number (namely N) of cycles to failure for each point P in one of the curves plotted in the above diagram.

where R_e is the elastic limit.

- *Gerber*:
$$f_3(\sigma_m) = 1 - \left(\frac{\sigma_m}{R_m}\right)^2$$
Note that Gerber's formula doesn't reproduce lifetime increasing for negative mean stress σ_m.

and are build with the help of the Haig's diagrams defined in figure (Fig. 1.11).

Damage and fatigue accumulation rule. We want define a relationship between the "lifetime fraction" of a specimen and the amplitude of the alternating load applied on it. To this end, we *introduce a variable* $\mathscr{D}$*, ranging between* 0 *and* 1*, referred to as damage* and defined as follows:

i) the damage generated *at the* n^{th} *alternating cycle of amplitude* σ_a is
$$\mathscr{D}(\sigma_a) = \frac{n}{N(\sigma_a)} \quad \text{(this number is lower than 1)}$$
where is $N(\sigma_a)$ is the number of cycles to failure associated with σ_a;

ii) Palmgren-Miner's rule assumes that the *damage caused by different cycles of alternating loads is accumulated in an additive way*: this means that the damage $\mathscr{D}(\sigma)$ caused by an arbitrary loading σ, which contains the alternating levels $(\sigma_{a_i})_{i=1}^p$ is defined as
$$\mathscr{D}(\sigma) = \sum_{i=1}^{p} \frac{n_i}{N(\sigma_{a_i})}$$
where
- n_i is the number of alternating cycles of magnitude σ_{a_i} occuring in the signal σ;
- and $N(\sigma_{a_i})$ is the number of cycles to failure corresponding to σ_{a_i};

iii) *we say that the loading cycle σ leads to failure when the so calculated damage $\mathscr{D}(\sigma)$ reaches* 1.

REMARK 1.3 This law reflects the fact that stresses lower than the fatigue limit are not damaging for the sample (we set $\frac{1}{N(\sigma_{a_i})} = 0$ for $\sigma_{a_i} \leq \sigma_d$) and that damage estimation depends only on the wöhler's curves for the loading conditions.

Let's conclude this sub-section by the following Remark, which points out a more "physical" approach of fatigue damage analysis; its implementation in the simulation loops requires however more advanced material characterizations and leads to solve non-linear mechanical problems.

REMARK 1.4 (*Approach based on the physics of materials*) CHABOCHE and LEMAITRE [**22**] consider *the damage $\mathscr{D}$ as a state variable describing the evolution of micro-defects*: it is zero in the initial state of a virgin material and reaches $\mathscr{D}_c \approx 1$ at failure. Introducing the concept of effective stress $\tilde{\sigma} = \frac{\sigma}{1-\mathscr{D}}$ as the stress which must be applied to a virgin the material to achieve the same strain ε as in the damaged one[4], they propose a law of damage accumulation, defined according to σ_a and σ_m by the differential equation[5]

$$d\mathscr{D} = \left(1-(1-\mathscr{D})^{\beta+1}\right)^{\alpha} \left[\frac{\sigma_a}{M_0(1-b\sigma_m)(1-\mathscr{D})}\right]^{\beta} dn$$

where β, M_0 and b are material constants and α is a parameter which depends on the loading. In this context, evolution of damage depends not only on the stress applied to the specimen, but also on its damaged state; as such, this law reflects the loading history. This damaging law, which is perfectly justified from the point of view of fracture mechanics and thermodynamics, is however more complicated to set up than the Palmgren-Miner's law because, by relaxing the material according to its damage, it is coupled with the structure equations and leads to a nonlinear system of equations. TIKRI et al. [**36**] use this law to identify the parameters of a Bastenaire's curve.

Cycles counting. In order to apply the Palmgren-Miner's rule to compute the total damage caused by an arbitrary loading, it remains to define a counting method of the alternating cycles, of magnitude σ_a at average σ_m, occurring in the loading signal. In other words : *we have to discretize the loading sequence into elementary cycles, evaluate the damage caused by each of these cycles and at last, with the help of an accumulation law, add the elementary damages to compute the total damage $\mathscr{D}$.*

Several methods have been developed to identify and count the elementary cycles, let's mention for instance the followings, which are compared in LALANNE [**21**] or ROSHANFAR [**33**]:

1) Peak count method

[4] In case of elastic material, this is equivalent to introduce the effective Young's modulus $\tilde{E}$ which is intended for describing the elastic behavior of a structure having reached the damage level $\mathscr{D}$; it is given by $\tilde{E} = E(1-\mathscr{D})$, where E is the Young modulus of the virgin material.

[5] Where n which is the number of cycles, plays role of time.

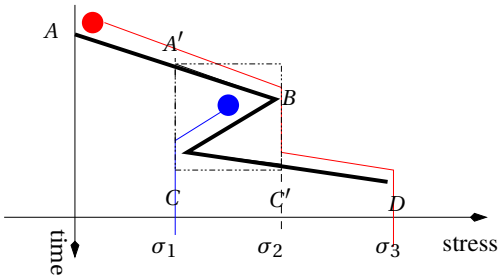


Fig. 1.12. **Basic principle of the Rain-flow counting algorithm.** The principle consists to follow the path of a red drop which, due to its weight, runs along the path $\overset{\frown}{AB}$, falls from the point B to the point C' (of abscissa σ_2) and follows the path $\overset{\frown}{C'D}$, where it ends its journey. Starting from the point B, a blue drop follows the path $\overset{\frown}{BC}$ and ends its journey at the point C of abscissa σ_1. This method allows to identify the oscillation AD, the local extrema B and C in the signal and thus to identify the alternating cycle $A'BCC'$ of magnitude $\frac{\sigma_2-\sigma_1}{2}$ at average $\frac{\sigma_2+\sigma_1}{2}$.

2) Level restricted peak count method
3) Mean crossing peak count method
4) Range pair count method
5) Level crossing method
6) Peak valley pair
7) Rain-flow method

It is commonly accepted that among all the counting methods enumerated above, *only the rain-flow counting method permits to identify and count both the alternating and the mean stresses.*

1.2. Formalization of the damage calculus

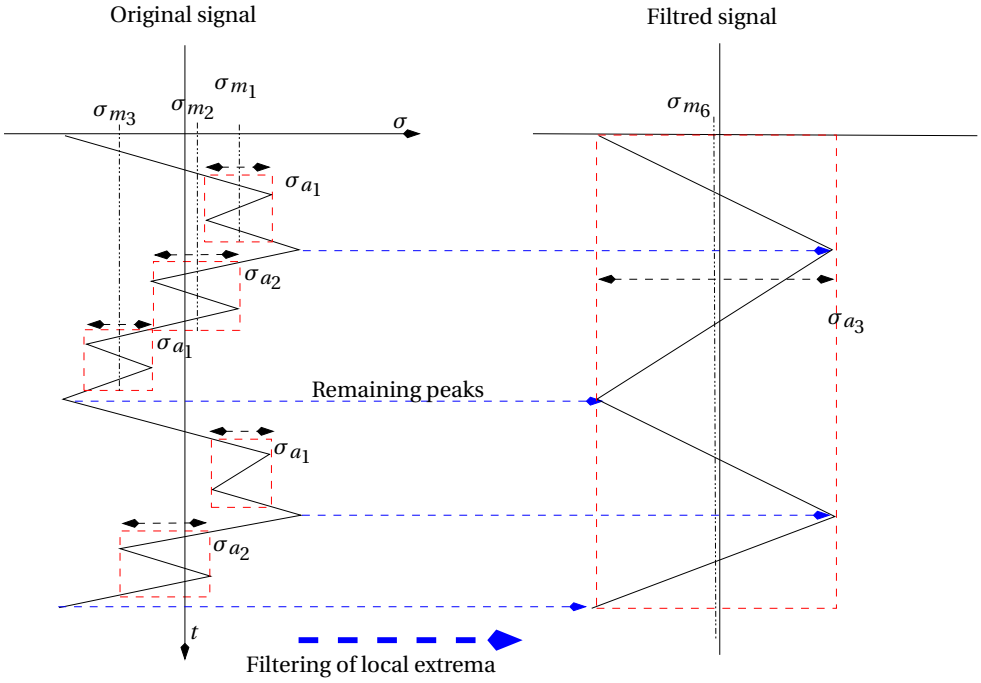
The rain-flow counting algorithm was introduced by ENDO [**26**]. From a signal-processing standpoint, the algorithm aims at encoding the loading signal by the sequence of its local extrema; it is often introduced in the literature devoted to fatigue analysis by the qualitative diagrams depicted in the figures (Fig. 1.12) and (Fig. 1.13).

The rain-flow algorithm aims at counting and classifying the local extrema of a sampled signal $v = (v(t_i))_{i=0}^{N}$. It will be formalized with the help of an algorithm which consists to simplify the signal in removing its monotone sections and counting the remaining oscillations by deleting them recursively.

Representing the "rain-flow" encoding of a signal under the form of a matrix $\left[R_{ij}\right]$, called *rain-flow matrix*, whose entry R_{ij} is the number of alternating cycles of magnitudes σ_{a_i}

Fig.a

Fig.b



Determination of alternating and mean stresses

– 3 alternate cycles at amplitude σ_{a_1}
– 2 alternate cycles at amplitude σ_{a_2}
– all the averages are different.

Rain-flow counting on the filtered signal

– 1.5 alternate cycles at amplitude σ_{a_3}
– the average is σ_{m_6}

Fig. 1.13. **Application of the rain-flow counting to a complex signal.** The principle consists to recursively filter the original signal to identify, with the help of the heuristic described in figure (Fig. 1.12), the alternating cycles in increasing order of magnitude. The process ends in counting the number of cycles of a residual signal which is shown in the figure Fig.b. A formal characterization of the residual signal is given in formula (1.9). This figure illustrates informally the fact that the operation of cycles counting by the rain-flow algorithm is a dissipative process, which depends only on the change of the direction of variation of the input signal. We show (see footnote 10 page 27) that this property of the counting function allows to represent this operation by a hysteresis.

at averages σ_{m_j}, damage computation consists to calculate the sum

$$\mathscr{D}(\sigma) = \sum_{ij} \frac{R_{ij}}{N_r\left(\sigma_{a_i}, \sigma_{m_j}\right)} \tag{1.7}$$

where, see the mapping plotted in figure (Fig. 1.5) page 13, $N_r\left(\sigma_{a_i}, \sigma_{m_j}\right)$ is the number of cycles to failure of a material which is submitted to the alternating stress σ_{a_i} at average σ_{m_j}.

From a formal point of view, *the rain-flow algorithm implements the following simplification rules*:

R_1/ signal simplification by *removing the middle point when three consecutive points are monotonically listed*, ie. the point v_i of a signal $v = (v_0, v_1, \cdots, v_N)$ is eliminated if it lies in the interval[6] $[v_{i-1}, v_{i+1}]$;

R_2/ the previous step leads to extract *the the local extrema from the original signal*; and it remains, see figures (Fig. 1.13) and (Fig. 1.14), to identify both their amplitudes and averages. To do this, we introduce the following simplification rules:

- Let be given four consecutive points $(v_k)_{k=i-2}^{i+1}$, we say that (v_{i-1}, v_i) is a *pair of Madelung*, see figure (Fig. 1.14), if the interval $[v_{i-1}, v_i]$ is contained in $[v_{i-2}, v_{i+1}]$. In this case, the processed portion of the signal is said to contain an *oscillation* of *amplitude* $\frac{|v_{i-1}-v_i|}{2}$ at *average* $\frac{v_{i-1}+v_i}{2}$.
- *The number* $a^{imp}(v)[v_{i-1}, v_i]$ *of Madelung's pairs* is incremented by 1 and *the signal is simplified* by deleting the pair (v_{i-1}, v_i).

At the end of this process, we obtain an irreducible residual signal v_R *and a integer valued function* $a^{imp}(v)[\rho_1, \rho_2]$, *defined on* $\mathbb{R}^2$, storing the number of Madelung's pairs of magnitude $\sigma_a = \frac{|\rho_1-\rho_2|}{2}$ at average $\sigma_m = \frac{\rho_1+\rho_2}{2}$ found in the sampled signal v.

As we want to not distinguish the pair (ρ_1, ρ_2) from the pair (ρ_2, ρ_1) in a counting function, we introduce the new *counting function*[7]

$$(\rho_1, \rho_2) := \rho \mapsto a(v)[\rho] = a^{imp}(v)[\rho_1, \rho_2] + a^{imp}(v)[\rho_2, \rho_1] \tag{1.8}$$

which is now defined on *the half-plane* $\mathscr{P}$ *of equation* $\rho_2 - \rho_1 \geq 0$, *referred to as Preisach plane*.

BROKATE [**7**] has proved, by induction on the number of samples of the signal v, the following consistency result for the rain-flow algorithm.

PROPOSITION 1.1 (Consistency result for the rain-flow algorithm) *Let a sampled signal* v *be given, there is an unique residual signal* v_R *which ends the rain-flow algorithm, regardless of the order in which the simplification sequences are applied. The counting function (1.8) (of the Madelung's pairs) doesn't depend on the order of the simplification sequences applied to* v *to compute it.*

We define in the following Remark the modifications which are to be made in the previous procedure to *count the extrema of a residual signal.*

[6] For convenience, we do not distinguish between the intervals

$$[a, b] = \{x \in \mathbb{R}\,;\ a \leq x \leq b\} \text{ and } [b, a] = \{x \in \mathbb{R}\,;\ b \leq x \leq a\}\,;$$

this means that, when talking about an interval $[a, b]$, we don't necessarily assume that $a \leq b$.

[7] On the example in figure Fig. 1.14 the pairs (v_2, v_3) and (v_8, v_9) are counted simultaneously.

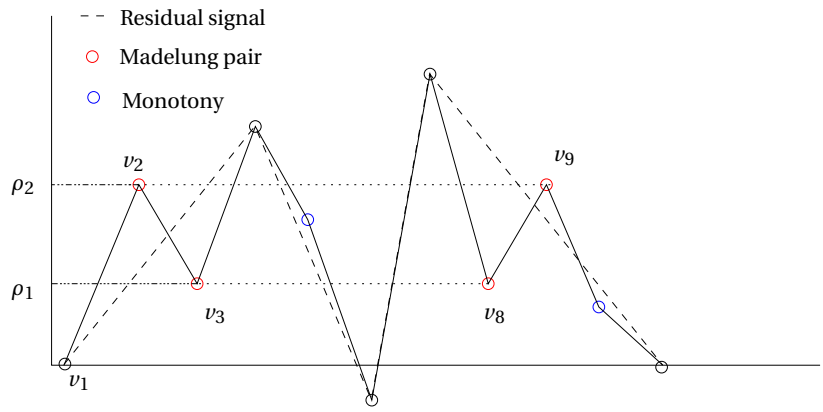


Fig. 1.14. **Illustration of the simplification processes applied in the rain-flow algorithm.** The blue samples are removed by an argument of monotony while red samples (which are Madelun's pairs) allow to count the oscillations of magnitude $\frac{\rho_2-\rho_1}{2}$ at average $\frac{\rho_2+\rho_1}{2}$ which are present in the signal. The residual signal, in dashed line, will in turn be post-processed by duplication of the signal, see figure (Fig. 1.15).

REMARK 1.5 (Characterization of an irreducible signal) Let $v = (v_i)_{i=1}^N$ a sampled signal be given; setting $d_i = v_{i+1} - v_i$, this signal v is irreducible if and only if it satisfies

$$\begin{aligned} &d_{i-1}\, d_i < 0 \text{ for each } 1 \le i \le N \text{ and there is an index } J \\ &\text{such that } |d_0| < |d_1| < \cdots < |d_J| \le |d_{J+1}| > \cdots > |d_N| \end{aligned} \tag{1.9}$$

The generic shape of a residual signal is plotted in figure (Fig. 1.15).

PROOF OF REMARK 1.5. This Remark is a consequence of the following results:

- a signal doesn't contain monotonic sections if an only if the first inequality of (1.9) holds;
- a pair (v_i, v_{i+1}) is Madelung if and only if

$$0 < |d_i| \le \min\{|d_{i-1}|\,, |d_{i+1}|\} \tag{1.10}$$

(have a look on the picture in figure (Fig. 1.14) to get convinced).

Let v be a signal satisfying the second condition of (1.9); the inequality (1.10) shows that the sequences $(v_0, \cdots, v_{J+1})$ and $(v_{J+1}, \cdots, v_n)$ do not contain Madelung's pair; since the pair (v_J, v_{J+1}) can't be Madelung, the signal v in its whole does not contain any Madelung's pair and is irreducible.

Conversely, if v doesn't contain any Madelung's pair, then denoting by J the smallest of the indices j such that

$$\left|d_j\right| > \left|d_{j+1}\right| > \cdots > |d_{N-1}|$$

and applying the characterization (1.10) of the Madelung's pairs (for $i = J-1, \cdots, 1$), we see that we must have

$$|d_0| < |d_1| < \cdots < |d_J|$$

and the second condition of (1.9) is satisfied. □

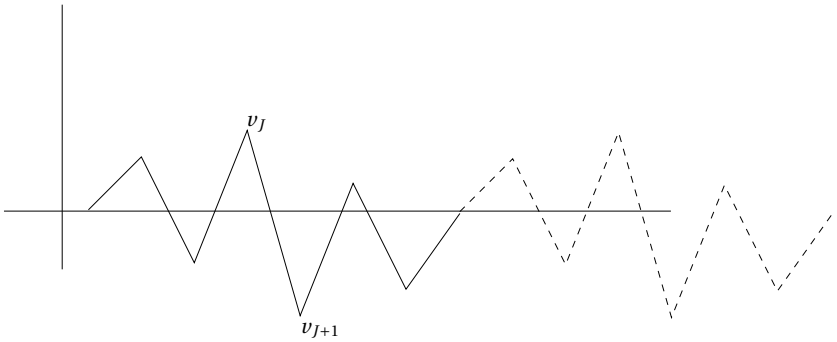


Fig. 1.15. **Generic shape of a residual signal obtained at the end of the rain-flow-algorithm**. The number of cycles of the residual signal is computed with the help of the rain-flow algorithm by processing the signal obtained by concatenating the points of the solid line to those of the dotted line curve. And this is precisely the purpose of the counting function a^{per}.

PROPOSITION 1.2 (Counting function adapted to process the residual signal) *In order to count the number of oscillations of the residual signal obtained at the end of the rain-flow algorithm, it is sufficient to apply rules R_1/ and R_2/ of the rain-flow algorithm to the concatenated signal $[v, v]$ with the following counting function:*

$$a^{per}(v) = a([v, v]) - a(v) \tag{1.11}$$

PROOF. We see, from the picture in figure (Fig. 1.15), that to count the oscillations of the residual signal it is sufficient to compute $a(v) + a([v_R, v_R])$. Using the following formula[8]:

$$a([u, v, w]) = a(v) + a([u, v_R, w])$$

with $u = v$ and $w = \emptyset$ on the first hand, and with $u = \emptyset$ and $w = v_R$ on the other hand; we obtain the formulas

$$a([v, v]) = a([v, v, \emptyset]) = a(v) + a([v, v_R, \emptyset]) = a(v) + a([v, v_R])$$

$$a([v, v_R]) = a([\emptyset, v, v_R]) = a(v) + a([\emptyset, v_R, v_R]) = a(v) + a([v_R, v_R])$$

which proof (1.11). □

The previous results justify the following fundamental definition, which formalizes damage computation by rain-flow counting algorithm and Palmgren-Miner's accumulation law.

DEFINITION 1.1 (Formal definition of damage) If the fatigue accumulation law is the Palmgren-Miner's rule the damage caused by a sampled loading v is defined by the following double sum

$$\boxed{\mathscr{D}(v) = \sum_{\rho_2 - \rho_1 \geq 0} \frac{a^{per}(v)[\rho_1, \rho_2)]}{N_r(\rho_1, \rho_2)}} \tag{1.12}$$

where :

[8] Which can be verified by starting the rain flow algorithm on the signal v and in applying the Proposition 1.1.

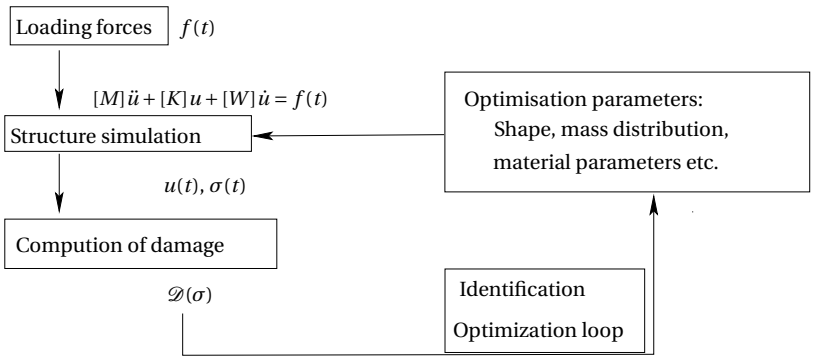


Fig. 1.16. **Principles of structure optimization a under fatigue life criterion.** Structure simulations allow to define the displacements $u(t)$ and the stresses $\sigma(t)$ according to the applied forces $f(t)$. Damage is then computed, see figure (Fig. 1.6) and Definition 1.1, from the computed stresses $\sigma(t)$. *Optimize consists*, knowing the loading F, to compute the entries of $[M]$, $[K]$ and $[W]$ which make the damage $\mathscr{D}$ lower than a given value (resp. which minimize for instance the mass of the structure under the constraint $\mathscr{D} \leq D_0$).

i) for any sampled signal v, the mapping $\rho \in \mathscr{P} \mapsto a^{per}(v)[\rho]$ is defined, according to the number of Madelung pairs of amplitude $\frac{\rho_1-\rho_2}{2}$ at average $\frac{\rho_1+\rho_2}{2}$ identified in the concatenated signal $[v, v]$ and in the signal v, by the formula (1.11);

ii) and $N_r(\rho)$ is the number of cycles to failure of the material for an alternating stress of amplitude $\sigma_a = \frac{\rho_2-\rho_1}{2}$ at average $\sigma_m = \frac{\rho_2+\rho_1}{2}$.

As the mapping $\rho \mapsto a^{per}(v)[\rho]$ is non zero on a finite number of points, *the sum* (1.12) *is actually carried out on a finite number of terms*[9].

Within this framework, *the rain-flow matrix* (1.7) *page 22 is a discretized representation of the counting function* $(\rho_1, \rho_2) \in \mathscr{P} \mapsto a^{per}(v)[\rho]$ once a discretization

$$\big(i\delta\rho_1, j\delta\rho_2\big)_{(i,j)\in\mathbb{Z}\times\mathbb{Z}}$$

of the half-plane $\mathscr{P}$ has been defined. In other words, the entries of rain-flow matrix $[R]$ are defined as

$$R_{ij} = a^{per}(v)\left[\big(i\delta\rho_1, j\delta\rho_2\big)\right]$$

1.3. Outline of the further results and scope of work

On the basis of the classical formalism introduced in Definition 1.1, the next aims at *generalizing the damage calculation procedure to time continuous signals.* This is the cornerstone allowing us to apply the methods *of the calculus of variations to formalize and solve the optimization problem introduced in figure* (Fig.1.16). We will establish,

[9] The sums and more generally the integrals are written on an infinite domains for convenience. There is no, in the matters dealt with here, any underlying convergence problem neither for the integrals nor for the sums.

see figure (Fig. 1.17), that the appropriate framework is that of the hysteresis modeling, which permits to

- reduce the computation of the damage caused on a structure by a time-continuous loading $t \mapsto v(t)$ to the energy dissipated in the hysteresis loops of the image of v by a Preisach operator[10] (Definitions 2.3 page 42 and 2.1 page 39) appropriately calibrated (Theorem 2.1 page 43) with the help of the analytical forms of the Wöhler's curves introduced in Section 1.1;
- and to formalize as follows (Theorem 2.3 page 72 and Remarks 2.8 page 75) the optimization problem depicted in figure (Fig. 1.16):

$$\begin{aligned} &\text{Minimize } \mathscr{D}(u) = \int_0^T j(X_u)\,dt \\ &\text{under the constraint } \frac{dX_u}{dt} = f(X_u, u, t) \quad \text{for } t \in [0, T] \end{aligned} \tag{1.13}$$

  We explain in Chapter 4 how to solve this problem by *gradient based methods for which the descent directions are computed via the integration of an adjoint equation* (see Proposition 4.10 page 165).

The numerical methods traditionally used to deal with the problem (1.13) are expensive to set up (see algorithm 4.7 page 173) and are inefficient to process FEM models. *To circumvent this difficulty, we introduce* in Section 3.1 page 96, *a forced response method* for the integration of second order linear systems, which permits

- to perform the integration of the state and adjoint equations on the same sampling as the loads;
- and, via a reduction of model, to significantly *reduce the amount of data which are to be stored to solve the adjoint equation.* We can indeed check that in case of fatigue analysis of a structure, where excitations occur at low frequencies and on a fairly long time, *less than* 10% *of the equations describing the dynamical behavior of the structure are actually needed to compute the criterion and its gradient.*

This integration method differs from the conventional ones implemented in the finite elements software insofar as *it reproduces the transients states of the dynamical system.* We will see that this property is particularly welcomed for the integration of the adjoint equation which, in the case of fatigue analysis, looks like a second-order system for which the right-hand member is a discontinuous function of time.

The theoretical results are illustrated with the help of numerous examples and algorithms (see Section 3.2 page 105 and Annexes B page 209) whose purposes are to support the relevance of the approach and serve as templates for its implementation in structure software.

[10] Basically, it is (see BERTOTTI- MAYERGOYZ [**27**], KRASNOSEL'SKII - POKORVSKII [**19**] and VISINTIN [**40**]) the mathematical way to characterize a signal processing operation which is covariant by time rescaling and therefore independent of the velocity, as it is the case for the cycle-counting operation.

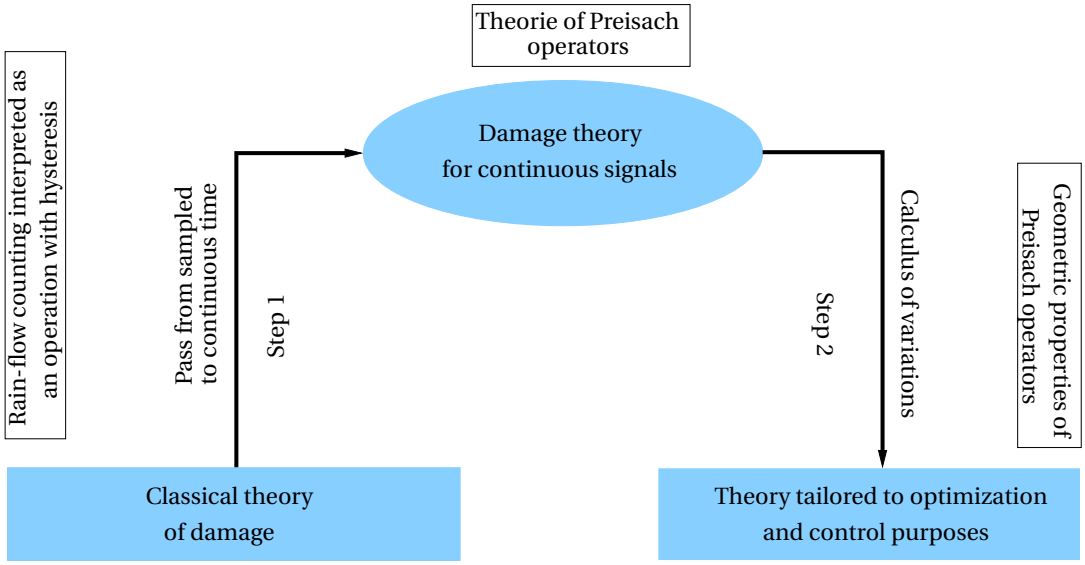


Fig. 1.17. **Overview on the mathematical problem.** As it only makes sense for sampled signals, the formula (1.12) doesn't permit to exploit the flexibility of "the calculus of variations" to write down and solve the optimization problem explained in the figure (Fig. 1.16). We will therefore extend the definition of damage calculation to continuous-time signals; the exercise will be processed within two steps: the first aims at showing that rain-flow counting process and accumulated damage computation can be performed with the help of relay and Preisach hysteresis operators, and the second uses the geometric properties of these operators to make explicit the integrand j in formula (1.13).

1.4. Exercises and complements

EXERCICE 1.1 For an aluminum alloy, fatigue tests have given the results shown in table Tab 1.1; two specimens were used for each stress level,

- Plot the $S-N$ curve,
- What is the fatigue limit σ_d?
- Identify Stromeyer and Bastenaire's coefficients for these $S-N$ curves
- Suppose that the specimen is cyclicly loaded between 50 and 350 MPa, compute its number of cycles to failure if $R_e = 450MPa$ and $R_m = 570MPa$
- Plot the $S-N$ surfaces as functions of σ_a and σ_m.

σ_a	400	350	300	250
N_r	$1.5E+04; 2.0E+04$	$4.E+04; 5.0E+04$	$2.1E+05; 2.0E+05$	$9.0E+05; 1.0E+06$
σ_a	220	180	170	160
N_r	$5.0E+06; 6.0E+06$	$5.1E+07; 5.0E+07$	$1.1E+08; 1.0E+08$	$7.0E+08; NF$

Tab. 1.1. **Experimental data obtained from fatigue tests carried out at mean stress** $\sigma_m = 0$.

EXERCICE 1.2

- Program in "Matlab" the rain-flow counting algorithm

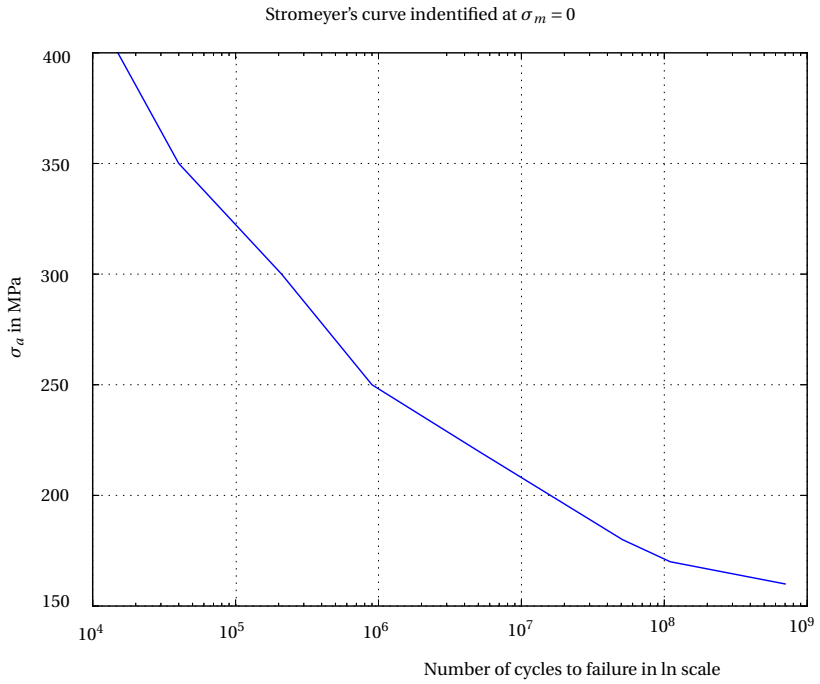


Fig. 1.18. **Identification of Stromeyer coefficients obtained from the experimental data given in table Tab 1.1**.

- Test it on signals of the form $200MPa * (\sin(t). * \sin(\omega * t))$, where t is the table $t = [0 : \delta_T : T]$
- Write a program to compute and plot the rain-flow matrix,
- Make a program to compute the damage.

Solutions and homework.

Solution of exercise 1.1. The Wöhler's curve is plotted in the figure Fig. 1.18, in semi-ln scales.

1/ Note that *this curve has no inflection point,*
2/ it can be interpolated by a *Stromeyer's formula* with coefficients

$$a_s = 7.70 \quad b_s = 0.227 \quad \sigma_d = 150\,MPa \quad C_s = 1.5 * 10^{15}.$$

These coefficients were obtained with the help of the python version of the Levenberg-Marquart's algorithm defined in LOURAKIS [**25**]. A simplified version of this identification program is given in the algorithm 1.1
3/ This curve *can't be interpolated by a Bastenaire* formula!

The Bastenaire formula is usually identified with the help of an *identification algorithm such as the algorithm 1.1.* See Section 4.1 page 140 for a better understanding of the optimization algorithms.

Wöhler's surface as a function of the variables σ_a and σ_m is plotted in figure (Fig. 1.19).

Homeworks.

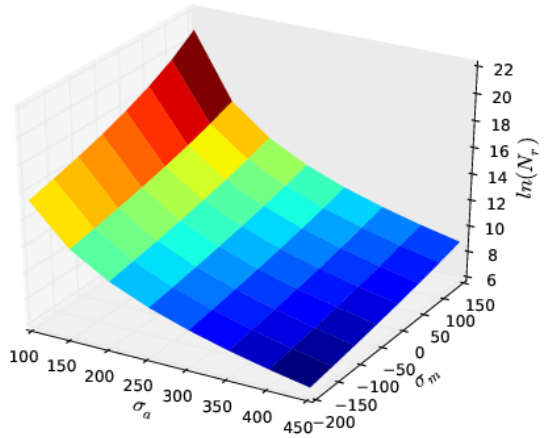


Fig. 1.19. **Wöhler's surface**.

Algorithm 1.1: *Basic algorithm for the identification of a parametric Wöhler's curve by a steepest descent method.*

Inputs :

- Experimental Wöhler curve data $\left(N_{r_i}\right)_{i=1}^{N}$ $\left(\sigma_{a_i}\right)_{i=1}^{N}$
- Parametric law

$$N_r(\sigma_a) = A\frac{e^{C(\sigma_d-\sigma_a)}}{\sigma_a-\sigma_d} - B \tag{1.14}$$

Outputs :

- Coefficients $\left(A^{opt}, B^{opt}, C^{opt}, \sigma_d^{opt}\right)$
- such that $N_{r_i} \approx N_r(\sigma_{a_i})$ for all $1 \leq i \leq N$, ie. *minimizing the criterion*

$$J(A,B,C,\sigma_d) = \sum_{i=1}^{N}\left(N_{r_i} - N_r(\sigma_{a_i}\right)^2 \tag{1.15}$$

begin

- Let the initial values $\left(A^0, B^0, C^0, \sigma_d^0\right)$ be given
- **while** $\|\nabla J\| > \varepsilon$ **do**

$$\left(A^k, B^k, C^k, \sigma_d^k\right) \leftarrow (A^{k-1}, B^{k-1}, C^{k-1}, \sigma_d^{k-1}) - c\nabla J(A^{k-1}, B^{k-1}, C^{k-1}, \sigma_d^{k-1})$$

end

- Set

$$\left(A^{opt}, B^{opt}, C^{opt}, \sigma_d^{opt}\right) = \left(A^k, B^k, C^k, \sigma_d^k\right)$$

end

- Explain with a picture the algorithm 1.1.
- Propose additional data allowing to identify a Bastenaire's curve.

- Modify the optimization algorithm in order to be able to identify the following curves:
  - Weibull: $\ln(N_r + B) = a - b\frac{\sigma_a - \sigma_d}{\sigma_u - \sigma_d}$
  - Stüssi: $\ln(N_r) = a - b\frac{\sigma_a - \sigma_d}{\sigma_u - \sigma_d}$

  where σ_u is a parameter defined in the figure Fig. 1.20.

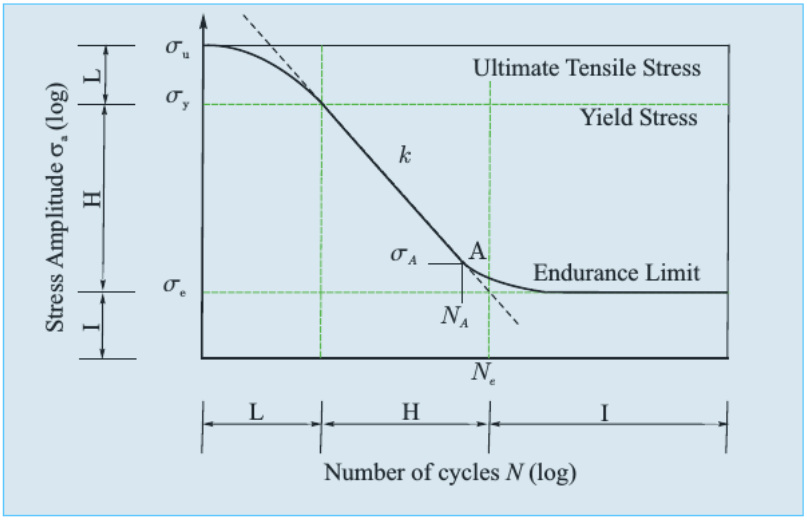


Fig. 1.20. **Physical interpretation of the parameter σ_u in Weibull and Stüs-Philips formulas**.

Solution of exercise 1.2. **Rain-flow algorithm**

- Counting function

```
function [y,rho_1,rho_2,Ind]=madelung(y,Nb_samp)
%
% Purpose: identify and remove the first Madelung pair in
% a sampled signal y.
%
    if Nb_samp>=4
        i=1;
        while (abs(y(i+1)-y(i+2))-min(abs(y(i)-y(i+1)),...
                                      abs(y(i+2)-y(i+3)))>=1.e-04)...
                && (i<Nb_samp-3)
            i=i+1;
        end
        % The index i is such that y(i+1), y(i+2) is
        % the first Madelung pair in y
       if i==Nb_samp-3
       % Case where there is no Madelung pair in y (ie. y is residual)
            Ind=0;% There is no Madelung pair
            rho_1=NaN;
            rho_2=NaN;
        else
            rho_1=y(i+1);
            rho_2=y(i+2);
```

```
            % The Madelung pair (ρ1, ρ2) is removed from y
            for j=i+1:Nb_samp-2
                y(j)=y(j+2);
            endfor
            Ind=1;% a Madelung pair has been found
        endif;
    else
        Ind=0;
        rho_1=NaN;
        rho_2=NaN;
    endif
endfunction
```

- Main program

```
time=0:0.01:120;% Time smapling
x_0=sin(time)+sin(time).*sin(4*time)+sin(0.5*time).*sin(6*time);
% x_0=sin(time);
% x_0=[0,1,-1,1.5,-2,2.5,-1.5,1,-0.5,0]
% time=[0,0.1,0.2,0.3,0.4,0.5,0.7,0.8,0.9,1]
```

1) Apply the simplification rule R_1)

```
% Remove monotonous sections in x
%
x=x_0;
Nb_samp=size(x,2);% Number of samples
for i=1:Nb_samp-2
if(x(i)-x(i+1))*(x(i+1)-x(i+2))>=0  % Apply rule R1) of the
        x(i+1)=x(i);                % rain-flow algirthm
    endif;
endfor
%
% Remove the duplicated entries in x to make y
j=1;
y(1)=x(1);
time2(1)=time(1);
for i=1:Nb_samp-1
    if abs(x(i)-x(i+1))>0
        j=j+1;
        y(j)=x(i+1);
        time2(j)=time(i+1);
    endif;
endfor;
```

2) Plot the obtained result

```
txt1=['Original signal (',num2str(size(x,2)),' samples)'];
txt2=['Simplified signal (',num2str(size(y,2)),' samples)'];
figure(1);
subplot(211)
plot(time,x_0)
title(txt1); xlabel('time'); ylabel('x(t)');
subplot(212)
plot(time2,y)
title(txt2);xlabel('time'); ylabel('y(t)');
Nb_samp_s=size(y,2) % Note that Nb_samp_s <<< Nb_samp!
```

see figure Fig. 1.21.

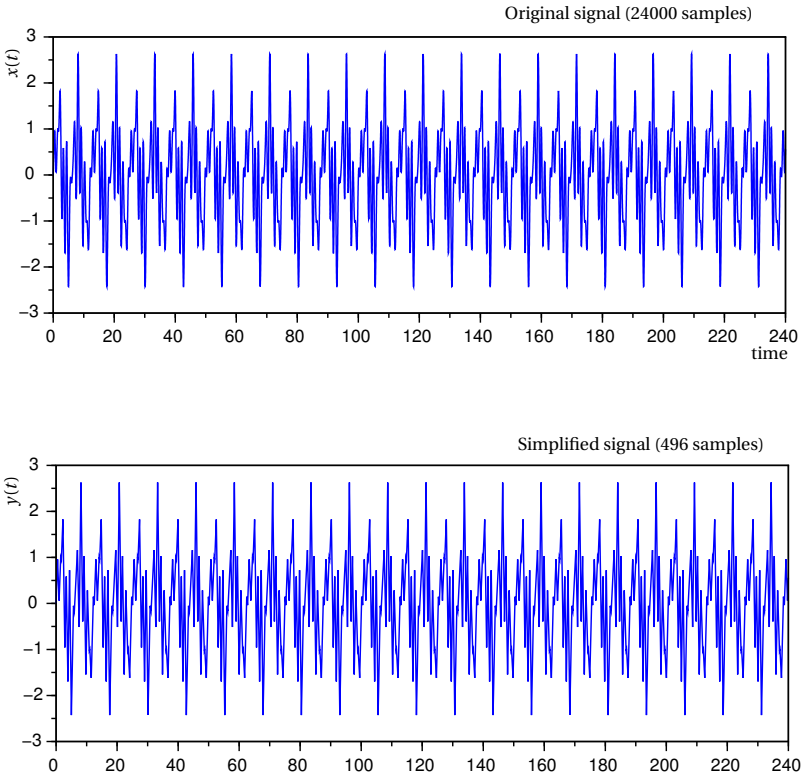


Fig. 1.21. **Simplification of the original signal in removing the monotone sequences**.

3) Apply rule R_2) to identify the Madelung's pairs in y

```
Ind=1;
i=1;
while Ind>0
    [y,rho_1,rho_2,Ind]=madelung(y,Nb_samp_s);
    if Ind>0  % If a Madelung pair has been found in y
        tab_1(i)=rho_1;% store the pair in tables tab_1 and tab_2
        tab_2(i)=rho_2;% to compute the rain-flow matrix
        i=i+1;
        Nb_samp_s=Nb_samp_s-2;%and reduce the number of samples used
    endif                    %for the next research
endwhile
% As the research stops when the number of Madelung pair is 0
% y(1:Nb_samp_s) is the residual signal v_R
% Plot it for checking!
figure(2);
subplot(211)
plot(x_0)
title('Original signal');
subplot(212)
plot(y(1:Nb_samp_s))
title('Residual signal obtained at the end of rain-flow algorithm');
```

see figure Fig. 1.22.

4) Compute the rain-flow matrix in the axes (ρ_1,ρ_2)[11]

```
if Nb_samp_s==size(y,2)
```

[11] The rain-flow matrix is often defined in the axes (σ_a,σ_m) in the literature.

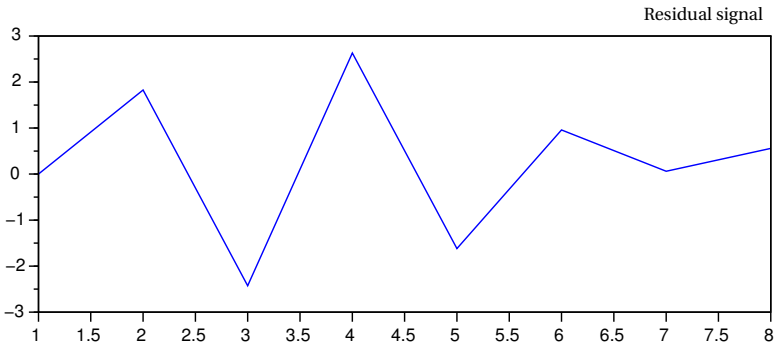


Fig. 1.22. **Example of residual signal obtained at the end of the rain-flow algorithm**.

```
    Nb_madelung_pair=0;% In this case, there is nothing to compute
else
    Nb_madelung_pair=size(tab_1,2);
    N=40;% Number of samples along the ρ1 axis
    % The Preisach plane is restricted to
    % the rectangle min(x)≤ ρ1, ρ2≤ max(x)
    delta_rho=(max(x_0)-min(x_0))/N;
    %
    for i=1:N+1
        for j=1:N+1
            tab(i,j)=0.0;
        endfor
    endfor
    for i=1:Nb_madelung_pair
        rho_1=tab_1(i);
        rho_2=tab_2(i);
        i_1=floor((rho_1-min(x_0))/delta_rho)+1;
        j_1=floor((rho_2-min(x_0))/delta_rho)+1;
        tab(i_1,j_1)=tab(i_1,j_1)+1;
    endfor
    %
    rho=min(x_0):delta_rho:max(x_0);
    % Compute a^per(ρ1,ρ2) and plot it (rain-flow matrix)
    for i=1:N+1
        for j=1:i
            tab2(i,j)=tab(i,j)+tab(j,i);
        endfor
        for j=i+1:N+1
            tab2(i,j)=NaN;
        endfor
    endfor
    %
    figure(3);
    surf(rho,rho,tab2)
    xlabel('rho_1'); ylabel('rho_2'); zlabel('Number of cycles');
endif
```

5) Now compute the total damage with help of the formula (1.12), where a^{per} is tabulated in the table $tab2$.

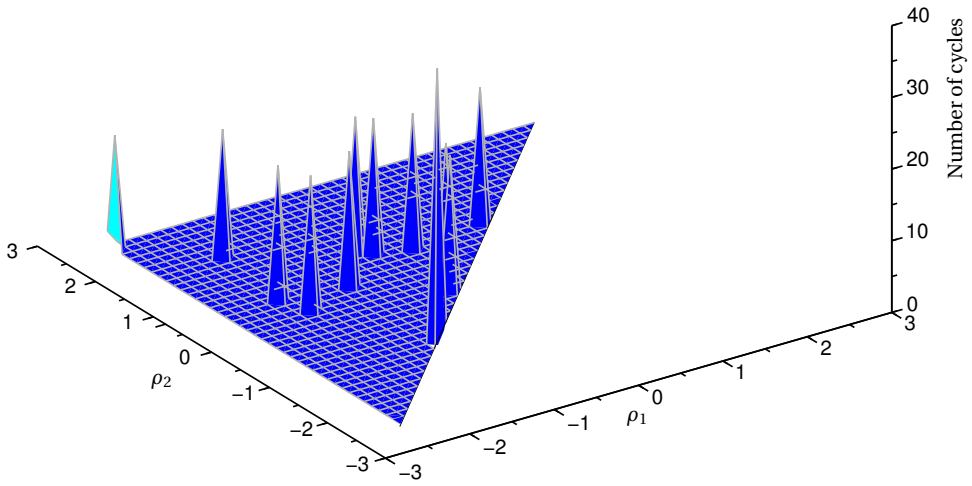


Fig. 1.23. **Rain-flow matrix obtained in applying the rain-flow algorithm on the signal shown in figure Fig. 1.21**.

Homeworks.

- Complete the program to compute the total damage.
- Run the program with $x_0 = \sin(time)$; did you expect that? Modify the input data to obtain better results and justify the modifications.

CHAPTER 2

DAMAGE CALCULUS FOR TIME-CONTINUOUS SIGNALS

PURPOSE of this Chapter is to extend to time-continuous signals the formula (1.12) introduced in Definition 1.1 which defines the damage generated by sampled loading. This will make us able to use *the classical methods of the calculus of variations* to write down and solve in Chapter 4 the optimization problem introduced in figure (Fig. 1.16) page 26.

This Chapter is organized as follows:

Contents

Starting from the results obtained by BROKATE, DREBLER and KREJCI [**7**] we show in Section 2.1 (Theorem 2.1 page 43) that we can calibrate the density μ of a Preisach operator $\mathcal{W}_\mu$ acting on the space of finite sequences to *compute the damage* $\mathcal{D}(v)$ *as the total variation* $V_T\left(\mathcal{W}_\mu(v)\right)$ of the image $\mathcal{W}_\mu(v)$ of the sequence v by the operator $\mathcal{W}_\mu$.

We make in Section 2.2 the passage from discrete to continuous time and we justify *the new definition*

$$\mathscr{D}(v) = \int_0^T \left| \mathscr{W}'_\mu(v,t) \right| dt \tag{2.1}$$

of the damage caused by a continuous loading $t \in [0,T] \mapsto v(t)$.

The geometric representation of a Preisach operator introduced in Section 2.3 (Theorem 2.2 page 66) will permit in Section 2.4 to

1^o/ explicit
- in Theorem 2.3 page 72, the integrand $\left|\mathscr{W}_\mu(v,t)'\right|$ in the formulation (2.1) of the damage;
- and, in Remarks 2.8 page 75, the computation of the derivatives of $\left|\mathscr{W}_\mu(v,t)'\right|$ with respect to the variables $v(t)$ and $\dot{v}(t)$, which are assumed to be independent variables;

2^o/ check that, *although the Preisach operator is not differentiable, in the sense of Frechet for instance, its outputs* $\mathscr{W}_\mu(v,t)$ *are almost every where differentiable with respect to the inputs* $v(t)$.

We proof in Chapter 4 (Proposition 4.10 page 165) *that this notion of "weak differentiability" of the Preisach operator is sufficient to define a descent direction for the structure optimization problem.*

2.1. Reformulation of the damage computation process

In this Section we formulate the damage computation process $v \mapsto \mathscr{D}(v)$ in terms of functional operations applied on the signal v. *At the end of this Section, we will have defined two equivalent ways to compute the damage caused by a sampled loading* v:

- the first one is the standard method (formula (1.12) page 25) based on the rain-flow counting algorithm and the Palmgren-Miner's rule;
- while the second, introduced in this Section (formula (2.12) page 43) leads us to *understand the total damage as the energy dissipated in the hysteresis loops* of the image $\mathscr{W}_\mu(v)$ of v by a Preisach operator $\mathscr{W}_\mu$ appropriately calibrated.

Besides the fact that the latter formulation of the damage computation method fits with the functional framework allowing to handle the optimization problem depicted in the figure (Fig. 1.16) with the help of the classical methods of the calculus of variations, it seems to *better correspond to the intuition of the failure mechanisms of a material* than the first one, which rather appears as a heuristic.

This Section is organized into two sub-sections aiming to

1^o/ show, in the first subsection, that cycle identification and cycle counting can be performed with the help of relays hysteresis, see Definition 2.1;

2^o/ calibrate the coefficients of a Preisach operator (see Definition 2.3 and Theorem 2.1 page 43) to perform the weighted cycle counting and the time integration required for the computation of the total damage.

Cycle counting by relay operator. If $\rho_1 > \rho_2$ are two given real numbers, a relay operator of thresholds ρ_1 and ρ_2 defined bellow, is a mathematical device tailored to *identify the oscillations of a sampled signal v crossing the interval* $[\rho_1, \rho_2]$.

DEFINITION 2.1 (Relay operator) Let $\rho_1 > \rho_2$ be two real numbers, we call *relay operator of thresholds* ρ_1 and ρ_2 the mapping $h_\rho : \mathbb{R}^n \to \{0,1\}^{n+1}$ defined by

$$\left[h_\rho(v)\right]_i := z_i = \begin{cases} 1 \text{ if } v_i \geq \rho_2 \\ 0 \text{ if } v_i \leq \rho_1 \\ z_{i-1} \text{ if } \rho_1 < v_i < \rho_2 \end{cases} \quad \text{for } 1 \leq i \leq n \tag{2.2}$$

$$z_0 = w_{-1}$$

where ρ is an abbreviation which refers to the couple $(\rho_1, \rho_2) \in \mathscr{P}$ and n is a given integer number.

To remove the ambiguity in formula (2.2) we impose, *arbitrarily for the time being*, the initial state $w_{-1} \in \{0,1\}$ of $\left[h_\rho(v)\right]_{-1}$. When will we have to specify this initializing state of the relay operator we will use the complete notation $h_\rho(v, w_{-1})$ instead of the simplified one $h_\rho(v)$.

A Relay operator is an elementary hysteresis operator, called hysteron, which changes of state when the signal v crosses the interval $[\rho_1, \rho_2]$:

- it switches from 0 to 1 (resp. from 1 to 0) when the signal crosses the interval $[\rho_1, \rho_2]$ in the increasing (resp. in the decreasing) direction,
- while it remains unmodified in the others cases.

An illustration is provided in figure (Fig. 2.2) page 41.

Now we proof that *counting the number of changes of state of the relay h_ρ is the same as counting the number of oscillations of the signal which cross the thresholds ρ_1 and ρ_2* and we *connect this operation with the cycle-counting function* $\rho \mapsto a^{per}(v)[\rho_1, \rho_2]$ introduced in Chapter 1 page 25 to formalize the rain-flow counting algorithm.

Let a sampled signal v be given, if (ρ_1', ρ_2') is a Madelung's pair such that $[\rho_1, \rho_2] \subset [\rho_1', \rho_2']$ then, see figure (Fig. 2.1), the simplified signal v' obtained in removing the pair (ρ_1', ρ_2') in v satisfies the equation[1]

$$V_T\left(h_\rho(v')\right) = V_T\left(h_\rho(v)\right) - 2$$

[1] The following Definition of the total variation of a finite sequence is extended to the continuous functions (Definition 2.6 page 51) where it is connected with the notion of derivative when the function v is sufficiently regular.

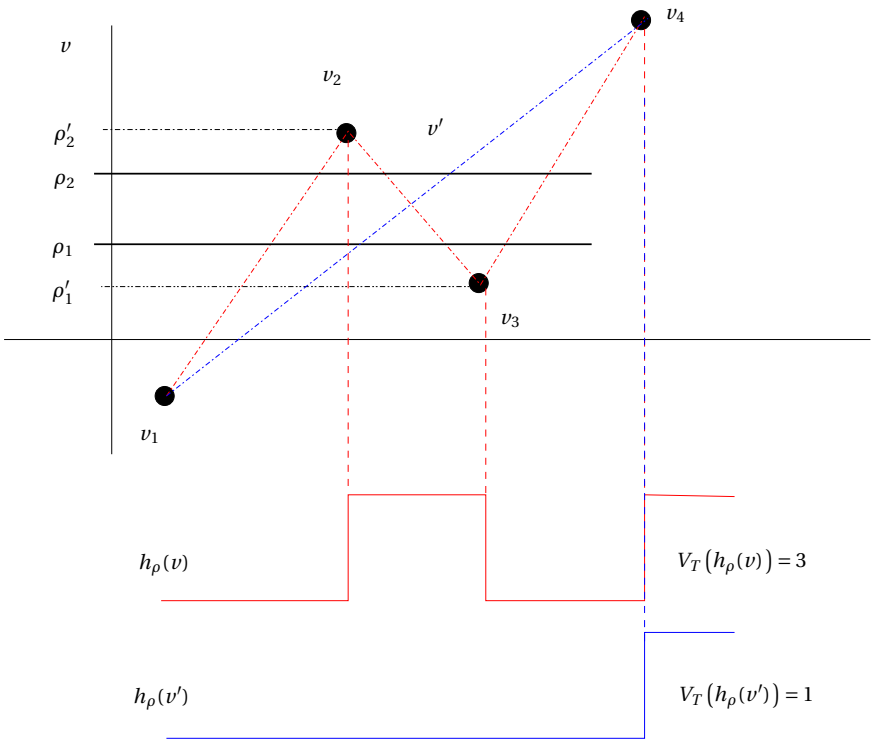


Fig. 2.1. **Relay filtering for counting the Madelung's pairs of a sampled signal**. Detect and remove an oscillation that crosses the thresholds ρ_1 and ρ_2 in the signal v reduces of 2 the total variation of the sequence $\left([h_\rho(v)]_i\right)_{i=0}^N$. In this figure, the signal v' in blue is obtained from v (in red) in deleting the points v_2 and v_3. Thus the total variation of the sequence $h_\rho(v)$, which is 3, is decremented by 2 in "removing" the Madelung's pair (v_2, v_3) in v.

continuing the simplification process until removing all the Madelung's pairs in the signal v we see that

$$V_T\left(h_\rho(v)\right) = 2 \sum_{\rho_1' \le \rho_1 < \rho_2 \le \rho_2'} a(v)[\rho_1', \rho_2'] + V_T\left(h_\rho(v_R)\right) \tag{2.4}$$

We are going to proof that *if the initial state* $\left[h_\rho(v)\right]_{-1}$ *is defined so that*

$$\left[h_\rho(v)\right]_{-1} = \left[h_\rho(v)\right]_0 = \left[h_\rho(v)\right]_N \tag{2.5}$$

the contribution of the residual signal to the total variation (2.4) *is canceled and we obtain the simpler formula*

$$V_T\left(h_\rho(v)\right) = 2 \sum_{\rho_1' \le \rho_1 < \rho_2 \le \rho_2'} a^{per}(v)[\rho_1', \rho_2'] \tag{2.6}$$

DEFINITION 2.2 (Total variation of a finite sequence) The total variation of a finite sequence $v = \left(v_i\right)_{i=0}^N$ is the positive number

$$V_T(v) = \sum_{i=0}^{N-1} \left|v_{i+1} - v_i\right| \tag{2.3}$$

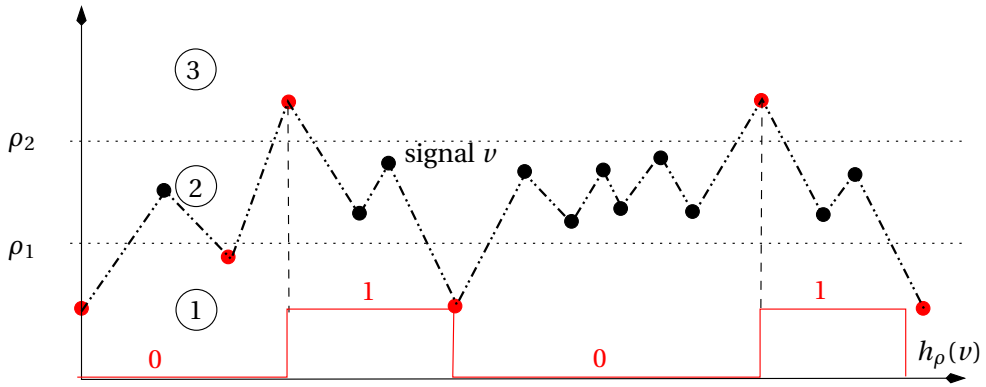


Fig. 2.2. **Outputs of the relay operator applied on a sampled signal.** The output of the relay switches when the signal crosses the zone 2. For instance, it switches from 0 to 1 when the signal passes from the zone 1 to the zone 3. If the signal remains confined in the zone 2, the relay stays stuck. The other interest of this figure is to show that when the signal v crosses $[\rho_1,\rho_2]$, the total variation of $t \mapsto h_\rho(v)(t)$ is incremented by 1. For example on this picture, the total variation of $h_\rho(v)$ is 4 and there are two alternating cycles of magnitude greater than $\frac{\rho_2-\rho_1}{2}$.

PROOF OF (2.6) UNDER THE HYPOTHESIS (2.5). By definition (1.11) page 25 of the counting function a^{per}, we have

$$V_T\left(h_\rho([v,v])\right) = 2 \sum_{\rho_1' \le \rho_1 < \rho_2 \le \rho_2'} (a^{per}(v) + a(v))[\rho_1',\rho_2'] + V_T\left(h_\rho([v,v]_R)\right)$$

As $[v,v]_R = v_R$, we can subtract (2.4) from the previous formula to obtain

$$V_T\left(h_\rho([v,v])\right) - V_T\left(h_\rho(v)\right) = 2 \sum_{\rho_1' \le \rho_1 < \rho_2 \le \rho_2'} a^{per}(v)[\rho_1',\rho_2']$$

To complete the proof, we just have to notice that the initialization (2.5) of the relay operator entails $V_T\left(h_\rho([v,v])\right) = 2\,V_T\left(h_\rho(v)\right)$. □

A way to satisfy the condition (2.5) consists to suppose that $v_0 = v_N$ and to initialize the relay at $w_{-1}^{per} = h_\rho(v,w_{-1})_N$, where w_{-1} is an arbitrary initial state. This leads to the following definition:

$$h_\rho^{per}(v,w_{-1}) = h_\rho(v,w_{-1}^{per}) \text{ where } w_{-1}^{per} \text{ is the last value of the sequence } h_\rho(v,w_{-1}) \tag{2.7}$$

of a "periodic relay" operator[2], which is tuned to satisfy[3]

$$\boxed{V_T\left(h_\rho^{per}(v)\right) = 2 \sum_{\rho_1' \le \rho_1 < \rho_2 \le \rho_2'} a^{per}(v)[\rho_1', \rho_2']} \tag{2.9}$$

regardless its initialization but under the condition $v_0 = v_N$.

Damage computation via a Preisach operator. In this subsection we calibrate a Preisach operator (see BERTOTTI [**2**], MAYERGOYZ [**27**], BROKATE [**7**], KREJCI [**20**] or VISINTIN [**40**]) to simultaneously perform the cycle counting operations and the weighted summations required by the computation of the total damage.

DEFINITION 2.3 (Preisach operator) Let be given a numerical mapping $\rho \mapsto \mu(\rho)$ defined on the half-plane $\mathscr{P}$ of equation $\rho_2 - \rho_1 > 0$. *We call Presach operator* the operator which associates to a finite sequence $v = (v_i)_{i=0}^N$ the sequence $\mathscr{H}_\mu(v)$ defined as

$$\left[\mathscr{H}_\mu(v)\right]_i = \int_{\mathscr{P}} \left[h_\rho(v)\right]_i \mu(\rho)\, d\rho \quad \text{for any index } i \tag{2.10}$$

where h_ρ is the relay operator defined in (2.2).

We denote, on the other hand, by $\mathscr{W}_\mu(v)$ *the Preisach operator associated with* the definition (2.7) of *the "periodic relay operator"*.

The Preisach operator $v \mapsto \mathscr{H}_\mu(v)$ *defined above is a hysteresis operator which is made up of parallel connection of hysterons switches* h_ρ, $(\rho \in \mathscr{P})$ *with the weights* $\mu(\rho)$. Purpose of Theorem 2.1 is to define the distribution of weights $\rho \in \mathscr{P} \mapsto \mu(\rho) \in \mathbb{R}$ allowing to compute the damage generated by a signal v as the total variation of the sequence $\mathscr{H}_\mu(v)$.

REMARKS 2.1 1^o/ A Preisach operator is often defined as follows in the literature

$$\mathscr{H}_\mu(v) = \int_{\mathscr{P}} h_\rho(v)\, d\mu(\rho)$$

where h_ρ is the relay operator of thresholds $\rho = (\rho_1, \rho_2) \in \mathscr{P}$ and μ is a measure, called *Preisach measure*, defined on the half-plane $\mathscr{P}$. The formula (2.10) corresponds to the case where the Preisach measure is a density measure with respect to the Lebesgue measure $d\rho_1 d\rho_2$ on the plane.

[2]Which, in other words, satisfies

$$h_\rho([v, v], w_{-1}) = \left[h_\rho(v, w_{-1}), h_\rho^{per}(v, w_{-1})\right]. \tag{2.8}$$

This means that the operator h_ρ^{per} maps the sequence v of length n to the last section of length n in the sequence $h_\rho([v, v], w_{-1})$, whose is length $2n$.

[3]This expression of the total variation of the discrete relay operator allows, in the proof of Theorem 2.1, to tune the parameter μ of a Preisach operator $v \mapsto \mathscr{H}_\mu(v)$, acting on the finite sequences, to compute the damage caused by a sampled signal v as the total variation of the sequence $\left(\left[\mathscr{H}_\mu(v)\right]_i\right)_{i=0}^N$.

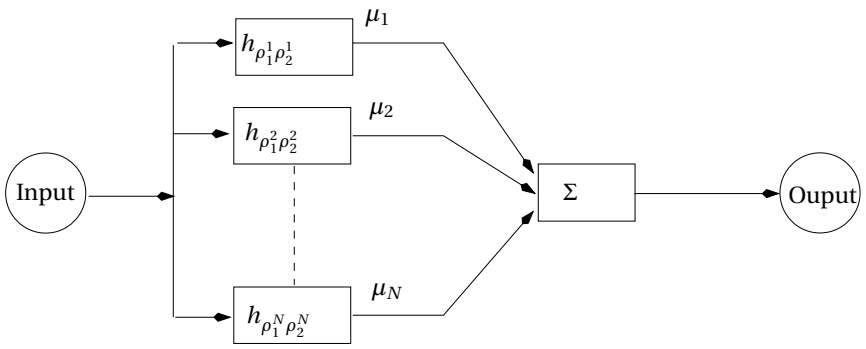


Fig. 2.3. **Representation of a Preisach operator for a discrete measure.** In this case, the Preisach operator can be understood as an electronic device which is made up of switches connected in parallel with the weights μ_i.

2^o/ When the measure μ is a discrete measure $\left(\mu(\rho^i)\right)_{i\in\mathbb{N}}$ on $\mathscr{P}$, the Preisach operator is the input-output system, plotted in figure (Fig. 2.3), which is made up of threshold switches connected in parallel with the weights $\mu(\rho^i)$, *where* ρ_2^i *is the opening threshold and* ρ_1^i *is the closing threshold of the* i*-th switch.*

We are now in position to proof the following Theorem, which provides a new way to compute the damage caused by a sampled loading v; we will see in Proposition 2.1 page 51 how to generalize this procedure to time-continuous signals.

THEOREM 2.1 (Calibration of a Preisach operator adapted to damage computation) *The damage* $\mathscr{D}(v)$ *caused by a sampled loading* $v=(v_i)_{i=0}^N$*, defined in* (1.12) *can be computed with the help of a Preisach operator* (2.10) *of density*

$$\mu(\rho)=-\frac{1}{2}\partial_{12}\left(\frac{1}{N_r(\rho)}\right) \tag{2.11}$$

by the formula

$$\mathscr{D}(v)=V_T(\mathscr{W}_\mu(v)) \tag{2.12}$$

where $N_r(\rho)$ *is the number of cycles to failure for an alternating loading* $\sigma_a=\frac{\rho_2-\rho_1}{2}\geq 0$*, at average* $\sigma_m=\frac{\rho_2+\rho_1}{2}$.

PROOF. The proof makes use of piecewize monotony of the relay and the Preisach operators:

DEFINITION 2.4 (Piecewize monotone operator) An operator $\mathscr{S}: v\mapsto\mathscr{S}(v)$ acting on the space of finite sequences is said to be *piecewize monotone* if it satisfies the following condition[4]

$$([\mathscr{S}(v)]_i-[\mathscr{S}(v)]_{i-1}).(v_i-v_{i-1})\geq 0 \quad \text{for any } i\geq 2 \tag{2.13}$$

[4]Meaning that the operator $\mathscr{S}$ preserves the direction of variation of the input signal. A definition of the concept of monotone operator valid in general normed vector spaces is introduced in footnote n^o 23 page 62.

1^o/ One can check on the diagram in figure (Fig. 2.4) that the relay operator $v \mapsto h_\rho^{per}(v)$ is monotone while the proof of the piecewize monotony of the Preisach operator, which makes use of the geometric properties, is given in the Remark 2.7 page 71.

2^o/ These monotony properties allow to write down the absolute values

$$|\left[\mathscr{W}_\mu(v)\right]_i - \left[\mathscr{W}_\mu(v)\right]_{i-1}| \quad \text{and} \quad \left|[h_\rho^{per}(v)]_i - [h_\rho^{per}(v)]_{i-1}\right|$$

as

$$\left|\left[\mathscr{W}_\mu(v)\right]_i - \left[\mathscr{W}_\mu(v)\right]_{i-1}\right| = \left(\left[\mathscr{W}_\mu(v)\right]_i - \left[\mathscr{W}_\mu(v)\right]_{i-1}\right).\mathrm{sign}(v_i - v_{i-1})$$
$$\left|[h_\rho^{per}(v)]_i - [h_\rho^{per}(v)]_{i-1}\right| = \left([h_\rho^{per}(v)]_i - [h_\rho^{per}(v)]_{i-1}\right).\mathrm{sign}(v_i - v_{i-1})$$

and by definition of the Preisach operator, this leads to[5]

$$\left|\left[\mathscr{W}_\mu(v)\right]_i - \left[\mathscr{W}_\mu(v)\right]_{i-1}\right| = \int\limits_{\mathscr{P}} \left|[h_\rho^{per}(v)]_i - [h_\rho^{per}(v)]_{i-1}\right| \mu(\rho)\, d\rho \quad 1 \le i \le N$$

Adding all these equations, we obtain the following expression

$$V_T(\mathscr{W}_\mu(v)) = \int\limits_{\mathscr{P}} V_T(h_\rho^{per}(v))\, \mu(\rho)\, d\rho$$

for total variation $V_T(\mathscr{W}_\mu(v))$.

3^o/ By virtue of (2.9), the total variation $V_T(\mathscr{W}_\mu(v))$ can then be computed according to the Madelung's pairs of the signal v as

$$\begin{aligned} V_T(\mathscr{W}_\mu(v)) &= 2\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\rho_2}\sum_{\rho_1'\le\rho_1<\rho_2\le\rho_2'} a^{per}(v)[\rho_1',\rho_2']\,\mu(\rho_1,\rho_2)\,d\rho_1\right)d\rho_2 \\ &= 2\sum_{\rho_2'\in\mathbb{R}}\sum_{\rho_1'<\rho_2'} a^{per}(v)[\rho_1',\rho_2']\int_{\rho_1'}^{\rho_2'}\left[\int_{\rho_1'}^{\rho_2}\mu(\rho_1,\rho_2)d\rho_1\right]d\rho_2 \end{aligned} \tag{2.14}$$

Using on one hand the definition of the density μ (which is the twice derivative ∂_{12} of the mapping $\rho \mapsto \Delta(\rho) := \frac{-1}{2\,N_r(\rho)}$) and, on the other hand, the following relationships[6]

$$\partial_1\Delta(\delta,\delta) = \partial_2\Delta(\delta,\delta) = \Delta(\delta,\delta) = 0 \quad \text{for all } \delta \in \mathbb{R}$$

we have

$$\begin{aligned} \int_{\rho_1'}^{\rho_2}\partial_{12}\Delta(\rho)\,d\rho_1 &= \partial_2\Delta(\rho_2,\rho_2) - \partial_2\Delta(\rho_1',\rho_2) \\ &= -\partial_2\Delta(\rho_1',\rho_2) \end{aligned}$$

and at last

$$\begin{aligned} \int_{\rho_1'}^{\rho_2'}\left[\int_{\rho_1'}^{\rho_2}\mu(\rho_1,\rho_2)d\rho_1\right]d\rho_2 &= -\int_{\rho_1'}^{\rho_2'}\partial_2\Delta(\rho_1',\rho_2)\,d\rho_2 = -\Delta(\rho_1',\rho_2') \\ &= \frac{1}{2\,N_r(\rho_1',\rho_2')} \end{aligned}$$

[5]Notice that piecewize monotony allows to permute integral and absolute value signs, which is generally a prohibited operation.

[6]Which reflects the fact that a specimen loaded by a constant stress has an infinite lifetime.

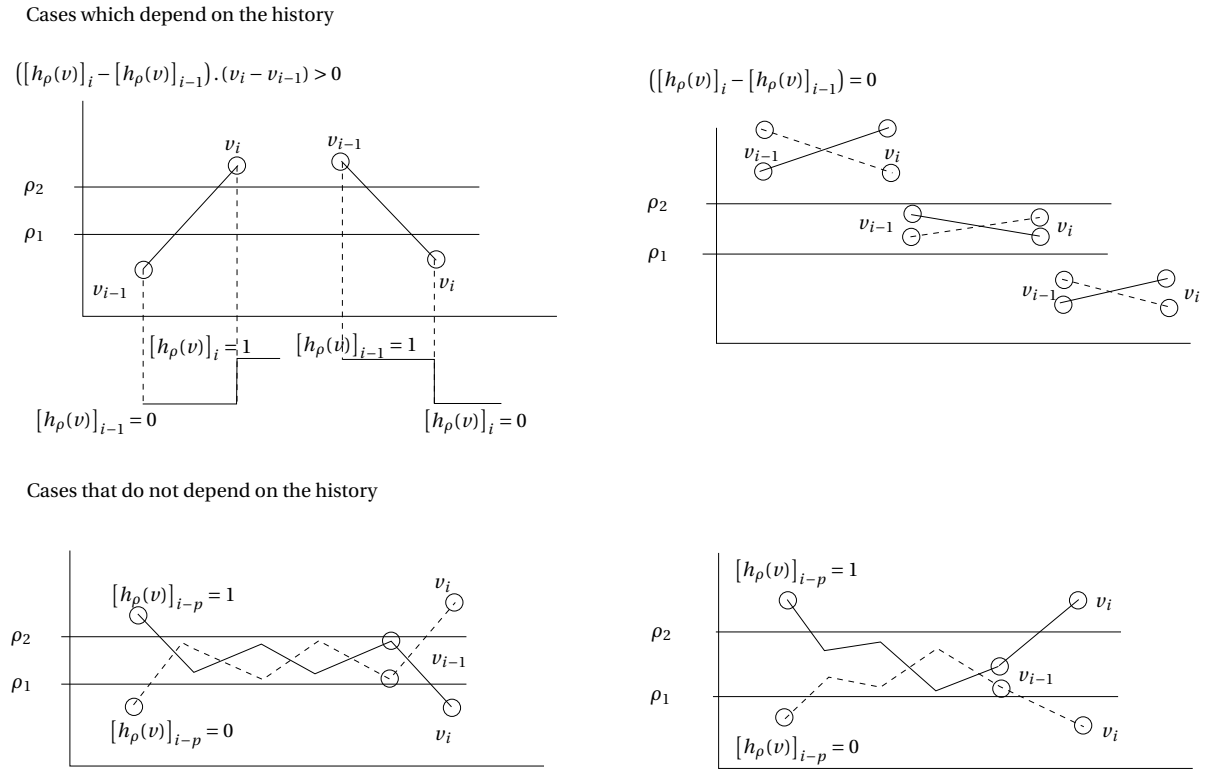


Fig. 2.4. **Proof of the monotony of relay operators.** We must check that $\left([h_\rho(v)]_i - [h_\rho(v)]_{i-1}\right).(v_i - v_{i-1}) \geq 0$; the 12 scenarios plotted in the figure above should be considered. We plots in the column on left the scenarios that lead to $\left([h_\rho(v)]_i - [h_\rho(v)]_{i-1}\right).(v_i - v_{i-1}) > 0$ and in the column on right those which lead to $[h_\rho(v)]_i - [h_\rho(v)]_{i-1} = 0$.

Given what has been said before, the formula (2.12) is obtained in identifying term by term the formulas (2.14) and (1.12) page 25. □

REMARKS 2.2 1^o/ *Representation* (2.12) *of damage requires the computation of the second order derivative* (2.11) *of the inverse of the number of cycles to failure identified from experimental data on Wöhler's curves.* If such a calculation does not ask any question for the parts BC of the curves in figure (Fig. 1.7) page 15, this is not the case for their asymptotic parts CD, where these derivatives may have non-physical singularities when the alternating stress σ_a approaches the fatigue limit σ_d; Examples 2.1 illustrate this situation.

2^o/ Restricting, if needed, $\mathscr{P}$ to a bounded part of the half-plane $\rho_2 \geq \rho_1$, we will assume in the following that $\mu \in L^1(\mathscr{P})$: this is a condition on the rate of pointwise divergence of $|\mu|$ when $\sigma_a = \frac{\rho_2 - \rho_1}{2}$ converges to the fatigue limit σ_d.

EXAMPLES 2.1 1^o/ When the Wöhler's curve is defined by a Stromeyer's formula (1.2) page 16, the density $\mu(\rho_1, \rho_2)$ is

$$\mu_s(\rho_1, \rho_2) = \begin{cases} \frac{(1-b_s)\delta^{\frac{1}{b_s}-2}}{8b_s^2 C_s} & \text{if } \delta \geq 0 \\ 0 & \text{else} \end{cases} \tag{2.15}$$

where

$$\sigma_a = \frac{\rho_2 - \rho_1}{2} \quad \text{and} \quad \delta = \sigma_a - \sigma_d$$

This function is positive if $b_s < 1$, while it is singular on the straight line of equation $\rho_2 - \rho_1 = 2\sigma_d$ when $b_s > \frac{1}{2}$. We see from the Remark 1.2-1/ that this singularity is not physical and means only that the interpolation formula used to define μ misrepresents the asymptotic behavior of the Wöhler's curve[7].

2^o/ Similar computations carried out on the Bastenaire's formula (1.3) give

$$\mu_b(\rho_1,\rho_2) = \begin{cases} \frac{e^{2C\delta}\left(\delta^2 ABC^2 + 2\delta ABC + 2AB\right) + e^{C\delta}\left(\delta(AC)^2 + 2A^2C\right)}{8\left(A - \delta B e^{C\delta}\right)^3} & \text{if } \delta \geq 0 \\ 0 & \text{else} \end{cases} \tag{2.16}$$

3^o/ When the Wöhler's curve is modified by the Goodman (1.5) or by the Soderberg (1.6) formula, to account for mean stress effect, the inverse of the number of cycles to failure can be written as follows[8] if $\sigma_m < R_m$

$$(\rho_1,\rho_2) \mapsto \begin{cases} f_0\left(\frac{R_m\sigma_a}{R_m - \sigma_m} - \sigma_d\right) & \text{if } \frac{R_m\sigma_a}{R_m - \sigma_m} \geq \sigma_d \\ 0 & \text{else} \end{cases} \tag{2.17}$$

where, setting $\sigma_a = \frac{\rho_2 - \rho_2}{2}$ and $\sigma_m = \frac{\rho_2 + \rho_2}{2}$, the mapping f_0 is defined by $f_0(x) = \frac{x^{\frac{1}{b_s}}}{C_s}$ for a Stromeyer's formula and by $f_0(x) = \frac{x}{Ae^{-Cx} - Bx}$ for the Bastenaire's one. The density μ is then defined by

$$\begin{cases} \frac{2R_m^2(R_m - \rho_1)(R_m - \rho_2)}{(2R_m - \rho_1 - \rho_2)^4} f_0''(\ldots) \\ \quad - \frac{R_m(\rho_2 - \rho_1)}{(2R_m - \rho_1 - \rho_2)^3} f_0'(\ldots) \end{cases} \quad \text{if } \frac{R_m\sigma_a}{R_m - \sigma_m} \geq \sigma_d \tag{2.18}$$

$$0 \qquad \text{else}$$

which makes sense for $\sigma_a + \sigma_m \leq R_m$, an example is plotted in figure (Fig. 2.6).

2.2. Generalization to continuous signals

In this Section we will extend the results of Theorem 2.1 to continuous signals. More specifically, we introduce the mathematical framework allowing to generalize the definition of damage $\mathscr{D}(v)$ as the integral (2.28) page 51 for a loading $t \mapsto v(t)$ defined in continuous time. *Two steps of generalization are required to this end.*

1^o/ *The first step consists to generalize to the continuous case the concept of the relay operator* introduced in the formula (2.2) page 39: this is the purpose of the following Definition.

[7] For these analytical formulas it is suggested to choose $b_s \approx \frac{1}{4}$ in code_Aster.

[8] If $\sigma_m \geq R_m$ failure occurs soon as the first loading cycle and the number of cycles to failure is zero.

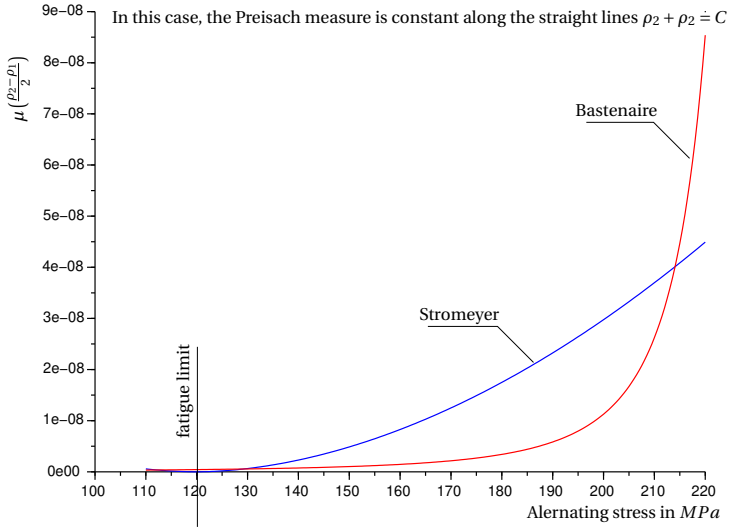


Fig. 2.5. **Examples of Preisach densities μ for the Bastenaire (red curve) and Stromeyer (blue curve) formulas**. This is an illustration of the formulas (2.15) and (2.16) with the coefficients identified in the figure (Fig. 1.8) page 17. This picture shows moreover that when the alternating stress σ_a goes to the fatigue limit σ_d, the density μ associated with the Stromeyer's formula converges more slowly to 0 than the measure associated with the Bastenaire's formula.

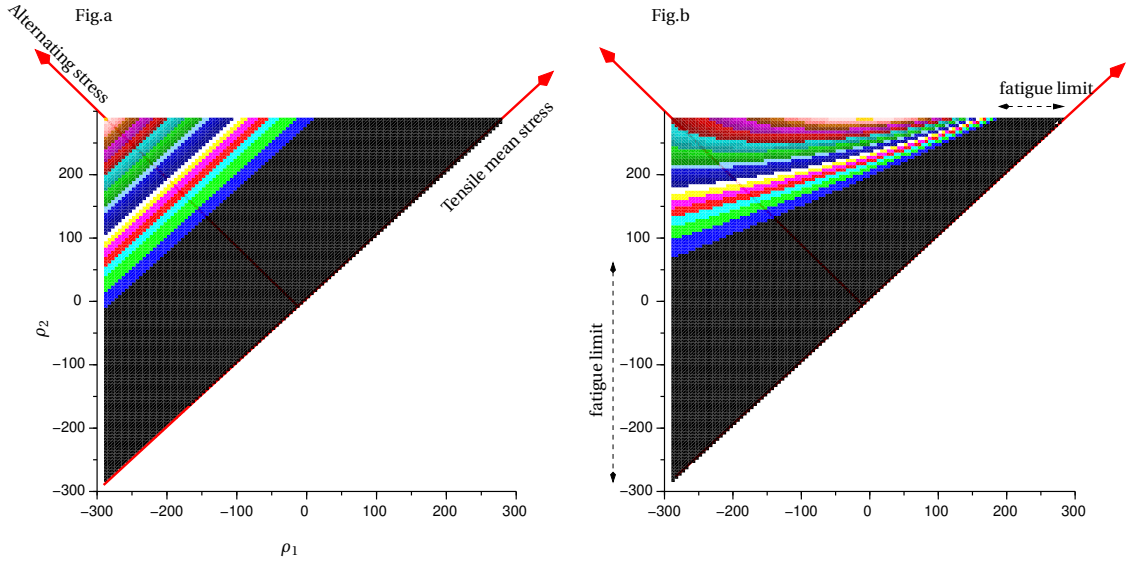


Fig. 2.6. **Iso-values of the Preisach densities μ for Stromeyer formulas.** We represent on the diagram (Fig.a) the iso-values of a Stromeyer's measure μ which does not account for mean stress effect; in this case the measure μ weights identically the two half-planes of equations $\sigma_m > 0$ and $\sigma_m < 0$ and, see figure (Fig. 2.5), is completely defined by its values on the line $\sigma_m = 0$. On the diagram (Fig.b) the Stromeyer's measure depends on the mean stress, in this case the half-plane $\sigma_m > 0$ is much more weighted than the half-plane $\sigma_m < 0$.

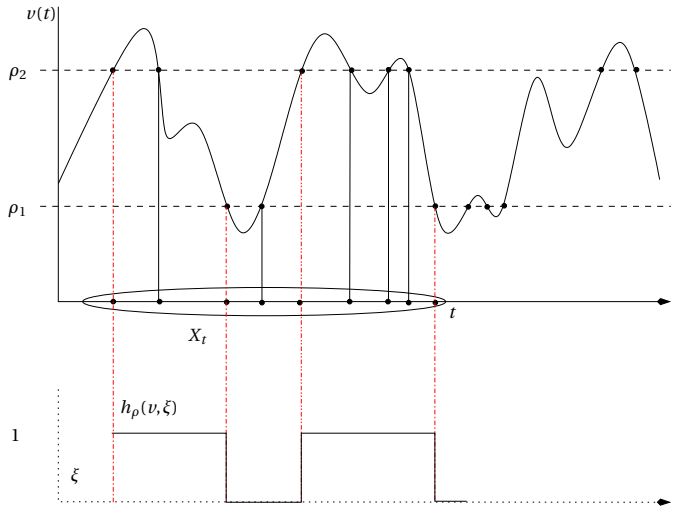


Fig. 2.7. **Definition of the relay filtering for continuous signals**. The set X_t is intended for identifying the times where the relay switches; similarly to the discrete relay, this operation allows to identify the oscillations of v which cross in the interval $[\rho_1, \rho_2]$.

DEFINITION 2.5 (Relay operator for continuous signals) Let v be a continuous numerical mapping defined $[0,T]$, and $\rho_2 > \rho_1$ two thresholds be given; for each $t \in]0,T]$, let's introduce the set

$$X_t = \left\{\tau \in [0,t]\ ;\ v(\tau) = \rho_1 \text{ or } v(\tau) = \rho_2\right\}$$

and define the relay operator $(v,\xi) \in C^0[0,T] \times \{0,1\} \mapsto h_\rho(v,\xi) \in \{0,1,\xi\}$ as follows:

$$h_\rho(v,\xi)(t) = \begin{cases} z_0 & \text{if } X_t = \emptyset \\ 0 & \text{if } X_t \neq \emptyset \text{ and } v(\max X_t) = \rho_1 \\ 1 & \text{if } X_t \neq \emptyset \text{ and } v(\max X_t) = \rho_2 \end{cases} \tag{2.19}$$

where

$$z_0 = \begin{cases} 1 & \text{if } v(0) \geq \rho_2 \\ 0 & \text{if } v(0) \leq \rho_1 \\ \xi & \text{if } \rho_1 < v(0) < \rho_2 \end{cases} \tag{2.20}$$

One can check that this version of relay operator satisfies the following properties:

i) the mapping $t \mapsto z(t) = h_\rho(v,\xi)(t)$ is well defined on $[0,T]$. Indeed:
- assume for instance that $v(0) < \rho_1$, then we have $z_0 = 0$ and $z(t)$ remains 0 as long as $v(t)$ doesn't cross the threshold ρ_2, where it switches to 1 and stays at this value until $v(t)$ crosses the threshold ρ_1 etc.
- the internal variable $\xi \in \{0,1\}$ is intended, see figure (Fig. 2.7), for unambiguously define the state of the relay when $\rho_1 < v(0) < \rho_2$;

ii) if $\lambda_1 \leq v(t) \leq \lambda_2$ for $t \in [t_1,t_2]$ then $t \in [t_1,t_2] \mapsto h_\rho(v,\xi)(t)$ is constant if either one of the following three conditions holds:

$$\begin{array}{ll} (a) & \rho_1 > \lambda_2 \\ (b) & \rho_1 < \lambda_1 \text{ and } \rho_2 > \lambda_2 \\ (c) & \rho_2 < \lambda_1 \end{array}$$

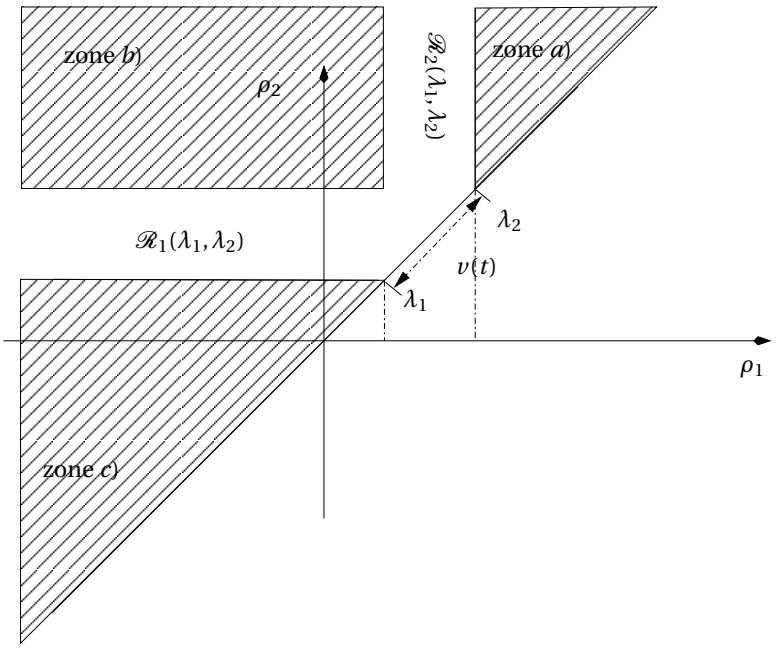


Fig. 2.8. **Partition of the Preisach plane in zones where the relay operators are constant**. If $\lambda_1 \le v(t) \le \lambda_2$ for $t \in [t_1,t_2]$ then the mappings $t \mapsto h_\rho(v,\xi)(t)$ are constant if ρ is in one of the hatched zones, while they vary between 0 and 1 when ρ lies in one of the strips $\mathscr{R}_i(\lambda_1,\lambda_2)$ for $i = 1,2$.

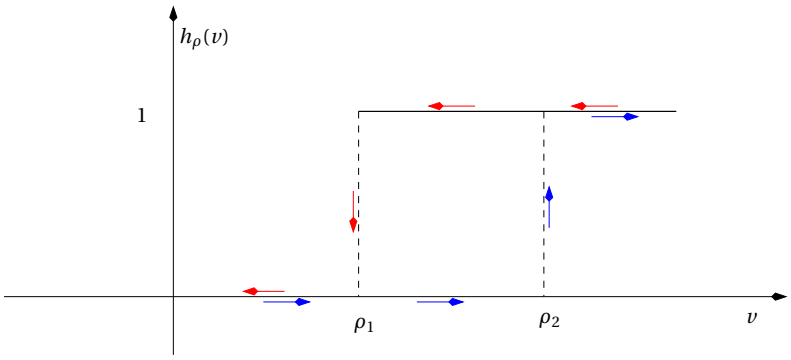


Fig. 2.9. **Hysteresis diagram for the relay operator.** When $v(t)$ goes increasingly from $v(t_0) < \rho_1$ to $v(t_1) > \rho_2$, the relay $h_\rho(v)$ switches between 0 and 1, along the blue arrows; next when v comes back decreasingly from $v(t_1)$ to $v(t_0)$, the state of $h_\rho(v)$ passes from 1 to 0 following the red arrows.

This allows to split up the Preisach plane $\mathscr{P}$ into the four zones described in figure (Fig. 2.8);

iii) the operator $v \mapsto h_\rho(v,\xi)$, which may be plotted in Lissajous' diagram in figure (Fig. 2.9) form a hysteresis loop.

2^o/ The second step consists to *extend as follows the definition of the Preisach operator*:

$$\mathscr{H}_\mu(v) = \int_{\mathscr{P}} h_\rho(v,\xi_\rho)\mu(\rho)d\rho \tag{2.21}$$

where $\rho \mapsto \xi_\rho \in \{0,1\}$ is defined by[9]

$$\xi_\rho = \begin{cases} 0 \text{ if } \rho_1+\rho_2>0 \\ 1 \text{ if } \rho_1+\rho_2<0 \end{cases} \tag{2.23}$$

REMARKS 2.3 1^o/ When v is a continuous piecewise affine function, defined with the help of a sampled sequence $(v(t_i))_{i=0}^N$, the definition (2.21) of a Preisach operator coincides with that which is given in Definition 2.3 in the meaning that the sequences $\left(\mathscr{H}_\mu(v)(t_i)\right)_{i=0}^N$ and (2.10) page 42 are the same if the discrete relay operators (2.2) are initialized as follows:

$$[h_\rho(v)]_{-1} = \begin{cases} 1 & \text{if } \rho_1+\rho_2>0 \\ 0 & \text{if } \rho_1+\rho_2<0 \end{cases}$$

2^o/ The continuous analogous of the "periodic" Preisach operator $\mathscr{W}_\mu$ introduced page 42 can be defined as follows:

- let's associate to a given mapping v defined on $[0,T]$ the mapping v^{per} defined on $[0,2T]$ as

$$v^{per}(t) = \begin{cases} v(t) & \text{if } t\in[0,T] \\ v(t-T) & \text{if } t\in[T,2T] \end{cases} \tag{2.24}$$

- then call $\mathscr{W}_\mu$ the operator which associates to v the following mapping:

$$t\in[0,T] \mapsto \mathscr{W}_\mu(v)(t) := \mathscr{H}_\mu(v^{per})(T+t) \tag{2.25}$$

  one can check that $v \mapsto \mathscr{W}_\mu(v)$ *maps the periodic functions onto periodic fonctions of same period*[10].

The regularity result sated in Proposition 2.1 allows to generalize the definition (2.12) of the damage caused by a smooth enough loading signal $t\in[0,T]\mapsto v(t)$ by one of the

[9] A more general definition consists to introduce an initializing function $\rho\in\mathscr{P}\mapsto\xi(\rho)=\xi_\rho\in\{0,1\}$ of the relay operators and to define the Preisach operator as the mapping

$$(\xi,v)\mapsto\mathscr{H}_\mu(v,\xi)=\int_{\mathscr{P}} h(v,\xi_\rho)\mu(\rho)d\rho \tag{2.22}$$

This definition makes sense only if v and ξ are in appropriate functional vector spaces. Theorem 2.1 says, among other things, that within the framework of fatigue analysis, this sophistication level is not necessary as long as we restrict ourselves to process signals satisfying $v(0)=v(T)$; in the following, we will even assume $v(0)=v(T)=0$. To connect together the definitions (2.21), (2.22) and (2.25) of a Preisach operator, one can show that a specific initialization ξ_v (which depends on v) of the relay operators may be defined so that

$$\mathscr{H}_\mu(v,\xi_v)=\mathscr{W}_\mu(v)$$

The definition (2.21) with the initialization (2.23) of the relays is a convenient manner of speaking because, see Theorem 2.2 page 66, it simplifies the geometric representation of the Preisach operator.

[10] We see in Exercise 2.3 page 76 that due to the initialization phase, this is not the case for the operator $v\mapsto\mathscr{H}_\mu(v)$.

following formulas[11]

$$\boxed{\mathscr{D}(v) = \int_0^T \left|\left(\mathscr{W}_\mu(v)\right)'(t)\right| dt = \int_T^{2T} \left|\left(\mathscr{H}_\mu(v^{per})\right)'(t)\right| dt} \tag{2.28}$$

In the following we will use interchangeably of one or the other of these two formulas to compute the damage generated by a loading $t \in [0,T] \mapsto v(t)$; being understood that if we choose the last one, we will implicitly assume that the argument v of $\mathscr{H}_\mu$ is defined on $[0,2T]$ and satisfies $v(t) = v(T+t)$ for $t \in [0,T]$.

A straightforward extension of the Proposition 2.1 below shows that the Preisach operator $\mathscr{W}_\mu$ is a non-linear operator which maps the space of continuous Lipschitz functions into itself. *The example depicted in figure (Fig. 2.19) page 71 shows that we can't expect more regular outputs even if the inputs are very smooth.* Note moreover that this Proposition is essential to insure well-definiteness of the adjoint equation introduced in Proposition 4.10 page 165.

PROPOSITION 2.1 (Regularity results for the Preisach outputs) *If the density μ is defined by the formula* (2.11) *of Theorem 2.1 page 43 then, restricting if needed the integration domain to a bounded part of the Preisach plane, the Preisach operator $\mathscr{H}_\mu$ maps the Sobolev space $W^{1,1}([0,T],\mathbb{R})$ onto itself.*[12].

PROOF. (can be omitted at first reading) To proof this Proposition we will

i) first show that the image $\mathscr{H}_\mu(v)$ of a continuous mapping v by the Preisach operator is continuous,
ii) and if v is furthermore assumed to be differentiable almost every where on $[0,T]$ and if its derivative $\dot{v}$ is in $L^1([0,T],\mathbb{R})$, it is the same for $\frac{d}{dt}\mathscr{H}_\mu(v)$.

[11] The total variation, defined on the finite sequences by the formula (2.3), can be extended as follows for time-continuous signals defined on $[0,T]$:

DEFINITION 2.6 (Total variation of a mapping taking its values in a normed space) Let $v : t \in [0,T] \to X$ be a mapping defined on $[0,T]$ and taking its values in a normed vector space X then, we call total variation of v the following number (finite or not)

$$V_T(v) = \sup\left\{\sum_{k=1}^{N-1} \|v(t_k) - v(t_{k+1})\|\right\} \tag{2.26}$$

where the sup is taken over all the finite sequences $(t_k)_{k=1}^N$ which start at 0 and end at T. We say that the function v is with bounded variation when the number $V_T(v)$ is finite.

When X is a reflexive normed space, each element $v \in W^{1,1}([0,T],X)$ has a representative $\tilde{v}$ of bounded variation, and:

$$V_T(\tilde{v}) = \int_0^T \left\|\frac{dv}{dt}\right\| dt \tag{2.27}$$

[12] VISINTIN [**40**], theorem 3.10 page 117, shows that this result is true in $W^{1,p}$ for $1 \le p < +\infty$, but this level of generality is not necessary because in the context of fatigue analysis we are only *interested in the "total variation of the outputs of a Preisach" operator and not in the behavior of a Preisach operator.*

We see form the diagram in figure (2.8) that if we set

$$\zeta(\lambda_2-\lambda_1)=\max_{i=1,2}\int\limits_{\mathscr{R}_i(\lambda_1,\lambda_2)}\mu(\rho)d\rho$$

the oscillation[13] $\underset{[t_1,t_2]}{\omega}(\mathscr{H}_\mu(v))$ is bounded above by $\zeta(\underset{[t_1,t_2]}{\omega}(v))$ for $[t_1,t_2]\subset[0,T]$.

As $\zeta(0)=0$, we see that if v is continuous at a time t_1 for instance, the oscillation of $\mathscr{H}_\mu(v)$ at t_1 is zero and thus that $t\mapsto\mathscr{H}_\mu(v)(t)$ is continuous at t_1. This shows that the image by the Preisach operator of a continuous function is continuous[14].

Using the definition (2.11) of μ we have for instance

$$\begin{aligned}\text{meas}\,[\mathscr{R}_1(\lambda_1,\lambda_2)]&=-\frac{1}{2}\int\limits_{\frac{\lambda_1}{\sqrt{2}}}^{\frac{\lambda_2}{\sqrt{2}}}\left(\int\limits_{-\rho_1^{max}}^{\rho_2}\partial_{12}\Delta(\rho_1,\rho_2)d\rho_1\right)d\rho_2\\&=\frac{1}{2}\int\limits_{\frac{\lambda_1}{\sqrt{2}}}^{\frac{\lambda_2}{\sqrt{2}}}\partial_2\Delta(-\rho_1^{max},\rho_2)d\rho_2\\&=\frac{1}{2}\left(\Delta(-\rho_1^{max},\frac{\lambda_2}{\sqrt{2}})-\Delta(-\rho_1^{max},\frac{\lambda_1}{\sqrt{2}})\right)\end{aligned}$$

As we can assume Δ continuously differentiable with the respect of ρ_2; the fact of being restricted to a bounded part of the Preisach plane allows to conclude that there is a positive constant C_1 such that

$$\text{meas}\,[\mathscr{R}_1(\lambda_1,\lambda_2)]\leq C_1(\lambda_2-\lambda_1)$$

A similar computation shows that we can define a positive constant C_2 such that $\text{meas}\,[\mathscr{R}_2(\lambda_1,\lambda_2)]\leq C_2(\lambda_2-\lambda_1)$ and therefore that

$$\zeta(\lambda_2-\lambda_1)\leq C(\lambda_2-\lambda_1)$$

We have proved the inequality:

$$\underset{[t_1,t_2]}{\omega}(\mathscr{H}_\mu(v))\leq C\,\underset{[t_1,t_2]}{\omega}(v)\text{ for all }[t_1,t_2]\subset[0,T]$$

When v is piecewize affine, this inequality shows that

$$\left|\frac{d}{dt}\mathscr{H}_\mu(v)\right|\leq C|\dot{v}|\text{ almost everywhere in }[0,T]$$

and by density, of the continuous piecewize affine functions in the space $W^{1,1}$, that $\frac{d}{dt}\mathscr{H}_\mu(v)\in L^1([0,T],\mathbb{R})$ when $v\in W^{1,1}([0,T],\mathbb{R})$. □

[13]The oscillation of a numerical function defined on interval I is the positive number

$$\underset{I}{\omega}(f)=\sup_{x\in I}|f(x)|-\inf_{x\in I}|f(x)|$$

let x_0 be given, the oscillation of f at x_0 is the limit

$$\underset{x_0}{\omega}(f)=\lim_{h\to 0}\underset{[x_0-h,x_0+h]}{\omega}(f)$$

and f is continuous at x_0 if and only if $\underset{x_0}{\omega}(f)=0$.

[14]To proof this property of the Preisach operator we have used the fact that the straight lines are null sets for the measure $\mu(\rho)d\rho$.

To conveniently use formula (2.28), *it remains to set up a computational method for* $\mathscr{W}_\mu(v)(t)$. This leads us to introduce the geometric representation (2.51) page 66 of the Preisach operator.

2.3. Geometric representation of the Preisach operator

Let a numerical mapping v defined on $[0,T]$ be given; at each time $t \in [0,T]$, the Preisach plane $\mathscr{P}$ is divided into the two complementary zones:

$$C_0(v,t) = \{\rho\ ;\ h_\rho(v,\xi)(t) = 0\} \quad \text{and} \quad C_1(v,t) = \{\rho\ ;\ h_\rho(v,\xi)(t) = 1\}$$

and the output $\mathscr{H}_\mu(v,\xi)(t)$ of the Preisach operator $\mathscr{H}_\mu$ is defined by

$$\mathscr{H}_\mu(v,t) = \int_{C_1(v,t)} \mu(\rho)\, d\rho$$

In order to compute of this integral, want to characterize the boundary $B(v,t)$ *between* C_0 *and* C_1. More specifically, we are intending to proof that

1^o/ the initialization (2.23) of the relays $h_\rho(v,\xi)$ allows to define the boundary $B(v,t)$ as the stair steps diagram plotted in figure (Fig. 2.11) ;

2^o/ the boundary $B(v,t) \subset \mathscr{P}$ is the graph of a numerical mapping defined by a recurrence equation when v is piecewise affine and, by a differential inequality in the general case.

To this end, we introduce the concept of $RMS(v,\tilde{t})$ *sequence (Reduced Memory Sequence) associated with* v at a given time $\tilde{t} \in [0,T]$. This notion permits indeed to identify "the corners of the boundary $B(v,t)$", and we will see moreover that

- the $RMS(v,\tilde{t})$ sequence stores the useful information contained in the history of input signal v to calculate $\mathscr{H}_\mu(v)(\tilde{t})$;
- if u and v are two numerical mappings such that $RMS(u,\tilde{t}) = RMS(v,\tilde{t})$ then $\mathscr{H}_\mu(u,\tilde{t}) = \mathscr{H}_\mu(v,\tilde{t})$.

DEFINITION 2.7 (Of a RMS sequence.) Let a numerical mapping v defined on $[0,T]$ and $\tilde{t} \in [0,T]$ be given. The $RMS(v,\tilde{t})$ sequence associated with v is a sequence $(v(t_i))_{i\in\mathbb{N}}$ of local extrema of v which is defined stepwise as follows:

Let

$$M = \max_{t\in[0,\tilde{t}]} |v(t)| \quad \text{and} \quad \bar{t} = \max\{t \in [0,\tilde{t}]\ ;\ |v(t)| = M\}$$

then, excluding the trivial case $M = 0$, the two following hypotheses are processed independently:

1^o/ If $v(\bar{t}) > 0$, we start the recurrence in setting

$$t_1 = \bar{t}\ ,\ \eta_1 = M \text{ and } \alpha_1 = \min_{t_1 \le t \le \tilde{t}} v(t)$$

As $|\alpha_1| < |\eta_1|$, we can define $t_2 = \max\{t \in [t_1, \tilde{t}]\ ;\ v(t) = \alpha_1\}$ and continue as follows:

i) the procedure stops if $t_2 = \tilde{t}$; else, *define the maximum*

$$\eta_2 = \max_{t_2 \le t \le \tilde{t}} v(t) < \eta_1 \quad \text{and set} \quad t_3 = \max\{t \in [t_2, \tilde{t}]\ ;\ v(t) = \eta_2\}$$

ii) the procedure ends if $t_3 = \tilde{t}$; else *define the manimum*

$$\alpha_2 = \min_{t_3 \le t \le \tilde{t}} v(t) > \alpha_1 \quad \text{and set} \quad t_4 = \max\{t \in [t_3, \tilde{t}]\ ;\ v(t) = \eta_2\}$$

iii) the procedure *continues from step i)* after substitution of t_4 to t_2.

$2^o/$ if $v(\bar{t}) < 0$, define first $t_0 = \bar{t}$ and $\alpha_0 = -M$ then set

$$\eta_1 = \max_{t_0 \le t \le \tilde{t}} v(t) \text{ and } t_1 = \max\{t \in [t_0, \tilde{t}]\ ;\ v(t) = \eta_1\}$$

i) the procedure completes if $t_1 = \tilde{t}$; else *define the minimum*

$$\alpha_1 = \min_{t_1 \le t \le \tilde{t}} v(t) > \alpha_0 \quad \text{and set} \quad t_2 = \max\{t \in [t_1, \tilde{t}]\ ;\ v(t) = \alpha_1\}$$

ii) the procedure completes if $t_2 = \tilde{t}$; else *define the maximum*

$$\eta_2 = \max_{t_2 \le t \le \tilde{t}} v(t) < \eta_1 \quad \text{and set} \quad t_3 = \max\{t \in [t_2, \tilde{t}]\ ;\ v(t) = \eta_2\}$$

iii) the procedure *continues from step i)* after substitution of t_3 to t_1.

We will denote $RMS(v, \tilde{t})$ this sequence.

REMARKS 2.4 $1^o/$ The RMS sequence $\big(v(t_j)\big)_{j\in\mathbb{N}}$ defined above is, see figure (Fig. 2.10), a sequence of extrema of v such that $\big(v(t_{2i})\big)_i$ (resp. such that $\big(v(t_{2i+1})\big)_i$) is an increasing sequence of minima (resp. decreasing sequence of maxima) satisfying the inequalities

$$\begin{aligned} \alpha_1 < \cdots < \alpha_i = v(t_{2i}) < \cdots < v(\tilde{t}) < \\ < \cdots < \eta_i = v(t_{2i-1}) < \cdots < \eta_1 \end{aligned} \tag{2.29}$$

$$[\alpha_{i+1}, \eta_{i+1}] \subset [\alpha_i, \eta_i] \quad \text{for all } i \tag{2.30}$$

$2^o/$ If the sequence $\big(t_j\big)_j$ is endless then, setting $t^* = \sup_{j\in\mathbb{N}} t_j$, the mapping v is constant on the interval $[t^*, \tilde{t}]$ and $\lim_{i\to\infty} \eta_i = \lim_{i\to\infty} \alpha_i = v(t^*) = v(\tilde{t})$.

Now assume that $t \in [0, T]$ is given, the mapping $\rho \in \mathscr{P} \mapsto h_\rho(v, \xi)(t) \in \{0, 1\}$ is defined as follows, with the help of the $RMS(v, t)$ sequence:

$$h_\rho(v,\xi)(t) = \begin{cases} 0 \text{ for } \alpha_{i-1} < \rho_1 \quad \forall \rho_2 > \eta_{i+1} \\ 1 \text{ for } \rho_2 < \eta_{i+1} \quad \forall \rho_1 < \alpha_i \\ \text{stays at its initialization } \xi \text{ if } \rho_2 > \eta_1 \text{ and } \rho_1 < \alpha_0 \end{cases} \tag{2.31}$$

PROOF OF FORMULA (2.31). Let an index $i \ge 0$ be given, we set, see figure (Fig. 2.11):

$$\mathscr{P}_i^1 = \{\rho \in \mathscr{P}\ ;\ \rho_1 < \alpha_{i+1} \text{ and } \rho_2 < \eta_{i+1}\} \quad \text{and} \quad \mathscr{P}_i^0 = \{\rho \in \mathscr{P}\ ;\ \rho_1 > \alpha_i \text{ and } \rho_2 > \eta_{i+1}\}$$

then by definition of the $RMS(v, t)$ sequence,

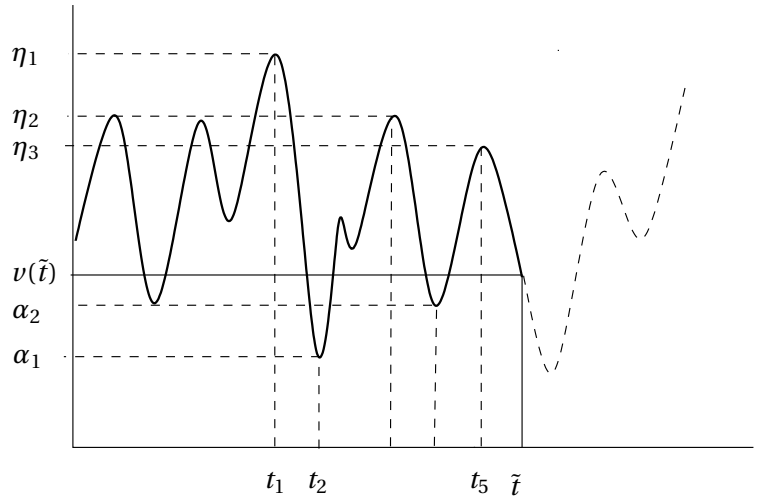


Fig. 2.10. **Definition of a** RMS **sequence**. It is a sequence made up of local extrema $\eta_1, \alpha_1, \cdots, \eta_n, \alpha_n \cdots$, listed so that the sequence $(\eta_i)_i$ is a decreasing sequence of maxima and $(\alpha_i)_i$ is an increasing sequence of minima.

- v reaches the minimum $\alpha_i = v(t_{2i})$ and the maximum $\eta_{i+1} = v(t_{2i+1})$ while $v(\tilde{t}) \in]\alpha_i, \eta_{i+1}[$ for $t_{2i+1} < \tilde{t} \leq t$. Thus if $\rho \in \mathscr{P}_i^1$, the state of the relays $h_\rho(v,\xi)(\tilde{t})$ remains blocked at $h_\rho(v,\xi)(t_{2i+1})$, which is 1, see the rectangle in blue in figure (Fig. 2.11);
- in the same manner the relays $h_\rho(v,\xi)(\tilde{t})$ remain blocked at $h_\rho(v,\xi)(t_{2i}) = 0$ when $\rho \in \mathscr{P}_i^0$, rectangle in red in the figure.

All of this allows to unambiguously define the mapping $\rho \mapsto h_\rho(v,\xi)(t)$ for ρ in the reunion $\bigcup_{i\geq 0}\left(\mathscr{P}_i^1 \cup \mathscr{P}_i^0\right)$. As α_0 and η_1 are respectively the absolute minimum and the absolute maximum of v on the interval $[0,t]$, the relays $h_\rho(v,\xi)(.)$ stay at their initialization states when $\rho_1 < \alpha_0$ and $\rho_2 > \eta_1$. □

Having initialized each relay operator $h_\rho(v,\xi)$ by the formula (2.23) page 50,

The Preisach plane $\mathscr{P}$ is split up into two complementary zones

$$C_0(v,t) = \left\{\rho\ ;\ h_\rho(v)(t) = 0\right\} \quad \text{and} \quad C_1(v,t) = \left\{\rho\ ;\ h_\rho(v)(t) = 1\right\}$$

and the boundary $B(v,t)$ between C_0 and C_1 is depicted by *the "stairs diagram" or " turning points" in figure (Fig. 2.11)*, with potentially an infinite number of stairs near the diagonal $\rho_1 = \rho_2$, if the sequence $RMS(v,t)$ is endless.

We prefer, see figure (Fig. 2.12), plot the diagram in figure (Fig. 2.11) in the physical axis

$$\sigma_a = \frac{\rho_2 - \rho_1}{2} \geq 0 \quad \text{and} \quad \sigma_m = \frac{\rho_2 + \rho_1}{2}$$

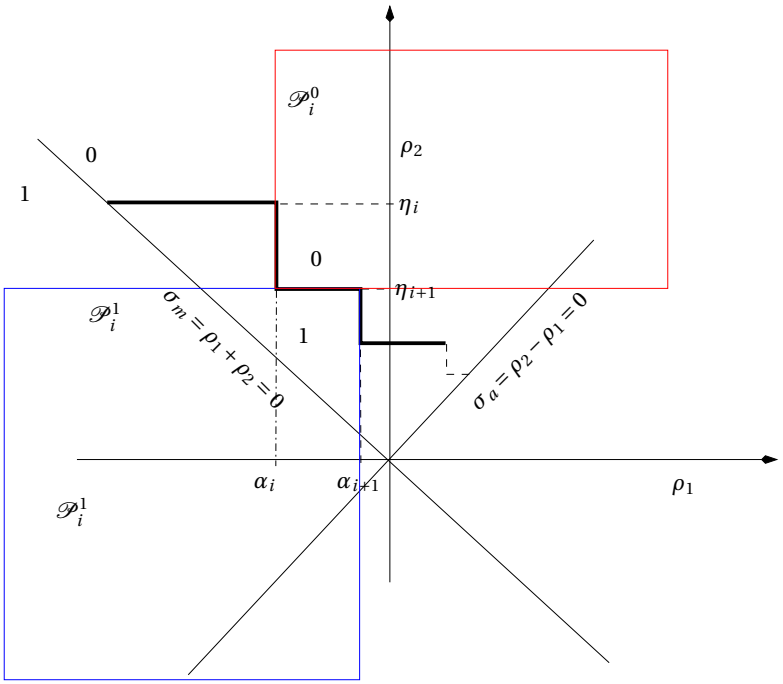


Fig. 2.11. **Boundary between the states** 0 **and** 1 **of the relay outputs in the Preisach plane**. In the plane (ρ_1, ρ_2),the boundary between the zone where the relay operator is 1 and the one where it is 0 is a broken line which consists of segments parallel to the axes, defined by the points of the $RMS(v,t)$ sequence. It may converge to the first diagonal of the Preisach plane with an infinite number of stair treads, for "pathological" signals having an infinite number of oscillations in a finite time interval: this is the case for $v(t) = (t-t_0)^2 \sin \frac{1}{t-t_0}$ if $t \neq t_0$ and $v(t_0) = 0$. It encounters the second diagonal of the plane at the initialization values of the relay operators.

The following Proposition, which is implicitly used by BROKATE [**7**] and proofed practically in the same manner by KRASNOSEL'SKII [**19**], allows to describe the boundary $B(v,t)$ as the graph of a function defined by the recurrence equation (2.33), when the processed signal $t \mapsto v(t)$ is piecewise affine.

PROPOSITION 2.2 *Let v be a continuous piecewise affine function defined on $[0,T]$ and $\tilde{t} \in [0,T]$ be given, then:*

- *there is an increasing sequence $\left(t_j\right)_{j=0}^N$ such that $\tilde{t} = t_N$ and $v(t)$ is monotonous between $v(t_i)$ and $v(t_{i+1})$;*

Fig.a Fig.b

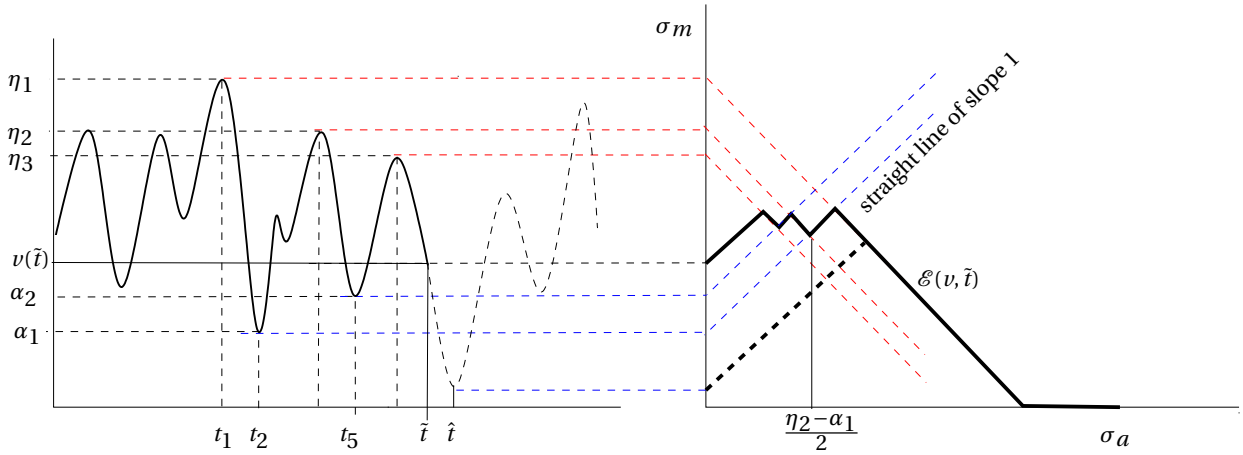


Fig. 2.12. **Turning points in the physical axis** (σ_a, σ_m)**.** For convenience reading reasons, we will prefer draw the boundary $B(v,t)$ in the $\left(\frac{\rho_1-\rho_2}{2}, \frac{\rho_1+\rho_2}{2}\right)$ coordinates rather than in the initial one. The graph in bold in the figure Fig.b represents the memory of the relay operator (or of the Preisach operator) it changes over the time: for instance, at time $\hat{t}$, it is depicted by the dotted line in bold, as at this time, the function reaches a global minimum, all the previous turning points are erased from the operator's memory. *Within the framework of fatigue analysis, this means that the counting operation has found all the "sub-cycles"of a cycle and that it starts a new main cycle.*

- *introducing the piecewise affine function defined at each sample t_i by the recurrence equation*[15]

$$\begin{aligned} &\mathscr{E}_{\sigma_a}(v,t_i) = \min\left[\, v(t_i)+\sigma_a, \max\left(v(t_i)-\sigma_a, \mathscr{E}_{\sigma_a}(v,t_{i-1})\right)\right] \\ &\mathscr{E}_{\sigma_a}(v,0) = 0 \end{aligned} \tag{2.33}$$

the boundary $B(v,t_N)$ at time $t_N=\tilde{t}$ may be characterized as the graph[16] *of the mapping $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,t_N)$.*

PROOF. In this particular case the $RMS(v,\tilde{t})$ sequence is a finite sequence which satisfies

$$\alpha_1 < \alpha_2 < \cdots < \alpha_j < \cdots < v(\tilde{t}) < \cdots < \eta_j < \cdots < \eta_2 < \eta_1$$

and such that $\eta_{i+1}-\alpha_{i+1} < \eta_i - \alpha_i$ for any i. To proof the Proposition, we have to proof that

[15] This equation is the compact form of the formula

$$\mathscr{E}_{\sigma_a}(v,t_i) = \begin{cases} \mathscr{E}_{\sigma_a}(v,t_{i-1}) & \text{if } \mathscr{E}_{\sigma_a}(v,t_{i-1}) \in [v(t_i)-\sigma_a, v(t_i)+\sigma_a] \\ v(t_i)-\sigma_a & \text{if } \mathscr{E}_{\sigma_a}(v,t_{i-1}) < v(t_i)-\sigma_a \\ v(t_i)+\sigma_a & \text{if } \mathscr{E}_{\sigma_a}(v,t_{i-1}) > v(t_i)+\sigma_a \end{cases} \tag{2.32}$$

which is the projection of $\mathscr{E}_{\sigma_a}(v,t_{i-1})$ onto the interval $[v(t_i)-\sigma_a,\, v(t_i)+\sigma_a]$.

[16] ie. The subset $\left\{\left(\sigma_a, \mathscr{E}_{\sigma_a}(v,t_N)\right)\,;\, \sigma_a > 0\right\}$ of the half plane $\sigma_a \geq 0$.

i) the "turning points" are caught by the recurrence equation (2.33) or, in other words that

$$\mathscr{E}_{\frac{\eta_i-\alpha_i}{2}}(v,t_N)=\frac{\eta_i+\alpha_i}{2}\quad\text{and}\quad\mathscr{E}_{\frac{\eta_{i+1}-\alpha_i}{2}}(v,t_N)=\frac{\eta_{i+1}+\alpha_i}{2}\text{ for any }i$$

ii) and that the mapping $\sigma_a\mapsto\mathscr{E}_{\sigma_a}(v,t_N)$ is affine between two consecutive "turning points".

The property i) demonstrated in this first step of the proof is a consequence of the definition of the $RMS(v,t)$ sequence and of the hypothesis v is piecewize affine. Under these assumptions, we can indeed find a point t_k (resp. a point $t_h\geq t_k$) of the subdivision such that $v(t_k)=\eta_i$ (resp. such that $v(t_h)=\alpha_i$).

1^o/ Assume that $\sigma_a=\frac{\eta_i-\alpha_i}{2}$, then we have

$$\mathscr{E}_{\sigma_a}(v,t_k)=\min\left(\frac{3\eta_i-\alpha_i}{2},\max\left(\frac{\eta_i+\alpha_i}{2},\mathscr{E}_{\sigma_a}(v,t_{k-1})\right)\right)\tag{2.34}$$

we intend to proof that $\mathscr{E}_{\sigma_a}(v,t_{k-1})\leq\frac{\eta_i+\alpha_i}{2}$ and that the mapping $t\mapsto\mathscr{E}_{\sigma_a}(v,t)$ stays constant for $t\geq t_k$.

The hypothesis $\mathscr{E}_{\sigma_a}(v,t_{k-1})>v(t_k)+\sigma_a$, which entails $v(t_{k-1})>\eta_i$, contradicts the fact that η_i is a local maximum of v at t_k, and can't occur. It remains thus to discuss the two following cases to compute $\mathscr{E}_{\sigma_a}(v,t_N)$:

a) if $\mathscr{E}_{\sigma_a}(v,t_{k-1})<v(t_k)-\sigma_a$ then $\mathscr{E}_{\sigma_a}(v,t)$ takes the value

$$v(t_k)-\sigma_a=\frac{\eta_i+\alpha_i}{2}$$

for $t=t_k$ and remains constant for $t\geq t_k$. This is proved by contradiction: If it were not the case, we could define $\bar{t}>t_k$ such that $\mathscr{E}_{\sigma_a}(v,t_k)$ should be either greater than $v(\bar{t})+\sigma_a$ or lower than $v(\bar{t})-\sigma_a$;

- the first case would imply, by (2.32), that $v(t_k)-2\sigma_a>v(\bar{t})$, or $v(\bar{t})<\alpha_i$ and would contradict the fact that α_i is the smallest of the local minima of v which are reached after t_k ;
- in the same way, the second case would imply the existence of a local maximum of v greater than $v(t_k)$ reached after t_k.

b) if $v(t_k)+\sigma_a\geq\mathscr{E}_{\sigma_a}(v,t_{k-1})\geq v(t_k)-\sigma_a$, the sequence $\left(\mathscr{E}_{\sigma_a}(v,t_j)\right)_{t_j=t_k}^{t_h}$ is stationary until meting a minimum $v(t_i)$ of v such that

$$\mathscr{E}_{\sigma_a}(v,t_j)>v(t_i)+\sigma_a$$

where it switches to $v(t_i)+\sigma_a$. As by definition of a RMS sequence, these minima are greater than $v(t_h)$, the last value reached by $\mathscr{E}_{\sigma_a}(v,t_j)$ is $v(t_h)+\sigma_a$,which is precisely $\frac{\alpha_i+\eta_i}{2}$. Then the same stationarity argument as the one previously invoked allows to conclude $\mathscr{E}_{\sigma_a}(v,t_N)=\frac{\eta_i+\alpha_i}{2}$.

2^o/ When $\sigma_a=\frac{\eta_{i+1}-\alpha_i}{2}$ we have

$$\mathscr{E}_{\sigma_a}(v,t_h)=\min\left(\frac{\eta_{i+1}+\alpha_i}{2},\max\left(\frac{3\alpha_i-\eta_{i+1}}{2},\mathscr{E}_{\sigma_a}(v,t_{h-1})\right)\right)$$

and we show that $\mathscr{E}_{\sigma_a}(v,t_N)=\frac{\eta_{i+1}+\alpha_i}{2}$ in the same way as before[17].

Proof of ii), let $\sigma_a>0$ be given, denote by

$$E_{\sigma_a}=\left\{t\in[0,\tilde{t}]\,;\,\mathscr{E}_{\sigma_a}(v,t)=v(t)\pm\sigma_a\right\}$$
$$\text{where } \bar{t}=\max E_{\sigma_a} \text{ when this set is not empty}$$

If the set E_{σ_a} is empty then $v(t)-\sigma_a<\mathscr{E}_{\sigma_a}(v,t)<v(t)+\sigma_a$ for any t in $[0,\tilde{t}]$. The recurrence equation (2.33) shows that these inequalities entail that the sequence $\left(\mathscr{E}_{\sigma_a}(v,t_i)\right)_{i=1}^N$ is identically 0 and we have $v(t)-\sigma_a<0<v(t)+\sigma_a$ for $t\in[0,\tilde{t}]$, which implies $\sigma_a>\max_{t\in[0,\tilde{t}]}|v(t)|$.

When $\sigma_a\le\max_{t\in[0,\tilde{t}]}|v(t)|$, the set E_{σ_a} is non-empty, $\bar{t}$ is well defined, the mapping $t\mapsto\mathscr{E}_{\sigma_a}(v,t)$ is stationary for $t\ge\bar{t}$ and satisfies the inequalities

$$v(t)-\sigma_a\le\mathscr{E}_{\sigma_a}(v,\bar{t})\le v(t)+\sigma_a\quad\text{for all}\quad t\ge\bar{t}$$

- if $\mathscr{E}_{\sigma_a}(v,\bar{t})=v(\bar{t})-\sigma_a$ the we have

  $$v(t)-\sigma_a\le v(\bar{t})-\sigma_a\le v(t)+\sigma_a \text{ for all } t\ge\bar{t}$$

  the definition of $\bar{t}$ and the first inequality shows that $v(\bar{t})$ is a maximum η_i in the $RMS(v,\tilde{t})$ sequence while the second one shows that we necessarily have $\sigma_a\ge\frac{v(\bar{t})-v(t)}{2}$ for all $t\in[\bar{t},\tilde{t}]$, and this means that $\sigma_a\ge\frac{\eta_i-\alpha_i}{2}$;
- we show in the same way that $\mathscr{E}_{\sigma_a}(v,\bar{t})=v(\bar{t})+\sigma_a$ entails that $v(\bar{t})$ is a minimum α_i in the $RMS(v,\tilde{t})$ sequence and that $\sigma_a\ge\frac{\eta_{i-1}-\alpha_i}{2}$.

This shows that the mapping $\sigma_a\mapsto\mathscr{E}_{\sigma_a}(v,\tilde{t})$ is decreasing, of slope $-\frac{1}{2}$ when σ_a is located on the right of $\frac{\eta_i-\alpha_i}{2}$, while it is increasing, of slope $\frac{1}{2}$ for σ_a is on the right of $\frac{\eta_{i-1}-\alpha_i}{2}$. □

We are now in position to generalize the results obtained so far to Lipschitz continuous signals $v\in W^{1,1}([0,T],\mathbb{R})$. To this end, *the recurrence equation* (2.33) *must be replaced by the differential inequality* (2.39), parameterized in $\sigma_a\ge0$.

Generalization to Lipschitz continuous signals. To define this extension, we notice that the mapping $t\mapsto\mathscr{E}_{\sigma_a}(v,t)$ introduced in the Proposition 2.2, which make sense for for $t\mapsto v(t)$ piecewize affine satisfies the following inequalities:

$$\begin{aligned}&v(t)-\sigma_a\le\mathscr{E}_{\sigma_a}(v,t)\le v(t)+\sigma_a \text{ for any } t\\ &\text{and } \dot{\mathscr{E}}_{\sigma_a}(v,t)=0 \text{ if } \mathscr{E}_{\sigma_a}(v,t)\in]v(t)-\sigma_a,v(t)+\sigma_a[\end{aligned}\tag{2.35}$$

[17] Now it is the hypothesis $\mathscr{E}_{\sigma_a}(v,t_{h-1})<v(t_h)-\sigma_a$ which need not be examined : indeed, it implies that $u(t_{h-1})<v(t_h)$ and contradicts the fact that α_i is a local minimum of v. Then a reasoning by contradiction shows that the case $\mathscr{E}_{\sigma_a}(v,t_{h-1})>v(t_h)+\sigma_a$ leads to

$$\mathscr{E}_{\sigma_a}(v,t_j)=v(t_h)+\sigma_a=\frac{\eta_{i+1}+\alpha_i}{2}\text{ for } j\ge h$$

At last, when

$$v(t_h)+\sigma_a\ge\mathscr{E}_{\sigma_a}(v,t_{h-1})[\sigma_a]\ge v(t_h)-\sigma_a$$

the sequence $\left(\mathscr{E}_{\sigma_a}(v,t_j)\right)_{j\ge h-1}$ remains constant until that v reaches a maximum $v(t_i)$ satisfying $v(t_i)-\sigma_a>\mathscr{E}_{\sigma_a}(v,t_j)$, where it switches to $v(t_i)-\sigma_a$. As η_{i+1} is the largest of these maxima, we have $\mathscr{E}_{\sigma_a}(v,t_N)=\frac{\eta_{i+1}+\alpha_i}{2}$.

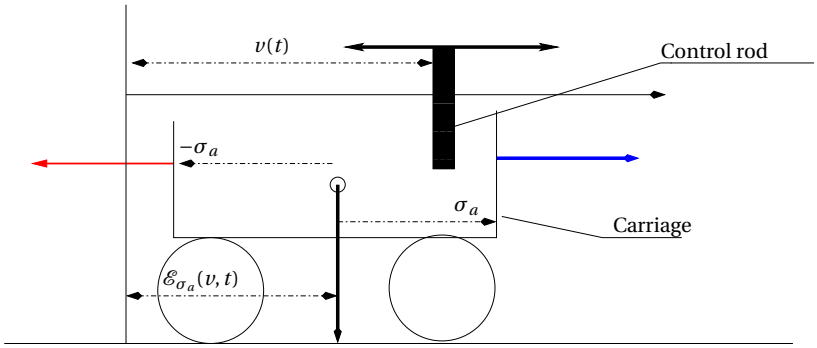


Fig. 2.13. **Mechanical representation of the play operator**. We can "see the solution $\mathscr{E}_{\sigma_a}(v,t)$ of the equation (2.39)" as the displacement of the center of the carriage when this one is pushed on its sides by a control rod; for instance, when the rod reaches the right hand edge of the carriage, it pushes the carriage toward the right if $v(t)$ is increasing but has no effect if $v(t)$ is decreasing. Moreover, the rod has no effect on the carriage when it is located between the two edges etc.

Then, if $I_{[-\sigma_a,\sigma_a]}$ is the characteristic function [18] of the interval $[-\sigma_a,\sigma_a]$, defined by

$$I_{[-\sigma_a,\sigma_a]} = \begin{cases} 0 & \text{if } x \in [-\sigma_a,\sigma_a] \\ +\infty & \text{else} \end{cases} \tag{2.36}$$

the inequalities (2.35) can be rewritten as the following variational inequality[19]

$$\begin{aligned} &\dot{\mathscr{E}}(t)\,(x-(\mathscr{E}(t)-v(t))) \geq I_{[-\sigma_a,\sigma_a]}(\mathscr{E}(t)-v(t)) - I_{[-\sigma_a,\sigma_a]}(x) \\ &\text{for all } x \in [-\sigma_a,\sigma_a] \end{aligned} \tag{2.37}$$

where, to simplify the notations, the dependency of $\mathscr{E}_{\sigma_a}(v,t)$ in σ_a and v is omitted.

PROOF. As the implication (2.35) $\Rightarrow$ (2.37) is straightforward , we have to proof that (2.37) entails (2.35). Let t be given, as the product $\dot{\mathscr{E}}(t)\,(x-(\mathscr{E}(t)-v(t)))$ is a finite number, we have $I_{[-\sigma_a,\sigma_a]}(\mathscr{E}(t)-v(t)) < +\infty$, or in other words:

$$\mathscr{E}(t)-v(t) \in [-\sigma_a,\sigma_a]$$

Then, if $-\sigma_a < \mathscr{E}(t)-v(t) < \sigma_a$, the relation

$$\dot{\mathscr{E}}(t)\,(x-(\mathscr{E}(t)-v(t))) \geq 0 \quad \forall\, x \in [-\sigma_a,\sigma_a]$$

implies $\dot{\mathscr{E}}(t)=0$.

If $\mathscr{E}(t)-v(t)=\sigma_a$ (resp. if $\mathscr{E}(t)-v(t)=-\sigma_a$) then $x-(\mathscr{E}(t)-v(t))$ is negative (resp. is positive) for all $x \in [-\sigma_a,\sigma_a]$ and the condition (2.37) entails $\dot{\mathscr{E}}(t)$ negative (resp. positive). Thus the condition (2.37) makes $\mathscr{E}(t)$ pointing toward the interior of $[v(t)-\sigma_a, v(t)+\sigma_a]$ but doesn't define the velocity $\dot{\mathscr{E}}(t)$. This result is somehow "moral" because the picture in figure (Fig. 2.13) shows that this velocity must depend on the direction of variation of $v(t)$. □

[18] Which is convex and lower semi-continuous (ie. such that the inverse image of an interval $[a,+\infty[$ is closed).

[19] Which make sense for any sufficiently regular numerical mapping $t \mapsto v(t)$.

If $v \in \mathbb{R} \mapsto A_{\sigma_a}(v) = \partial I_{[-\sigma_a,\sigma_a]}(v) \in 2^{\mathbb{R}}$ is the sub-differential[20] at v of the characteristic function of the interval $[-\sigma_a,\sigma_a]$, the variational inequality (2.37) can be written as the differential inequality

$$\boxed{\begin{array}{l}\dot{\mathscr{E}}(t) \in -A_{\sigma_a}(\mathscr{E}(t) - v(t)) \text{ for } t \in [0,T] \\ \text{with the initial condition } \mathscr{E}(0) = 0\end{array}} \tag{2.39}$$

and we have the following results:

1^o/ A straightforward computation shows that A_{σ_a} is defined by

$$A_{\sigma_a} v \underset{def}{=} \partial I_{[-\sigma_a,\sigma_a]}(v) = \begin{cases} \emptyset & \text{if } |v| > \sigma_a \\ \mathbb{R}^- & \text{if } v = -\sigma_a \\ 0 & \text{si } v \in]-\sigma_a,\sigma_a[\\ \mathbb{R}^+ & \text{if } v = \sigma_a \end{cases} \tag{2.40}$$

when $\sigma_a \neq 0$, and as follows if $\sigma_a = 0$

$$A_0 v \underset{def}{=} \partial I_0(v) = \begin{cases} \emptyset & \text{if } v \neq 0 \\ \mathbb{R} & \text{if } v = 0 \end{cases} \tag{2.41}$$

2^o/ Noticing that the mapping $u \in [-\sigma_a,\sigma_a] \mapsto u + hA_{\sigma_a} u \in \mathbb{R}$ is onto and that its right inverse is[21]

$$y \mapsto \begin{cases} \sigma_a & \text{if } y \geq \sigma_a \\ y & \text{if } y \in [-\sigma_a,\sigma_a] \\ -\sigma_a & \text{if } y \leq -\sigma_a \end{cases}$$

the equation

$$\frac{\mathscr{E}_h(t+h) - \mathscr{E}_h(t)}{h} + A_{\sigma_a}(\mathscr{E}_h(t+h) - v(t+h)) \ni 0$$

which is obtained in discretizing the equation (2.39) by the backward Euler method can be solved as follows

$$\mathscr{E}_h(t+h) = \begin{cases} v(t+h) + \sigma_a & \text{if } \mathscr{E}_h(t) \geq \sigma_a + v_{t+h} \\ \mathscr{E}_h(t) & \text{if } \mathscr{E}_h(t) \in]v(t+h) - \sigma_a, v(t+h) + \sigma_a[\\ v(t+h) - \sigma_a & \text{if } \mathscr{E}(t) \leq v(t+h) - \sigma_a \end{cases} \tag{2.42}$$

and leads to interpolate the solution of the equation (2.39) by a recurrence equation identical to (2.33). *The equation* (2.39) *is thus a natural extension to the continuous time case of the recurrence equation* (2.33).

---

[20] The sub-differential $\partial f(x_0)$ at x_0 of a convex numerical mapping f, defined on a normed space E, is the convex hull of the affine minorant of f at x_0. Thus a linear functional $\xi \in E^*$ is in the sub-differential $\partial f(x_0)$ if and only if

$$f(x) - f(x_0) \geq < \xi, x - x_0 > \text{ for any } x \tag{2.38}$$

where $< .,. >$ denote the duality bracket between E and E^*, which associates to a pair $(\xi, x) \in E^* \times E$ the value of ξ on the vector x. The convex mapping f is differentiable at x_0 if and only if its sub-differentail $\partial f(x_0)$ reduces to a point which is then the derivative of f at x_0. *While the definition given here is very general, it is sufficient for the moment to think E as the vector space of the real numbers, identified to its dual, and to understand the duality bracket as the ordinary product on* $\mathbb{R}$.

[21] This mapping is the resolvant of the operator A_{σ_a}, in this case, it does not depend on h because, by definition, $A_{\sigma_a}(u)$ takes arbitrary positive (resp. negative) value when u is σ_a (resp. $-\sigma_a$). BREZIS [**5**] denote J_h the resolvant of A_{σ_a} and calls Yosida regularization of A_{σ_a} the mapping $\frac{1}{h}(I - J_h)$ plotted in dotted line in the diagram on left in the figure (Fig. 2.14).

Now we proof that *the condition* $v \in W^{1,1}([0,T],\mathbb{R})$ *is a sufficient condition which ensures an existence and a regularity result for the equation* (2.39)[22].

If we set

$$\mathscr{G}(t) = \mathscr{E}(t) - v(t)$$

the equation (2.39) is equivalent to

$$\begin{cases} \dot{\mathscr{G}} + A_{\sigma_a}(\mathscr{G}) \ni -\dot{v} \\ \mathscr{G}(0) = v_0 \end{cases} \tag{2.43}$$

and we are intending to prove an existence result for this differential equation when the right hand member $t \mapsto \dot{v}(t)$ is in the space $L^1([0,T],\mathbb{R})$. This is the purpose of the following Proposition.

PROPOSITION 2.3 *Let be given* $v \in W^{1,1}([0,T],\mathbb{R})$ *and* v_0 *in the domain of* A_σ*, ie. such that* $A_{\sigma_a}(v_0) \neq \emptyset$*, then the equation* (2.43) *has an unique solution which is in* $W^{1,1}([0,T],\mathbb{R})$.

PROOF. This Proposition is special case of a result demonstrated by BREZIS [**5**] (proposition 3.8 page 82) which aims at establishing existence and uniqueness results for a differential equation of form

$$\frac{du}{dt} + A\,u \ni f \quad u(0) = u_0 \tag{2.44}$$

defined on a Hilbert space H, where:

- the mapping $u \in H \mapsto A\,u \in 2^H$ is a maximal monotone operator[23]
- and $f \in L^1([0,T],H)$[24].

The proof given by BREZIS is carried out within two steps:

1^o/ The first step consists (see [**5**] theorem 3.4 page 65) to show that the equation (2.44) has a weak solution in the sense of the Definition 2.8;

2^o/ and the second one aims at showing (see proposition 3.8 page 82) that if H is a finite dimensional space, the weak solution obtained in the first step is actually a strong solution.

[22]This will show that the algorithm (2.42) has a chance to converge to a result which makes sense when the time step size h goes to 0.

[23]Let H be a Hilbert space, for scalar product $<.;.>$. A mapping $A : u \in H \mapsto Au \in 2^H$ is said to be monotone if

$$\langle y_1 - y_2\,;\, u_1 - u_2\rangle \geq 0 \quad \text{for all} \quad y_1 \in A\,u_1 \text{ and } y_2 \in A\,u_2$$

As the set of monotone operators is ordered by the inclusion, we say that A is maximal monotone if it is maximal with respect to this ordering (that is: if for every monotone operator B on H, the relation $B \supset A$ entails $B = A$. When $u \mapsto \varphi(u)$ is a convex numerical mapping defined on H, the mapping $u \in H \mapsto \partial\varphi(u) \in 2^H$ is monotone; it is maximal monotone if φ is proper (not identically equal to $+\infty$) and lower semi-continuous. This is the case for the characteristic function of a closed convex subset of H.

[24]Space of the H-valued functions $t \mapsto f(t)$ defined on $[0,T]$, such that the norm $t \mapsto \|f(t)\|_H$ is integrable and satisfies $\int_0^T \|f(t)\|_H < +\infty$.

DEFINITIONS 2.8 (Strong and weak solutions for the equation (2.44)) Let $f \in L^1([0,T],H)$ be given, we call *strong solution for the equation* $\dot{u} + Au \ni f$ any mapping u in the space $C^0([0,T],H)$ witch is moreover absolutely continuous[25] on any compact subset of $]0,T[$ and satisfies the differential inclusion

$$\frac{du}{dt}(t) + Au(t) \ni f(t) \quad \text{almost every where on } [0,T]$$

We say that $u \in C^0([0,T],H)$ is *a weak solution for the equation* $\dot{u} + Au \ni f$ if their are two sequences $\left(f_n\right)_n \subset L^1([0,T],H)$ and $(u_n)_n \subset C^0([0,T],H)$ such that:

i) the sequence f_n converges to f in $L^1([0,T],H)$,
ii) for all n, the mapping u_n is a strong solution for the equation $\dot{u}_n + Au_n \ni f_n$,
iii) and the sequence $(u_n)_n$ converges uniformly to u on $[0,T]$.

□

REMARKS 2.5 1^o/ Existence of a weak solution for the equation (2.43) is a straightforward consequence of the Proposition 2.2 (ie. which can be obtained without invoking the theorem 3.4 of BREZIS). Indeed, if v is in the space $W^{1,1}([0,T],\mathbb{R})$ there is a sequence $\left(g_n\right)_n$ of step functions which converges to $\dot{v}$ in $L^1([0,T],\mathbb{R})$ and we can define the sequence $(v_n)_n$ of piecewise affine functions such that $\dot{v}_n = g_n$. From Proposition 2.2 and discussion of the first part of this Section, the following sequence:

$$\mathscr{G}_n(t) = \mathscr{E}_{\sigma_a}(v_n,t) - v_n(t)$$

is a sequence of strong solutions of (2.43) such that $\mathscr{G}_n(0) = v_n(0) = v_0$. Inequality (2.47) shows that

$$|\mathscr{G}_n(t) - \mathscr{G}_m(t)| \leq \int_0^t \left|g_n(s) - g_m(s)\right| ds \quad \text{for all } t \in [0,T]$$

thus the sequence $(\mathscr{G}_n)_n$ converges uniformly to a continuous function $\mathscr{G}$, which is then, by definition, a weak solution of (2.43).

If $(\mathscr{G}_n)_n$ and $(\tilde{\mathscr{G}}_n)_n$ are two sequences of strong solutions of (2.43) with the right hand members g_n and $\tilde{g}_n$ respectively, they define the weak solutions $\mathscr{G}$ and $\tilde{\mathscr{G}}$, the inequality (2.47) shows that:

$$\left|\mathscr{G}_n(t) - \tilde{\mathscr{G}}(t)\right| \leq \left|\mathscr{G}_n(0) - \tilde{\mathscr{G}}(0)\right| + \int_0^t \left|g_n(s) - \tilde{g}_n(s)\right| ds \tag{2.45}$$

Since, by hypothesis, the initial conditions $\mathscr{G}_n(0)$ and $\tilde{\mathscr{G}}(0)$ (resp. the sequences g_n and $\tilde{g}_n$) converge (resp. converge in $L^1([0,T],\mathbb{R})$) to the same limit we have (in passing to the limit in the inequality (2.45)) $\lim_{n\to\infty} \left\|\mathscr{G}_n - \tilde{\mathscr{G}}\right\|_{C^0([0,T],\mathbb{R})} = 0$, and this entails that $\mathscr{G} = \tilde{\mathscr{G}}$.

[25] Such a function is differentiable almost every where on $]0,T[$ and satisfies

$$u(\tilde{t}) - u(\delta) = \int_\delta^{\tilde{t}} \mathscr{G}'(t)dt \quad \text{for all} \quad \delta, \tilde{t} \in]0,T[$$

2^o/ *The arguments used above to define the weak solution for the equation* (2.43) *is a consistency result for the algorithm* (2.33) because they allow to proof that the operator $v \mapsto \mathscr{E}_{\sigma_a}(v)$, defined in Proposition 2.2 for v piecewise affine can be extended to the functions $v \in W^{1,1}([0,T],\mathbb{R})$ in passing to the uniform limit and setting

$$\mathscr{E}_{\sigma_a}(v) = v + \mathscr{G}(\dot{v}) \quad \text{where } \mathscr{G}(\dot{v}) \text{ is the solution of the equation (2.43)}$$

when $v \in W^{1,1}([0,T],\mathbb{R})$.

3^o/ VISINTIN [**40**] extends the definition of the operator $v \mapsto \mathscr{E}_{\sigma_a}(v)$ by a continuity argument together with the density of the piecewize affine function in the space $W^{1,p}([0,T],\mathbb{R})$. Doing so, he obtains an operator which takes its values in $C^0([0,T],\mathbb{R})$. This condition is not sufficient to ensure that the graph of the function

$$\sigma_a \in \mathbb{R}^+ \mapsto \mathscr{E}_{\sigma_a}(v,t) \in \mathbb{R}$$

represents the boundary $B(v,t)$ defined page 53.

LEMMA 2.1 *Let f and g in $L^1([0,T],\mathbb{R})$ be given, denoting by u and v two strong solutions of the equations*

$$\frac{du}{dt} + A_{\sigma_a}(u) \ni f \quad \textit{and} \quad \frac{dv}{dt} + A_{\sigma_a}(v) \ni g \tag{2.46}$$

then we have:

$$|u(t) - v(t)| \le |u(s) - v(s)| + \int_s^t \big|f(x) - g(x)\big|\, dx \quad \textit{for all} \quad 0 \le s \le t \le T \tag{2.47}$$

PROOF. Subtracting the equations (2.46) member to member and multiplying the obtained result by $(u-v)$, we have

$$\frac{d(u-v)^2}{dt} + (A_{\sigma_a}(u) - A_{\sigma_a}(v))(u-v) \ni (f-g)(u-v)$$

As $A_{\sigma_a}(u)$ is the sub-differential at u of a convex function, the characterization given in the footnote n^o 20 page 61 shows that $(A_{\sigma_a}(u) - A_{\sigma_a}(v))(u-v) \ge 0$ and the above equation may be rewritten as

$$\frac{d(u-v)^2}{dt}(t) \le (f(t)-g(t))(u(t)-v(t)) \le \big|f(t)-g(t)\big|\,|u(t)-v(t)|$$

for all $t \in [0,T]$. Integrating this inequality between s and t and using the fact that the mapping $x \mapsto (u(x)-v(x))^2$ is absolutely continuous, we have

$$(u(t)-v(t))^2 - (u(s)-v(s))^2 \le \int_s^t \big|f(x)-g(x)\big|\,|u(x)-v(x)|\, dx \quad \text{for } 0 \le s \le t \le T$$

To infer (2.47) from this last inequality, it remains to apply lemma A.5 of BREZIS [**5**] page 157, reproduced below for reading convenience. □

LEMMA 2.2 *Let $m \in L^1([0,T],\mathbb{R})$ such that $m \ge 0$ p.p. on $]0,T[$ and a a positive constant be given. Let φ be a continuous numerical mapping defined on $[0,T]$ such that:*

$$\frac{1}{2}\varphi^2(t) \le \frac{1}{2}a^2 + \int_0^t m(s)\varphi(s)\, ds \quad \textit{for all} \quad t \in [0,T]$$

then we have:

$$\left|\varphi(t)\right| \leq a + \int_0^t m(s)\,ds \quad \textit{for all} \quad t \in [0,T]$$

Numerical treatment of the variational inequality. Another way to solve numerically the differential equation (2.39) or (2.43) is to replace the graph A_{σ_a} by its Yosida regularization (see footnote 21 page 61) and to integrate the obtained differential equation by an implicit method in time, which is unconditionally stable.

In the one-dimensional case, this amounts to replace the vertical bold lines in figure (Fig. 2.14) by the dotted ones[26]. In other words, we must solve the following differential equation.

$$\begin{aligned} &\dot{\mathscr{E}}(t) + F_k(\mathscr{E}(t) - v(t)) = 0 \text{ on } [0,T] \\ &\text{with the initial condition } \mathscr{E}(0) = 0 \end{aligned} \tag{2.48}$$

Where, denoting k the slope of the dashed line[27], the mapping $x \mapsto F_k(x)$ is defined by

$$F_k(x) = \begin{cases} k(x+\sigma_a) & \text{if } x \leq -\sigma_a \\ 0 & \text{if } -\sigma_a \leq x \leq \sigma_a \\ k(x-\sigma_a) & \text{if } x \geq \sigma_a \end{cases}$$

And the implicit Euler method leads to solve the following equation:

$$\mathscr{E}(t+h) = -h\,F_k\left(\mathscr{E}(t+h) - v(t+h)\right) + \mathscr{E}(t) \tag{2.49}$$

where h is the discretization step size of the approximation equation (2.48). Equation (2.49) can be solved as follows:

- setting

$$y_1 = \frac{\mathscr{E}(t) + hk\left(v(t+h) + \sigma_a\right)}{1+hk} \qquad y_2 = \frac{\mathscr{E}(t) - hk\left(\sigma_a - v(t+h)\right)}{1+hk}$$

- the output $\mathscr{E}(t+h)$ is defined by

$$\begin{aligned} &\mathscr{E}(t+h) = y_2 && \text{if} \quad y_2 \leq v(t+h) - \sigma_a \\ &\mathscr{E}(t+h) = y_1 && \text{if} \quad y_1 \geq v(t+h) + \sigma_a \\ &\mathscr{E}(t+h) = \mathscr{E}(v,t) && \text{if} \quad v(t+h) - \sigma_a < y_2 \leq y_1 < \sigma_a + v(t+h) \end{aligned} \tag{2.50}$$

REMARKS 2.6 1^o/ The algorithm (2.50) and the recurrence equation (2.32) are two discretization methods for the continuous equation (2.39). The recurrence equation (2.32) is obtained in explicitly solving the variational inequality (2.37) while the algorithm (2.50) consists to solve a regularized version of (2.37) in which we have substituted the functional J_k depicted in figure (Fig. 2.14) (which is differentiable) to the characteristic function of the convex $[-\sigma_a, \sigma_a]$. *Ability to explicitly solve* (2.37) *is specific to the dimension* 1 and in the general case, *the numerical treatments of* (2.37) *are carried out after regularization.*

[26] Using the analogy introduced in figure (Fig. 2.13), this is the same as linking rod and center of the carriage by a soft spring while the rod does not reach the edges and by a very stiff spring to simulate the contact between rod and carriage.

[27] Which is intended for going to $+\infty$.

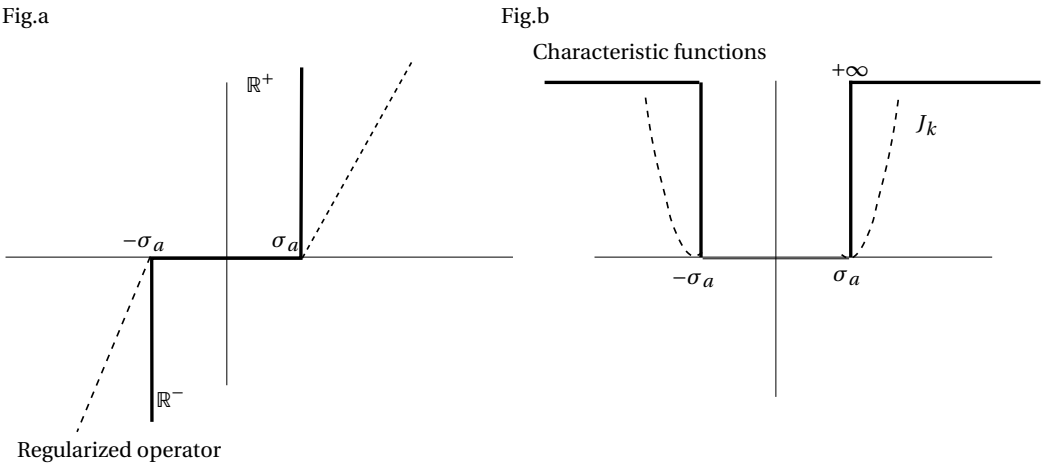


Fig. 2.14. **Mathematical definitions of the play operator.** It is the set-valued graph A_{σ_a} plotted in bold lines on the diagram Fig.a, which can be regularized by the function in dotted lines. These operators are the sub-differential of the mappings plotted on the diagram in Fig.b; the mapping φ_k, which regularizes the characteristic function $I_{[-\sigma_a,\sigma_a]}$ being differentiable, its sub-differential is an single-valued mapping.

2^o/ As the mapping $x \mapsto F_k(x)$ is not differentiable, the solution of the differential equation (2.48) doesn't have the required regularity properties to efficiently use an integration scheme of order higher than 1.

3^o/ We can make the algorithm (2.50) dimensionless with respect to the time step size h by setting $k = \frac{1}{h}$. We then obtain an algorithm which only depends on the discretization step size of the signal v on $[0, T]$ and converges to the solution of the continuous model when the number of samples increases.

The above results justify the following geometric representation of the Preisach operator.

THEOREM 2.2 *Denoting by $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,t)$ the mapping defined by the recurrence equation* (2.33) *of the Proposition 2.2 or as the solution of the differential inequality* (2.39)*, whose the graph is the boundary $B(v,t)$ between the zones of the Preisach plane where the relays are* 1 *and that where they are* 0*, defined page 53, the outputs of Preisach operator are defined for each $t \in [0,T]$ by following double integral*

$$\mathscr{H}_\mu(v,t) = 2\int_0^{+\infty}\left[\int_{-\infty}^{\mathscr{E}_{\sigma_a}(v,t)} \mu(\sigma_m - \sigma_a, \sigma_m + \sigma_a)\, d\sigma_m\right] d\sigma_a \tag{2.51}$$

where μ is the Preisach measure[28] *defined by the formula* (2.11) *page 43*

[28] In the context of fatigue analysis, the support of the measure μ can be supposed to be compact, and the bounds $\pm\infty$ are only intending for simplifying the notations. We will explain in Chapter 3 a way to define this support.

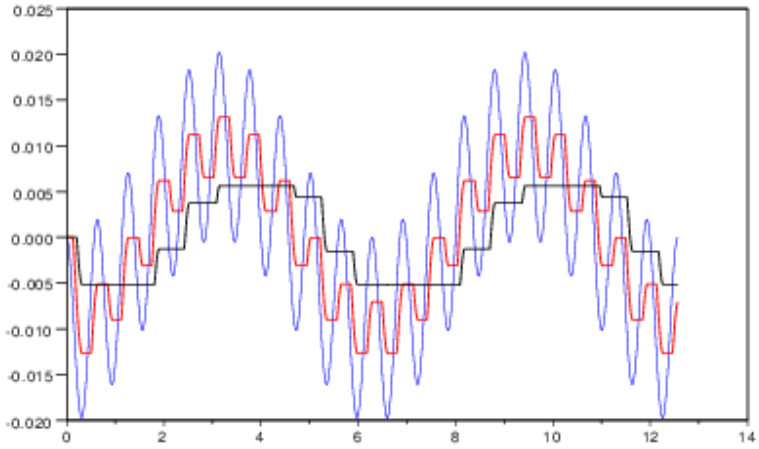


Fig. 2.15. **Representation of $t \mapsto \mathscr{E}(v,t,\sigma_a)$ (bold lines) for some values of σ_a.** The curve in blue is the input signal; the filtering effect increases with σ_a. Further note that regularization of equation (2.39) rounds the outputs, which are theoretically rectangular.

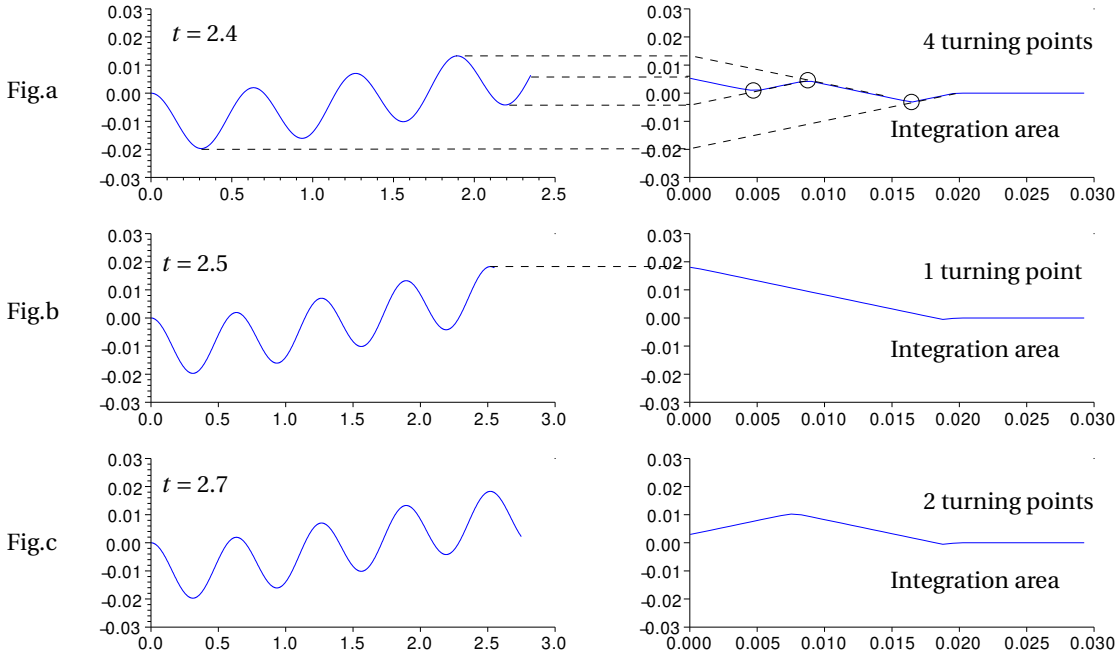


Fig. 2.16. **Representation of $\sigma_a \mapsto \mathscr{E}(v,t,\sigma_a)$ at t given.** Comparison between the figures Fig.a and Fig.b shows that all or part of the memory of the Preisach operator is erased when the signal reaches an extremum. We will see that this phenomenon causes the non-differentiability of the outputs of the Preisach operator with respect to the inputs.

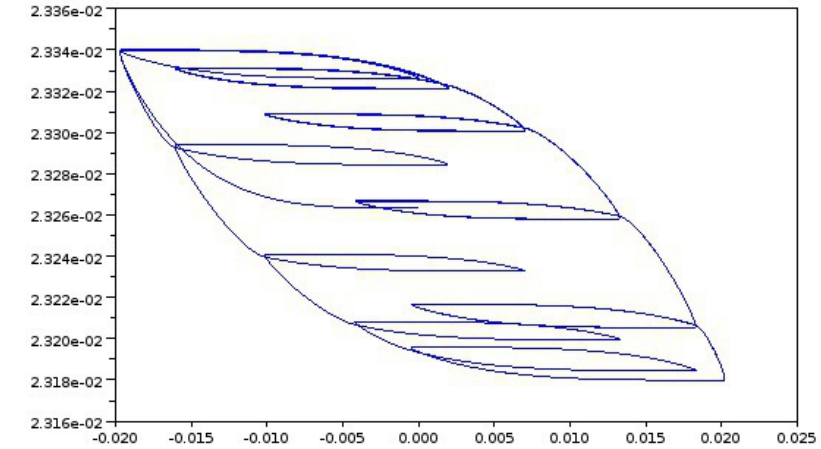


Fig. 2.17. **Hysteresis loops of the outputs of a Preisach operator.** The computation doesn't account for mean stress effect, this explains why the hysteresis loops are geometrically similar to each other. A hysteresis loop closes once a cycle has been browsed.

The algorithm (2.50) is used to compute the "turning points" (see figure (Fig. 2.16)) of the signal defined in figure (Fig. 2.15) and deduce, by integration in the Preisach plane, the outputs of the Preisach operator (see Theorem 2.2), which are plotted in function of the input signal as the Lissajous' diagram plotted in figure (Fig. 2.17).

As they require the computation of the twice integral (2.51), the values of $t \mapsto \mathscr{H}(v,t)$ are extremely expensive to sample; *it is therefore unreasonable to approach the total variation of* $\mathscr{H}(v)$ *by the formula* (2.26) *page 51 and this justifies the developments carried out in the next Section.*

2.4. Damage accumulation for Lipschitz continuous loadings

We have proved in Section 2.2 that for a smooth enough loading $t \mapsto v(t)$, the damage might be computed as the total variation of the image of v by the periodic Preisach operator, or in other words, by one of the equivalent formulas:

$$\mathscr{D}(v) = \int_T^{2T} \left| \frac{d\,\mathscr{H}_\mu(v^{per})}{dt} \right| dt = \int_0^T \left| \frac{d\,\mathscr{W}_\mu(v)}{dt} \right| dt$$

For the homogeneity of the presentation, we will use the first formula. To simplify the notations we will moreover replace v^{per} *by* v *and assume, if necessary, that this mapping is defined on* $[0,2T]$ *and satisfies*

$$v(t) = v(t+T) \quad \text{for any } t \in [0,T] \tag{2.52}$$

Let a time $\tilde{t}$ be given, the question is to define a procedure to compute the derivative $\frac{d\,\mathscr{H}_\mu(v)}{dt}(\tilde{t})$.

Using the representation (2.51) of the Preisach operator, we see that this reduces to the computation of the limit

$$\lim_{h\to 0}\frac{1}{h}\int_0^{+\infty}\left[\int_{\mathscr{E}_{\sigma_a}(v,\tilde{t})}^{\mathscr{E}_{\sigma_a}(v,\tilde{t}+h)}\mu(\sigma_m-\sigma_a,\sigma_m+\sigma_a)\,d\sigma_m\right]d\sigma_a \tag{2.53}$$

where $\mathscr{E}_{\sigma_a}(v,\tilde{t})$ is the solution at $t=\tilde{t}$ of the variational inequality (2.37) or (2.39), parameterized in $\sigma_a\geq 0$.

1^o/ Assume that $\dot{v}(\tilde{t})\neq 0$, then setting $v(\tilde{t}):=v$, we have to calculate the variation of surface below the curve $\sigma_a\mapsto\mathscr{E}_{\sigma_a}(v,\tilde{t})$ for perturbations $\delta v:=h\,\dot{v}(\tilde{t})$ of v; and *this leads to calculate the derivative with respect to v of the surfaces $a_i(v)$ of the hatched triangles plotted in figure* (Fig. 2.18).

$1^o.a$/ If $\dot{v}(\tilde{t})\geq 0$, the surface $a_1(v)$ is defined by

$$a_1(v)=2\int_0^{\tilde{\sigma}_a(v)}\left[\int_{\alpha_n+\sigma_a}^{v-\sigma_a}\mu(\sigma_m-\sigma_a,\sigma_m+\sigma_a)d\sigma_m\right]d\sigma_a$$

thus, using the Leibniz formula[29], we have:

$$\frac{da_1}{dv}=2\int_0^{\tilde{\sigma}_a(v)}\frac{\partial}{\partial v}\left[\int_{\alpha_n+\sigma_a}^{v-\sigma_a}\mu(\sigma_m-\sigma_a,\sigma_m+\sigma_a)d\sigma_m\right]d\sigma_a \tag{2.54a}$$

$$+2\frac{\partial\tilde{\sigma}_a(v)}{\partial v}\int_{\alpha_n+\tilde{\sigma}_a}^{v-\tilde{\sigma}_a}\mu(\sigma_m-\tilde{\sigma_a},\sigma_m+\tilde{\sigma_a})d\sigma_m \tag{2.54b}$$

The relationship $\alpha_n+\tilde{\sigma}_a=v-\tilde{\sigma}_a$ shows that (2.54b) is zero. To compute (2.54a), we note that

$$\frac{\partial}{\partial v}\left[\int_{\alpha_n+\sigma_a}^{v-\sigma_a}\mu(\sigma_m-\sigma_a,\sigma_m+\sigma_a)d\sigma_m\right] \tag{2.55a}$$

$$=\int_{\alpha_n+\sigma_a}^{v-\sigma_a}\frac{\partial}{\partial v}\mu(\sigma_m-\sigma_a,\sigma_m+\sigma_a)d\sigma_m \tag{2.55b}$$

$$+\frac{\partial(v-\sigma_a)}{\partial v}\mu(v-2\sigma_a,v) \tag{2.55c}$$

As μ doesn't depend on v the term (2.55b) is zero, thus (2.55a) reduces to (2.55c), which is actually $\mu(v-2\sigma_a,v)$. Reporting this value in equation (2.54a) we conclude that

$$\frac{da_1}{dv}=2\int_0^{\tilde{\sigma}_a(v)}\mu(v-2\sigma_a,v)d\sigma_a \tag{2.56}$$

[29] Let $f(x,t)$ be a function such that both $f(x,t)$ and its partial derivative $\partial_x f(x,t)$ are continuous in t and x. Suppose on the other hand that $a(x)$ and $b(x)$ are continuous and have continuous derivatives, then:

$$\frac{d}{dx}\left[\int_{a(x)}^{b(x)}f(x,t)dt\right]=\int_{a(x)}^{b(x)}\partial_x f(x,t)dt+f(x,b(x))b'(x)-f(x,a(x))a'(x)$$

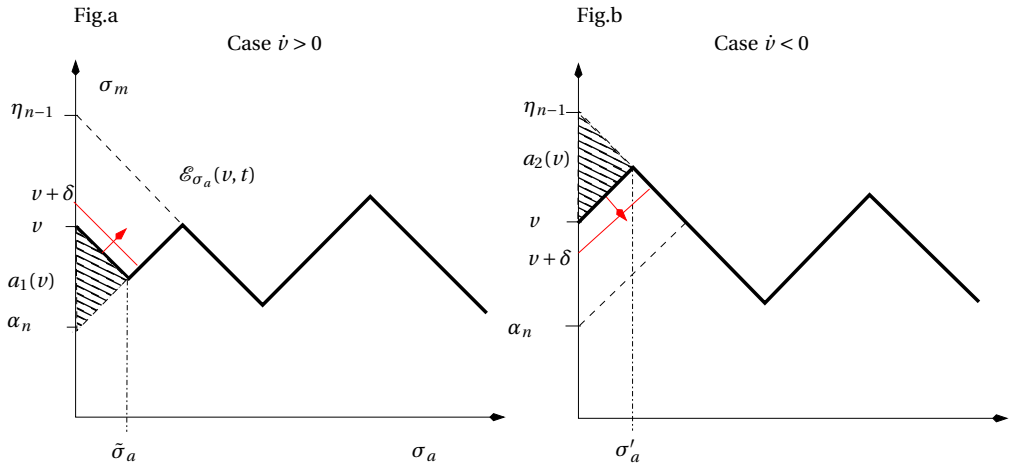


Fig. 2.18. **Evolution of the graph $\left(\sigma_a, \mathscr{E}_{\sigma_a}(v,\tilde{t})\right)$ for a perturbation $h\,\dot{v}(\tilde{t})$ of $\mathscr{E}_0(v,\tilde{t})$.** With the notations of the Definition 2.7, v is bounded by the local extrema α_n and η_{n-1} of the RMS sequence. When $\dot{v}(\tilde{t}) > 0$, the variations of boundary follow straight lines parallel to the second diagonal and the variation $\delta\mathscr{W}_\mu(v)$ can be evaluated in computing the variation $\delta a_1(v)$ of the surface of the hashed triangle in figure Fig.a. In the same way (see figure Fig.b), when $\dot{v}(\tilde{t}) < 0$, variations of boundary are parallel to the first diagonal and $\delta\mathscr{W}_\mu(v)$ is $-\delta a_2(v)$.

$1^o.b/$ When $\dot{v}(\tilde{t}) \le 0$, we show in the same way that

$$\frac{d\,a_2}{dv} = 2\int_0^{\sigma'_a(v)} \mu(v, v+2\sigma_a)d\sigma_a \tag{2.57}$$

where $\sigma'_a(v)$ is defined by $\eta_{n-1} - \sigma'_a(v) = v + \sigma'_a(v)$.

These two formulas show that

$$\frac{d\mathscr{H}_\mu(v)}{dt}(\tilde{t}) = \begin{cases} 2\,\dot{v}(\tilde{t}) \displaystyle\int_0^{\sigma_0(v)} \mu(v-2\sigma_a, v)d\sigma_a & \text{si } \dot{v}(\tilde{t}) \ge 0 \\ 2\,\dot{v}(\tilde{t}) \displaystyle\int_0^{\sigma_0(v)} \mu(v, v+2\sigma_a)d\sigma_a & \text{si } \dot{v}(\tilde{t}) \le 0 \end{cases} \tag{2.58}$$

where $\sigma_0(v)$ is the abscissa of the first extremum of the mapping $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,\tilde{t})$: It is a minimum when $\dot{v} \ge 0$ and a maximum when $\dot{v} \le 0$.

$2^o/$ When $\dot{v}(\tilde{t}) = 0$, one of the two terms α_n or η_{n-1} of the $RMS(v,\tilde{t})$ sequence is v and we have

$$\sigma'_a(v) = 0 \text{ when } v \text{ is a maximum}$$
$$\tilde{\sigma}_a(v) = 0 \text{ when } v \text{ is a minimum}$$

Form the formulas (2.56) and (2.57), this means that the derivatives $\frac{\partial\mathscr{H}_\mu(v,\tilde{t})}{\partial v(\tilde{t})}$ is not defined when $v(\tilde{t})$ is a local extremum of $t \mapsto v(t)$; this situation is illustrated in the figure (Fig. 2.19).

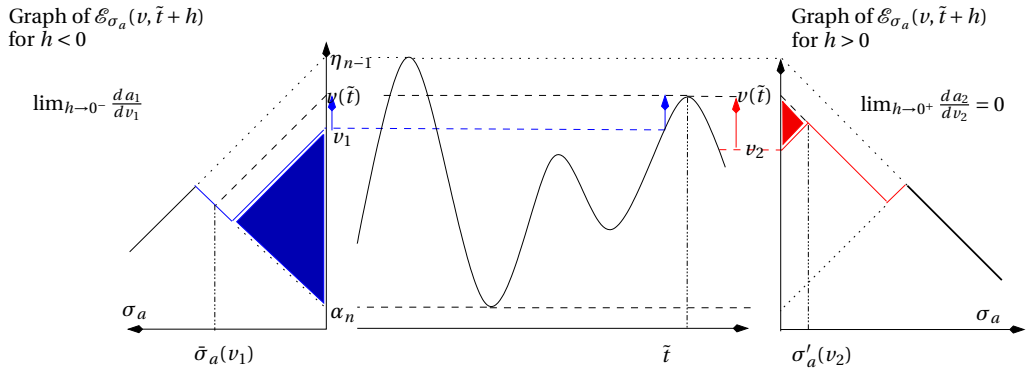


Fig.a Fig.b Fig.c

Fig. 2.19. **Evolution of the graph of $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,t)$ when t varies in a neighborhood of a point $\tilde{t}$ where $v(\tilde{t})$ is a maximum**. When t converges to $\tilde{t}$ by lower values (figure Fig.b), the graph $\big(\sigma_a, \mathscr{E}_{\sigma_a}(v,t)\big)$ (in figure Fig.a) converges to the graph $\big(\sigma_a, \mathscr{E}_{\sigma_a}(v,\tilde{t})\big)$ along the blue broken lines and the derivative of the blue area $a_1(v_1)$ in the figure Fig.a converges, according to (2.56), to $2\int_0^{\bar{\sigma}_a}\mu\big(v(\tilde{t})-2\sigma_a, v(\tilde{t})\big)\,d\sigma_a$ with $\bar{\sigma}_a=\frac{v(\tilde{t})-\alpha_n}{2}$. When t converges to $\tilde{t}$ by upper values (cf. figure Fig.b), as $v(\tilde{t})$ is a maximum, it is also a point of the kind η_n of the sequence $RMS(v,\tilde{t}+h)$ and $\sigma'_a(v_2)$ goes to 0 with h; formula (2.57) shows that the same is true for the derivative of the area $a_2(v_2)$ in red (in the figure Fig.c). These results show that if $v(\tilde{t})$ is a maximum (resp. a minimum) of v, than the right and the left derivatives of the mapping $t\mapsto w(v(t),\dot{v}(t))$ do not agree at $\tilde{t}$.

When μ is defined by the formula (2.11) page 43, the derivative (2.58) may be written down as

$$\boxed{\frac{d\,\mathscr{H}_\mu(v)}{dt}(\tilde{t})=\begin{cases}\frac{1}{2}\partial_2\Delta\left(v-2\sigma_0(v),v\right)\dot{v}(\tilde{t}) & \text{if } \dot{v}(\tilde{t})>0\\ -\frac{1}{2}\partial_1\Delta\left(v,v+2\sigma_0(v)\right)\dot{v}(\tilde{t}) & \text{if } \dot{v}(\tilde{t})<0\end{cases}} \tag{2.59}$$

REMARK 2.7 *End of proof of Theorem 2.1*; to complete the proof of this Theorem, it remains to show that the Preisach operator defined by the formula (2.10) page 42, is a piecewize monotone operator when μ is defined in terms of inverse of the number of cycles to failure Δ by the formula (2.11) and satisfies the following additional conditions[30]:

$$\boxed{\partial_1\Delta(\rho_1,\rho_2)\le 0 \quad\text{and}\quad \partial_2\Delta(\rho_1,\rho_2)\ge 0 \text{ for any } \rho=(\rho_1,\rho_2)\in\mathscr{P}} \tag{2.60}$$

Let $(v_i)_{i=0}^N$ a sampled signal be given, as it can be written as the sampling of a piecewise affine function v, the formula (2.59) shows that for each index i, there is a positive constant σ_i such that

$$\big[\mathscr{W}_\mu(v)\big]_i-\big[\mathscr{W}_\mu(v)\big]_{i-1}=\begin{cases}\frac{1}{2}\partial_2\Delta\left(v_i-2\sigma_i,v_i\right)(v_i-v_{i-1}) & \text{if } v_i>v_{i-1}\\ -\frac{1}{2}\partial_1\Delta\left(v_i,v_i+2\sigma_i\right)(v_i-v_{i-1}) & \text{if } v_i<v_{i-1}\end{cases}$$

[30] Which mean t hat the number of cycles to failure of the material increases when the mean stress increases while it decreases when alternating stress increases.

Taking account of assumption (2.60), we have

$$\left(\left[\mathscr{W}_\mu(v)\right]_i - \left[\mathscr{W}_\mu(v)\right]_{i-1}\right)(v_i - v_{i-1}) \geq 0$$

which means that *the discrete version of Preisach is a piecewize monotone operator in the meaning of the* Definition 2.4 page 43.

The results obtained in this Section are summarized in the following Theorem:

THEOREM 2.3 *Assume that* $t \in [0,2T] \mapsto v(t)$ *belongs to* $W^{1,1}([0,2T],\mathbb{R})$ *and satisfies* (2.52) *the damage caused by the loading* $t \in [0,T] \mapsto v(t)$ *reduces to the following integral:*

$$\mathscr{D}(v) = \int_T^{2T} w(v(t),\sigma_0(t),\dot{v}(t))\,|\dot{v}(t)|\,dt \tag{2.61}$$

where

- $\sigma_0(t)$ *is the abscissa of the first extremum of the mapping* $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,t)$ *defined by the variational inequality* (2.39)*, parametrized by* σ_a*;*
- *the function* $(v_1,v_2,v_3) \in \mathbb{R}^3 \mapsto w(v_1,v_2,v_3) \in \mathbb{R}$ *is defined by*[31]*:*

$$w(v_1,v_2,v_3) = \begin{cases} \frac{1}{2}\partial_2\Delta(v_1-2v_2,v_1) & \text{if } v_3 > 0 \\ -\frac{1}{2}\partial_1\Delta(v_1,v_1+2v_2) & \text{if } v_3 < 0 \\ 0 & \text{if } v_3 = 0 \end{cases} \tag{2.62}$$

where (see Theorem 2.1 page 43) the mapping $(\rho_1,\rho_2) \in \mathscr{P} \mapsto \Delta(\rho_1,\rho_2) \in \mathbb{R}$ *is the inverse of the number of cycles to failure defined by the Wöhler's curve, for an alternating load* $\sigma_a = \frac{\rho_2-\rho_1}{2} \geq 0$*, at average* $\sigma_m = \frac{\rho_2+\rho_1}{2}$*:*

In practice, computation of damage by formula (2.61) is carried out within three steps:

1^o/ the first one "is a cycles identification process" which consists,see figure (Fig. 2.20), to define the mapping $t \mapsto \sigma_0(t)$ in identifying the abscissa the first extremum of the function $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(t,v)$, obtained in solving the differential inequality (2.39) for different values of σ_a; an algorithm is provided in Section 3.3, page 115;

2^o/ compute then the function (2.62) in setting

$$v_1 = v(t),\ v_2 = \sigma_0(t) \text{ and } v_3 = \dot{v}(t)$$

to *define the contribution to the total damage of each part of the cycles identified in the previous step*, an example is depicted in figures (Fig. 2.21-*a*) and (Fig. 2.21-*b*);

3^o/ the thirth step consists to carry out *the time integration of the obtained results to compute the total damage.*

[31] In view of what is stated above, it would be more consistent to define w as a set-valued mapping in saying that $w(v_1,v_2,0)$ is the interval $[v_2,0]$ or $[0,v_2]$ according to the sign of v_2. The choice (2.62), that doesn't change the computation of damage, has the advantage to simplify the computation of the partial derivatives of $(v_1,v_2,v_3) \mapsto j(v_1,v_2,v_3) = w(v_1,v_2,v_3)|v_3|$ but does not change the fact that j is not differentiable in the plane $v_3 = 0$ when $v_2 \neq 0$.

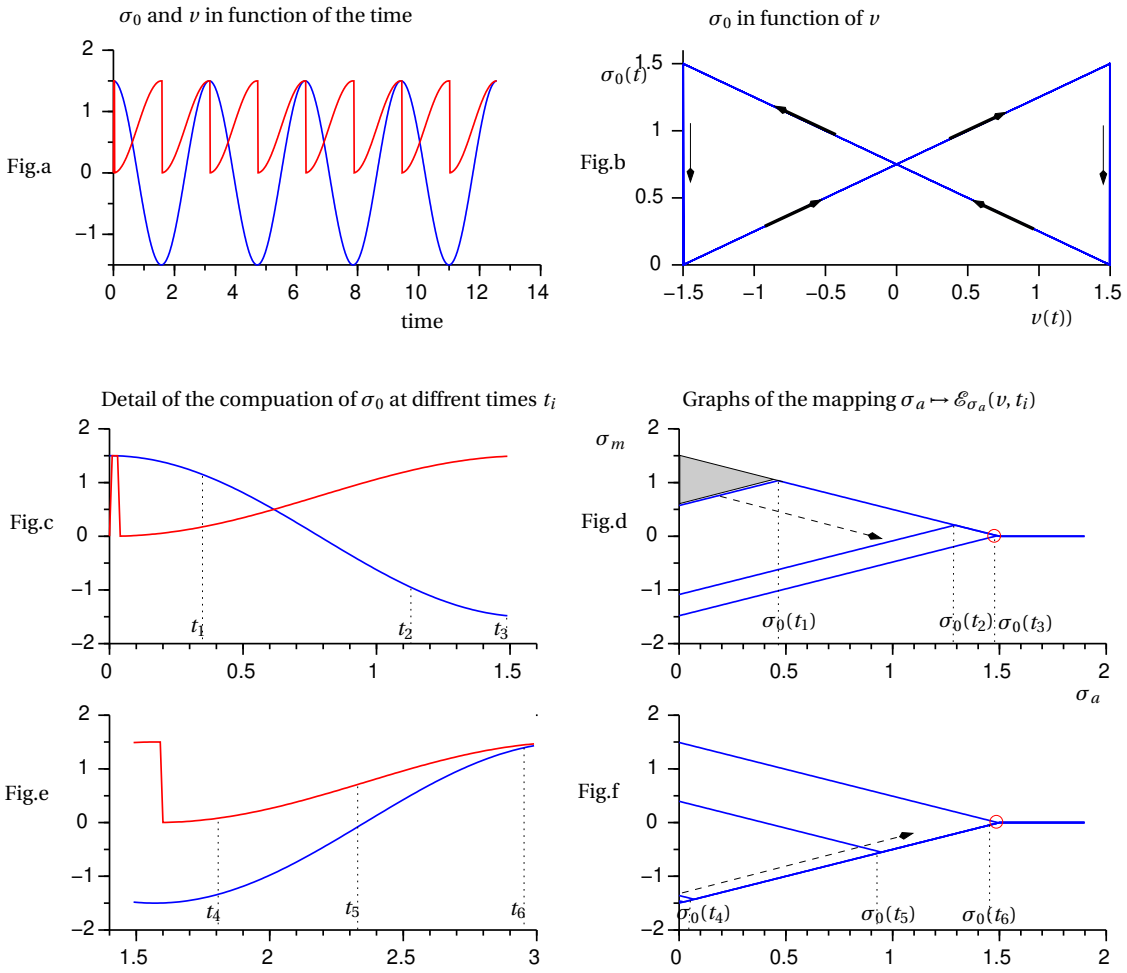


Fig. 2.20. **Computation of $t \mapsto \sigma_0(t)$, when the signal $v(t)$ is a simple function.** Figure (Fig.a) represents on the same picture the signal $v(t) = 1.5 * \cos(2\,t)$ (for t in $[0, 4\pi]$), in blue and the signal $\sigma_0(t)$ in red; σ_0 has a discontinuity when $v(t)$ changes its direction of variation; this is illustrated in figure (Fig.b) on right, when one represents in the form of a Lissajous diagram $\sigma_0(t)$ as a function of $v(t)$. This figure also shows that apart from discontinuity points, the derivative of σ_0 with respect to v is $\pm\frac{1}{2}$, according to the direction of variation of v. Figures (Fig.c) to (Fig.f) (which are not plotted in square scales) provide an illustration of the theoretical results depicted in figure (Fig. 2.18), they have been obtained numerically with the help of the integration algorithm of the equation (2.39) defined by the formula (2.50) page 65. In this simple case, there are at most two truning points; one of them is surrounded by a red circle in the figures (Fig.d) and (Fig.f), has the abscissa $\tilde{\sigma}_0 = 1.5$, which corresponds to the absolute value of the extrema of v. Figures (Fig.c) and (Fig.e) show that $\lim_{t\to\frac{\pi}{2}^-} \sigma_0(t) = \tilde{\sigma}_0$ while $\lim_{t\to\frac{\pi}{2}^+} \sigma_0(t) = 0$, which is in agreement with the theoretical results shown in the figure (Fig. 2.19)

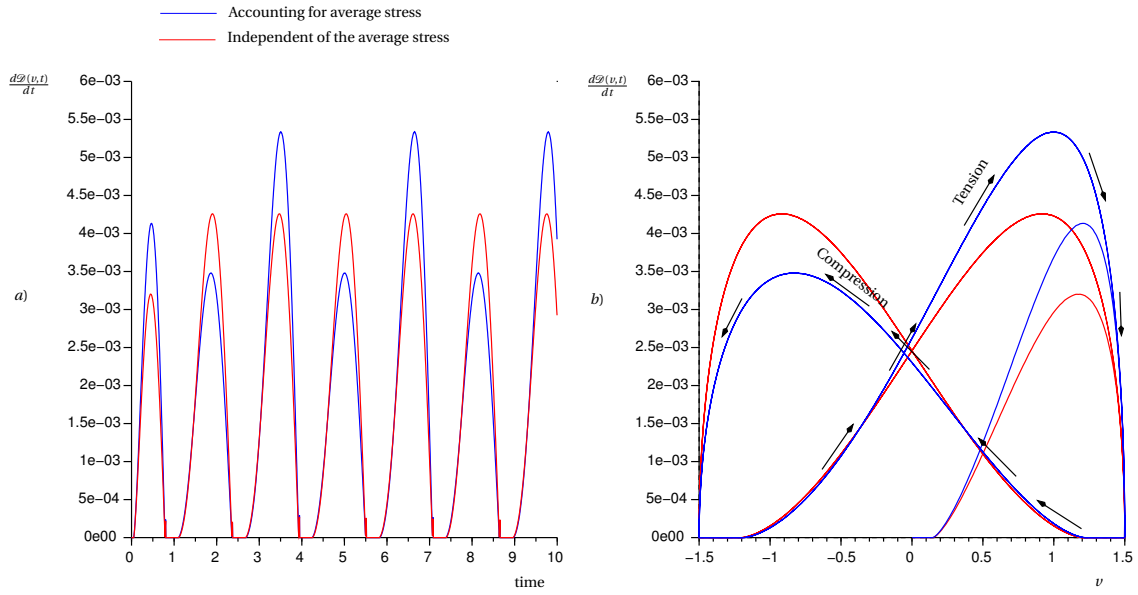


Fig. 2.21. **Computation of $t \mapsto w(v(t),\sigma_0(t),\dot{v}(t))$ in case of the Strohmeyer's formula, with and without mean stress effect, when $v(t)$ is a pure sine**. Note that insofar as each cycle is decomposed into sub-elements where the signal v is monotone, we account the parts of the cycle which are in tension (ie. $\dot{v} \geq 0$) in a different way than these which are in compression; this case is plotted on the curves in blue.

EXAMPLES 2.2 1^o/ When $\rho \mapsto \Delta(\rho)$ is defined by a Stromeyer formula without accounting for mean stress effect (formula (1.2), page 16) the formula (2.62) simplifies and does not depend on v_1; it can be written down as

$$w(v_2,v_3) = \begin{cases} \frac{1}{4\,b_s C_s}\,[\max\{(v_2-\sigma_d),0\}]^{\frac{1}{b_s}-1} & \text{if } v_3 \neq 0 \\ 0 & \text{else} \end{cases}$$

and the Theorem 2.3 allows to compute the damage as

$$\mathscr{D}(v) = \frac{1}{4\,b_s C_s}\int_T^{2T} [\max\{(\sigma_0(t)-\sigma_d),0\}]^{\frac{1}{b_s}-1}\,|\dot{v}(t)|\,dt \tag{2.63}$$

A numerical application is proposed on the signal $t \in [0,16\pi] \mapsto 1.5 * \sin(2\,t)$ with the following numerical data $b_s = 0.42,\quad C_s = 359,\quad \sigma_d = 0.15$, which is equivalent to process a loading of 16 alternate cycles of amplitude $\sigma = 1.5 * 800 MPa$ with the data identified in the Wöhler curves of figure (Fig 1.10) page 18. The damage obtained by the method described above is $\mathscr{D} = 0.0901$ while the exact result, which is given by the formula

$$\mathscr{D} = \frac{16}{N_r(\sigma)} = \frac{4\,(\sigma-\sigma_d)^{\frac{1}{b_s}}}{C_s}$$

is $\mathscr{D} = 0.0910$; *this gives a relative error of* -1% *compared with the theoretical results.*

2^o/ If we take into account the mean stress effect in the Strohmeyer formula, the function w defined in (2.62) depends on the variables v_1, v_2 and v_3; using for example

the corrective formula of Goodman (1.5) page 18, the function w is defined by

$$w(v_1,v_2,v_3)=\begin{cases}\dfrac{(R_m+2\,v_2-v_1)}{4\,C_s b_s v_2\,(R_m+v_2-v_1)}\left[\max\left(\dfrac{R_m v_2}{R_m+v_2-v_1}-\sigma_d,0\right)\right]^{\frac{1}{b_s}} & \text{if } v_3>0\\ \dfrac{(v_1-R_m)}{4\,C_s b_s v_2\,(v_2+v_1-R_m)}\left[\max\left(\dfrac{R_m v_2}{R_m-v_2-v_1}-\sigma_d,0\right)\right]^{\frac{1}{b_s}} & \text{if } v_3<0\end{cases} \tag{2.64}$$

A numerical simulation is carried out with the data given in the first example, complemented by $R_m = 10.0$; the computation of the damage depends now on the direction of variation of $v(t)$, this situation is illustrated by the lines in blue in figure (Fig. 2.21). However, as the signal v remains symmetric with respect to 0 *the cumulative damage computed over several periods must remain identical to that of a calculation carried out without taking into account the average stress.* In this case the numerical simulation leads to $\mathscr{D} = 0.0922$; which gives a relative error of +1.3% compared to the theoretical value.

The following Remarks are used in Section 4.3 page 176 to compute the right hand member of the adjoint state to the structure optimization problem.

REMARKS 2.8 1^o/ The derivatives of the mapping

$$(v_1,v_2,v_3)\in\mathbb{R}^3 \mapsto j(v_1,v_2,v_3)=w(v_1,v_2,v_3)|v_3|\in\mathbb{R}$$

are given by the formulas

$$\frac{\partial j}{\partial v_1}=\frac{v_3}{2}\begin{cases}\partial_{12}\Delta(v_1-2v_2,v_1)+\partial_{22}\Delta(v_1-2v_2,v_1) & \text{if } v_3>0\\ \partial_{11}\Delta(v_1,v_1+2v_2)+\partial_{12}\Delta(v_1,v_1+2v_2) & \text{if } v_3<0\\ 0 & \text{if } v_3=0\end{cases} \tag{2.65}$$

$$\frac{\partial j}{\partial v_2}=v_3\begin{cases}-\partial_{12}\Delta(v_1-2v_2,v_1) & \text{if } v_3>0\\ \partial_{12}\Delta(v_1,v_1+2v_2) & \text{if } v_3<0\\ 0 & \text{si } v_3=0\end{cases} \tag{2.66}$$

$$\frac{\partial j}{\partial v_3}=\frac{1}{2}\begin{cases}\partial_2\Delta(v_1-2v_2,v_1) & \text{if } v_3>0\\ \partial_1\Delta(v_1,v_1+2v_2) & \text{if } v_3<0\\ 0 & \text{si } v_3=0\end{cases} \tag{2.67}$$

2^o/ Let t given, the partial derivatives of the mapping

$$(v(t),\dot v(t))\mapsto j(v(t),\sigma_0(t),\dot v(t))$$

with respect to the independent variables $v(t)$ and $\dot v(t)$ are defined respectively by

$$\frac{\partial j}{\partial v_1}(v(t),\sigma_0(t),\dot v(t))+\frac{\partial j}{\partial v_2}(v(t),\sigma_0(t),\dot v(t))\frac{\partial \sigma_0(t)}{\partial v(t)}$$

and

$$\frac{\partial j}{\partial v_3}(v(t),\sigma_0(t),\dot v(t))$$

with, taking into account the results shown on the diagrams of the figure (Fig. 2.18)

$$\frac{\partial \sigma_0(t)}{\partial v(t)} = \frac{1}{2} sign(\dot{v}(t))$$

The formulas (2.65) et (2.66) show that setting $v = v(t)$ et $\dot{v} = \dot{v}(t)$ we have

$$\boxed{\frac{\partial j}{\partial v}(v, \sigma_0, \dot{v}) = \frac{\dot{v}}{2} \begin{cases} \partial_{22}\Delta(v - 2\sigma_0, v) - \partial_{12}\Delta(v - 2\sigma_0, v) & \text{if } \dot{v} > 0 \\ \partial_{11}\Delta(v, v + 2\sigma_0) - \partial_{12}\Delta(v, v + 2\sigma_0) & \text{if } \dot{v} < 0 \\ 0 & \text{if } \dot{v} = 0 \end{cases}} \tag{2.68}$$

While (2.67) leads to

$$\boxed{\frac{\partial j}{\partial \dot{v}}(v, \sigma_0, \dot{v}) = \frac{1}{2} \begin{cases} \partial_2\Delta(v - 2\sigma_0, v) & \text{if } \dot{v} > 0 \\ \partial_1\Delta(v, v + 2\sigma_0) & \text{if } \dot{v} < 0 \\ 0 & \text{if } \dot{v} = 0 \end{cases}} \tag{2.69}$$

EXAMPLE 2.3 When $\rho \mapsto \Delta(\rho)$ is defined with the help of the Strohmeyer's formula, without accounting for mean stress effect, the formulas (2.68) and (2.69) depend only on σ_0 and $\dot{v}$, they have the following extremely simple forms:

$$\frac{\partial j}{\partial v}(v, \sigma_0, \dot{v}) = \frac{1 - b_s}{4 b_s^2 C_s} \dot{v} \left[\max\{(\sigma_0 - \sigma_d), 0\}\right]^{\frac{1}{b_s} - 2}$$
$$\frac{\partial j}{\partial \dot{v}}(v, \sigma_0, \dot{v}) = \frac{1}{4 b_s C_s} sign(\dot{v}) \left[\max\{(\sigma_0 - \sigma_d), 0\}\right]^{\frac{1}{b_s} - 1}$$

2.5. Exercises and complements

EXERCICE 2.1 1^o/ Make a program to set up the "input-output" diagram given in figure (Fig. 2.3) where (ρ_1^i, ρ_2^i) is a given finite sequence such that $\rho_2^i \geq \rho_1^i$. Set $\mu_i = (\rho_1^i - \rho_2^i)^2$ and $\mu_i = \sin(\rho_2^i - \rho_1^i)$ for instance.

2^o/ Plot the outputs S as a function of the inputs E and explain the obtained results.

EXERCICE 2.2 (Simple verification of the formula (1.11)) 1^o/ Let a monotone sequence $v = (v_i)_{i=1}^n$ be given, compute the total variation $V_T(v)$ of v (apply formula (2.3))

2^o/ Compute the total variation $V_T(v)$ of a continuous piecewise affine function v with the help of the formula (2.3) and proof that

$$V_T(v) = \int_0^T \left|v'(t)\right| dt$$

Is the mapping $t \mapsto v(t)$ everywhere differentiable?

3^o/ Compute the total variation of the function $t \in [0, 2\pi] \mapsto \sin(t) \in [-1, 1]$ and verify the "magic formula"

$$V_T(\sin) = \int_0^{2\pi} |\cos(t)| dt = \int_0^{2\pi} |\sin'(t)| dt$$

EXERCICE 2.3 1^o) Compute the set X_t introduced in Definition 2.5 for:

$$(\rho_1, \rho_2) = \begin{cases} (-0.5, 0.5) \\ (-1, 1) \\ (1.001, 1.1) \end{cases}$$

if v is the mapping

$$t \in [0,2\pi] \mapsto \sin(t)$$

2^o) In each previous case, plot the function $t \mapsto h_\rho(v,\xi)(t)$ for $\xi = 0$ and $\xi = 1$

EXERCICE 2.4 1^o/ Compute the RMS sequence of the function $t \in [0,2\pi] \mapsto \sin(t)$ for $t \in [0,2\pi]$

2^o/ Try to do the same for the mapping

$$t \in [-\pi,\pi] \mapsto \begin{cases} t^2 \sin\left(\frac{1}{t}\right) & \text{if } t \neq 0 \\ 0 & \text{if } t = 0 \end{cases} \tag{2.70}$$

Is this mapping in the Sobolev space $W^{1,1}([-\pi,\pi],\mathbb{R})$?

EXERCICE 2.5 1^o/ Write an algorithm to compute the function $t \mapsto \mathscr{E}_{\sigma_a}(v,t)$

2^o/ Let v be the sequence $\left(\sin(\frac{2k\pi}{N})\right)_{k=0}^{N}$, plot the graph of the mapping $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,\frac{2k_0\pi}{N})$ (where $0 \leq k_0 \leq N$ is a given integer) and identify the RMS sequence on this graph.

3^o/ Test the algorithm on the mapping (2.70); what happens when you increase N?

EXERCICE 2.6 1^o/ Proof the formulas (2.40) and (2.41) page 61

2^o/ Proof that the operator

$$x \mapsto \operatorname{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ [-1,1] & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{2.71}$$

is the sub-differential of the mapping $x \mapsto |x|$.

3^o/ Make a program to solve the differential equation

$$m\ddot{x} + k\,x + c\operatorname{sign}(\dot{x}) \ni f(t)$$

and verify numerically that the mapping $f \mapsto x$ is a hysteretic damping.

4^o/ What is the mechanical interpretion of the multivalued character of the sign function?

Solution of the exercises & homework.

Solution of exercise 2.1. You can easily write the program (see listing bellow), basically you have to

- write a function to implement the relays operators $h_\rho(v,\xi)$ defined in Definition 2.1
- initialize the relays as you like
- and compute the sum according to the formula shown in figure (Fig. 2.3) page 43.

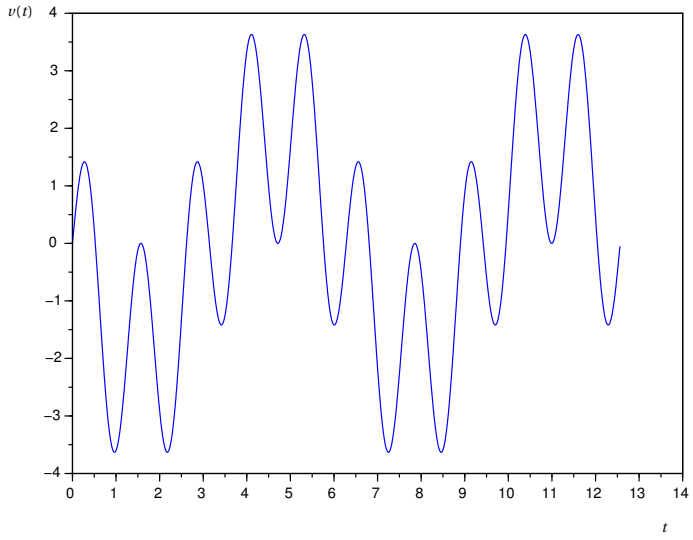


Fig. 2.22. **Input signal.** The signal sampled on 1257 samples

You can use the following program, but be careful when you use the "copy-paste" function of your computer.

Prorgram : implementation of "input-output" diagram given in figure (Fig. 2.3).

- The relay operator

```
function z=h_rho(rho_1,rho_2,v,xi)
% Compute the output of the relay operator  h_ρ(v,ξ)
% for a given initial state ξ
    if v<=rho_1
        z=0;
    else if v>=rho_2
            z=1;
        else
        % At first call ξ, is the initialization
        % after it, is the previous value of the realy.
            z=xi;
        endif
    endif
endfunction
```

- Main program
  - Sampling of the input signal:

    ```
    % The sampled input signal
    time=0:0.01:4*pi;
    signal=4*sin(time).*(cos(2*time)+cos(4*time));
    %
    figure(1);
    plot(time,signal)
    title('Input signal'); xlabel('time');ylabel('v(t)');
    ```

    which is plotted in figure (Fig. 2.22) and obviously satisfies $v(0) = v(4\pi) = 0$.
  - Initialization of the computation

    ```
    % The computation is performed on a bounded part of the Preisach plane.
    ```

```
rho_max=max(signal)*1.20;
rho_min=min(signal)*1.20;
% N_rho^2 hysterons will be used to compute the outputs of the Preisach operator
% they are equally distributed in the Preisach plane
N_rho=20;
% N_rho=50;
sample_rho=(rho_max-rho_min)/(N_rho+1);
Rho=rho_min:sample_rho:rho_max;% Sampling
```

– Initialization of relays and computation of the measure μ_i.

```
R_m=8.5;% For instance
for i=1:size(Rho,2)
    rho_2=Rho(i);
    for j=1:size(Rho,2)
        rho_1=Rho(j);
        if rho_1>rho_2
            Relay(i,j)=NaN;
            mu_p(i,j)=NaN;
        else if (rho_1+rho_2)>0
        % Relay(i,j)=0; % Standard initialization
                Relay(i,j)=round(rand());% Random initialization
                mu_p(i,j)=sample_rho**2*(rho_2-rho_1)**2/(1 -(rho_1+rho_2)/R_m);
                % mu_p(i,j)=sample_rho**2*(rho_2-rho_1)**2;
                % mu_p(i,j)=sample_rho**2*sin(rho_2-rho_1);
            else
            % Relay(i,j)=1;% Standard initialisation
                Relay(i,j)=round(rand());% Random intitialization
                mu_p(i,j)=sample_rho**2*(rho_2-rho_1)**2/(1-(rho_1+rho_2)/R_m);
                % mu_p(i,j)=sample_rho**2*(rho_2-rho_1)**2;
                % mu_p(i,j)=sample_rho**2*sin(rho_2-rho_1);
            endif
        endif
    endfor
endfor
%
% Plot the initial states of the Relays see figure (Fig.2.23)
figure(2);
surf(Rho,Rho,Relay)
view(0,90);
title('Initial states of the relays');
xlabel('rho_1');ylabel('rho_2');zlabel('h(rho_1,rho_1)');
% Plot the weights mu_i, which depend on rho_1 and rho_2
% see figure (Fig. 2.24)
figure(3);
surf(Rho,Rho,mu_p)
view(0,90);
title('Weights according to rho_1 and rho_2');
xlabel('rho_1');ylabel('rho_2');zlabel('mu(rho_1,rho_1)');
```

– Compute the output of the Preisach operator

```
for k=1:size(signal,2)
    z=0;
    for i=1:size(Rho,2)
        rho_2=Rho(i);
        for j=1:size(Rho,2)
            rho_1=Rho(j);
            if rho_1>rho_2
```

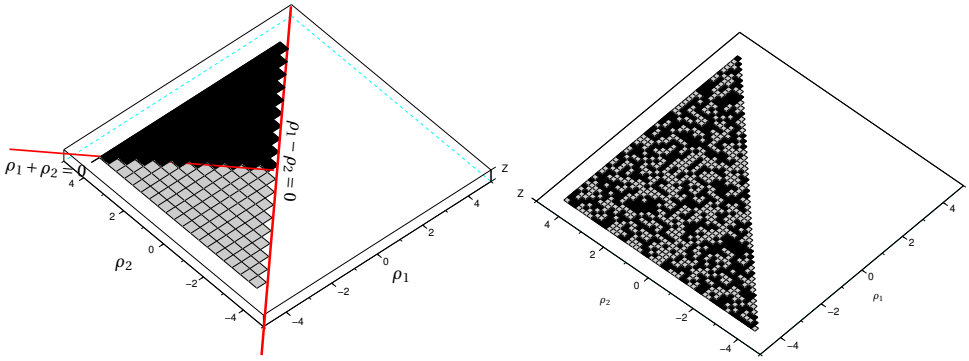


Fig. 2.23. **Initialization of the relays operators.** The relay operators are uniformly distributed in the half Preisach plane; on the left hand diagram they are initialized at 1 if $\rho_2+\rho_1<0$ and at 0 if $\rho_2+\rho_1>0$, while on the right hand one they are randomly initialized.

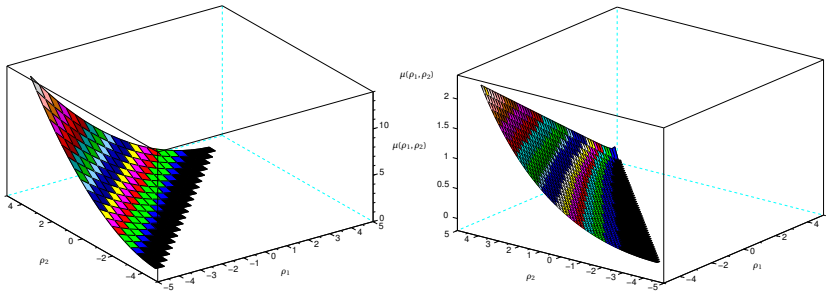


Fig. 2.24. The weights are $\mu=(\rho_2-\rho_1)^2$ on the right and $\frac{(\rho_2-\rho_1)^2}{1-\frac{rho_1+rho_2}{Rm}}$ on the left. Note that in the first case, the distribution is symmetrical with respect to the line $\rho_1+\rho_2=0$.

```
                break
            else
                mu=mu_p(i,j);
                xi=Relay(i,j);
                z=z+h_rho(rho_1,rho_2,signal(k),xi)*mu;
                Relay(i,j)=h_rho(rho_1,rho_2,signal(k),xi);
            endif
        endfor
    endfor
    h_mu(k)=z;
endfor
%
% Plot the output under the form of a Lissajous diagram see
% figures (Fig. 2.25) and (Fig. 2.26).
figure(4);
plot(signal,h_mu)
title('Hysteresis loops of the Preisach operator');
xlabel('Input signal v');ylabel('Output h_mu(v)');
%
% Plot the initialization of the relays to compute the output of W_mu
figure(5);
surf(Rho,Rho,Relay)
```

```
view(0,90);
title('Relays initialization tuned for the computation of W_mu');
xlabel('rho_1');ylabel('rho_2');
% Compute the output of W_μ
for k=1:size(signal,2)
    z=0;
    for i=1:size(Rho,2)
        rho_2=Rho(i);
        for j=1:size(Rho,2)
            rho_1=Rho(j);
            if rho_1>rho_2
                break
            else
                mu=mu_p(i,j);
                xi=Relay(i,j);
                z=z+h_rho(rho_1,rho_2,signal(k),xi)*mu;
                Relay(i,j)=h_rho(rho_1,rho_2,signal(k),xi);
            endif
        endfor
    endfor
    W_mu(k)=z;
endfor
% Plot the output
figure(6);
plot(signal,W_mu)
title('Hysteresis loops of W_mu');
xlabel('Input signal v');ylabel('Output h_mu(v)');
% Note that the W_μ  maps the periodic signals into periodic signals
figure(7);
hold on;
plot(time,W_mu)
rescal=50.;% scaling factor used for post-processing purpose, see figure (Fig. 2.28)
plot(time,rescal*signal)
xlabel('time');ylabel('v(t)');
%
% Compute the total variation of W_μ(v)
%
V_T=0;
for i=1:size(W_mu,2)-1
    V_T=V_T+abs(W_mu(i+1)-W_mu(i));
end
%
% Print the total variation
V_T
%
% Note that the computations are very expensive, a way to simplify is to use
% the geometric representation of the Preisach operator.
```

Solution of exercise 2.2. 1^o/ Suppose for instance that $v = (v_i)_{i=1}^N$ is an increasing sequence then $v_i \leq v_{i+1}$ and

$$V_T(v) = \sum_{i=1}^{N-1} |v_{i+1} - v_i| = \sum_{i=1}^{N-1} v_{i+1} - v_i = v_N - v_1$$

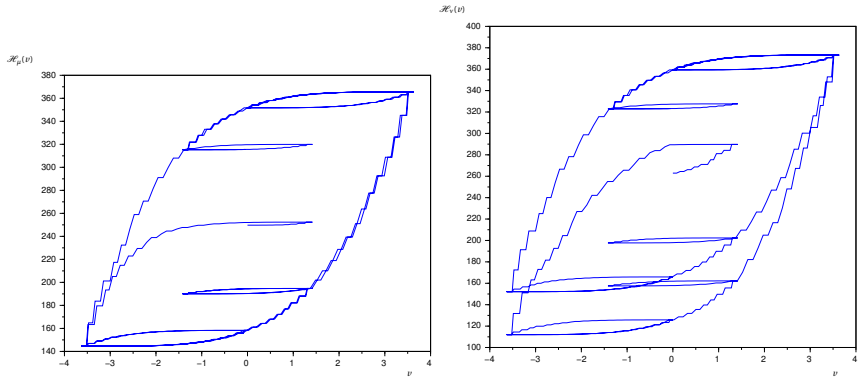


Fig. 2.25. **Outputs of the Preisach plotted under the form of a Lissajous diagram**. The computations are carried out on the signal v^{per}, which is defined on $[0,26s]$, see figure (Fig. 2.22). On the left hand diagram the relay operators are initialized in the standard way while in the left hand diagram the relays randomly initialized. We see that after an initialization phase, the outputs of the Preisach operator are a periodic; this is explained in figure (Fig. 2.27).

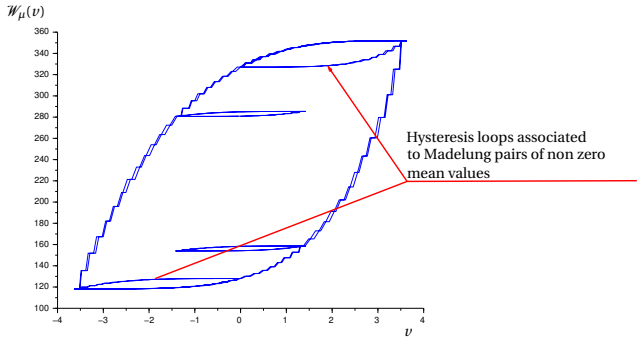


Fig. 2.26. **Outputs of the Preisach plotted under the form of a Lissajous diagram**. In this case, the distribution of weights μ_i is not symmetrical with respect to the straight line $\rho_1+\rho_2=0$ and the contribution to the total variation of $\mathscr{W}_\mu(v)$ of the hysteresis loops associated to the Madelumg's pairs depends on their mean values.

$2^o/$ Plot the function $t\in[0,2\pi]\mapsto \sin t$ and you see that, if you compute the total variation with the help of the previous formula you get $V_T(\sin)=4$. On the other hand, the derivative of sin is cos and we have:

$$\int_0^{2\pi}|\cos t|dt=\int_0^{\frac{\pi}{2}}\cos t dt-\int_{\frac{\pi}{2}}^{\frac{3\pi}{2}}\cos t dt+\int_{\frac{3\pi}{2}}^{2\pi}\cos t dt=4$$

$3^o/$ If $t\in[0,T]\mapsto v(t)$ is piecewise affine you can define a finite increasing sequence $(t_k)_{k=1}^N$ starting at $t_1=0$, ending at $t_N=T$ and such that the mapping

$$v_i : t\in[t_i,t_{i+1}]\mapsto v(t)$$

is affine and thus monotone. Using on the one hand the first part of the exercise, we have

$$V_T(v_i)=|v(t_{i+1})-v(t_i)|$$

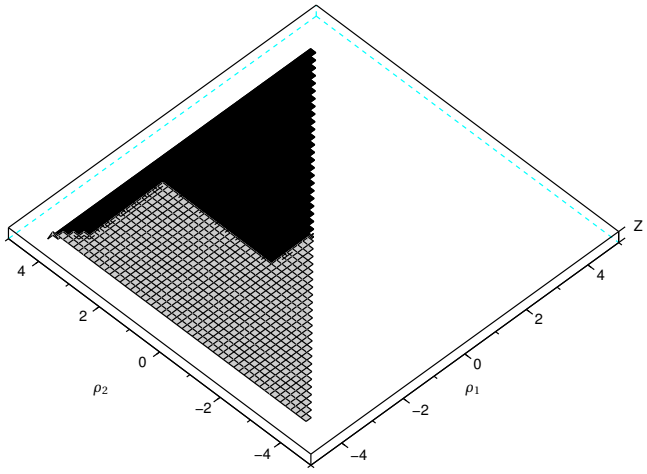


Fig. 2.27. **State of the relays at the end of the first loop**. The two initializations of the relays plotted in the figure (Fig. 2.25) lead to the same result at the end of the first computation loop, these sates of the relays are used to compute the outputs of the operator $\mathcal{W}_\mu$ defined by formula (2.25) page 50.

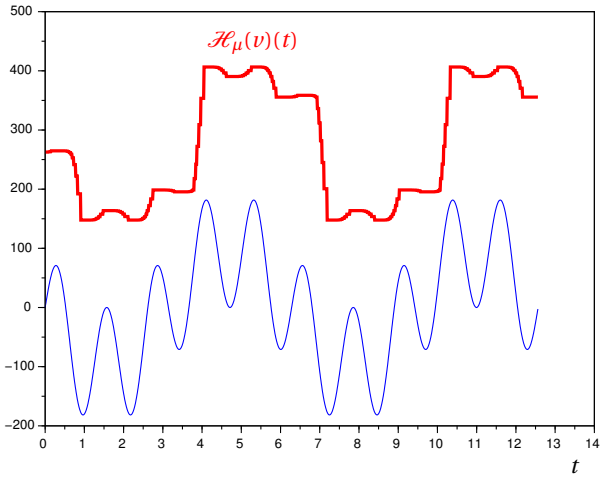


Fig. 2.28. We show on the same picture the output of the Preisach operator and the input $50 * v$ (for scaling reasons).

and, on the other hand, the formula

$$|v(t_{i+1}) - v(t_i)| = \int_{t_i}^{t_{i+1}} |\dot{v}_i(t)|dt$$

we get

$$V_T(v) = \sum_{i=1}^{N-1} |v(t_{i+1}) - v(t_i)| = \sum_{i=1}^{N-1} \int_{t_i}^{t_{i+1}} |\dot{v}_i(t)|dt \tag{2.72}$$

As $v_i'(t) = v'(t)$ on each interval $]t_i, t_{i+1}[$ we can define the total variation (2.72) by

$$V_T(v) = \int_0^T |\dot{v}(t)|dt$$

Notice that the derivative $\dot{v}(t)$ is not necessarily defined at time t_i but that wasn't a matter to compute the integral.

Homework.

- I remember you that a mapping $v : [0,T] \to \mathbb{R}$ is said to be Lipschitz continuous if there is a positive constant C such that:
$$|v(t_1) - v(t_2)| \leq C|t_1 - t_2| \quad \text{for } t_1, t_2 \in [0,T]$$
Show that the total variation of a Lipschitz continuous function is bounded.
- Is the total variation of the mapping
$$t \in [-\pi,\pi] \mapsto \begin{cases} t^2 \cos\left(\frac{1}{t^2}\right) & \text{if } t \neq 0 \\ 0 & \text{else} \end{cases}$$
bounded? Hint. Plot the derivative

Solution of exercise 2.3.

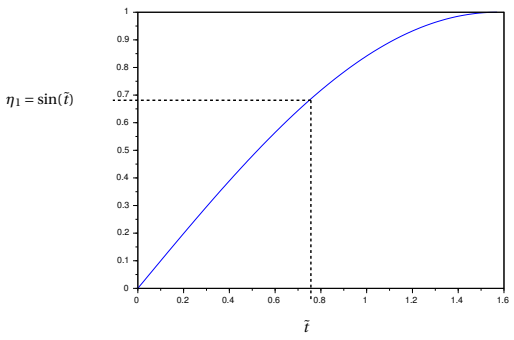
$1^o/$ When $\rho_1 = -0.5$ and $\rho_2 = 0.5$
- Note that that $\sin(\frac{\pi}{6}) = 0.5$ and $\sin(\frac{5\pi}{6}) = 0.5$ thus
  - if $t < \frac{\pi}{6}$ then $X_t = \emptyset$
  - if $\frac{\pi}{6} \leq t < \frac{5\pi}{6}$ then $X_t = \{\frac{\pi}{6}\}$
- Note on the other hand that $\sin(\frac{7\pi}{6}) = -0.5$ and $\sin(\frac{11\pi}{6}) = -0.5$, from the previous results we conclude that
  - if $\frac{5\pi}{6} \leq t < \frac{7\pi}{6}$ then $X_t = \{\frac{\pi}{6}, \frac{5\pi}{6}\}$
  - if $\frac{7\pi}{6} \leq t < \frac{11\pi}{6}$ then $X_t = \{\frac{\pi}{6}, \frac{5\pi}{6}, \frac{7\pi}{6}\}$
  - if $\frac{11\pi}{6} \leq t \leq 2\pi$ then $X_t = \{\frac{\pi}{6}, \frac{5\pi}{6}, \frac{7\pi}{6}, \frac{11\pi}{6}\}$

$2^o/$ When $\rho_1 = -1$ and $\rho_2 = 1$
- if $t < \frac{\pi}{2}$ then $X_t = \emptyset$
- if $\frac{\pi}{2} \leq t < \frac{3\pi}{2}$ then $X_t = \{\frac{\pi}{2}\}$
- if $\frac{3\pi}{2} \leq t \leq 2\pi$ then $X_t = \{\frac{\pi}{2}, \frac{3\pi}{2}\}$

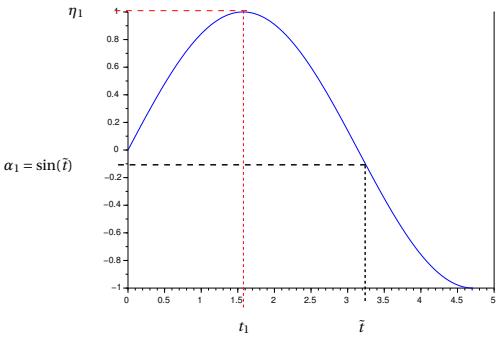
$3^o/$ At last if $\rho_1 = 1.001$ and $\rho_2 = 1.1$ then $X_t = \emptyset$ for all $t \in [0, 2\pi]$

This allows to compute the mapping $t \mapsto h_\rho(v,\xi)(t)$ as follows:

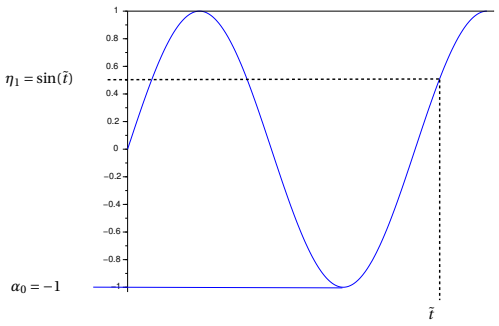
- Case $1^o/$
$$h_\rho(v,\xi)(t) = \begin{cases} \xi & \text{if } 0 \leq t < \frac{\pi}{6} \\ 1 & \text{if } \frac{\pi}{6} \leq t < \frac{5\pi}{6} \\ 0 & \text{if } \frac{5\pi}{6} \leq t \leq 2\pi \end{cases}$$
- Case $2^o/$
$$h_\rho(v,\xi)(t) = \begin{cases} \xi & \text{if } 0 \leq t < \frac{\pi}{2} \\ 1 & \text{if } \frac{\pi}{2} \leq t < \frac{3\pi}{2} \\ 0 & \text{if } \frac{3\pi}{2} \leq t \leq 2\pi \end{cases}$$
- Case $3^o/$ $h_\rho(v,\xi)(t) = \xi$ for all $t \in [0, 2\pi]$.



Case 1



Case 2



Case 3

Fig. 2.29. **Computation of the RMS sequence associated with the** sin **function.**

Solution of exercise 2.4. Go back to the procedure given in Definition 2.7 page 53 and remember that the RMS sequence depends on the time $\tilde{t}$.

1^o/ If $0 \leq \tilde{t} \leq \frac{\pi}{2}$ (diagram "Case 1" in figure Fig. 2.29). As sin is increasing we have

$$\max_{t\in[0,\tilde{t}]} |\sin(t)| = \sin(\tilde{t})$$

so we can set

$$t_1 = \tilde{t} \text{ and } \eta_1 = \sin(\tilde{t})$$

The RMS sequence reduces to $\eta_1 = \sin(\tilde{t})$.

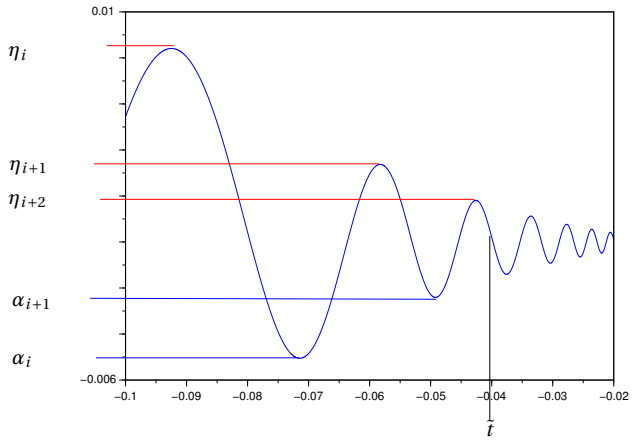


Fig. 2.30. **Graph of the function $t^2 \sin\left(\frac{1}{t}\right)$ for $t \in [-0.1, -0.002]$.**

$2^o/$ If $\frac{\pi}{2} < \tilde{t} < \frac{3\pi}{2}$ (diagram "Case 2" in figure (Fig. 2.29)). The function sin is now decreasing, as

$$\sin(\frac{\pi}{2}) = 1 = \underset{t\in[0,\tilde{t}]}{max} |\sin(t)|$$

we can set

$$t_1 = \bar{t} = \frac{\pi}{2} \text{ and } \eta_1 = 1$$

the case $1^o/$ of the procedure gives

$$\alpha_1 = \min_{\frac{\pi}{2} \le t \le \tilde{t}} \sin(t) = \sin(\tilde{t})$$

As $t_2 = \tilde{t}$ the procedure stops and $\{\eta_1 = 1, \alpha_1 = \sin\tilde{t}\}$ is the RMS sequence.

$3^o/$ If $\frac{3\pi}{2} < \tilde{t} < \frac{5\pi}{2}$ (diagram Case 3 in figure (Fig. 2.29)) the function sin is increasing and we have

$$1 = \underset{t\in[0,\tilde{t}]}{max} |\sin(t)|$$

which is achieved at $\bar{t} = \frac{3\pi}{2}$; as $t_0 = \sin(\bar{t}) = -1$ is negative the case $2^o/$ of the procedure gives $\alpha_0 = -1$ but as the function sin is increasing we can set

$$t_1 = \tilde{t} \text{ and } \eta_1 = \sin(\tilde{t})$$

Thus $\{\alpha_0 = -1, \eta_1 = \sin(\tilde{t}\}$ is the RMS sequence...

$4^o/$ When $\tilde{t} = \frac{3\pi}{2}$ only $\alpha_0 = -1$ exists.

Now you are comfortable with the concept of RMS sequence, we can see what happens for the mapping

$$\tilde{t} \mapsto \tilde{t}^2 \sin\left(\frac{1}{\tilde{t}}\right)$$

let us plot this function for $\tilde{t}$ ranging between -0.1 and -0.002. You see in figure (Fig. 2.30) that when $\tilde{t}$ goes to 0 the number of terms of the RMS sequence increases and goes to ∞. This is due to the fact that the extrema of $\tilde{t}^2 \sin\left(\frac{1}{\tilde{t}}\right)$ monotonically decrease when $\tilde{t}$ goes to 0, and are kept in the RMS sequence.

I claim that the mapping $t \in [-\pi,\pi] \mapsto f(t) = t^2 \sin\left(\frac{1}{t}\right)$ is in $W^{1,1}([-\pi,\pi],\mathbb{R})$! To proof this, we have to cheek that

$$\int_{-\pi}^{\pi} |f(t)|dt + \int_{-\pi}^{\pi} |f'(t)|dt < +\infty$$

- It is quite easy to proof the inequality $\int_{-\pi}^{\pi} |f(t)|dt < +\infty$: as $|f(t)| \leq t^2$, we have thus
$$\int_{-\pi}^{\pi} \left| t^2 \sin\left(\frac{1}{t}\right) \right| dt \leq \frac{2\pi^3}{3}$$
- On the other hand $f'(t) = 2t\sin\left(\frac{1}{t}\right) - \cos\left(\frac{1}{t}\right)$ and it remains to show that
$$\int_{-\pi}^{0} \left| \cos\left(\frac{1}{t}\right) \right| dt < +\infty$$
for instance. Making the change of variable $u = \frac{1}{t}$, we have to show that
$$\int_{-\infty}^{-\frac{1}{\pi}} \left| \frac{\cos(u)}{u^2} \right| du < +\infty$$
this inequality being obvious because
$$\left| \frac{\cos(u)}{u^2} \right| \leq \frac{1}{u^2}$$

This example was intending to show you that functions in $W^{1,1}$ may have "pathological" behaviors. Fortunately, such a scenario does not occur in mechanics!

Homework.

- Give necessary and sufficient conditions on the real numbers α and β insuring that the mapping
$$t \in [-\pi,\pi] \mapsto \begin{cases} t^\alpha \sin\left(\frac{1}{t^\beta}\right) & \text{if } t \neq 0 \\ 0 & \text{else} \end{cases}$$
is in $W^{1,1}([-\pi,\pi],\mathbb{R})$.
- Proof that the closure of vector space of piecewize affine functions defined on $[0,T]$ for the norm
$$v \mapsto \int_0^T |v(t)| + |v'(t)|dt$$
is the Sobolev space $W^{1,1}([0,T],\mathbb{R})$. Hint: You can easily see that the space of piecewize affine function is a sub-space of $W^{1,1}([0,T],\mathbb{R})$, if you can't do the rest of the proof, have a look in Rudin [**39**] or Brezis [**6**].
- If you have read Rudin [**39**] you can now proof that a Lipschitz continuous function on $[0,T]$ is in $W^{1,1}([0,T],\mathbb{R})$, what about the converse?

Solution of exercise 2.5. This exercise is the given below implementation of the formula (2.50) page 65:

- Make a function allowing to compute $\mathscr{E}_{\sigma_a}(t+h)$ as function of $v(t+h)$ and $\mathscr{E}_{\sigma_a}(t)$

```
function E_1=E_sigma(E_0,v,sigma_a,delta)
% Implementation of the formula (2.50)
%
% inputs: E_0:=\mathscr{E}_{\sigma_a}(t), v:=v(t+h)
% sigma_a:=\sigma_a and delta:=hk where h is the step size of the
% integration method and k is the slope of regularization.
% output : E_0:=\mathscr{E}_{\sigma_a}(t+h)
%
    y1=(E_0+delta*(v+sigma_a))/(1+delta);
    y2=(E_0-delta*(sigma_a-v))/(1+delta);
    if y2<=(v-sigma_a)
        E_1=y2;
    else if y1>=(v+sigma_a)
            E_1=y1;
        else
            E_1=E_0;
        endif
    endif
endfunction
```

- We first study the effect of the discretization parameters:
  - *integration step size h*,
  - and *slope of the regularization k* (see definition in figure (Fig. 2.14))

  on the time integration of a regularized version of the differential inequality (2.39) page 61.

```
N=10*10*10;% Nunber of samples of the input signal
h=2*pi/(N-1);% Step size for the time integration
time=0:h:2*pi;% Integration is carried out between 0 and 2π
v=sin(time);% Sampling of input signal
sigma_a=0.5;
k=5*10%*10;% Slope of the regularization
delta=h*k;
% Integration of the differential inequality (2.39) by an implicit Euler method
% E:= sampling of the solution \mathscr{E}_{\sigma_a}(t) for a given value of \sigma_a
E(1)=0;
for j=1:N-1
    E(j+1)=E_sigma(E(j),v(j+1),sigma_a,delta);
endfor
% Plot on the same graphic the inputs t\mapsto v(t) and
% the output t\mapsto \mathscr{E}_{\sigma_a}(v,t)
figure(1);
hold on
plot(time,v); plot(time,E)% An example is given in figure (2.31)
% Plot the hysteresis loop
figure(2);
plot(v,E)% See figure (Fig.2.32)
```

  - Figure (Fig. 2.31) shows that when the step size h goes to 0, the approached solution (by the implicit Euler method) converges to the exact solution of the equation (2.39).
  - And figure (Fig. 2.32) shows that when k goes to $+\infty$ the solution of equation (2.48) approaches the solution of the variational inequality (2.39).
- Let t_0 be given, can now compute as follows the graph of $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v, t_0)$, we suppose first that $v(t) = \sin t$ for $0 \le t \le 2\pi$.

```
N0=N/4+100;% We will assume π/2 < t0 < 3π/2
```

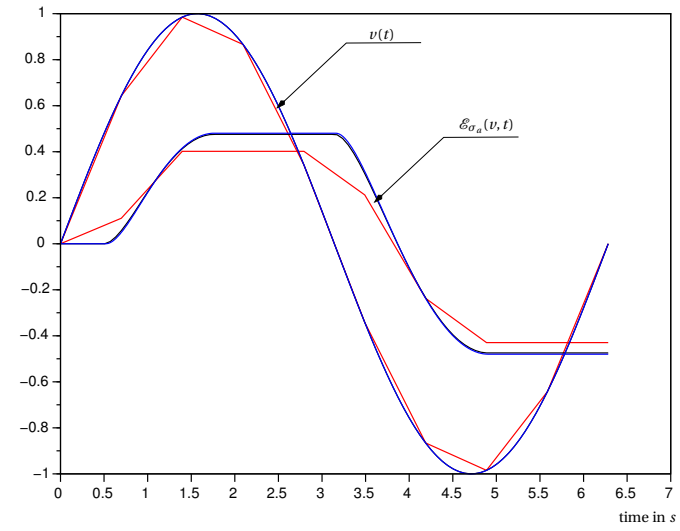


Fig. 2.31. **Resolution of the equation** (2.48) **by the implicit Euler method.** In this case: $\sigma_a = 0.5$, $k = 5$; the Euler method is carried out for several values of h: for the curves in red $h = 0.6$ seconds; h is 0.06 (resp. 0.006) for the curves in black (resp. in blue).

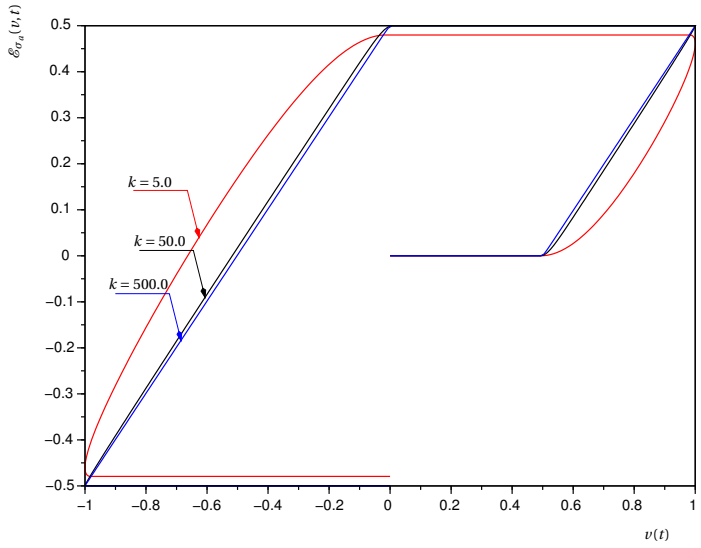


Fig. 2.32. **Hysteresis loops for several values of the slope k.** In this case the integration is carried out for $k = 5.0$ (red), $k = 50.0$ (black) and $k = 500$ (blue). The integration step is chosen small enough to suppose the outputs of Euler's method converged to the exact solution of the equation (2.48).

```
delta_sigma=0.01;
sigma=0:delta_sigma:1.20;
E(1)=0;
for i=1:size(sigma,2)
    sigma_a=sigma(i);
    % Time integration between 0 and t0
    for j=1:N0
```

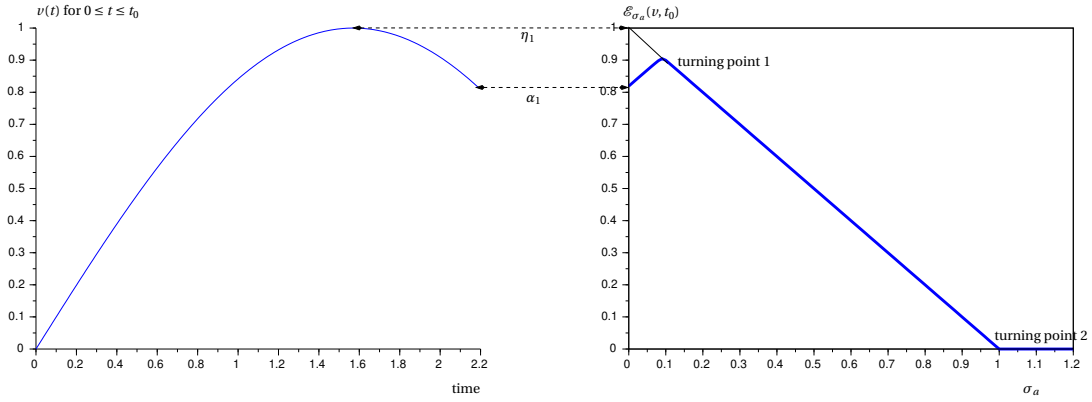


Fig. 2.33. **Turning points of the function** $t \in [0, t_0] \mapsto \sin t$**.** In this case the RMS sequence has two points η_1, α_1 as explained in exercise 2.4.

```
        E(j+1)=E_sigma(E(j),v(j+1),sigma_a,delta);
    end
    G(i)=E(NO+1);
end
%
figure(3);% See figures (Fig. 2.33) and (Fig. 2.34)
subplot(121)
plot(time(1:NO+1),v(1:NO+1))% Plot the signal
subplot(122)
plot(sigma,G);% Plot the graph of σa ↦ E_σa(v,t0)
```

Analogue computations can be carried out for the mapping

$$t \in [0, \frac{\pi}{2}] \mapsto \left(t - \frac{\pi}{4}\right)^2 \sin \frac{4}{4t - \pi}$$

and $t_0 = \frac{\pi}{4}$; we know from Exercise 2.4 that in this case the RMS sequence is endless and figure (Fig. 2.34) shows that the numerical solution of the variational inequality can't reproduce this phenomena.

Homework. 1^o/ Use the algorithm previously developed to compute the outputs of the Preisach operator $\mathscr{H}_\mu$ where μ is defined by Bastenaire and Stromeyer formulas

2^o/ Compute the total variation of the function $t \in [0, 2\pi] \mapsto \mathscr{H}_\mu(\sin, t)$ with the help of the formula (2.3)

3^o/ Use the results of the Theorem 2.3 to do the same computation.

4^o/ What do you conclude?

5^o/ Test the algorithm on the random function (white noise).

Solution of exercise 2.6. 1^o/ Assume to simplify that $\sigma_a = 1$. Let $x_0 \in \mathbb{R}$ be given, by definition (2.38) page 61 of the sub-differential, $\xi \in \mathbb{R}$ lies in $\partial I_{[-1,1]}(x_0)$ if and only if the inequality

$$I_{[-1,1]}(x) - I_{[-1,1]}(x_0) \geq \xi.(x - x_0) \tag{2.73}$$

holds for any $x \in \mathbb{R}$.

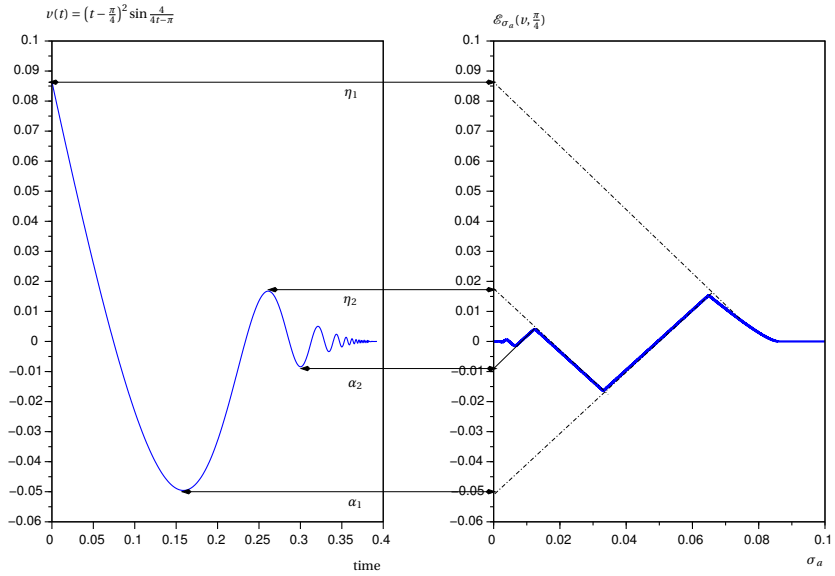


Fig. 2.34. **Turning points of the function** $t \in [0, \frac{\pi}{4}] \mapsto \left(t - \frac{\pi}{4}\right)^2 \sin \frac{4}{4t-\pi}$**.** In this case the RMS sequence $(\eta_1, \alpha_1, \eta_2, \alpha_2, \cdots)$ is endless as explained in exercise 2.4 and the numerical method allows to find 9 turning points only.

a/ For $|x_0| > 1$ we have $I_{[-1,1]}(x_0) = +\infty$ by definition (2.36) page 60 of a characteristic function. Using (2.73), we see that ξ must satisfy

$$-\infty \geq -\xi.x_0$$

for instance. As this inequality is false for any $\xi \in \mathbb{R}$, the sub-differential $\partial I_{[-1,1]}(x_0)$ is the empty set.

b/ For $|x_0| \leq 1$, we have $I_{[-1,1]}(x_0) = 0$, inequality (2.73) rewrites as

$$\begin{array}{cc} +\infty \geq \xi.(x - x_0) & \text{for } |x| > 1 \\ 0 \geq \xi.(x - x_0) & \text{for } |x| \leq 1 \end{array}$$

- If $|x_0| < 1$ the second inequality entails $\xi = 0$ so that $\partial I_{[-1,1]}(x_0) = \{0\}$.
- While if $x_0 = -1$ (resp. $x_0 = 1$) it entails $\xi \leq 0$ (resp. $\xi \geq 0$) which leads to

$$\partial I_{[-1,1]}(-1) = \mathbb{R}^- \quad \partial I_{[-1,1]}(1) = \mathbb{R}^+$$

2^o/ Let f be the mapping $x \mapsto |x|$. As f is convex, differentiable for $x \neq 0$ we have

$$\partial f(x) = \{f'(x)\} = \left\{ \begin{array}{cc} -1 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{array} \right.$$

and it remains to compute the "generalized derivative $\partial f(0)$" of f at $x = 0$. Using once more the definition (2.38), we see that it is the set

$$\partial f(0) = \{\xi \in \mathbb{R}\,;\, |x| \geq \xi.x \; \forall x \in \mathbb{R}\}$$

or in other words the interval $[-1, 1]$. *We summarize all of that in saying that if the function sign is defined by the formula* (2.71)*, the derivative of the "absolute value" function is the "sign" function.*

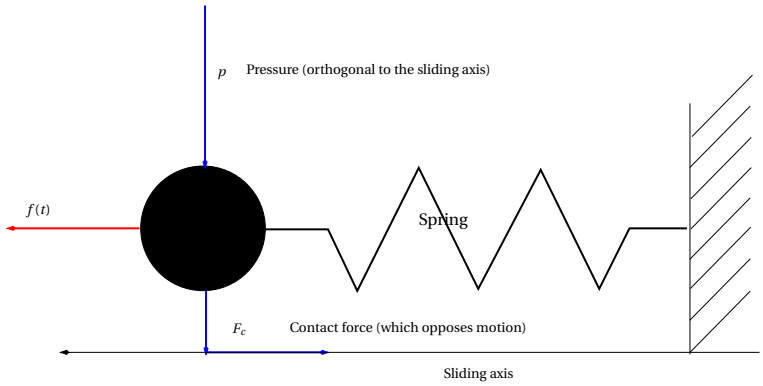


Fig. 2.35. **Hysteretic damping.** We are considering a sliding mass m which is attached by a spring of stiffness k. We assume moreover that the contact between the particle and the sliding axis x is defined by a Coulomb's law of friction coefficient μ and that mass is submitted to a pressure p (normal to the sliding axe). The contact generates thus a constant force $F_c = -\mu p \,\mathrm{sign}(\dot{x})$ opposed to the motion, notice that this force can be defined by the Rayleigh dissipation function $\mu p|\dot{x}|$.

3^o/ Let's now consider the differential equation

$$m\ddot{x} + k\,x + c\,\mathrm{sign}(\dot{x}) \ni f(t) \tag{2.74}$$

which models the dynamical behavior of a particle submitted to the conditions depicted in figure (Fig.2.35). If we discretize this equation by the Euler implicit method, knowing the displacement x_t and the velocity v_t at time t, we have to solve the variational inequality

$$(m + h^2 k)\,v_{t+h} + ch\,\mathrm{sign}(v_{t+h}) \ni h f(t+h) - hkx_t + mv_t \tag{2.75}$$

to define the displacement $x_{t+h} := x_t + hv_{t+h}$ at time $t+h$. Setting

$$F_{t+h} := hf(t+h) - hkx_t + mv_t$$

it is easy to check that v_{t+h}, defined as follows:

$$v_{t+h} = \begin{cases} 0 & \text{if} \quad F_{t+h} \in [-ch, ch] \\ \frac{F_{t+h}+ch}{m+h^2k} & \text{if} \quad F_{t+h} \le -ch \\ \frac{F_{t+h}-ch}{m+h^2k} & \text{if} \quad F_{t+h} \ge ch \end{cases}$$

is solution of (2.75). Now we can run the following program to have a deeper understanding of the hysteretic damping.

- Main program

```
clear all
k=100.0;
m=0.1;
c=80.0; % Damping coefficient
% c=c/10;
T=10.;
h=0.0001; % The time step is small enough to assume the Euler method converged
time=[0:h:T];
f=500*sin(2*time); % Excitation force
f=0*f;
```

```
% x(1)=0;
x(1)=10.0; %Initial condition (displacement)
v(1)=0.0; %Initial condition (velocity)
%
% Numerical integration of equation (2.74)
%
for i=2:size(time,2)
 F=h*f(i)+m*v(i-1)-h*k*x(i-1);
 if abs(F)<=c*h
  v(i)=0.0;
  x(i)=x(i-1);
 elseif F<=-c*h
  v(i)=(F+c*h)/(m+h^2*k);
  x(i)=x(i-1)+h*v(i);
 else
  v(i)=(F-c*h)/(m+h^2*k);
  x(i)=x(i-1)+h*v(i);
 endif
endfor
```

- Numerical simulations:

  1^o/ We first study the equation (2.74) with an identically zero right hand member but with the initial conditions $x_0 = 10$ and $\dot{x}(0) = 0$. The results are plotted and analyzed in figure (Fig. 2.36).

  2^o/ Now we can compute the solution of (2.74) submitted to a harmonic excitation $f(t) = \sin\omega t$ (in this case $\omega = 2$) with the initial conditions $x(0) = \dot{x}(0) = 0$. We see in figure (Fig. 2.37) that the mapping $f \mapsto x$ presents a hysteresis loop.

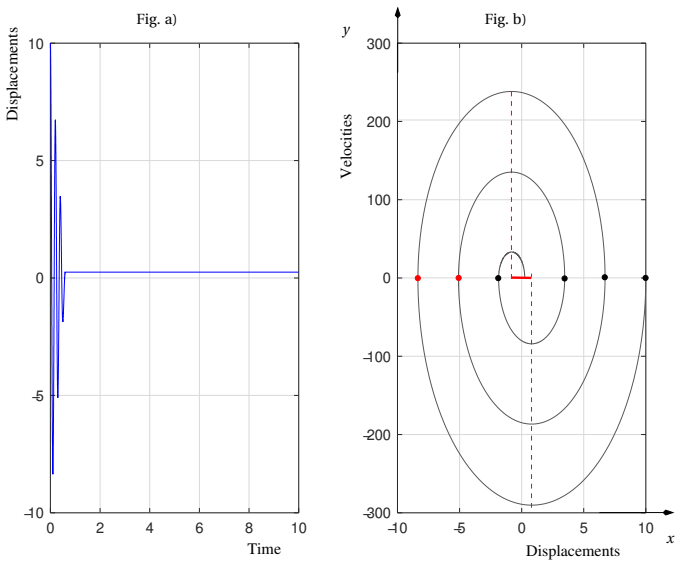


Fig. 2.36. **First simulation.** Displacements in function of time are plotted in figure Fig.a). We see that particle stops after a finite number of oscillations, namely 3 at a position which may be non zero and that the amplitude of the oscillations decreases linearly. This phenomena is explained in figure Fig.b) by integrating the equation (2.74) in the phase plane: Assuming for instance the velocity negative, the motion equation rewrites $m\ddot{x} + k\,x - c = 0$. Multiplying this equation by $\dot{x}$, we see that the particle goes form a black bullet to a red one along a curve of equation $my^2 + kx^2 - cx = Const$ (which is an ellipse centered on the point $(\frac{c}{2k}, 0)$). In the same manner, when the velocity is positive, the particle goes form a red bullet to black one along an ellipse centered on $(-\frac{c}{2k}, 0)$. Arguing so, we verify that the motion stops when the particle reaches the interval $[-\frac{c}{2k}, \frac{c}{2k}]$ of the axis $y = 0$.

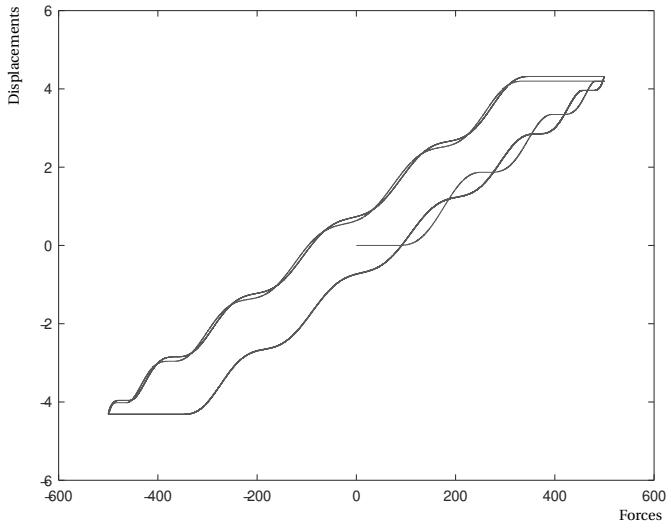


Fig. 2.37. **Hysteresis loop for the Coulomb's damping.** The reader should compare this result with the one that would have been obtained with linear dissipation (ie. if the Rayleigh dissipation is quadratic function of velocity).

CHAPTER 3

IMPLEMENTATION IN STRUCTURE ANALYSIS

THE principle consists to apply a system of forces $F(t)$ defined on a time interval $[0, T]$ (referred to as time horizon T) on a linear model of structure and to calculate the stresses $\sigma_{\alpha\beta}(t)$ at some interesting points. These stresses, which are tensor data, must be reduced to a scalar signal $t \mapsto \Sigma_e(t)$ (referred to as equivalent stress[1]) to which we apply the rain-flow counting algorithm and the Palmgren-Miner's rule.

In other words, we have to set up the following numerical process:

A) solve a second-order differential system, set up in $\mathbb{R}^n$ to fix the ideas

$$[M]\,\ddot{x} + [W]\,\dot{x} + [K]\,x = F(t) \tag{3.1}$$

to compute the displacements $x(t)$ of the structure;

B) compute the equivalent stress $\Sigma_e(t)$, which is actually a function $\Sigma_e(x(.))$;

C) and, according to the results obtained in Chapter 2, the damage is obtained in calculating the integral

$$\mathscr{D} = \int_T^{2T} w\left(\Sigma_e(t), \Sigma_0(t), \dot{\Sigma}_e(t)\right) \left|\dot{\Sigma}_e(t)\right| \, dt \tag{3.2}$$

where

- w is defined by the formula (2.62) page 72, according to the inverse of the number of cycles to failure identified with the help of the Wöhler's abacus;
- and, to simplify the notations, we do not distinguish between the signal Σ_e and its "periodic" counterpart Σ_e^{per} defined on $[0, 2T]$ by the formula (2.24) page 50.

[1] Obtained in computing for instance a deviatoric invariant.

This chapter, organized as follows:

Contents

aims at defining and illustrating with the help of simple examples the algorithms which are to be set up for carrying out the steps A) to C) of the damage computation process. We will see in Chapter 4 a way to *adapt these algorithms to compute the descent direction of the structure optimization problem* introduced in figure (Fig. 1.16) page 26.

3.1. Integration of the state equation

Let's first rewrite the state equations (3.1) as the first-order system (ie. set $y = \dot{x}$)

$$\frac{d}{dt}\left\{\begin{array}{c} x \\ y \end{array}\right\} = -[A]\left\{\begin{array}{c} x \\ y \end{array}\right\} + G(t)$$

(3.3) where $[A]$ is the matrix $\left[\begin{array}{cc} 0 & -I \\ [M]^{-1}[K] & [M]^{-1}[W] \end{array}\right]$

and $G(t)$ the vector $\left\{\begin{array}{c} 0 \\ [M]^{-1}F(t) \end{array}\right\}$

Several methods may be used to solve this system of differential equations:

1^o/ Setting $X = \left\{\begin{array}{c} x \\ y \end{array}\right\}$, *the transient integration methods* consist to approximate the time derivative of the left hand member of (3.3) by the finite difference[2]

$$\frac{dX}{dt}(t_i + h) \approx \frac{X(t_i + h) - X(t_i)}{h}$$

and its right hand member by

$$-[A]\,X(t_i + h) + G(t_i + h) \quad \text{or by} \quad -[A]\,X(t_i) + G(t_i)$$

[2]Called Euler numerical scheme; other discretization methods, such as Newmark methods, implicit or explicit, may be considered but basically they lead to the same conclusions.

according to the numerical scheme (*implicit or explicit*) used to calculate $X(t_{i+1}) := X(t_i + h)$ by one of the following recurrence equations:

$$X(t_{i+1}) = X(t_i) + (I_d + h[A])^{-1} G(t_i + h) \text{ or by} \tag{3.4a}$$

$$X(t_{i+1}) = X(t_i) - h\left([A]X(t_i) - G(t_i)\right) \tag{3.4b}$$

- *the recurrence* (3.4b) *is stable if the time step size* $h := h_{exp}$ *is small enough*: its order of magnitude must be the inverse of the highest natural frequency of the system (3.1)[3]; as *for fatigue analysis, the time horizon is about* 100 *seconds*, this leads to perform between $1.0E^9$ and $1.0E^{10}$ integration steps to compute the criterion;
- *while the recurrence* (3.4a) *is unconditionally stable*, but as it produces a numerical dissipation proportional to the step size, suitable results are obtained for a step size $h_{imp} \approx 100\, h_{exp}$;

2^o/ *An alternative to the above mentioned transient methods* consists to exploit the *analytic solution* (3.5) *of a linear system of differential equations* to compute the solution of the system (3.3); this method will be referred to as *"short time" forced response method*.

We will see in this Section that this method allows

- to significantly reduce the dimension of the state equation by retaining only the relevant eigenmodes with respect to the excitations;
- and to perform the time integration on the same time step size as the sampling of the measured input signals (ie. without any re-sampling of the excitations).

As it is linear, the *state equation can be solved with the help of the analytic formula*

$$X(t) = e^{-[A]t} X_0 + \int_0^t \left[e^{-[A](t-s)}\right] G(s)\, ds \tag{3.5}$$

The objective is to detail a numerical method to compute the integral (3.5); this task is achieved within three steps:

1^o/ The first one is intended for explaining the computations which are to be carried out to solve a second order differential equation submitted to arbitrary excitations.

2^o/ We then show that *if the damping matrix satisfies the Basile's hypothesis*, the system (3.1) decouples in a basis, referred to as *modal basis*.

3^o/ At last, we *compare the performances* of the obtained algorithm to the transient algorithms for the discretization of a system of differential equations.

[3] For standard FEM model, this leads to choose a time step-size $h_{exp} \approx 1.0E^{-7}\, s$.

Uni-dimensional case. In this first step we are focusing on the second order equation, *written in its canonical form*

$$\begin{aligned} &\ddot{\xi}+\omega^2\xi+c\dot{\xi}=f(t) \quad \text{for } t\in[0,T] \\ &\xi(0)=\xi_0 \quad \text{and} \quad \dot{\xi}(0)=\dot{\xi}_0 \end{aligned} \tag{3.6}$$

Setting $\Xi=\left\{\begin{array}{c}\xi\\ \dot{\xi}\end{array}\right\}$, this equation is equivalent to the *first order system*

$$\begin{aligned} &\frac{d\Xi}{dt}=\left[\begin{array}{cc}0&1\\ -\omega^2&-c\end{array}\right]\Xi+\left\{\begin{array}{c}0\\ f(t)\end{array}\right\} \\ &\Xi(0)=\Xi_0 \end{aligned} \tag{3.7}$$

which can be solved with the help of formula (3.5). The eigenvalues of the matrix

$$[A]=\left[\begin{array}{cc}0&1\\ -\omega^2&-c\end{array}\right]$$

are

$$\lambda_1=-\frac{1}{2}\left(c+\sqrt{c^2-4\omega^2}\right) \quad \text{and} \quad \lambda_2=-\frac{1}{2}\left(c-\sqrt{c^2-4\omega^2}\right) \tag{3.8}$$

Introducing the change of bases of matrices[4]

$$[P_1]=\left[\begin{array}{cc}1&1\\ \lambda_1&\lambda_2\end{array}\right] \quad \text{and} \quad [P_2]=\frac{1}{\lambda_2-\lambda_1}\left[\begin{array}{cc}\lambda_2&-1\\ -\lambda_1&1\end{array}\right]$$

the exponential $e^{[A]t}$ is factored as

$$e^{[A]t}=[P_1]\left[\begin{array}{cc}e^{\lambda_1 t}&0\\ 0&e^{\lambda_2 t}\end{array}\right][P_2] \tag{3.9}$$

Formulas (3.9) and (3.5) allow then to perform the time integration of (3.6); all computations carried out we obtain

$$\begin{aligned} \xi(t)&=\frac{e^{\lambda_2 t}-e^{\lambda_1 t}}{\lambda_2-\lambda_1}\dot{\xi}_0+\frac{\lambda_2 e^{\lambda_1 t}-\lambda_1 e^{\lambda_2 t}}{\lambda_2-\lambda_1}\xi_0 \\ &\qquad+\frac{1}{\lambda_2-\lambda_1}\int_0^t\left(e^{\lambda_2(t-s)}-e^{\lambda_1(t-s)}\right)f(s)ds \\ \dot{\xi}(t)&=\frac{\lambda_2 e^{\lambda_1 t}-\lambda_1 e^{\lambda_2 t}}{\lambda_2-\lambda_1}\dot{\xi}_0+\frac{\lambda_1\lambda_2\left(e^{\lambda_1 t}-e^{\lambda_2 t}\right)}{\lambda_2-\lambda_1}\xi_0 \\ &\qquad+\frac{1}{\lambda_2-\lambda_1}\int_0^t\left(\lambda_2 e^{\lambda_2(t-s)}-\lambda_1 e^{\lambda_1(t-s)}\right)f(s)ds \end{aligned} \tag{3.10}$$

Sub-critical damping. When $c^2-4\omega^2<0$ *the damping is said to be sub-critical*; λ_1 and λ_2 are complex conjugate numbers; setting

$$\delta=\frac{1}{2}\sqrt{4\omega^2-c^2}$$

[4]The matrix $[P_1]$ is obtained in writing in columns the eigenvectors of $[A]$ and $[P_2]$ is the inverse of $[P_1]$.

the formulas (3.10) simplify as:

$$
\begin{aligned}
\xi(t) &= \frac{e^{-\frac{c}{2}t}\sin\delta t}{\delta}\dot{\xi}_0 + \frac{e^{-\frac{c}{2}t}\left(c\sin\delta t + 2\delta\cos\delta t\right)}{2\delta}\xi_0 \\
&\qquad + \frac{1}{\delta}\int_0^t e^{-\frac{c}{2}(t-s)}\sin\left(\delta(t-s)\right) f(s)ds \\
\dot{\xi}(t) &= \frac{e^{-\frac{c}{2}t}\left(2\delta\cos\delta t - c\sin\delta t\right)}{2\delta}\dot{\xi}_0 - \frac{\left(2\omega^2 - c^2\right)e^{-\frac{c}{2}t}\sin\delta t}{2\delta}\xi_0 \\
&\qquad + \frac{1}{2\delta}\int_0^t e^{-\frac{c}{2}(t-s)}\left(2\delta\cos(\delta(t-s)) - c\sin(\delta(t-s))\right) f(s)ds
\end{aligned}
\tag{3.11}
$$

REMARKS 3.1 1^o/ If $c = 0$ and $f(t) = \cos\omega_0 t$, the formula (3.11) may be rewritten as

$$
\xi(t) = \frac{\sin\omega t}{\omega}\dot{\xi}_0 + \cos\omega t\,\xi_0 + \begin{cases} \frac{1}{\omega^2-\omega_0^2}\left(\cos\omega_0 t - \cos\omega t\right) & \text{if } \omega_0 \neq \omega \\ \frac{t}{2\omega}\sin\omega t & \text{if } \omega_0 = \omega \end{cases}
$$

We see that:

- the response contains the angular velocity ω_0 of the excitation and the angular velocity ω of the harmonic oscillator, we see on the other hand that the effect of the initial conditions is permanent;
- when the angular velocity of the excitation coincides with the angular velocity of the oscillator, the solution $\xi(t)$, which is bounded on any time horizon $[0, T]$, diverges when T goes to $+\infty$; the divergence ratio being linear with respect to T;
- the response of the oscillator turns its phase when ω_0 crosses the angular velocity of the undamped equation.

2^o/ Assume that $c > 0$ and that the damping remains sub-critical then, always for an excitation of the form $f(t) = \cos\omega_0 t$, the convolution product ξ_c in formula (3.11) is

$$
\begin{aligned}
\xi_c(t) &= \frac{1}{(\omega^2-\omega_0^2)^2 + c^2\omega_0^2}\left(c\,\omega_0\sin\omega_0 t + \left(\omega^2 - \omega_0^2\right)\cos\omega_0 t\right) \\
&\quad + \frac{e^{-\frac{c}{2}t}}{(\omega^2-\omega_0^2)^2 + c^2\omega_0^2}\left(\left(\omega_0^2 - \omega^2\right)\cos\delta t - \frac{c\left(\omega_0^2+\omega^2\right)}{\sqrt{4\omega^2 - c^2}}\sin\delta t\right)
\end{aligned}
$$

This formula shows that $\xi(t)$ has the following asymptotic behavior when t goes to infinity:

$$
\xi_c(t) \approx \xi_p(t) = \frac{\cos(\omega_0 t - \varphi)}{\omega^2\sqrt{\left(1 - \left(\frac{\omega_0}{\omega}\right)^2\right)^2 + \left(\frac{c\,\omega_0}{\omega^2}\right)^2}} \tag{3.12}
$$

where φ is the phase angle defined by

$$
\tan\varphi = \frac{c\,\omega_0}{\omega^2 - \omega_0^2}
$$

We see that

- the impact of the initial conditions decreases exponentially with the time and the response of the oscillator converges to a steady state;

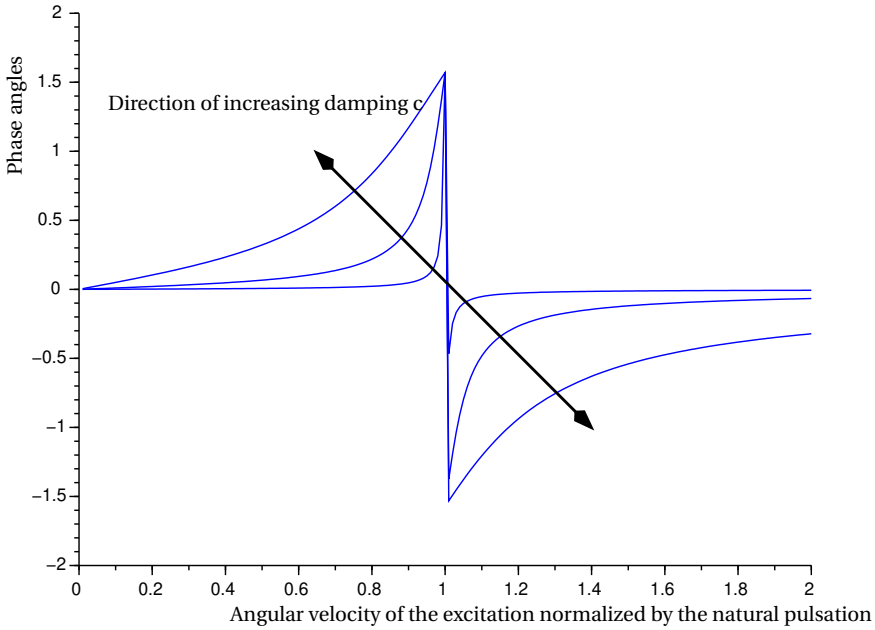


Fig. 3.1. **Evolution of the phase angle φ vs the damping parameter c.** This picture shows phase shifting of the response of the oscillator at resonance crossing. The reversal is all the more sudden as the damping is small.

- the response of the oscillator is (see figure (Fig. 3.1)) out of phase with respect to the excitation and there is a phase reversal when the angular velocity crosses a resonance;
- the amplitude of the response reaches its maximum when the angular velocity of the excitation is $\omega_0 = \sqrt{4\omega^2 - c^2}$;
- introducing the coefficient of dynamic amplification

$$a(\omega_0) = \frac{1}{\sqrt{\left(1 - \left(\frac{\omega_0}{\omega}\right)^2\right)^2 + \left(\frac{c\,\omega_0}{\omega^2}\right)^2}} \tag{3.13}$$

we have $a(\omega_0) \geq 1$ for $\omega_0 \leq \omega$ while $a(\omega_0) \leq 1$ for $\omega_0 \gg \omega$; this means that the amplitude of the response of the oscillator is greater than (resp. is lower than) the amplitude of its quasi-static response before (resp. after) the resonance. We show in figure (Fig. 3.2) the evolution of this coefficient with respect to damping coefficient c.

Super-critical damping. When $c^2 - 4\omega^2 > 0$ the damping is said to be super-critical; the eignvalues λ_1 and λ_2 are both real and negative, to obtain the literal expression of the solution of (3.6), it suffices to replace the trigonometric functions in formulas (3.11) by their hyperbolic counterparts, after having set $\delta = \frac{1}{2}\sqrt{c^2 - 4\omega^2}$. Then, always for an excitation of the form $f(t) = \cos\omega_0 t$, the convolution product ξ_c is

$$\xi_c(t) = \xi_p(t) + \frac{1}{\lambda_2 - \lambda_1}\left(\frac{\lambda_2 e^{\lambda_2 t}}{\omega_0^2 + \lambda_2^2} - \frac{\lambda_1 e^{\lambda_1 t}}{\omega_0^2 + \lambda_1^2}\right)$$

and we see that the effect of the initial conditions decreases to 0 without oscillation.

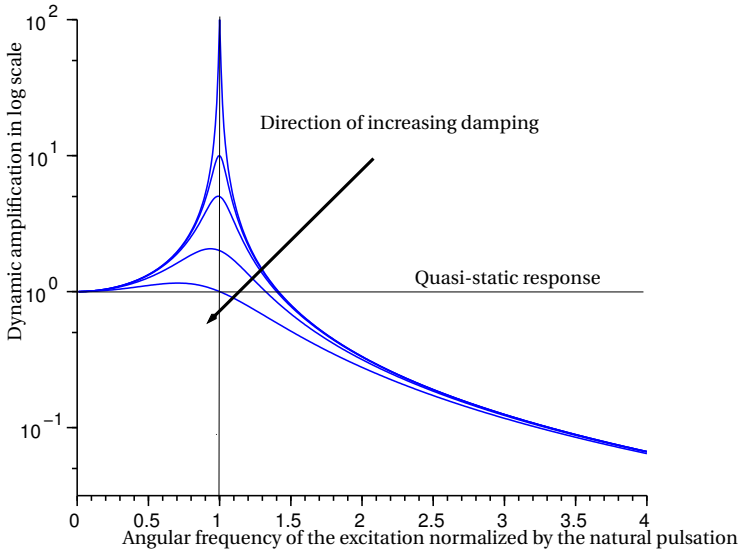


Fig. 3.2. **Evolution of $a(\omega_0)$ as a function of c.**

Thus, *in contrast with what happens for the sub-critical damping, the convergence of $\xi_c(t)$ to the stationary state $\xi_p(t)$ occurs without oscillating,* but the convergence can be very slow.

Critical damping. The damping is said to be critical if $c^2 - 4\omega^2 = 0$. In this case we have $\lambda_1 = \lambda_2 = -\frac{c}{2}$ and the matrix $[A]$ can't be diagonalized. It can however be made triangular of the form

$$[\tilde{A}] = \begin{bmatrix} -\frac{c}{2} & 1+\frac{c^2}{4} \\ 0 & -\frac{c}{2} \end{bmatrix} \quad \text{in the basis} \quad [P] = \frac{1}{\sqrt{1+\frac{c^2}{4}}} \begin{bmatrix} 1 & \frac{c}{2} \\ -\frac{c}{2} & 1 \end{bmatrix}$$

As the matrix $[\tilde{A}]$ is the sum of a diagonal matrix and a nilpotent one, the exponential $e^{[\tilde{A}]t}$ reduces to the matrices product

$$e^{[\tilde{A}]t} = \begin{bmatrix} e^{-\frac{c}{2}t} & 0 \\ 0 & e^{-\frac{c}{2}t} \end{bmatrix} \begin{bmatrix} 1 & \left(1+\frac{c^2}{4}\right)t \\ 0 & 1 \end{bmatrix} = e^{-\frac{c}{2}t} \begin{bmatrix} 1 & \left(1+\frac{c^2}{4}\right)t \\ 0 & 1 \end{bmatrix}$$

then applying the formula $e^{[A]t} = [P]e^{[\tilde{A}]t}[P]^t$ we get

$$e^{[A]t} = e^{-\frac{c}{2}t} \begin{bmatrix} \frac{ct+2}{2} & t \\ -\frac{c^2 t}{4} & -\frac{ct-2}{2} \end{bmatrix}$$

and the solution $\xi(t)$ of the differential equation (3.6) is defined by[5]

$$\xi(t) = e^{-\frac{c}{2}t}\left(t\dot{\xi}_0 + \frac{ct+2}{2}\xi_0\right) + \int_0^t e^{-\frac{c}{2}(t-s)}(t-s)f(s)ds \tag{3.14}$$

This formula shows that effect of the initial conditions decreases along the time. On the other hand, when the excitation is of the form $f(t) = \cos\omega_0 t$ we see that *the convolution product converges to the steady-state $\xi_p(t)$ without any oscillation* and this, faster than in case of super-critical damping.

[5] If $c = 0$ then $\xi(t)$ is the solution of the equation $\ddot{\xi}(t) = f(t)$ satisfying the conditions $\xi(0) = \xi_0$ and $\dot{\xi}(0) = \dot{\xi}_0$.

Convolution formula. When the excitation is arbitrary and given under the form of a sampled signal $\big(f(t_k)\big)_{k=0}^{N_{samp}}$ defined on a time horizon T, the principle consists to replace the sampled signal by the following stair-stepped one, which is piecewise continuous:

$$f_{cont}(t) = \sum_{k=0}^{N_{samp}-1} f(t_k) 1_{[t_k t_{k+1}]}$$

The integral $x(t) = \int_0^t g(t-s) f_{cont}(s) ds$ in then sampled at time t_k by the sum

$$x(t_k) = \sum_{j=0}^{k} f(t_j) \int_{t_j}^{t_{j+1}} g(t_k - s) ds \tag{3.15}$$

If we approximate the integral by

$$\int_{t_j}^{t_{j+1}} g(t_k - s) ds \approx (t_{j+1} - t_j) g(t_k - t_j)$$

and assume that $t_{j+1} - t_j = \delta_{ech}$ is constant, we have $g(t_k - t_j) = g\big((k-j)\delta_{ech}\big)$ which is just the $(k-j)^{th}$ term of a sampling of the convolution kernel, sampled at the frequency $\frac{1}{\delta_{ech}}$. One can thus approximate the sum (3.15) by the following discrete convolution product:

$$\boxed{x(t_k) \approx \delta_{ech} \sum_{j=0}^{k} f(t_j) g(t_{k-j})} \tag{3.16}$$

REMARKS 3.2 1^o/ Lot of fast convolution algorithms are available, see HENRI [**15**], they allow to process extremely long signals so, *they are well suited for the integration of the state equation on a large time horizon.* The most basic of them exploits the fact that the "Fast Fourier Transform" converts convolution products into ordinary products, but in the complex field, and the formula (3.16) may be computed via the following sequence:

- make a FFT of the signals $\big(f(t_k)\big)_{k=0}^{N_{samp}-1}$ and $\big(g(t_k)\big)_{k=0}^{N_{samp}-1}$;
- calculate term by term the product of the fast Fourier transforms;
- recenter the result by framing it with 0, to avoid circular convolution, and apply an inverse FFT to the obtained signal; *the sequence made of the N_{samp} first samples is then the expected sampling* $(x(t_k))_{k=0}^{N_{samp}-1}$ of the convolution product.

2^o/ As the signal $g(t)$ is of the form $e^{-ct}\sin\omega t$, we can reduce the length of the signal $\big(g(t_k)\big)_k$ by restricting it to "the characteristic time" t_N for which the amplitude of the sinusoid is sufficiently attenuated. The sampling frequency of g must however remain the same as that of f.

3^o/ When the angular velocity ω of the kernel g is greater than the largest of the angular velocities found in the excitation signal f, we can apply the formula (3.12) and assume that $x(t) \approx \frac{f(t)}{\omega^2}$, without computing any convolution product; in practice, we can even assume that this term is negligible compared to the terms of lower angular velocities.

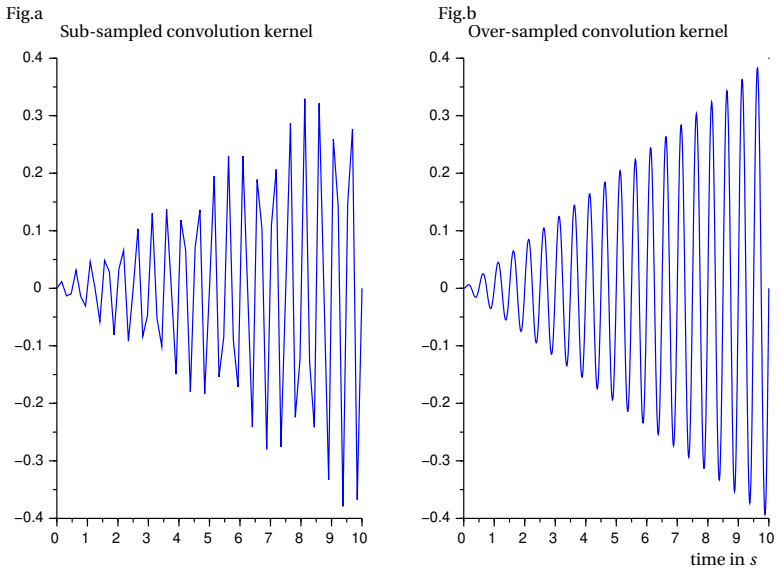


Fig. 3.3. **Integration of an undamped oscillator at resonance.** The figure Fig.a shows the effect of a sub-sampling of the excitation with respect to the characteristic frequency of the convolution kernel. Note on the other hand (on the figure Fig.b) that the numerical integration by convolution allows to accurately reproduce the divergence of the solution of a second order equation excited at the resonance.

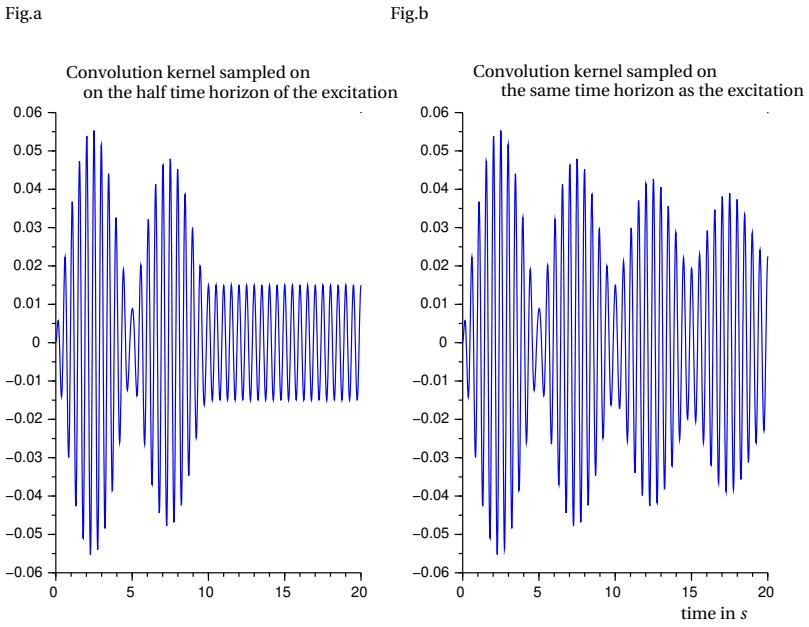


Fig. 3.4. **Integration of a damped oscillator** (1% **of the critical damping)** The figure Fig.a shows the effect of a reduction of the time horizon on which the convolution kernel is sampled; the response of the oscillator is over-damped.

Diagonalization process. When *the damping matrix* $[W]$ *is proportional to the stiffness matrix* $[K]$ *or when it can be diagonalized in the modal basis of the undamped system*[6], one can define a basis $[Q]$ which decouples the oscillators of the equations (3.1).

This basis can be defined as follows[7]:

$1^o/$ The mass matrix $[M]$ being symmetric, it can be diagonalized in an orthonormal basis $[\tilde{Q}]$ so that $[M]$ can be written as $[M] = [\tilde{Q}]^{\ulcorner} m_{ii}{}^{\lrcorner} [\tilde{Q}]^t$. As the mass matrix is positive definite, the entries m_{ii} are positives and we can define a square root[8] $[M]^{-\frac{1}{2}} = [\tilde{Q}]^{\ulcorner} \frac{1}{\sqrt{m_{ii}}}{}^{\lrcorner} [\tilde{Q}]^t$, which satisfies $[M]^{-\frac{1}{2}}[M]^{-\frac{1}{2}} = [M]^{-1}$. Multiplying (3.1) by this square root matrix, we obtain the system of equations

$$[M]^{\frac{1}{2}}\ddot{x} + [M]^{-\frac{1}{2}}[K]x + [M]^{-\frac{1}{2}}[W]\dot{x} = [M]^{-\frac{1}{2}}F(t)$$

$2^o/$ Setting $\tilde{x} = [M]^{\frac{1}{2}}x$ in the previous system, this one can be written

$$\ddot{\tilde{x}} + [M]^{-\frac{1}{2}}[K][M]^{-\frac{1}{2}}\tilde{x} + [M]^{-\frac{1}{2}}[W][M]^{-\frac{1}{2}}\dot{\tilde{x}} = [M]^{-\frac{1}{2}}F(t) \tag{3.17}$$

Since the matrices $[M]^{-\frac{1}{2}}[K][M]^{-\frac{1}{2}}$ and $[M]^{-\frac{1}{2}}[W][M]^{-\frac{1}{2}}$ are symmetric, there is an orthonormal base $[\hat{Q}]$ which diagonalizes both of them under the forms $^{\ulcorner}\hat{k}_{ii}{}^{\lrcorner}$ and $^{\ulcorner}\hat{w}_{ii}{}^{\lrcorner}$. At last, setting $\hat{x} = [\hat{Q}]^t\tilde{x}$ and multiplying the equation (3.17) by $[\hat{Q}]^t$, we get the uncoupled (or diagonal) system of equations

$$\ddot{\hat{x}}_{ii} + \hat{k}_{kk}\hat{x}_{ii} + \hat{w}_{kk}\dot{\hat{x}}_{ii} = \hat{F}_i(t) \tag{3.18}$$

$3^o/$ To diagonalise (3.1), we have at last performed the change of bases

$$\hat{x} = [\hat{Q}]^t[M]^{\frac{1}{2}}x$$

which is not orthogonal. Note that

- the matrix $^{\ulcorner}\hat{k}_{ii}{}^{\lrcorner}$ is made up of the natural frequencies of the undamped system and the basis $[M]^{-\frac{1}{2}}[\hat{Q}]$ is the basis of the associted eigen-modes;
- it is the vector $[M]^{-\frac{1}{2}}F(t)$ which is projected on the modal basis, and not the vector $F(t)$!

REMARKS 3.3 $1^o/$ When the stiffness matrix $[K]$ est is semi-definite, some of the entries $\hat{k}_{kk}$ are zero and the integration formula used to compute the solution of (3.17) corresponds to the critical damping (resp. super-critical damping) case, according to the value, zero or positive, of the corresponding damping coefficient $\hat{w}_{kk}$. For instance, formula (3.14) shows that $t \mapsto \hat{x}_{kk}(t)$ satisfies the differential equation $\frac{d^2\hat{x}_{kk}}{dt^2}(t) = \hat{f}_k(t)$ if $\hat{w}_{kk} = 0$.

[6]This assumption is satisfied when the damping is proportional to the stiffness, mass or to the critical damping per mode.

[7]There are more efficient methods to compute the eigen-values and the eigen-modes of the generalized eigen-value problem $(\lambda M + K)X = 0$, but they use methods of numerical analysis of matrices which are beyond the scope of this course.

[8]If we define the square root of a matrix $[M]$ as a matrix $\sqrt{[M]}$ such that $\sqrt{[M]}\sqrt{[M]} = [M]$, a given semi-definite positive matrix $[M]$ has several square roots, but only one of them is semi-definite positive and is called *principal square root.* For instance the matrices

$$\begin{bmatrix} \sin\theta & \pm\cos\theta \\ \pm\cos\theta & -\sin\theta \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -\sin\theta & \pm\cos\theta \\ \pm\cos\theta & \sin\theta \end{bmatrix} \quad \text{for } \theta \in [0, 2\pi]$$

are symmetric square roots of the identity matrix, but none of them is definite positive.

2^o/ *The modal basis which diagonalizes the system of equations* (3.1) *is orthonormal for the scalar product*

$$(x, x') \mapsto \langle x, x' \rangle_{[M]} := \langle x, [M]x' \rangle$$

3^o/ *The forced response method proposed here*[9] differs from those which are traditionally implemented in the FEM software insofar as it *permits to reproduce the transient states of the dynamic system.* For instance, the coefficients of dynamic amplification are used here only to eliminate from the integration process the oscillators whose the impact on the final result may be considered as negligible. The price to pay for this level of generality is to replace the standard products "of complex numbers" by convolution products, which are more expensive to compute; but the example given in Section 3.2 shows that they must be carried out on a very limited number of oscillators[10].

Integration algorithm. The integration method for the state equation described previously is summarized under the form the algorithm given in figure (Fig. 3.5) and may be implemented in any FEM software.

3.2. Implementation on an example

To illustrate the implementation of the algorithm introduced in figure (Fig. 3.5) we study the example described in figure (Fig. 3.6), which deals with a beam submitted to torsional loads. More specifically, the purpose of the Section is to explain the articulation of the computations allowing to preform the fatigue analysis of a structure which is carried out in the Section 3.3. The step to step programming of this example is given in Annex B.1 page 209.

Continuous model. The state equation governing the shear behavior of a beam of variable cross section is

$$\begin{aligned} & I(s)\frac{\partial^2\theta}{\partial t^2} - \mu\frac{\partial}{\partial s}\left(J(s)\frac{\partial\theta}{\partial s}\right) = m(s,t) \text{ (Equilibrium equation)} \\ & \frac{\partial\theta}{\partial s}(S_i) = \frac{m_i(t)}{\mu J} \text{ for } i = 0,1 \text{ (Torques at the ends of the beam)} \\ & \theta(0,s) = \frac{\partial\theta}{\partial t}(0,s) = 0 \text{ for } s \in [S_0, S_1] \text{ (Initial conditions)} \end{aligned} \tag{3.19}$$

where:

- $\theta(s)$ is the rotation angle of the cross section Σ_s, of abscissa s, around the neutral line;
- μ is the second Lamé coefficient of the constitutive the material;
- $I(s)$ and $J(s)$ are respectively the inertia and the modulus of rigidity in torsion of the cross section Σ_s;

[9] Which could be qualified of "short time integration method".

[10] The integration of the state equation is thus performed on a reduced model; we will see in the Chapter 4 that the same is true for the adjoint equation.

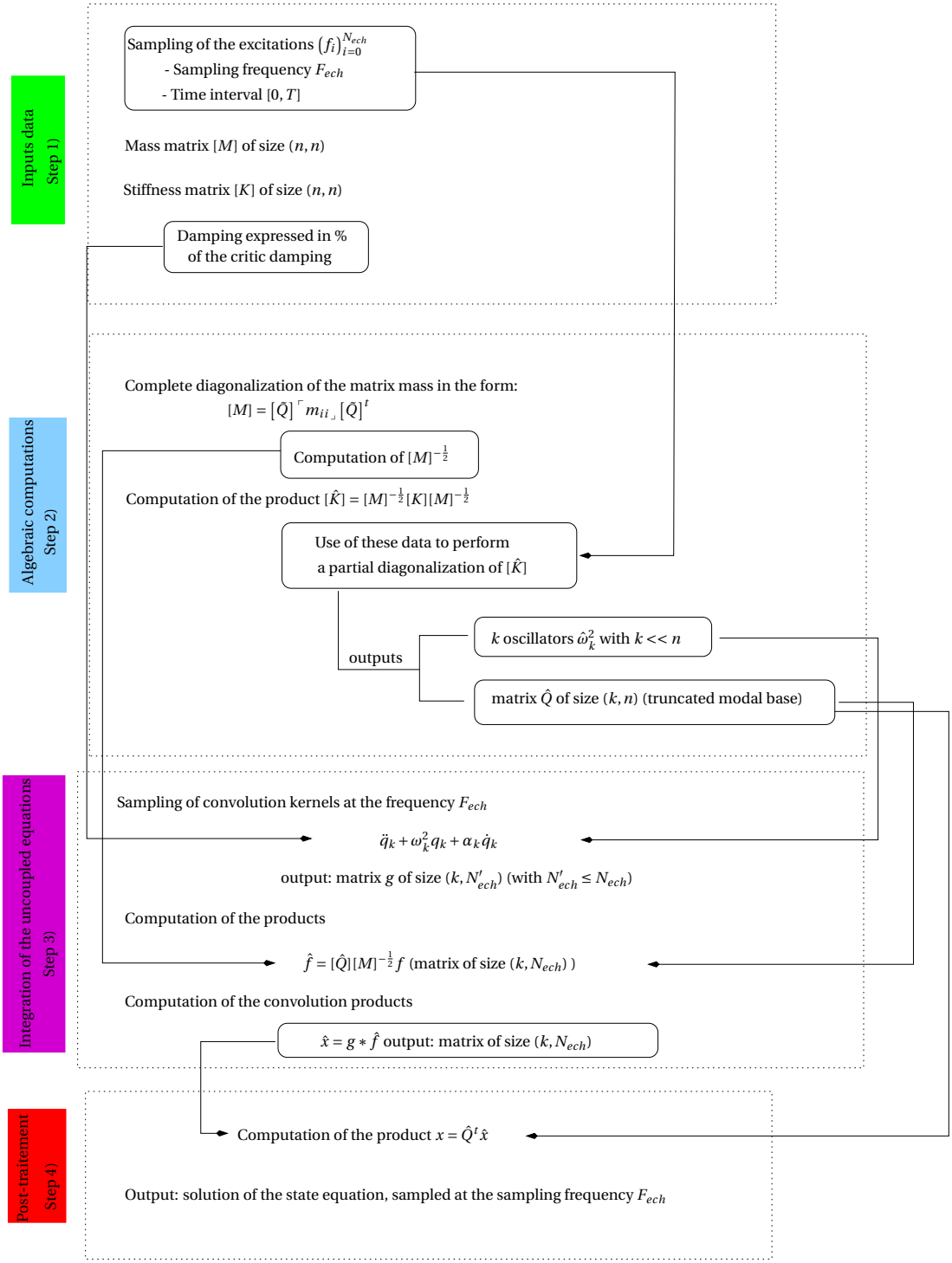


Fig. 3.5. **Integration algorithm for the state equation.** The main difficulty of this algorithm is the choice of the modal truncation; we use the Nyquist-Shannon theorem to eliminate in the modal basis the eigenmodes whose frequencies are higher than $\frac{F_{ech}}{2}$. On the other hand, to reduce the computational cost, we can truncate the convolution kernels at N'_{samp} samples, where N'_{samp} is defined according to the heuristic given in the Remark 3.2-2^o/. Note further that the integration is carried out under the hypothesis $x(0) = \dot{x}(0) = 0$.

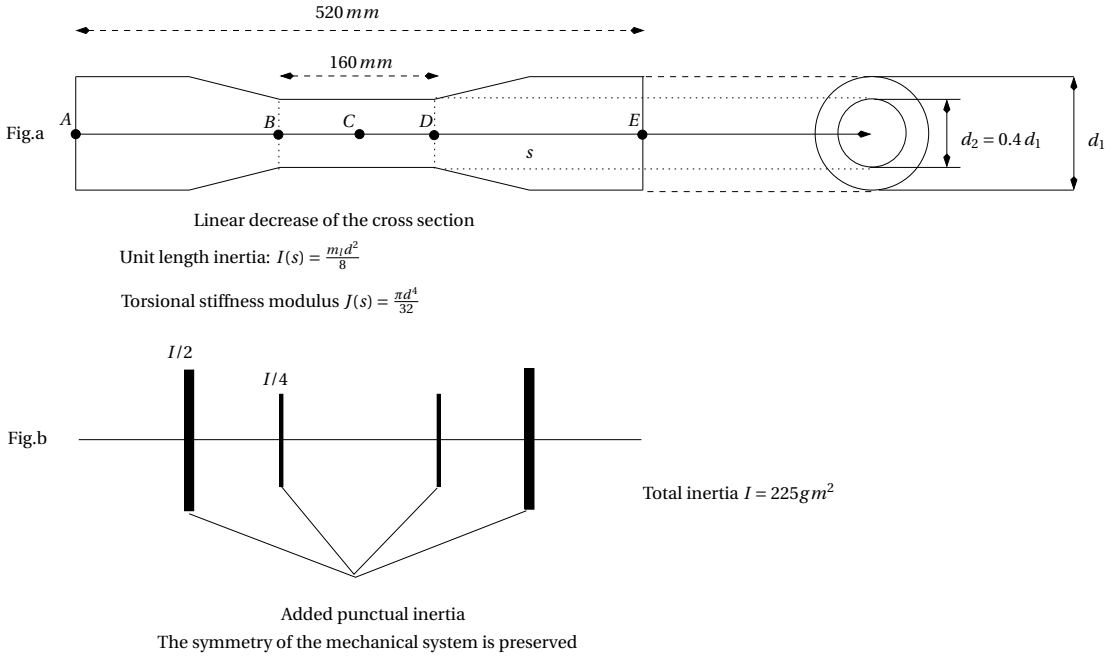


Fig. 3.6. **Description of the mechanical device**. It is a beam of variable cross-section, whose geometric parameters are shown on the figure Fig.a). It is discretized into 26 elements, and submitted at its ends to two opposite torques about the neutral line. We add moreover the 4 punctual inertia shown on the figure Fig.b).

- $m(s,t)$ (resp. $m_i(t)$) is a distributed torque (resp. punctual torque on the ending cross-sections) around the neutral line of the beam.

REMARK 3.4 Within the framework of curvilinear modelling of a three-dimensional body, the cohesion forces are described by distributor tensor fields, whose elements of reduction $\vec{T}(s)$ and $\vec{M}(s)$ satisfy the equilibrium equations

$$\frac{d\vec{T}}{ds}(s) + \vec{f}(s) = 0 \quad \text{and} \quad \frac{d\vec{M}}{ds}(s) + \vec{e}_1 \wedge \vec{T}(s) + \vec{m}(s) = 0$$

where $\vec{e}_1$ is the tangent vector to the neutral line and $\vec{f}$ (resp. $\vec{m}$) is the resultant (resp. the resultant moment) on the neutral line of the tri-dimensional forces applied to the beam.

It can be shown that under these conditions, only the components σ_{12} and σ_{13} of the stress tensor $[\sigma]$ are nonzero in the cross-sections and if they are circular[11], these

[11] It can be shown that in the case of an arbitrary cross-section, the stresses are defined with the help of a stress distribution function ψ by the formulas

$$\sigma_{12} = \frac{M_1}{J}\partial_3\psi \quad \text{and} \quad \sigma_{13} = -\frac{M_1}{J}\partial_2\psi$$

where $J = 2\int_\Sigma \psi$ and ψ is defined by the partial differential equation:

$$\triangle\psi = -2 \text{ in the cross-section } \Sigma$$
$$\psi = 0 \text{ on the boundary } \partial\Sigma$$

In this case, the location the maximum stresses must be defined numerically!

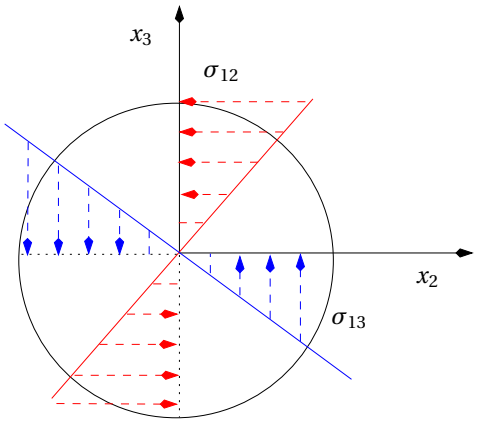


Fig. 3.7. **Stresses in the cross sections of the beam.** When the cross section of the beam is circular, the stresses are maximal in norm on the boundary of the cross sections; but this is generally false for other shapes of the cross section (think for instance to a L shaped cross section).

stresses are, see figure (Fig. 3.7), maximal on the boundary and connected to the strain torsor $\frac{d\theta_s}{ds}$ by the relationships

$$\sigma_{12} = -x_3\mu\frac{d\theta}{ds} \quad \text{and} \quad \sigma_{13} = x_2\mu\frac{d\theta}{ds}$$

Under these conditions, *computation of damage can be performed by processing the variable*

$$\boxed{R\mu\frac{d\theta}{ds} \text{ where } R \text{ is the radius of the cross-section.}} \tag{3.20}$$

which is representative of the maximal shear stresses in the cross-sections.

FEM approximation. By discretizing the equation (3.19) with the help of linear finite elements, we obtain a state equationof the form (3.1) where $[M]$ and $[K]$ are tridiagonal matrices build as follows:

$1^o/$ multiplying the first equation of (3.19) by a test function ψ and integrating the result by part on $[S_0, S_1]$, we see that the equation (3.19) is equivalent to

$$\begin{aligned}\int_{S_0}^{S_1} I\frac{\partial^2\theta}{\partial t^2}(t,s)\psi(s)ds - \int_{S_0}^{S_1}\mu J\frac{\partial\theta}{\partial s}\frac{\partial\psi}{\partial s}ds - \int_{S_0}^{S_1} m(t,s)\psi(s)ds \\ = \mu J\left(\frac{\partial\theta}{\partial s}(S_0)\psi(S_0) - \frac{\partial\theta}{\partial s}(S_1)\psi(S_1)\right) \\ = m_0(t)\psi(S_0) - m_1(t)\psi(S_1)\end{aligned} \tag{3.21}$$

the last equality in this equation reduces to 0 when the beam is not loaded on the ending cross-sections and we obtain the classical variational formulation of the beam equation;

$2^o/$ the discretization of this equation is then performed in introducing a subdivision $(s_i)_{i=0}^N$ of the interval $[S_0, S_1]$ and assuming that θ and ψ are piecewise affine

functions defined by their values[12] $\theta(s_i)=\theta_i$ and $\psi_i(s_i)=\psi_i$ at the points s_i, that is:

$$\theta(s)=\frac{1}{s_{i+1}-s_i}\left(\theta_i(s_{i+1}-s)+\theta_{i+1}(s-s_i)\right) \text{ for } s\in[s_i,s_{i+1}] \tag{3.22}$$

this gives

$$\int_{s_i}^{s_{i+1}}\mu J(s)\frac{d\theta}{ds}\frac{d\psi}{ds}ds=\mu\frac{(\psi_{i+1}-\psi_i)(\theta_{i+1}-\theta_i)}{(s_{i+1}-s_i)^2}\int_{s_i}^{s_{i+1}}J(s)ds$$

which is written in matrix form as

$$(\psi_i,\psi_{i+1})\frac{\mu}{\delta_i^2}\int_{s_i}^{s_{i+1}}J(s)ds\begin{bmatrix}1&-1\\-1&1\end{bmatrix}\begin{Bmatrix}\theta_i\\\theta_{i+1}\end{Bmatrix} \tag{3.23}$$

where $\delta_i=s_{i+1}-s_i$. In the same way, the integral $\int_{s_i}^{s_{i+1}}I(s)\frac{d^2\theta}{dt^2}(s)\psi(s)ds$ is

$$(\psi_i,\psi_{i+1})\frac{1}{\delta_i^2}\int_{s_i}^{s_{i+1}}\begin{bmatrix}I(s)(s_{i+1}-s)^2&I(s)(s_{i+1}-s)(s-s_i)\\sym&I(s)(s-s_i)^2\end{bmatrix}ds\begin{Bmatrix}\ddot{\theta}_i\\\ddot{\theta}_{i+1}\end{Bmatrix} \tag{3.24}$$

at last, the torque $m(t,s)$ is discretized as

$$(\psi_i,\psi_{i+1})\delta_i\int_{s_i}^{s_{i+1}}\begin{Bmatrix}m(s,t)(s_{i+1}-s)\\m(s,t)(s-s_{i+1})\end{Bmatrix}ds \tag{3.25}$$

assuming that $m(s,t)$ is $\frac{m(t)}{\delta_i}$ on the element $[s_i,s_{i+1}]$, the previous formula reduces to

$$(\psi_i,\psi_{i+1})\frac{m(t)}{2}\begin{Bmatrix}1\\1\end{Bmatrix}$$

in the case of a circular beam, whose diameter varies linearly in function of s we have $J(s)=\frac{\pi}{32\delta_i^4}\left(d_i(s_{i+1}-s)+d_{i+1}(s-s_i)\right)^4$ and

$$\int_{s_i}^{s_{i+1}}J(s)ds=\frac{\pi}{160}\left(d_{i+1}^4+d_id_{i+1}^3+d_i^2d_{i+1}^2+d_i^3d_{i+1}+d_i^4\right)\delta_i$$

we can check in the same way that

$$\int_{s_i}^{s_{i+1}}I(s)(s_{i+1}-s)^2ds=\frac{\rho\pi\delta_i^3}{3360}\left(d_{i+1}^4+3d_id_{i+1}^3+6d_i^2d_{i+1}^2+10d_i^3d_{i+1}+15d_i^4\right)$$

$$\int_{s_i}^{s_{i+1}}I(s)(s_{i+1}-s)(s-s_i)ds=\frac{\rho\pi\delta_i^3}{6720}\left(5d_{i+1}^4+8d_id_{i+1}^3+9d_i^2d_{i+1}^2+8d_i^3d_{i+1}+5d_i^4\right)$$

$3^o/$ denoting $[k^i]$ (resp. $[m^i]$) the elementary stiffness matrix (of the i^{th} element) defined in (3.23) (resp. the elementary mass matrix defined in (3.24)) and denoting Ψ

[12] Which implicitly depends upon the time; this means that the solution of (3.21) has separated variables.

	Unit	Numerical value
Young's modulus	daN/mm^2	$0.16500E+05$
Poisson ratio	without	0.3
Mass density	$kg*10^{-4}/mm^3$	$0.79E-09$
Diameter d_1	mm	$4.05E+01$

Tab. 3.1. **Numerical data used for the simulations**. The units may appear strange at first glance, but they allow to express the lengths in mm and forces in (daN) or in kg.

(resp. Θ) the vector obtained in listing columnwise the test functions ψ_i (resp. the unknown θ_i) we can rewrite the equation (3.21) in the form:

$$\Psi^t[M]\ddot{\Theta}+\Psi^t[K]\Theta=\Psi^t F(t) \text{ or any test function } \Psi$$

$$\text{with the initial condition } \Theta(0)=\dot{\Theta}(0)=0$$

where $[M]$ and $[K]$ are built with thee help of the assembly process described below

$$[K]=\begin{bmatrix} k_{11}^1 & k_{12}^1 & 0 & & \\ k_{21}^1 & k_{22}^1+k_{11}^2 & k_{12}^2 & 0 & \\ 0 & k_{21}^2 & k_{22}^2+k_{11}^3 & k_{12}^3 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & k_{12}^n & k_{22}^n \end{bmatrix}$$

$4^o/$ formula (3.22) allows to calculate the derivative $\frac{d\theta}{ds}$ as

$$\frac{d\theta}{ds}=\frac{\theta_{i+1}-\theta_i}{\delta s_i}$$

which is actually the derivative on the right of $\frac{d\theta_i}{ds}$;

$5^o/$ at last, the stress Σ_e is computed in the middle of the element of nodes $p-1$ and p by a formula of the form

$$\Sigma_e(p,t)=(V_p\,;\Theta(t)) \tag{3.26}$$

where V_p is the vector

$$V_p=\frac{\mu(R_{p-1}+R_p)}{2\delta_{s_p}}(\underbrace{0,\cdots,0,}_{p-1}-1,1,0,\cdots,0) \tag{3.27}$$

and R_p is the radius of the beam at the node p.

Numerical computations and mechanical analysis. Applications given in this example are carried out on the basis of the numerical data specified in table (Tab. 3.1) and we assume that the beam to be discretized in 26 elements of identical lengths.

Quasi-static loading. In this case the beam is loaded on the endings elements by two opposite torques at $\frac{200}{\delta s}\,dN*m$ per mm given in the figure (Fig 3.9-a). The stresses, which are maximal in the segment BD, are plotted in the figure (Fig 3.9-b) for two numerical values of the diameters of the cross sections. Note that:

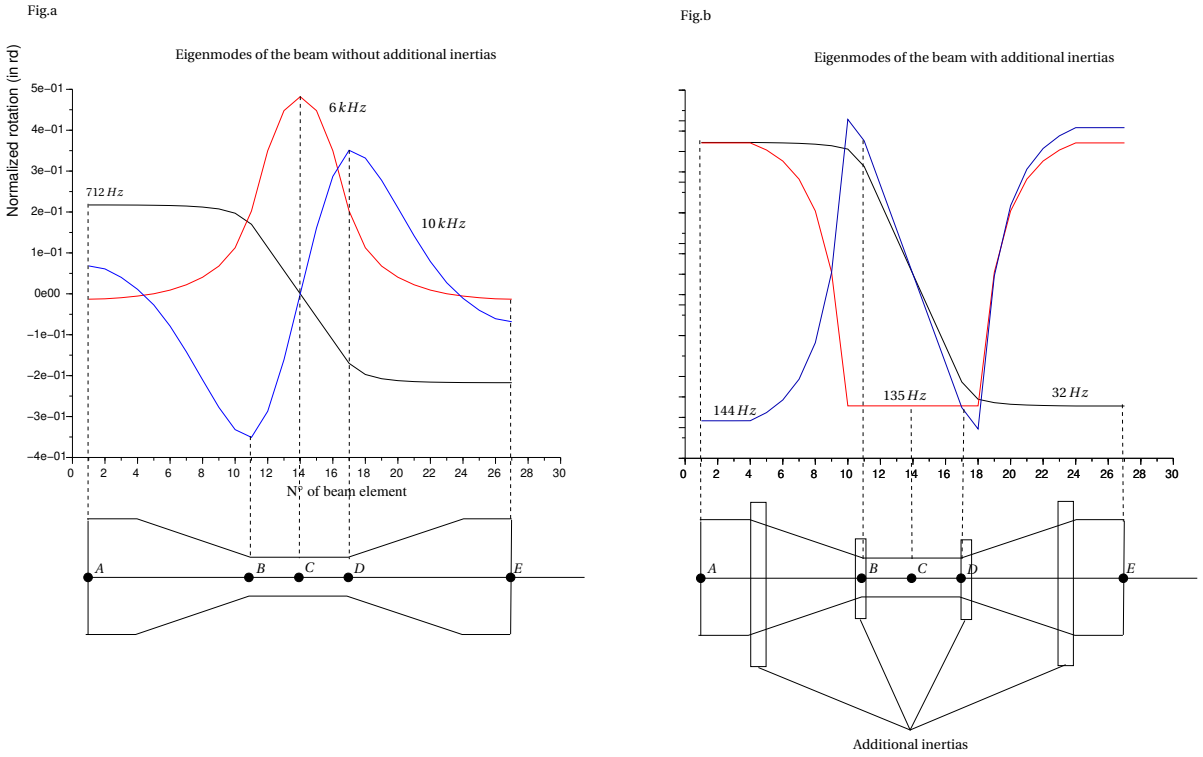


Fig. 3.8. **Eigenmodes of the beam in configurations " with and without additional inertia"**. The three eigenmodes plotted in this figure are global modes of the beam; the mode at 712 Hz (without additional inertia) and at 32 Hz (on the equipped beam) is a global mode (the phases of the torsional vibrations at the points A and E are opposite and these points are antinodes) this mode is particularly well excited by the quasi-static loading plotted in the figure (Fig. 3.9). The modes shown in red (at 6 kHz in Fig.a, at 135 Hz in Fig.b) and in blue (at 10 kHz in Fig.a, at 144 Hz in Fig.b) are deformation modes of the notch; in the absence of additional inertia, they are quite difficult to excite (the vibration nodes of these mode are located near A and E) by the torques at the ends of the beam while they become when the beam gets equipped with the additional inertias.

- *the natural frequencies of a torsion beam depend only on the material coefficients* (second coefficient of Lamé, mass density) and on the length of the beam. The figure (Fig. 3.8) shows that the interest of the additional inertia is to change the natural frequencies and the mode shapes of the beam, in order to enrich the spectrum of the structure at low frequencies;
- although reduction of diameter doesn't change the natural frequencies of the beam, the integration method allows to reproduce the effect of stiffness change;
- and the effects of discontinuous loads are well reproduced; *this aspect of the method is welcomed to integrate in the* Section 4.3 *the adjoint state of the structure optimization problem.*

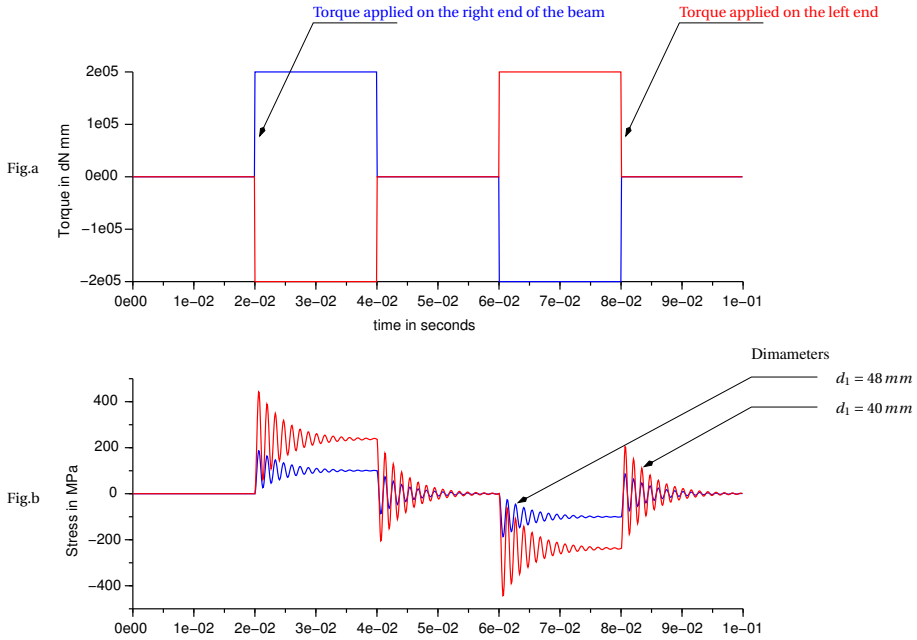


Fig. 3.9. **Simulation of the beam without additional inertia (damping** 20% **of the critical damping)** Figure Fig.a) shows the torques applied on the beam and Figure Fig.b) shows the results of the simulation (stresses at point C) for two given diameters; in this case, the first natural frequency is 712 Hz, the convolutions are performed on 512 samples and computed on the first two oscillators associated with the rigid body and the first eigenmode of the beam; taking into account the chosen sampling frequency (5.12 kHz), the response of the other egeinmodes is quasi-static and the table 3.2 shows that their contribution to global the response of the structure, which may be computed by the formula (3.13), is negligible.

N^o **mode**	**Frequency** Hz	**Generalized stiffness** (rd^2/s^2)	**Damping**
1	0	0	0
2	712	$2.01E+07$	$4.5E+02$
3	$6.7E+03$	$1.8E+09$	$4.2E+03$
4	$10.7E+03$	$4.5E+09$	$6.7E+03$
5	$11.9E+03$	$5.6E+09$	$7.45E+03$

Tab. 3.2. **Characteristics of the** 5 **first modes of the beam without additional inertia (damping:** 20% **of the critical damping)** This table shows for instance that to obtain an accurate response of the third mode, the loads must be sampled at 14 kHz and that before its resonance, the amplitude of the response of this mode is 90 times lower than that of the mode N^o2.

Dynamical loading. In this case we apply the torques given in figure (Fig 3.10) which are obtained from those shown in figure (Fig 3.9.a) by superposition of a pulsed torque at 140 Hz, sampled at 1024 Hz. The stresses are computed at the middle of the beam (node N^o14) and at the node N^o18 of the meshing of plotted in figure (Fig. 3.8). The

N^o mode	Frequency	Stiffness	Damping
1	0	0	0
2	32	$4.17E+04$	$10.21E+00$
3	135.3	$7.22E+06$	$4.25E+01$
4	144.5	$8.24E+06$	$4.53E+01$
5	$8.91E+03$	$3.14E+09$	$2.80E+03$

Tab. 3.3. **Characteristics of the 5 first modes of the equipped beam (5% of the critical damping)** This table shows that to obtain an accurate response of the modes 3 and 4 the loads must be sampled at 300 Hz and to observe the response of the mode N^o1 we have to perform the simulation during at last 0.6 seconds.

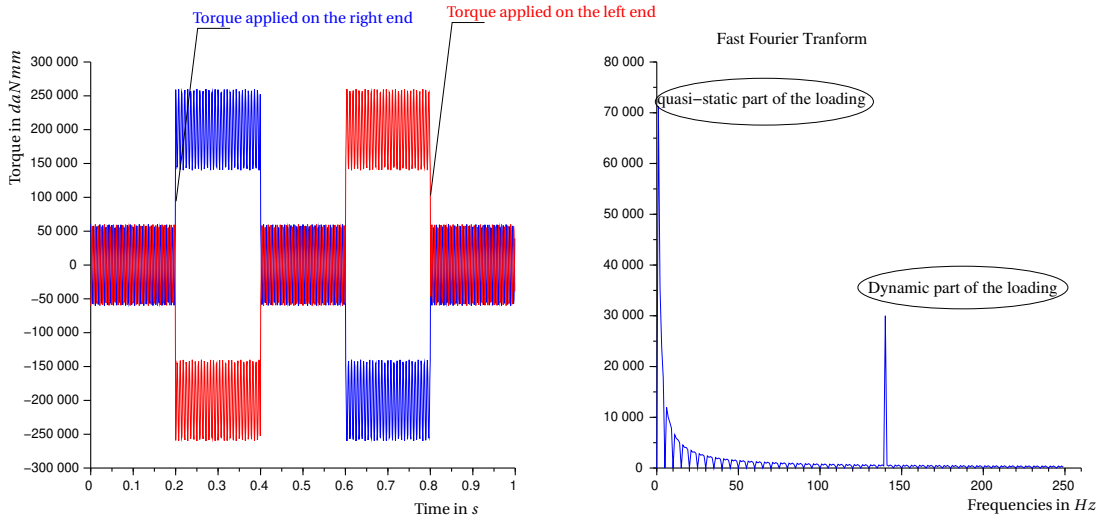


Fig. 3.10. **Dynamic loading of the beam.** As in the case shown on the figure (Fig 3.9.a) the beam is loaded at points A and E by two opposite torques; the magnitude of the pulsed torque at 140 Hz is about 40% of the quasi-static loading.

computations are performed with four oscillators at frequencies 0, 32, 135.3 et 144.5 Hz because, see table (Tab.3.3), the contributions of the other egein-modes to the overall response of the beam are negligible.

Comparison with the transient methods. The comparisons between the methods of integration are performed on the basis of the computations performed within the framework of the previous example, the beam being submitted at its ends by the torques depicted in figure (Fig 3.9-a). The state equation is solved with the help of the forced response method and by the implicit finite difference method (3.4a). Staying at iso-damping, 5% of the critical damping per mode[13], the results of these comparisons are summarized in figures (Fig. 3.12) which show that

[13] According to the Remark 3.1 page 101 critical damping is defined on the basis of the square roots of the eigenvalues λ_i of the matrix $[\tilde{K}] = [M]^{-\frac{1}{2}}[K][M]^{-\frac{1}{2}}$, to define the damping matrix $[W]$ corresponding to a fraction c of the critical damping per mode we have to set

$$[W] = c[M]^{\frac{1}{2}}[Q]^{\ulcorner}\sqrt{\lambda_i}_{\lrcorner}[Q]^t[M]^{\frac{1}{2}}$$

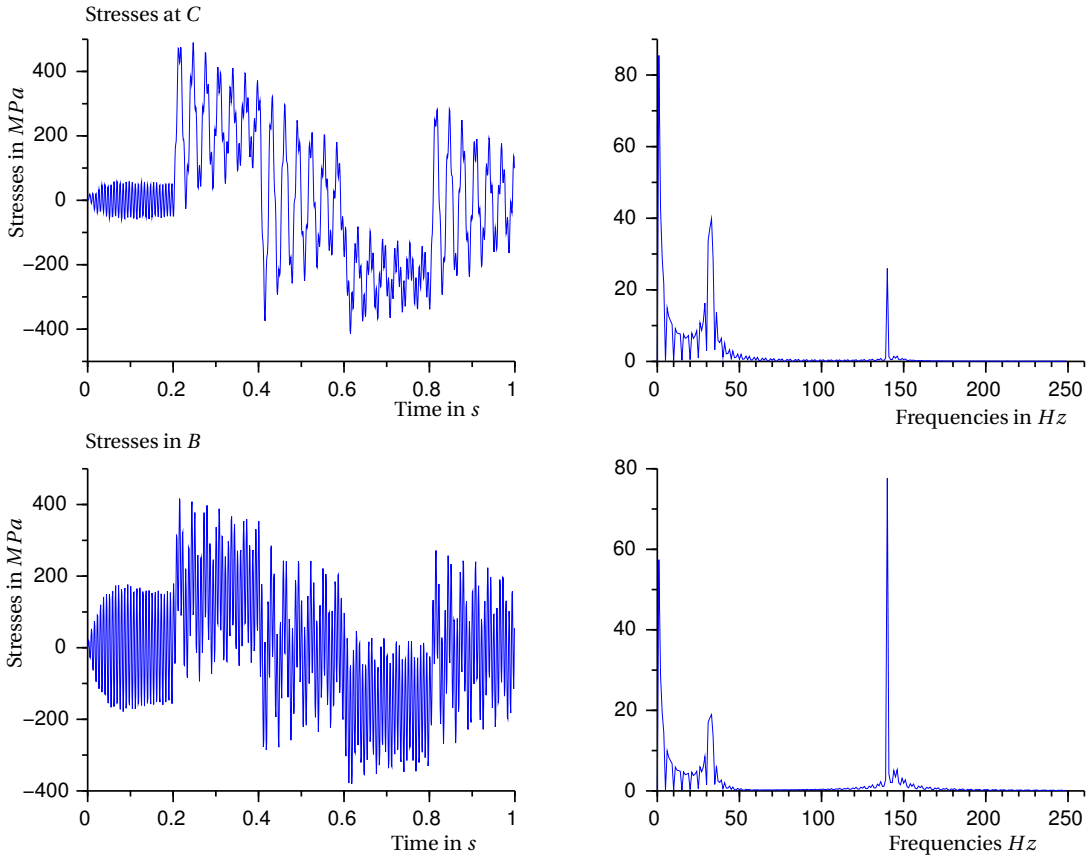


Fig. 3.11. **Computation of the stresses for quasi-static and dynamic loading**. The beam, equipped with its additional inertias, is loaded on its ends by the torques shown on the figure (Fig 3.10). Computations are performed with 5% of the critical damping per mode. We see that the quasi-static component of the loads excites the mode at 32 Hz and generates a high level of stress in C; this is, on the first hand, due to the fact that the mode N^o2, in black on the figure (Fig 3.8), is particularly excited by opposite loads at the ends A and E of the beam and, on the second hand, because C is the point where the strain of the mode N^o2 is maximum. The dynamic part of the loads are quite legible on the stresses computed in B because this point is the point where the strain of the mode N^o4, in blue on the figure (Fig 3.8), is maximum. Notice at last that the mode N^o3, in red, can't be excited by solicitations at the ends of the beam.

the two methods lead to similar results but the implicit finite difference method produces an extra-damping proportional to the step size of the time discretization. As such, it requires a finer sampling of the excitations to converge to the same result as the forced response method.

where $[Q]$ is the modal basis of $[\tilde{K}]$. In other words, $[W]$ is the matrix

$$[W] = c[M]^{\frac{1}{2}} \sqrt{[M]^{-\frac{1}{2}}[K][M]^{-\frac{1}{2}}}\,[M]^{\frac{1}{2}} \tag{3.28}$$

which is semi-definite positive.

	Forced response	**Finite diffrences**
Advantages	– Converge without introducing numerical damping with low sampling of the excitations; this allows the integration of the structural equations on long time horizon. – Not expensive in terms of state variables because the computations are carried out on reduced models with less than 10% of the initial variables. – In the context of structural optimization, *the integration cost of the adjoint equation is identical to that of the state equation.*	– Easy to set up from a digital point of view. – Unconditionally stable in case of implicit scheme – Easily adapts to the treatment of nonlinear problems, especially for the explicit versions.
Drawbacks	– Requires manipulations of complex matrix; examples: SVD, eigenmodes extraction, etc. – Do not generalize to the treatment of all nonlinear equations. – Differentiation of the state equation difficult in the framework of optimization.	– Difficulty to control the numerical damping. – *The integration of the adjoint equation requires a very large volume of data and makes the method inefficient for large systems.*

Tab. 3.4. **Comparison between finite difference and forced response methods for the numerical integration of structural dynamical systems.**

Table (Tab. 3.4) summarizes the advantages and drawbacks of the integration methods to numerically process a problem of vibrations of structures.

3.3. Application to damage computation

Once the state equation solved, the stresses $t \in [0,T] \mapsto \Sigma_e(t)$ *computed and sampled* by the methods outlined in the previous Section, it remains to *define a calculation method of the damage* in some parts of the structure. The Theorem 2.1 page 43 shows that this can be carried out in computing the total variation $\mathscr{D}(\Sigma_e)$ of the function[14] $t \in [0,T] \mapsto \mathscr{H}_\mu(\Sigma_e^{per})(t+T)$ where μ is defined, from the Wöhler's curves of the material, by the formula (2.11) page 43.

Then the Theorem 2.3 page 72 defines, under some regularities conditions on the mapping $t \mapsto \Sigma_e^{per}(t)$, a way to compute this total variation.

Purpose of this Section is to *explain with the help of an algorithm the computations that must be carried out to calculate the integral* (2.61).

[14] We recall that if v is a numerical mapping defined on $[0,T]$, then v^{per} is defined on $[0,2T]$ by

$$v^{per}(t) = \begin{cases} v(t) & \text{if } 0 \le t \le T \\ v(t-T) & \text{if } T \le t \le 2T \end{cases}$$

this definition makes sense if $v(0) = v(T) = 0$ for instance; as we are talking about a function v which is the solution of a differential equation, this reduces to assume that the excitations of the dynamic system are zero on a sub-interval $[T_0,T]$ of $[0,T]$ and that the damping is sufficient to have $\lim_{t\to T} v(t) < \sigma_d$.

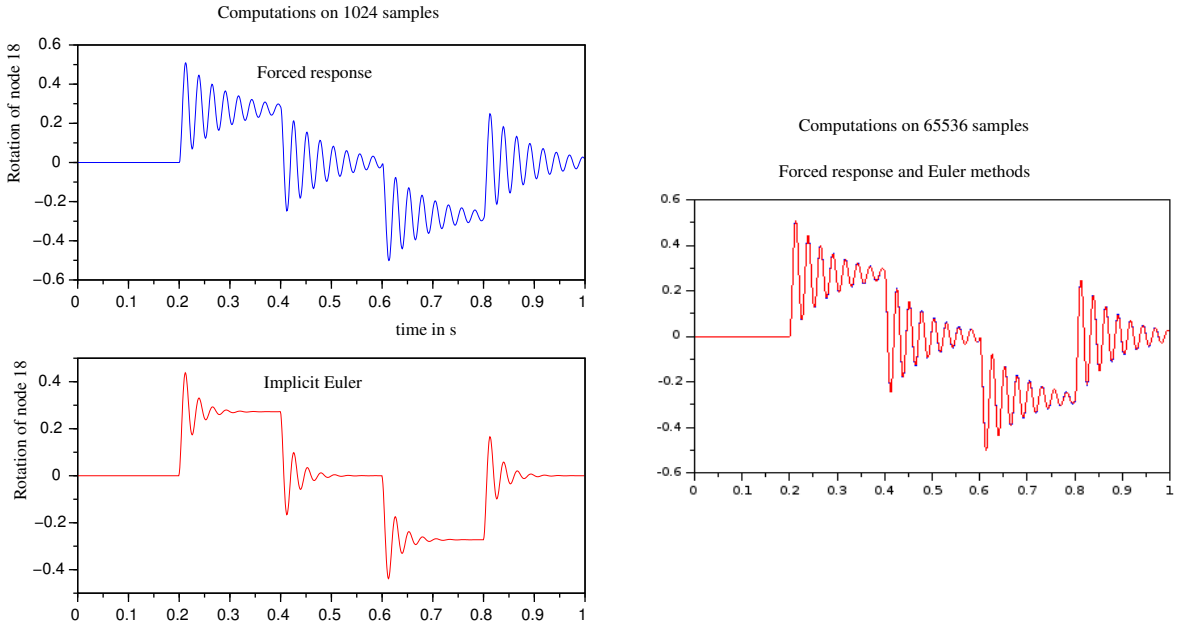


Fig. 3.12. **Comparison between the finite difference method and the forced response method for the integration of the beam equation.** We show in these figures the rotations of the beam computed at B (see figure (Fig. 3.6)) for different time discretizations of the state equation. The blue curves show the computation results for the forced response method while the reds one are the results obtained by the implicit Euler method. We see that 1024 samples are sufficient to obtain the convergence with the forced response method while 64 times more are necessary to achieve convergence for the Euler method. The Euler and the forced response methods converge to the same solution: figure on the right, where the red and blue curves are superimposed.

$1^o/$ The first step consists to identify the "turning-point" of the mapping $t \mapsto \Sigma_e^{per}(t)$; to this end, we have to compute the graph

$$\left\{\left(\sigma_a, \mathscr{E}_{\sigma_a}(\Sigma_e^{per})(t_k)\right) \quad \sigma_a \geq 0\right\}$$

where $\mathscr{E}_{\sigma_a}(\Sigma_e^{per})(t_k)$ is the solution of the equation (2.39) computed at each sampling time t_k of the sampled solution $\Sigma_e^{per}(t_k)$ and $\dot{\Sigma}_e^{per}(t_k)$ of the state equation. This can be carried out in giving a subdivision $\left(\sigma_{a_i}\right)_{i=1}^m$ of the interval $[0, \sigma_{max}]$ and in using the fact that the sampled solution of (2.39) is defined by the recurrence equation (2.50) page 65, this amounts (see the algorithm 3.1) to fill in the matrix $[E]$ defined as follows[15]

$$E_{ij} = \mathscr{E}_{\sigma_{a_i}}(\Sigma_e^{per})(t_j) \quad \text{for } 1 \leq j \leq 2k \quad \text{and } 1 \leq i \leq m$$

$2^o/$ The second step, which is intended for sampling the function $t \mapsto \Sigma_0(t)$, consists to identify as explained in the algorithm 3.2 the first extremum of each column E_j of the matrix $[E]$.

[15]The algorithm only stores the column E_j of $[E]$.

3^o/ The last step consists to compute the integral (2.61) by a trapeze formula[16], and this done by updating the value $\mathscr{D}(\Sigma_e, t_i)$ of the damage in writing that

$$\begin{aligned}\mathscr{D}(\Sigma_e, t_i) &= \mathscr{D}(\Sigma_e, t_{i-1}) \\ &\quad + \delta_t\, w\left(\Sigma_e(t_i), \Sigma_0(T+t_i), \dot{\Sigma}_e(t_i)\right)\left|\dot{\Sigma}_e(t_i)\right| \quad \text{for } 1 \le i \le k\end{aligned} \tag{3.29}$$

where

- w is the mapping defined by the formula (2.62) page 72
- and δ_t is the sampling time of the state equation.

Algorithm 3.3 summarizes the computations which are to be carried out to calculate the damage $\mathscr{D}(v)$ caused by a signal $t \in [0,T] \mapsto v(t)$ with the help of the theoretical results given in the Theorem 2.3.

Algorithm 3.1: *Partial integration of equation* (2.39) *with the help of the recurrence* (2.50) *page 65.*

input :

- Table $(\sigma_{a_i})_{i=1}^m$ containing the sampling points of $\mathscr{E}_{\sigma_a}(v,t)$.
- Sampling of $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,t_k)$ in the column E_1, of size m ;
- Value $v(t_{k+1})$ of $v(t)$ at the sampling time point $t = t_{k+1}$;
- Step size δ_t of the sampling of $v(t)$.

output : Column E_1 updated $E_1 \leftarrow \left(\mathscr{E}_{\sigma_{a_i}}(v,t_{k+1})\right)_{i=1}^m$

begin

  for $i = 1$ **to** m **do**

    $y_1 := \frac{E_1(i)+k\delta_t(v(t_{k+1})+\sigma_{a_i})}{1+k\delta_t}$ and $y_2 := \frac{E_1(i)-k\delta_t(\sigma_{a_i}-v(t_{k+1})}{1+k\delta_t}$;

    if $y_2 - v(t_{k+1}) \le -\sigma_{a_i}$ **then**

      $E_1(i) \leftarrow y_2$

    else

      if $y_1 - v(t_{k+1}) \ge \sigma_{a_i}$ **then**

        $E_1(i) \leftarrow y_1$

      end

    end

  end

end

Practical application. Starting from the example given in Section 3.2 we show how to set up the previously explained algorithms to calculate the damage caused at some critical points of the beam when this one is loaded at the ends by the opposite torques depicted in figure (Fig. 3.10).

1^o/ It is assumed that the Wöhler's curve is defined by a Stromeyer[17] formula with the numerical coefficients:

$$b_s = 0.42\ C_s = 3.6E+09, \text{ with the fatigue limit } \sigma_d = 80\,MPa$$

[16] Which consists to write that $\int_0^T g(t) \approx \delta_t \left[\frac{1}{2}g(x_1) + \sum_{k=2}^{n-1} g(t_k) + \frac{1}{2}g(x_n)\right]$, where $\delta_t = \frac{T}{n}$.

[17] Without accounting for mean stress effect.

Algorithm 3.2: *Computation of* $\Sigma_0(t_k)$. Identification of the first extremum of the table E_1 and sampling of $\Sigma_0(t)$.

input :
- Table $\left(\sigma_{a_i}\right)_{i=1}^{m}$ containing the sampling points of $\mathscr{E}_{\sigma_a}(v,t)$;
- Sampling of the mapping $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v,t_k)$ in the table E_1 of size m ;

output : Function $t \mapsto \Sigma_0(t)$ sampled at time t_k.

begin
  for $i = 1$ **to** $m-2$ **do**
    $p := (E_1(i) - E_1(i+1))\,(E_1(i+1) - E_1(i+2))$;
    if $p \leq 0$ **then**
      Break : exit of the loop
    end
  end
  $\boxed{\Sigma_0(t_k) \leftarrow \sigma_{a_{i+1}}}$
end

Algorithm 3.3: *Computation of the damage caused* by a sampled signal $(v(t_i))_{i=1}^{n}$, defined on a time interval $[0,T]$.

input :
- Sampling $\left(v(t_i)\right)_{i=1}^{n}$ of v and $\left(\dot{v}(t_i)\right)_{i=1}^{n}$ of $\dot{v}$;
- Sampling $\left(\sigma_{a_j}\right)_{j=1}^{m}$ he sampling points of the function $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(v)(t_i)$;
- Material coefficients of the Wöhler's curves (see the Examples 2.2 page 74).

output : Damage $\mathscr{D}(v)$ caused by the loading $t \in [0,T] \mapsto v(t)$.

begin
  Initialize the variable E_1 at 0 ;
  for $i = 1$ **to** n **do**
    - **Update** $\mathscr{P}_1 \leftarrow \mathscr{E}_{\sigma_a}(v)(t_i)$ by the algorithm 3.1
    - **Compute** σ_0 by the algorithm 3.2

  end
  Initialize the damage $\mathscr{D}(v)_0$ at 0
  for $i = 1$ **to** n **do**
    - **Update** $E_1 \leftarrow \mathscr{E}_{\sigma_a}(v)(t_i)$, which is now $\mathscr{E}_{\sigma_a}(v^{per})(T+t_i)$
    - **Compute** σ_0, which is $\sigma_0(T+t_i)$
    - **Set** $v_1 := v(t_i)$, $v_2 := \sigma_0$, $v_3 := \dot{v}(t_i)$ and **compute the integrand** $w(t_i) = w(v_1,v_2,v_3)|\dot{v}(t_i)|$ by the formula (2.62) page 72

    if $i \neq n$ **then**
      Update $\mathscr{D}(v)_i \leftarrow \mathscr{D}(v)_{i-1} + \delta_t\, w(t_i)$
    else
      $\mathscr{D}(v)_n \leftarrow \mathscr{D}(v)_{n-1} + \frac{\delta_t}{2}\, w(t_n)$
    end
  end
end

2^o/ The damage generated on the elements of the beam is computed in applying the algorithm 3.3 on the signals $t \mapsto \Sigma_e(p,t)$ and $t \mapsto \dot{\Sigma}_e(p,t)$ defined by the formulas (3.26); the mapping $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(\Sigma_e, t_k)$ is sampled on the 100 points $\left(\sigma_{a_i}\right)_{i=1}^{100}$ equally distributed between 0 and $1.1 \max_k |\Sigma_e(t_k)|$. A practical implementation

is given in the Steps 6) and 7) of the program whose listing is given in Annex B.1 page 209.

3^o/ The numerical applications depicted in figures (Fig. 3.13), (Fig. 3.14) and in figure (Fig. 3.15) show that

- when the angular velocity of the pulsed torque approaches that of an eigen-frequency of the structure, the damage is maximal at the vibrating loops of this mode (here zones 2 of the beam) and that the modifications affecting the dynamical behavior of the structure are the most relevant: decrease by 10% the additional inertias is less impacting than increasing by the same amount the diameter of the notch ;
- if the response of the beam is obtained by quasi-static computation (this is the case for the simulations shown on the figure (Fig. 3.15)) the damage is maximal in the zone 3 and the damage levels achieved in zone 2, see figure (Fig. 3.13), are not obtained.

3.4. Exercises and complements

EXERCICE 3.1 (Classical results on numerical schemes for the integration of an ODE)

1^o/ Assume that $t \mapsto x(t)$ is a solution of the differential equation $\dot{x} = f(t,x)$; let x_{t_i} be the output of the Euler algorithm, compute the difference $x(t_i) - x_{t_i}$ as a function of $x(t_{i-1}) - x_{t_{i-1}}$ and deduce an estimation of $\|x(t_i) - x_{t_i}\|$ as a function of the discretization step size h.

2^o/ Define and compare the stability properties of the explicit and implicit Euler algorithms.

EXERCICE 3.2 Proof the formula (3.5) page 97.

EXERCICE 3.3 (Existence result for the wave equation (3.19) page 105) Use the method of separation of variables to proof an existence result for the equation (3.19). You can first study the following PDE:

$$\begin{aligned} &\frac{\partial^2 u}{\partial t^2} - c\frac{\partial^2 u}{\partial x^2} = m(t,x) \quad \text{for } x \in [0,1] \\ &u(t,0) = u(t,1) = 0 \\ &u(0,x) = f(x) \quad \frac{\partial u}{\partial t}(0,x) = g(x) \end{aligned} \tag{3.30}$$

where c is a positive constant.

EXERCICE 3.4 Use the function “eig” of Matlab to compute the eigen-modes in the algorithm defined in figure 3.5 page 106 and explain the obtained results.

EXERCICE 3.5 Make an iterative algorithm to compute the square root of a symmetric definite positive matrix.

EXERCICE 3.6 Proof the mechanical results given in the footnote 11 page 107.

Solutions & homeworks.

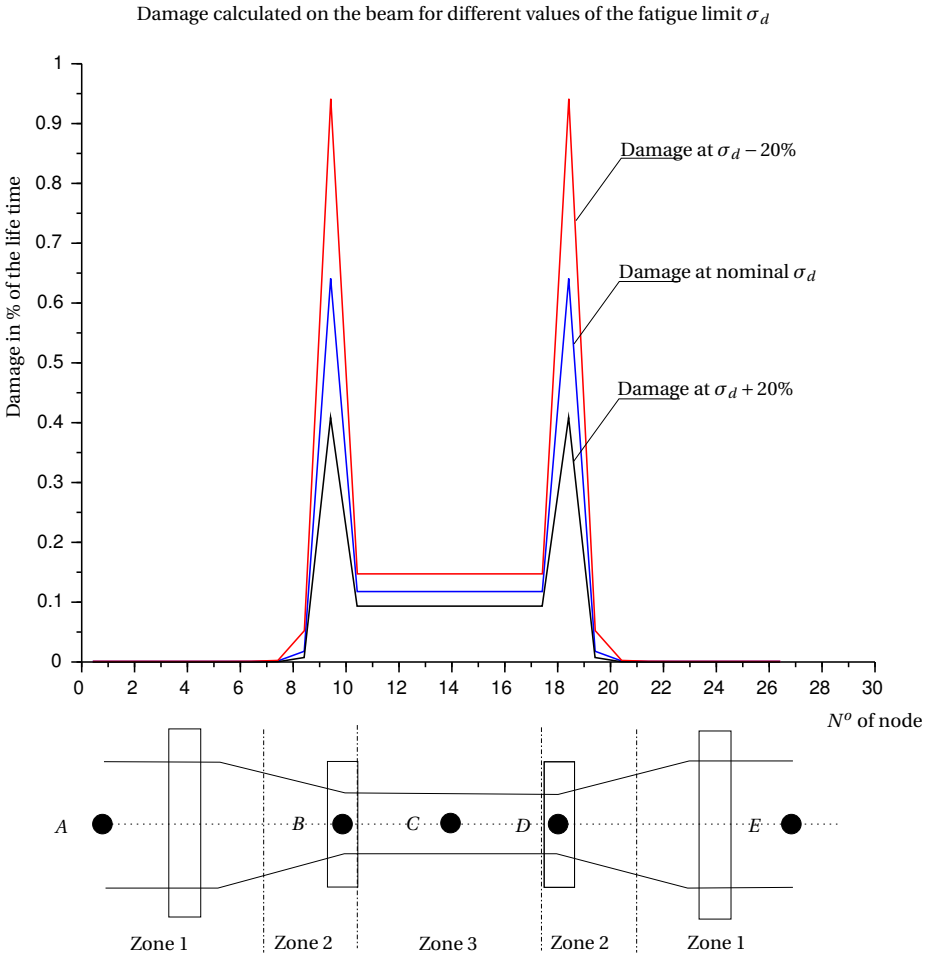


Fig. 3.13. **Computation of the damage along the beam**. This figure shows the damage generated on the beam by the torques given in figure (Fig. 3.10). We see that there are tree interesting zones: the zone 1 where the stress level does not exceed the fatigue limit σ_d and which is not damaged; the zone 2 where, see the figure (Fig. 3.11), the stress level generated by the pulsed term at 140 Hz, which solicits the natural mode N^o3 of the figure (Fig. 3.8),is sufficient to significantly cause damage and the zone 3, where the quasi-static part of the excitations causes the damage. Comparing the curves in blue and in red, for example, one sees on the other hand that the damage in the zone 2 depends primarily on the fatigue limit σ_d because it allows to count in the accumulated damage all or part of high frequency stress cycles. In zone 3 the stress level is high enough to be always above the fatigue limit σ_d.

Solution of exercise 3.1. We first study the case of the explicit Euler method, which consists to discretize the differential equation

$$\dot{x} = f(t,x) \text{ on } [0,T] \tag{3.31}$$

by the numerical scheme

$$x_{t_i} = x_{t_{i-1}} + hf(x_{t_{i-1}}, t_{i-1}) \tag{3.32}$$

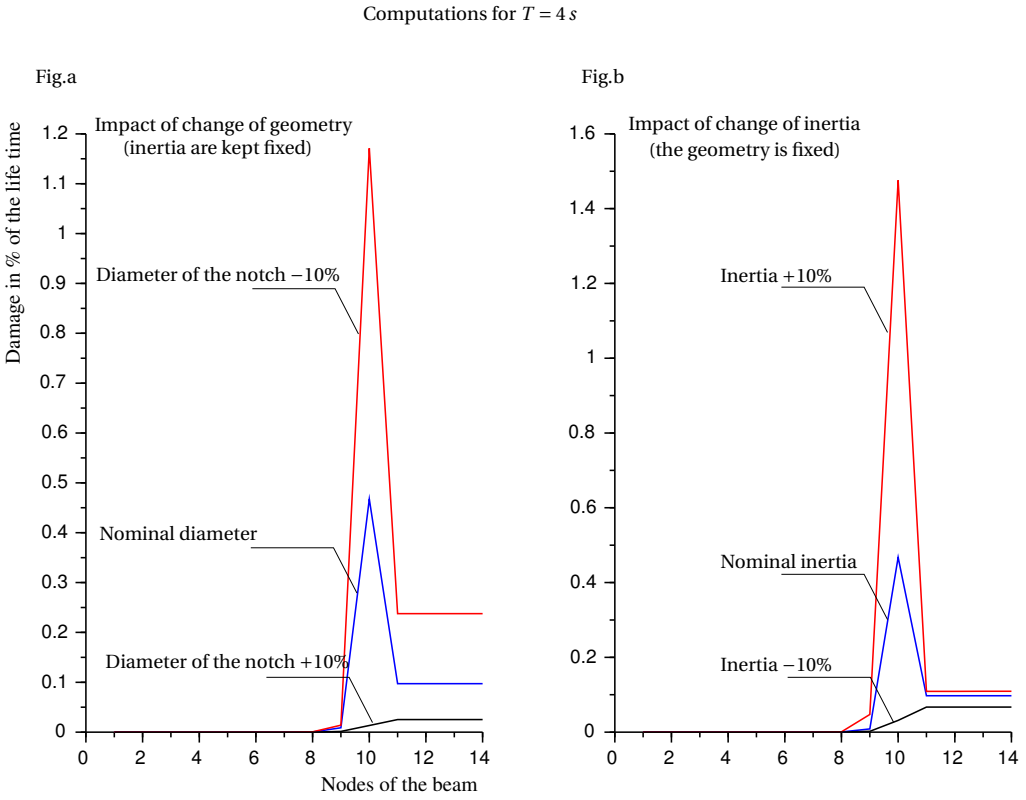


Fig. 3.14. **Impact of design modifications on the damage**. We show on figure Fig.a, the impact of design modifications affecting the stiffness of the beam (diameter of the notch) in this case the modifications impact the damage in the zones 2 and 3 of the beam; the figure Fig.b shows modifications affecting the dynamic behavior of the beam; in this case, the damage in the zone 2 is mainly impacted.

where, $(t_i)_{i=0}^N$ is a sampling sequence contained in $[0,T]$, such that $t_{i+1} - t_i = h$ for $0 \leq i \leq N-1$. In this case, the equation (3.32) can be rewritten as

$$\begin{aligned} x_{t_i} - x(t_i) &= \left(x_{t_{i-1}} - x(t_{i-1})\right) + (x(t_{i-1}) - x(t_i)) + hf(x_{t_{i-1}}, t_{i-1}) \\ &= \left(x_{t_{i-1}} - x(t_{i-1})\right) + (x(t_{i-1}) - x(t_i) + h\dot{x}(t_{i-1})) \\ &\quad - hf(x(t_{i-1}), t_{i-1}) + hf(x_{t_{i-1}}, t_{i-1}) \end{aligned}$$

Assume that $t \in [0,T] \mapsto x(t)$ is a solution of the equation (3.31) which is twice differentiable, the Taylor-Lagrange formula shows that there is a positive constant M such that

$$\|x(t_{i-1}) - x(t_i) + h\dot{x}(t_{i-1})\| \leq Mh^2$$

On the other hand, if $f(.,t)$ is assumed to be uniformly Lipschitz[18] with respect to t, there is a constant C such that

$$\left\|f(x_{t_{i-1}}, t_{i-1}) - f(x(t_{i-1}), t_{i-1})\right\| \leq C\left\|x_{t_{i-1}} - x(t_{i-1})\right\|$$

[18]There is a constant $C > 0$ such that

$$\|f(x,t) - f(y,t)\| \leq C\|x - y\| \text{ for any } t$$

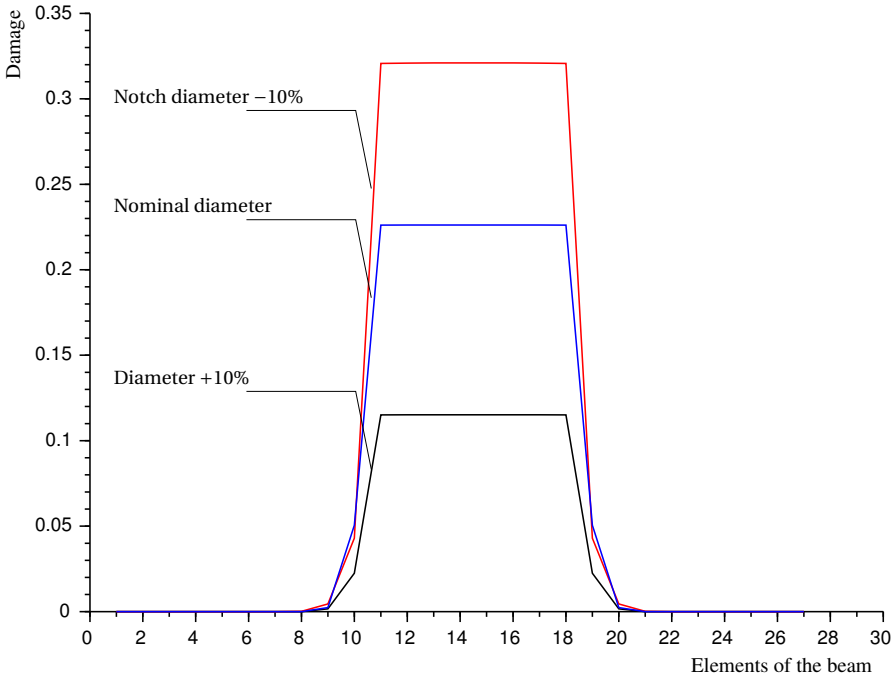


Fig. 3.15. **Quasi-static simulations**. In this case, only the modifications that affect the stiffness of the beam have an effect on the damage, and this justifies to compute the damage with the help of dynamic simulations.

Combining these inequalities, we obtain

$$\left\| x_{t_i} - x(t_i) \right\| \leq (1+Ch) \left\| x_{t_{i-1}} - x(t_{i-1}) \right\| + Mh^2 \tag{3.33}$$

Setting

$$\delta_i = \frac{\left\| x_{t_i} - x(t_i) \right\|}{(1+Ch)^i}$$

the inequality (3.33) can be written as

$$\delta_i - \delta_{i-1} \leq \frac{Mh^2}{(1+Ch)^i}$$

summing these inequalities up to n, we get:

$$\delta_n \leq \delta_0 + \sum_{k=1}^{n-1} \frac{Mh^2}{(1+Ch)^k} = \delta_0 + \frac{Mh^2}{(1+Ch)} \left(\frac{1 - \frac{1}{(1+Ch)^n}}{1 - \frac{1}{1+Ch}} \right) \leq \delta_0 + \frac{M}{C} h$$

and

$$\left\| x_{t_i} - x(t_i) \right\| \leq (1+Ch)^i \left(\delta_0 + \frac{M}{C} h \right) \tag{3.34}$$

Let a number N of samples be given and assume that $h = \frac{T}{N}$, then we have:

$$(1+Ch)^i \leq (1+Ch)^{\frac{T}{h}} \leq e^{\frac{T}{h}Ch} = e^{CT} \quad \text{for any } i \leq N$$

Combining this inequality with (3.34) we obtain the following error estimation for the explicit Euler scheme

$$\boxed{\left\| x_{t_i} - x(t_i) \right\| \leq e^{CT} \left\| x_{t_0} - x(t_0) \right\| + \frac{Me^{CT}}{C} h} \tag{3.35}$$

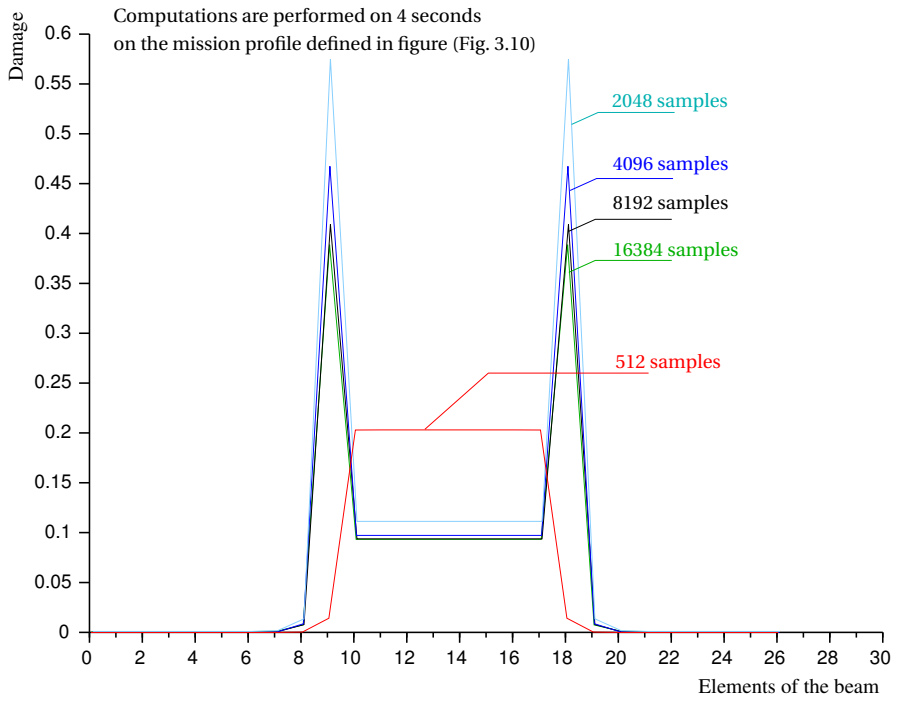


Fig. 3.16. **Computation's accuracy according to the sampling**. This figure shows that computation of damage carried out with 8192 samples on 4 seconds can be regarded as converged (this corresponds to a sampling step size $dt = 5E^{-04}\ s$) but the computations performed on 2048 or 4096 samples can already be considered as satisfactory. Note that 512 samples on 4 seconds are not sufficient to reproduce the frequency 140 Hz of the excitation. The fact that a sub-sampling overestimates the damage is due to the fact that the numerical integration (3.29), by the trapezes method, converges by higher values to the exact integral (2.63) page 74.

A similar inequality can be established for the implicit Euler method.

To conclude, we compare the behaviors of the implicit and explicit schemes for the numerical resolution of the following differential equation

$$\dot{x} = ax \quad x(0) = x_0 \tag{3.36}$$

Let $T > 0$ and a number of samples N be given; define the time step size $h = \frac{T}{N}$, the explicit and the implicit Euler's methods lead to interpolate the solution $x(t_n)$ of (3.36) at time $t_n = \frac{nT}{N}$ by the recurrence equations:

$$\begin{aligned} &x^{exp}_{t_{n+1}} = \left(1 + \frac{T}{N}a\right) x^{exp}_{t_{n-1}} \quad \text{and} \quad x^{imp}_{t_{n+1}} = \frac{1}{1 - \frac{T}{N}a} x^{imp}_{t_{n-1}} \\ &\text{with the initial conditions } x^{exp}_0 = x^{imp}_0 = x_0 \end{aligned} \tag{3.37}$$

so that

$$x^{exp}_T = \left(1 + \frac{T}{N}a\right)^N x_0 \quad \text{and} \quad x^{imp}_T = \left(\frac{1}{1 - \frac{T}{N}a}\right)^N x_0$$

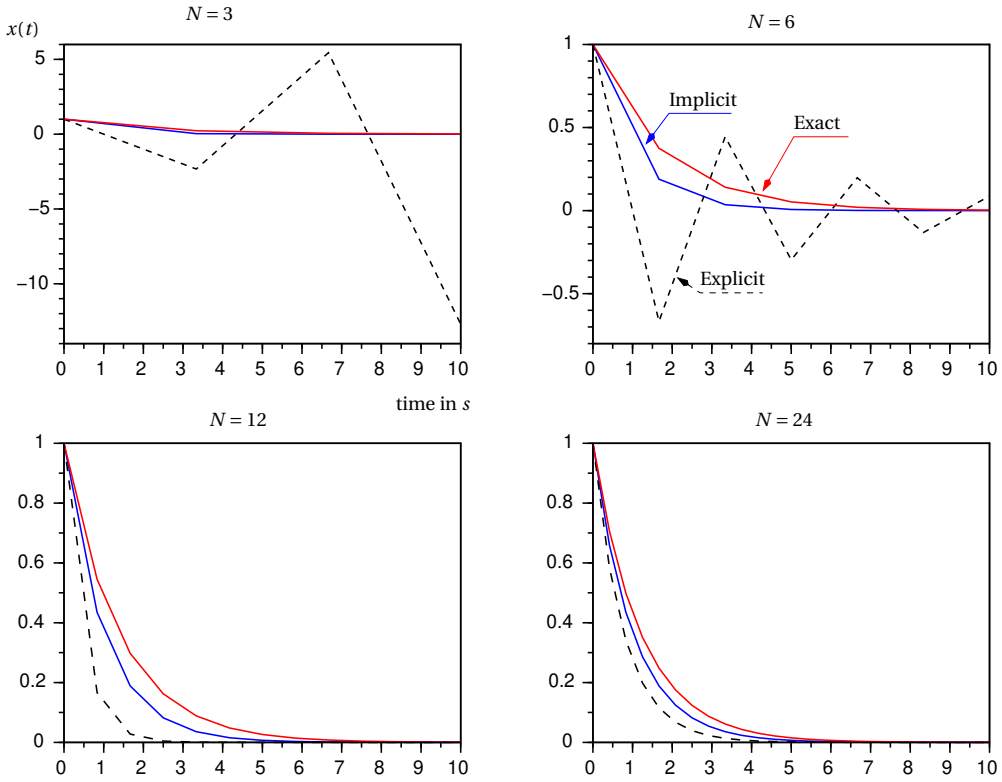


Fig. 3.17. **Comparisons between implicit and explicit schemes**. In this case we assume that $a = -1$ and we compare the outputs of the algorithms (3.37) (plotted in dashed line for the explicit method and in blue for the implicit one) to the exact solution of (3.36), plotted in red. We can see that for $\frac{T}{N} > a$, the outputs of explicit method oscillate about the exact solution of (3.36), so that the *explicit scheme is considered as an unstable integration scheme for the large values the step size* $h = \frac{T}{N}$. Notice that if $a > 0$ the implicit Euler method should be considered as unstable for the large values of the step size h. *Stability of a numerical scheme always refers to the expected behavior of the equation we wish interpolate.*

and we can easily check that

$$\lim_{N\to\infty} x_T^{exp} = \lim_{N\to\infty} x_T^{imp} = e^{aT} x_0$$

is the exact solution of (3.36) at time T. We show in figure (Fig. 3.17) that the approximate solution obtained by the explicit scheme can oscillate about the exact solution and *cannot be used as a numerical approximation for the solution of the differential equation* (3.36).

Homeworks.

1^o/ Do the same stability analysis for the equation

$$\ddot{x} + kx = \sin\omega t \quad x(0) = 0$$

and explain the footnote 3 page 97.

2^o/ Compute the order of convergence of the following modified Euler method

$$x_{t_i} = x_{t_{i-1}} + \frac{h}{2}\left(f(x_{t_{i-1}}, t_{i-1}) + f(x_{t_i}, t_i)\right)$$

3^o/ In the case of the implicit Euler method, the difficulty is to solve the non linear equation

$$x_{t_i} - hf(x_{t_i}, t_{i-1}) = x_{t_{i-1}} \tag{3.38}$$

to compute x_{t_i} as a function of $x_{t_{i-1}}$. Show that if h is small enough and if $f(.,t)$ is Lipschitz continuous, then the equation (3.38) has one and only one solution which can be obtained with the help of a fixed point algorithm.

Solution of exercise 3.2. The exponential of a matrix $[A]$ is defined as the sum of the series

$$e^{[A]t} = \lim_{N\to\infty} \sum_{k=0}^{N} \frac{[A]^k t^k}{k!} \tag{3.39}$$

which is normally convergent (and thus absolutely convergent) for any square matrix $[A]$ and $t \in \mathbb{R}$. The derivative of the mapping $t \mapsto e^{[A]t}$ can be defined, at last formally, by

$$\frac{d}{dt}e^{[A]t} = \sum_{k\geq 0} k\frac{[A]^k t^{k-1}}{k!} = \sum_{k\geq 1} \frac{[A]^k t^{k-1}}{(k-1)!} = [A]e^{[A]t} \tag{3.40}$$

Setting $X_1(t) = \int_0^t e^{[A](t-s)} f(s)ds$, we can compute $\Delta_h = \frac{X_1(t+h)-X_1(t)}{h}$ as follows:

$$\Delta_h = \frac{1}{h}\int_0^t \left(e^{[A](t+h-s)} - e^{[A](t-s)}\right) f(s)ds + \frac{1}{h}\int_t^{t+h} e^{[A](t+h-s)} f(s)ds$$

The formula (3.40) shows that the first term of right hand member of the previous equation converges to

$$[A]\int_0^t e^{[A](t-s)} f(s)ds$$

when h goes to 0. By the mean value theorem, one can find $t_h \in [t, t+h[$ such that

$$\int_t^{t+h} e^{[A](t+h-s)} f(s)ds = he^{[A](t+h-t_0)} f(t_h)$$

Combining these results we see that

$$\frac{dX_1}{dt}(t) = \lim_{h\to 0} \Delta_h = [A]X_0(t) + f(t)$$

satisfies $X_1(0) = 0$; to take into account the initial condition $X(0) = X_0$ it remains to set

$$X(t) = e^{[A]t}X_0 + X_1(t)$$

which, due to linearity, satisfies $\frac{dX}{dt} = [A]X + f$.

Homeworks.

1^o/ Proof an uniqueness result for the equation $\frac{dX}{dt} = [A]X + f(t)$; Lemma 4.4 page 167 generalizes this result to non-linear differential equations.

2^o/ Exponential of matrices is defined in the same manner as the exponential for real or a complex numbers but proof that:

i) the relationship $e^{[A]+[B]} = e^{[A]}.e^{[B]}$ is generally false.

- compute the exponentials of the matrices

$$[A]=\begin{bmatrix}0&0&-\omega_1\\0&0&0\\\omega_1&0&0\end{bmatrix}\quad[B]=\begin{bmatrix}0&0&0\\0&0&-\omega_2\\0&\omega_2&0\end{bmatrix}\quad\text{and}\quad[C]=[A]+[B]$$

  where ω_1 and ω_2 are two real numbers; give a geometric interpretation of the obtained result[19];
- proof however that if $[A][B]=[B][A]$ then $e^{[A]+[B]}=e^{[A]}.e^{[B]}$.

ii) Is the mapping $[A]\mapsto e^{[A]}$ invertible in the space $\mathscr{M}_n(\mathbb{R})$ of the real square matrices of order n? If it is the case, make an algorithm to compute the logarithm of a square matrix. Same questions in the space $\mathscr{M}_n(\mathbb{C})$.

Solution of exercise 3.3. Assume that the undeformed configuration of the beam is the interval $[0,1]$, we will solve the partial differential equation

$$I(s)\frac{\partial^2\theta}{\partial t^2}-\mu\frac{\partial}{\partial s}\left(J(s)\frac{\partial\theta}{\partial s}\right)=m(t,s) \tag{3.41-a}$$

$$\frac{\partial\theta}{\partial s}(t,0)=m_1(t)\quad\frac{\partial\theta}{\partial s}(t,1)=m_2(t) \tag{3.41-b}$$

$$\frac{\partial\theta}{\partial t}(0,s)=\theta(0,s)=0\quad\text{for } s\in[0,1]\quad\text{(initial conditions)} \tag{3.41-c}$$

as follows, within the three steps:

1°/ Apply the method of separation of variables to the homogeneous equation

$$\begin{aligned}&I(s)\frac{\partial^2\theta}{\partial t^2}-\mu\frac{\partial}{\partial s}\left(J(s)\frac{\partial\theta}{\partial s}\right)=0\\&\frac{\partial\theta}{\partial s}(0)=\frac{\partial\theta}{\partial s}(1)=0\end{aligned} \tag{3.42}$$

consists to seek solutions of (3.42) under the following particular form:

$$\theta(s,t)=u(s)f(t)$$

Using this formula in (3.42), we get the equations

$$u(s)f''(t)=\frac{\mu}{I(s)}\frac{d}{ds}\big(J(s)u'(s)\big)f(t)\quad\text{and}\quad u'(0)=u'(1)=0$$

which show that (3.42) splits into the ordinary differential equation (3.43-a) and the boundary value problem (3.43-b)

$$f''(t)=\lambda f(t) \tag{3.43-a}$$

$$\begin{cases}\frac{d}{ds}\big(J(s)u'(s)\big)=\lambda\frac{I(s)}{\mu}u(s)\\u'(0)=u'(1)=0\end{cases} \tag{3.43-b}$$

where λ is a real number.

- Solve (3.43-b) amounts to find a mapping

$$s\in[0,1]\mapsto u(s)\in\mathbb{R}\text{ such that }u'(0)=u'(1)=0$$

[19] Hint: proof that the exponentials of the matrices $[A]$ and $[B]$ are rotations in the group $SO(3)$, which is not a commutative group.

satisfying the variational equation

$$\int_0^1 \frac{d}{ds}\left(J(s)u'(s)\right)v'(s)ds - \lambda\int_0^1 \frac{I(s)}{\mu}u(s)v(s)ds = 0 \quad \text{for any mapping } v \text{ defined on } [0,1] \tag{3.44}$$

If we are looking for solutions u in the space[20] $H^1([0,1])$, an integration by parts shows that solve (3.44) with the boundary conditions $u'(0) = u'(1) = 0$ is equivalent to

$$\text{find } u \in H^1([0,1]) \text{ such that } \int_0^1 J(s)u'(s)v'(s)ds + \lambda\int_0^1 \frac{I(s)}{\mu}u(s)v(s)ds = 0 \quad \text{for all } v \in H^1([0,1]) \tag{3.45}$$

As J and I are positive, continuous functions of $s \in [0,1]$ this problem makes sense and we can easily show[21] that it has only the solution $u = 0$ for $\lambda > 0$; the other solutions are defined via the following Lemma[22], which will be proofed later on.

LEMMA 3.1 *There are a decreasing sequence $(\lambda_n)_n$ such that $\lim_n \lambda_n = -\infty$ and a Hilbert basis $\left(\varphi_n\right)_n$ of $L^2([0,1])$, endowed with the scalar product*

$$(u,v) \mapsto < u,v >= \frac{1}{\mu}\int_0^1 I(s)u(s)v(s)ds$$

such that the equation (3.45) *has*

- *only the solution $u = 0$ for $\lambda \in \complement(\lambda_n)_n$*
- *and the non-zero solution $u = \varphi_n$ for $\lambda = \lambda_n$.*

- Deferring $\lambda := \lambda_n$ in (3.43-a) and setting $\nu_n = \sqrt{-\lambda_n}$ we see that the generic form of $f(t)$ is

$$a_n \cos \nu_n t + b_n \sin \nu_n t \quad \text{if } \nu_n \neq 0$$
$$a_0 + b_0 t \quad \text{if } \nu_0 = 0$$

[20] It is the space of square-integrable mapping having square-integrable derivative; endowed with the scalar product

$$(u,v) \mapsto \int_0^1 u(s)v(s)ds + \int_0^1 u'(s)v'(s)ds,$$

the space $H^1([0,1])$ is a Hilbert space.

[21] It is indeed the Euler-Lagrange equation associated with the minimization problem (see Section 4.1 page 140)

$$\mathcal{J}(u) = \inf_{v \in H^1([0,1])} \mathcal{J}(v), \tag{3.46}$$

where $v \in H^1([0,1]) \mapsto \mathcal{J}(v) \in \mathbb{R}$ is the convex mapping

$$\mathcal{J}(v) = \int_0^1 J(s)(v'(s))^2 ds + \lambda\int_0^1 \frac{I(s)}{\mu}(v(s))^2 ds,$$

which is positive if $v \neq 0$. As this functional is corecive for $\lambda > 0$ (ie. there is a positive constant c such that $\mathcal{J}(v) \geq c\|v\|_1^2$) the only solution of the minimization problem (3.46) is $u = 0$.

[22] Which can be easily checked for the equation (3.30).

which allows to define $\theta(t,s)$ (at least formally) by the following series[23]

$$\theta(t,s) = \sum_n (a_n \cos \nu_n t + b_n \sin \nu_n t)\varphi_n(s)$$

where the constants a_n and b_n must be defined according to the initial conditions. For instance if we assume $\theta(0,s) = \theta_0(s)$ and $\frac{d\theta}{dt}(0,s) = 0$, then

$$a_n = \frac{1}{\mu}\int_0^1 I(s)\theta_0(s)\varphi_n(s)ds \quad \text{and} \quad b_n = 0$$

2°/ If we want solve the system (3.41) with the additional hypothesis $m_1 = m_2 = 0$, let's rewrite the equation (3.41-a) as

$$\frac{\partial^2\theta}{\partial t^2} - \frac{\mu}{I(s)}\frac{\partial}{\partial s}\left(J(s)\frac{\partial\theta}{\partial s}\right) = \frac{m(t,s)}{I(s)}$$

and seek a solution under the form

$$\theta(t,s) = \sum_n g_n(t)\varphi_n(s)$$

then computing scalar product $<\theta(t,.),\varphi_m>$ of $s \mapsto \theta(t,s)$ with a basis function φ_m in $L^2([0,1])$ defined in Lemma 3.1, we must have

$$\sum_n \left[\frac{d^2 g_n}{dt^2}(t)\int_0^1 \frac{I(s)}{\mu}\varphi_n(s)\varphi_m(s)ds - g_n(t)\int_0^1 \frac{d}{ds}\left(J(s)\varphi_n'(s)\right)\varphi_m(s)\right] = \frac{1}{\mu}\int_0^1 m(t,s)\varphi_m(s)ds$$

Now use the fact that φ_n is the solution of the variational equation (3.45) for $\lambda = \lambda_n$ to rewrite the previous equation under the form

$$\sum_n (\frac{d^2 g_n}{dt^2}(t) - \lambda_n g_n(t))\delta_{mn} = \frac{1}{\mu}\int_0^1 m(t,s)\varphi_m(s)ds$$

which means that for each m the mapping $t \mapsto g_m(t)$ is solution of the second order differential equation

$$\frac{d^2 g_m}{dt^2}(t) - \lambda_m g_m(t) = \frac{1}{\mu}\int_0^1 m(t,s)\varphi_m(s)ds$$

with the initial conditions $g_m(0) = \frac{dg_m}{dt}(0) = 0$.

3°/ To solve (3.41) in its general form, it remains to make the change of variables

$$\tilde{\theta}(t,s) = \theta(t,s) - \frac{s^2}{2}m_2(t) + \frac{(s-1)^2}{2}m_1(t)$$

which satisfies $\frac{d\tilde{\theta}}{ds}(t,0) = \frac{d\tilde{\theta}}{ds}(t,1) = 0$.

4°/ There are several ways to proof the results stated in Lemma 3.1:

- the first one consists to notice that this Lemma is (see ZETTI [**43**]) just a reformulation of the Sturm–Liouville theory concerning existence results for second order differential equations submitted to end-points conditions;

[23] There is no warranty that this series converges, but in the classical of applications only a finite number of its terms are non-zero.

- but we can also use the regularizing effect of an elliptic PDE to diagonalize the space equation in a suitable Hilbert space, *and as such this method can be generalized to other situations such as plates, shells etc.* Its implementation is however technical because it makes use of Sobolev spaces embedding theorems (see BREZIS [**6**]) and spectral theory of compact operators[24],see DUNFORD & SCHWARTZ [**12**] & [**13**].

PROOF OF LEMMA 3.1. Let V (resp. F) be the space of the mappings $u \in H^1([0,1])$ (resp. $f \in L^2([0,1])$) having zero mean values. *We will proof that for any $f \in F$, the equation*

$$\begin{aligned} &\frac{d}{ds}\left[J(s)\frac{du}{ds}\right] = f \\ &\frac{du}{ds}(0) = \frac{du}{ds}(1) = 0 \end{aligned} \tag{3.47}$$

has one and only one solution u_f in V and that the operator $f \in F \mapsto W(f) = u_f \in V$ is a self-adjoint continuous operator. As the embedding $H^1([0,1]) \subset L^2([0,1])$ is compact[25], the operator W can be seen as a compact operator acting on F so that we can use the standard results of the spectral theory of compact operators to proof that

- the spectrum[26] of W is discrete, with the only possible accumulation point 0 and made up of eigenvalues of finite multiplicities μ_i;
- moreover, the associated eigenvectors φ_i are an orthonormal basis of F.

As this means that the differential equation

$$\frac{d}{ds}\left[J(s)\frac{du}{ds}\right] = \frac{1}{\mu}u \tag{3.48}$$

has only the solution $u = 0$ for $\mu \neq \mu_i$ and the non-zero solution $u = \varphi_i$ for $\mu = \mu_i$, we have proofed the Lemma with the additional condition $\int_0^1 u(s)ds = 0$. Introduce for convenience the continuous linear functional defined on $L^2([0,1])$ by

$$l(u) = \int_0^1 u(s)ds$$

as $(\ker l)^\perp$ is the one-dimensional subspace of $L^2([0,1])$ spanned by the constant mappings, which are solutions of the equation (3.48) for $\frac{1}{\mu} = 0$, we can complete the orthonormal sequence $\left(\varphi_i\right)_i$ defined previously into a Hilbert basis of $L^2([0,1])$ which satisfies the properties stated in the Lemma.

---

[24] Which basically map bounded sequences into a convergent sequences.

[25] From any bounded sequence $(u_n)_n$ for the norm $\|.\|_{H^1}$ we can extract a sub-sequence $(u_m)_m$ which converges for the norm $\|.\|_{L^2}$.

[26] The spectrum of a continuous linear operator T acting on a normed space H is the set

$$\sigma(T) = \left\{\mu \in \mathbb{R} \,/\, \mu I - T \text{ is not invertible}\right\}$$

Notice that if H is not a finite dimensional space, the condition $\mu \in \sigma(T)$ doesn't entail the existence of an non-zero vector $v \in H$ such that $\mu v = Tv$, see for instance the right shift operator

$$(x_1, x_2, \cdots, x_n, \cdots) \mapsto (0, x_1, x_2, \cdots, x_n, \cdots)$$

acting on the space l^2 of the sequences $\left(x_k\right)_k$ such that $\sum_k x_k^2 < +\infty$.

To complete the proof, it remains to show that the equation (3.47) has only one solution in V if the right hand member f is in F ; to this end, we want show that the functional

$$(u,v)\in V\times V\mapsto \int_0^1 J(s)u'(s)v'(s)ds-\int_0^1 f(s)v(s)ds$$

satisfies the hypothesis of the Lax-Milgram Lemma[27], or in other words that there is a constant $c>0$ such that

$$\int_0^1 J(s)\left(u'(s)\right)^2 ds\geq c\|u\|_{H^1}^2 \tag{3.49}$$

If (3.49) were false, we could find a sequence $(u_n)_n$ contained in the unit sphere of V such that

$$\int_0^1 J(s)\left(u_n'(s)\right)^2 ds\leq \frac{1}{n} \tag{3.50}$$

As the unit sphere of V is weakly compact[28], we can assume that there is a subsequence $(u_m)_m$ which is weakly convergent. Setting u the weak limit, if we can proof that (3.50) entails $u'=0$, we get a contradiction because we will have found $u\in V$ which is at the same time zero and in the unit sphere.

Computing the derivative u' of u in the sense of distributions, the following equations must be satisfied for any $\varphi\in C^\infty([0,1])$ such that $\varphi(0)=\varphi(1)=0$.

$$\begin{aligned}\int_0^1 u'(s)\varphi(s)ds&=-\int_0^1 u(s)\varphi'(s)ds\\&=-\lim_m\int_0^1 u_m(s)\varphi'(s)ds\\&=\lim_m\int_0^1 u_m'(s)\varphi(s)ds\end{aligned}$$

Using the inequality (3.50) we see that, setting $c_0=\sqrt{\inf_s J(s)}$, we have

$$\left|\int_0^1 u_m'(s)\varphi(s)ds\right|\leq \|u_m'\|_{L^2}\|\varphi\|_{L^2}\leq\frac{1}{c_0\sqrt{n}}\|\varphi\|_{L^2}$$

and we conclude that $u'=0$ in the sense of distributions. □

Homeworks.

1/ Poof in the same manner an existence result for the damped equation

$$I(s)\frac{\partial^2\theta}{\partial t^2}+C(s)\frac{\partial\theta}{\partial t}-\mu\frac{\partial}{\partial s}\left(J(s)\frac{\partial\theta}{\partial s}\right)=m(t,s) \tag{3.41-a$'$}$$

where $s\mapsto C(s)>0$ is a given damping coefficient.

2/ How to adapt the previous mathematical machinery to proof an existence result for the bending beam equation?

[27] *Let $(u,v)\in H\times H\mapsto B(u,v)\in\mathbb{R}$ a continuous symmetric bilinear form defined on a Hilbert space H and $v\in H\mapsto l(v)\in\mathbb{R}$ a continuous linear form be given. If B is coercice in the sense that there is a constant $c>0$ such that $B(v,v)\geq c\|v\|^2$ for all $v\in H$ then, the variational equation*

$$B(u,v)=l(v)\quad\forall v\in H$$

has one and only one solution $u\in H$.

[28] Because V is a reflexive Hilbert space: it is a closed subspace of $H^1([0,1])$, which is reflexive.

Solution of exercise 3.4. Use for instance the mass and the stiffness matrices defined in step 2) of the algorithm given page 209. You will see that the algorithm explained in figure (Fig. 3.5) and the function "eig" of Matlab lead to the same results. Actually the function "$eig([M],[K])$ " compute the generalized eigenvalues (resp. eigenvectors) of the pencil $([M],[K])$ ie. the vectors v and the complex numbers λ such that:

$$\lambda[M]v-[K]v=0 \quad v\neq 0$$

This is performed in the following way :

$1^o/$ Assume that $[M]$ can be written in the form

$$[M]=[L][L]^t \quad \text{where } [L] \text{ is a lower triangular matrix} \tag{3.51}$$

$2^o/$ then the generalized eigenvalue problem can be written as the symmetric eigenvalue problem

$$\lambda y-[L]^{-1}[K][L]^{-t}y=0 \quad \text{where } y:=[L]^t x \tag{3.52}$$

and solved, for instance, by the Givens-Householder method, which is well suited to the research of selected eigenvalues of a symmetric matrices, for example all the eigenvalues which are in a given interval[29].

The form (3.51) *of the matrix* $[M]$ *is called Cholesky factorization* of $[M]$; we can actually proof the following result:

THEOREM 3.1 *Let* $[A]\in\mathcal{M}_n(\mathbb{R})$ *be symmetric definite positive then there is a lower triangular matrix* $[L]$ *such that:*

$$[A]=[L][L]^t \tag{3.53}$$

Moreover, this factorization is unique if the diagonal coefficients l_{ii} *of* $[L]$ *are assumed to be positive.*

PROOF. As the matrix $[A]$ is symmetric definite positive, the mapping

$$(x,y)\mapsto y^t[A]x:=<x,y>_A \tag{3.54}$$

is a scalar product on $\mathbb{R}^n$, we will denote $\|.\|_A$ its associated Euclidean norm. We can use the Gram–Schmidt process to othonormalize the canonical basis $(e):=(e_i)_{i=1}^n$ of $\mathbb{R}^n$ in the sense of the scalar product (3.54) ; ie. to define a basis $(f):=\left(f_i\right)_i^n$ such that

$$<f_i,f_j>_A=\delta_{ij}$$

As the basis (f) is defined step by step as follows:

$$\begin{array}{ll} g_1=e_1 & f_1=\frac{g_1}{\|g_1\|_A} \\ g_2=e_2-<e_2,f_1>_A f_1 & f_2=\frac{g_2}{\|g_2\|_A} \\ \vdots & \vdots \\ g_i=e_i-\sum_{k=1}^{i-1}<e_i,f_k>_A f_k & f_i=\frac{g_i}{\|g_i\|_A} \\ \vdots & \vdots \end{array}$$

[29] The function "eigs" of Matlab do the job!

the change of bases matrix $[P]$ from the basis (e) to (f) is upper triangular and satisfies

$$[P]^t[A][P] = [Id]$$

The matrix $[L] := [P]^{-t}$ is lower triangular and such that $[A] = [L][L]^t$.

To proof uniqueness of this factorization, assume that $[L_1]$ is an other lower triangular matrix such that $[L][L]^t = [L_1][L_1]^t = [A]$. As the diagonal entries of $[L]$ and $[L_1]$ are assumed to be positive, $[L]$ and $[L_1]$ are both invertible and we have

$$\left([L_1]^{-1}[L]\right)\left([L]^t[L_1]^{-t}\right) = \left([L_1]^{-1}[L]\right)\left([L_1]^{-1}[L]\right)^t = [Id]$$

As the matrix $[L_1]^{-1}[L]$ is lower triangular[30] this equation shows actually that $[L_1]^{-1}[L] = [Id]$. □

REMARK 3.5 (Cholesky's algorithm) Note that, if the matrix $[A]$ is invertible, the diagonal entries l_{ii} of $[L]$ are necessarily positive and we can use the following algorithm to compute step by step the entries l_{ij} of $[L]$. Let's set

$$[L] = \begin{bmatrix} l_{11} & & & \\ l_{12} & l_{22} & & \\ . & . & . & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$

From the equation $[L][L]^t = [A]$ we deduce that

$$a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik}l_{jk}$$

As the matrix $[A]$ is symmetric, we can assume $i \leq j$ and we have to solve the equations

$$a_{ij} = \sum_{k=1}^{i} l_{ik}l_{jk} \quad \text{for } 1 \leq i \leq j \leq n$$

Setting $i = 1$ we have

$$\begin{array}{lll} l_{11}^2 = a_{11} & \Rightarrow & l_{11} = \sqrt{a_{11}} \\ l_{11}l_{21} = a_{12} & \Rightarrow & l_{21} = \frac{a_{21}}{l_{11}} \\ \vdots & & \vdots \\ l_{11}l_{n1} = a_{1n} & \Rightarrow & l_{n1} = \frac{a_{n1}}{l_{11}} \end{array}$$

which permits to compute the first column of $[L]$; the other columns are calculated step by step as follows:

$$\begin{aligned} a_{ii} = \sum_{k=1} l_{ik}^2 \Rightarrow l_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} \\ a_{i(i+1)} = \sum_{k=1}^{i} l_{ik}l_{(i+1)k} \Rightarrow l_{(i+1)i} &= \frac{a_{i(i+1)} - \sum_{k=1}^{i-1} l_{ik}l_{(i+1)k}}{l_{ii}} \\ &\vdots \\ a_{in} = \sum_{k=1}^{i} l_{ik}l_{nk} \Rightarrow l_{ni} &= \frac{a_{in} - \sum_{k=1}^{i-1} l_{ik}l_{nk}}{l_{ii}} \end{aligned}$$

[30] One can easily check that a product of lower triangular matrices remains a lower triangular matrix.

Homeworks.

1^o/ Why can't we use Cholesky's algorithm to proof the Theorem 3.1?

2^o/ Use the Cholesky algorithm to solve the linear equation $[A]X = b$, where $[A]$ is a symmetric definite positive matrix and compute the number of floating point operations needed to implement the method.

3^o/ Reformulate the algorithm given in figure (Fig. 3.5) with a Cholesky factorization of the mass matrix.

Solution of exercise 3.5. The algorithm consists to use the Newton's iterative method to solve equation

$$[X]^2 - [A] = 0 \tag{3.55}$$

where $[A]$ is a given symmetric definite positive matrix. More generally, let's consider a non linear mapping $X \mapsto F(X)$ defined on a normed space V. Basically the Newton's method consists to solve the equation $F(X) = 0$ with the help of the following iterative scheme

$$X_{k+1} = X_k - (DF(X_k))^{-1}.F(X_k) \quad \text{the starting point } X_0 \text{ being given} \tag{3.56}$$

where the derivative $X \in V \mapsto DF(X) \in \mathscr{L}(V,V)$ of F at X is assumed to be invertible. Noticing that the derivative of the mapping $[X] \mapsto F([X]) = [X]^2$ which is defined on the space of square matrices $\mathscr{M}_n(\mathbb{R})$ of order n is the linear mapping

$$[H] \mapsto DF([X]).[H] = [X][H] + [H][X]$$

the implementation of algorithm (3.56) for solving the equation (3.55) consists in the following steps:

$$\begin{array}{l} \text{Let a starting point } [X_0] \quad \text{be given} \\ \text{Solve the equation } [X_k][H_k] + [H_k][X_k] = [A] - [X_k]^2 \\ \text{Set } [X_{k+1}] := [X_k] + [H_k] \end{array} \tag{3.57}$$

If we assume that $[X_k][H_k] = [H_k][X_k]$ then the second step of (3.57) reduces to solve the equation

$$[X_k][H_k] = [H_k][X_k] = \frac{1}{2}\left([A] - [X_k]^2\right)$$

and the square root of $[A]$ can be computed with help of the following (well-known) iterative scheme

$$[Y_{k+1}] = \frac{1}{2}\left([Y_k] + [Y_k]^{-1}[A]\right) \quad [Z_{k+1}] = \frac{1}{2}\left([Z_k] + [A][Z_k]^{-1}\right) \tag{3.58}$$

HIGHAM [**16**] has proofed that if $[X_0] = [Y_0] = [Z_0]$ commute with $[A]$ then the iterations (3.58) are well-defined and the sequences (3.58) converge toward the principal square root of $[A]$; he has moreover pointed out that this method is numerically unstable.

Homeworks.

1^o/ Proof that the solution of the equation $F(X_1) = Y_1$ can be obtained form a solution X_0 of the equation $F(X_0) = Y_0$ by integrating the differential equation

$$DF(X_\lambda).\frac{dX_\lambda}{d\lambda} = Y_1 - Y_0 \tag{3.59}$$

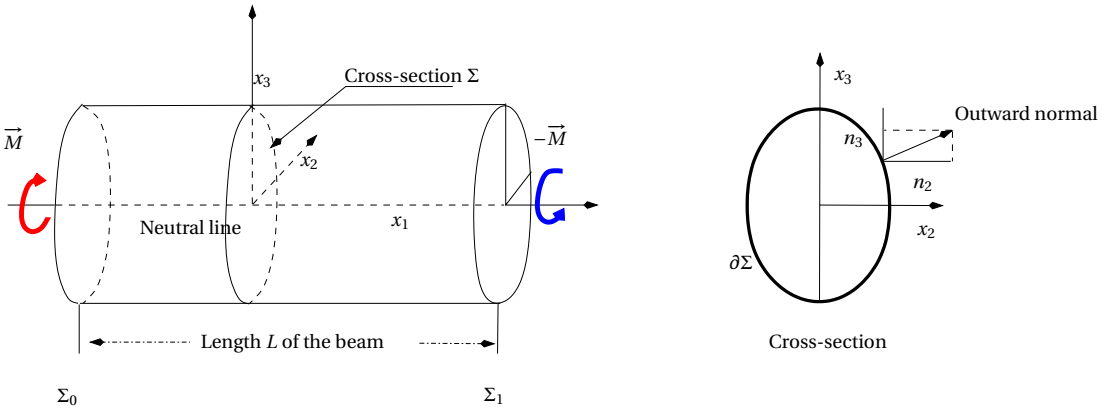


Fig. 3.18. **Torsional loading for a cylindrical beam**. The beam is a cylinder of basis Σ_0 and length L, the centers of gravity of the cross-sections are located on the neutral line and the product moment of area $I_{23} = \int_\Sigma x_2 x_3$ is zero. We assume that the beam is loaded at the ends Σ_0 and Σ_1 by a balanced system of distributed forces having a null resultant and prescribed moments $\vec{M}$ (resp. $-\vec{M}$) on Σ_0 (resp. Σ_1) around the x_1-axis; moreover, the lateral face is assumed to be not loaded.

between 0 and 1, with the initial condition $X_{\lambda=0} = X_0$.

- Use the implicit function theorem[31] to proof that, at last locally (ie. for λ small enough) the equation (3.59) has an unique solution $\lambda \in [0, \varepsilon[\mapsto X_\lambda$;
- use the Euler methods to compute an approximation of X_1;
- compare the obtained algorithm to the Newton iterative method;
- what happens if you try to solve the equation $x^2 = -1$ (for $x \in \mathbb{R}$) with the starting point $x_0 = 1$? Explain the obtained result.

2^o/ Use the alogorithm (3.57) to show that the iterative method

$$\left.\begin{aligned} [P_0] &= [A] \quad [Q_0] = [I] \\ [P_{k+1}] &= \frac{1}{2}\left([P_k] + [Q_k]^{-1}\right) \\ [Q_{k+1}] &= \frac{1}{2}\left([Q_k] + [P_k]^{-1}\right) \end{aligned}\right\} \quad \text{for } k = 1, 2, \cdots, n, \cdots$$

converges to $\sqrt{[A]}$ and $\sqrt{[A]^{-1}}$ respectively; analyze the stability of this method.

Solution of exercise 3.6. We are going to identify the torsion law of an elastic beam by explicitly solving the three-dimensional elasticity equations posed on the cylinder defined in the figure (Fig. 3.18). Assuming that the components of the stress tensor in the plane containing the cross-sections are zero, this lead us to solve the de Saint Venant's problem to calculate the shearing stresses in the three-dimensional body as a function of the torques applied on the ending cross-sections. Then using Hooke's law, we will be able to compute the displacement field in the three-dimensional medium and compare it to the one which would be obtained in applying the beams theory.

[31] Given in footnote 23 page 168.

The de Saint Venant's problem. As the cross-sections are assumed to be undeformable, the stresses in the three-dimensional beam take the particular form

$$[\sigma] = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & 0 & 0 \\ \sigma_{13} & 0 & 0 \end{pmatrix}$$

They satisfy the equilibrium equations

$$\begin{aligned} &\partial_1\sigma_{11} + \partial_2\sigma_{12} + \partial_3\sigma_{13} = 0 \\ &\partial_1\sigma_{12} = 0 \\ &\partial_1\sigma_{13} = 0 \end{aligned} \tag{3.60}$$

and the Baltrami's equations which, in this particular case, can be written as follows:

$$\begin{aligned} \triangle\sigma_{12} + \partial_{11}\sigma_{11} &= 0 \\ \triangle\sigma_{12} + \frac{1}{1+\nu}\partial_{12}\sigma_{11} &= 0 \\ \triangle\sigma_{13} + \frac{1}{1+\nu}\partial_{11}\sigma_{11} &= 0 \\ \partial_{22}\sigma_{11} - \nu\triangle\sigma_{11} &= 0 \\ \partial_{23}\sigma_{11} &= 0 \\ \partial_{33}\sigma_{11} - \nu\triangle\sigma_{11} &= 0 \end{aligned} \tag{3.61}$$

The two last equations of (3.60) show that the stresses σ_{12} and σ_{13} don't depend on x_1 and are defined on the cross-sections of the beam. Then the system (3.61) is a system of linear equations, which links together the second order derivatives of the stresses σ_{1i} ($i = 1,3$), whose resolution shows that the second order derivatives of σ_{11} are 0 and that σ_{11} is the polynomial

$$\sigma_{11} = a_1x_1 + a_0 + (b_1x_1 + b_0)x_2 + (c_1x_1 + c_0)x_3$$

The boundary conditions defined in figure (Fig. 3.18) show that we must have

$$\begin{aligned} &\int_{\Sigma_0} \sigma_{1j} = \int_{\Sigma_1} \sigma_{1j} = 0 \quad \text{for } 1 \le j \le 3 \\ &\int_{\Sigma_0} x_2\sigma_{13} - x_3\sigma_{12} = \int_{\Sigma_1} x_2\sigma_{13} - x_3\sigma_{12} = M \\ &\int_{\Sigma_0} x_3\sigma_{11} = \int_{\Sigma_1} x_3\sigma_{11} = 0 \\ &\int_{\Sigma_0} x_2\sigma_{11} = \int_{\Sigma_1} x_2\sigma_{11} = 0 \end{aligned}$$

These boundary condition provide thus six equations which show that the integration constants a_0, a_1, b_0, b_1, c_0, c_1 are zero and this leads to

$$\sigma_{11} = 0 \tag{3.62}$$

Deferring (3.62) in the first equation of (3.60) we see that the derivatives $\partial_2\sigma_{12}$ and $\partial_3\sigma_{13}$ satisfy the equation

$$\partial_2\sigma_{12} + \partial_3\sigma_{13} = 0$$

If we introduce the potential $\varphi(x_2, x_3)$ defined on the cross-section Σ by

$$\sigma_{12} = \partial_3\varphi \quad \text{and} \quad \sigma_{13} = -\partial_2\varphi \tag{3.63}$$

The Baltrami's equations show that φ satisfies the equations

$$\partial_3\triangle\varphi = 0 \quad \text{and} \quad \partial_2\triangle\varphi = 0$$

which can be integrated to define φ as a solution of the following partial differential equation set up on Σ

$$\triangle\varphi = A$$

where A a constant, which will be defined according to the shape of Σ and M.

As the beam is not loaded on its lateral surface, the normal stresses are zero, thus denoting by (n_2, n_3) the components of the outward-pointing normal vector to the boundary $\partial\Sigma$ of the cross-section, we must have $\sigma_{12}n_2 + \sigma_{13}n_3 = 0$ on $\partial\Sigma$. Using the formulas (3.63), this means that

$$\partial_3\varphi n_2 - \partial_2\varphi n_3 = 0 \tag{3.64}$$

or in other words, that the tangential derivative of φ along $\partial\Sigma$ is 0. If $\partial\Sigma$ is connected, we can even assume that[32]

$$\varphi = 0 \text{ on } \partial\Sigma \quad \text{and} \quad 2\int_\Sigma \varphi = M$$

The stress distribution function $\psi = -\frac{2\varphi}{A}$ defined by

$$\triangle\psi = -2 \text{ in } \Sigma \quad \textit{and} \quad \psi = 0 \text{ on } \partial\Sigma \tag{3.65}$$

depends only on the shape of the cross-section and is such that:

$$M = -A\int_\Sigma \psi$$

Thus, setting

$$J = 2\int_\Sigma \psi \tag{3.66}$$

[32]Equation (3.64) shows that φ is constant on the connected components of the boundary $\partial\Sigma$. Let's introduce the vector field $\overrightarrow{Y}$ defined on Σ by

$$\overrightarrow{Y}(x_2, x_3) = \left\{ \begin{array}{c} 0 \\ \varphi x_3 \\ \varphi x_2 \end{array} \right\}$$

we have $\operatorname{div}\overrightarrow{Y} = 2\varphi + \partial_2\varphi x_2 + \partial_3\varphi x_3$ and, by the divergence formula

$$\begin{aligned} \int_\Sigma \operatorname{div}\overrightarrow{Y} = 2\int_\Sigma \varphi - M &= \int_{\partial\Sigma} Y_2 n_2 + Y_3 n_3 \\ &= \int_{\partial\Sigma} \varphi(x_2 n_2 + x_3 n_3) \end{aligned}$$

this formula shows that we must have

$$M = 2\int_\Sigma \varphi + \sum_k \varphi_k \int_{\partial\Sigma_k} x_3 dx_2 - x_2 dx_3$$

where $(\partial\Sigma_k)_k$ are the connected components of $\partial\Sigma$ and φ_k is the value of φ on $\partial\Sigma_k$. If $\partial\Sigma$ has only one connected component we can assume that $\varphi = 0$ on $\partial\Sigma$ so that $M = 2\int_\Sigma \varphi$.

the stresses in the three-dimensional beam don't depend on x_1 and are defined by

$$\sigma = \frac{M}{J}\begin{bmatrix} 0 & \partial_3\psi & -\partial_2\psi \\ \partial_3\psi & 0 & 0 \\ -\partial_2\psi & 0 & 0 \end{bmatrix} \tag{3.67}$$

REMARK 3.6 We can easily check that if the cross-section is elliptical, of equation

$$x_2^2 + hx_3^2 \leq R^2$$

the stress distribution function ψ is defined by

$$\psi(x_2, x_3) = \frac{-1}{1+h}\left(x_2^2 + hx_3^2 - R^2\right)$$

so that $J = \frac{\pi R^4}{\sqrt{h}(1+h)}$.

Computation of the displacements. The Hooke's law shows that the entries ϵ_{ij} of linearized strain tensor $[\epsilon]$ are

$$\epsilon_{12} = \frac{M}{2\mu J}\partial_3\psi \quad \epsilon_{13} = -\frac{M}{2\mu J}\partial_2\psi \tag{3.68-a}$$

$$\epsilon_{11} = \epsilon_{22} = \epsilon_{33} = \epsilon_{23} = 0 \tag{3.68-b}$$

As $\epsilon_{ii} = 0$ for $1 \leq i \leq 3$, the displacement $\overrightarrow{X}(x_1, x_2, x_3)$ of the three-dimensional media is of the following form:

$$X_1 = f(x_2, x_3), \quad X_2 = g(x_1, x_3), \quad X_3 = h(x_1, x_2)$$

The condition $\epsilon_{23} = 0$ leads to

$$\partial_2 h(x_1, x_2) + \partial_3 g(x_1, x_3) = 0$$

differentiating this equation with respect to x_2 and x_3, we see that the second order derivatives $\partial_{22}h$ and $\partial_{33}g$ are zero so that *we can write h and g as follows:*

$$h(x_1, x_2) = C_1 x_2 h_1(x_1) \quad g(x_1, x_3) = C_2 x_3 g_1(x_1)$$

with

$$h_1 = \frac{C_2}{C_1} g_1$$

The expressions (3.68-a) of ϵ_{1i} $(i = 2, 3)$ allow to write down the following equations to define the mapping f, h_1 and to compute the constant C_1

$$\partial_2 f - C_1 x_3 h_1' = \frac{M}{\mu J}\partial_3\psi \quad \text{and} \quad \partial_3 f + C_1 x_2 h_1' = -\frac{M}{\mu J}\partial_2\psi \tag{3.69}$$

differentiating one of these equations with respect to x_1 we verify that the second derivative of h_1 is 0 and we can assume that $h_1(x_1) = x_1$. Differentiating the first equation of (3.69) with respect to x_3, the second one with respect to x_2 and subtracting member to member the obtained results we get

$$C_1 = -\frac{M}{2\mu J}\triangle\psi = \frac{M}{\mu J}$$

we can then define the components X_2 and X_3 of the displacement $\overrightarrow{X}$ by

$$X_2 = -\frac{M}{\mu J}x_1 x_3 \quad X_3 = \frac{M}{\mu J}x_1 x_2 \tag{3.70}$$

and the component X_1 as the solution[33] of the differential equations

$$\partial_2 X_1 = \frac{M}{\mu J}\left(\partial_3\psi + x_3\right) \quad \partial_3 X_1 = -\frac{M}{\mu J}\left(\partial_2\psi + x_2\right) \tag{3.71}$$

REMARK 3.7 If the cross-section is elliptic (see Remark 3.6) the system (3.71) has the analytic solution

$$X_1(x_2, x_3) = \frac{M(1-h)}{\mu J(1+h)} x_2 x_3$$

which shows that if $h \neq 1$ *the deformed configuration of a cross-section doesn't remain flat.*

The solution

$$(x_2, x_3) \mapsto X_1(x_2, x_3)$$

of the differential equations (3.71) is called *warping mapping*: if we introduce the rotation

$$\theta_1(x_1) = \frac{M}{\mu J} x_1$$

the displacement $\overrightarrow{X}$ of the three-dimensional beam is defined by

$$\overrightarrow{X} = \left[\theta_1 \overrightarrow{e}_1 \wedge (x_2 \overrightarrow{e}_2 + x_3 \overrightarrow{e}_3)\right] + X_1(x_2, x_3)\overrightarrow{e}_1$$

and it is easy to check that the term between brackets is, up to an additive constant, the solution of the beam equation (3.19) page 105 with the boundary conditions $m_1 = m_2 = M$.

Homeworks.

1^o/ Compute in the same way the displacement field in the cylindrical bar submitted to the following loading conditions:

- *Tension / compression*: the ends of the bar are submitted to a balanced system of forces F_1 and $-F_1$ along the axis x_1;
- *Pure bending*: the ends of the bar are submitted to a balanced system of torques M_2 and $-M_2$ (resp. M_3 and $-M_3$) about the axis x_2 (resp. x_3);
- *Searing forces*: the ending cros-sections are submitted to a balanced system of forces along the axes x_2 and x_3.

2^o/ What happens if you no longer assume that the x_2 and x_3 axes are the principal axes of inertia of the cross-sections?

[33] Which is defined up to an additive constant.

CHAPTER 4

APPLICATION TO OPTIMAL DESIGN OF STRUCTURES

ASSUME that the mass, the stiffness and the damping matrices of an elastic structure depend on a design parameter $u \in U_{ad}$ and let $t \in [0,T] \mapsto F(t)$ a mission profile[1], defined on a time horizon T be given. The damage caused by the loading $t \mapsto F(t)$ on a given zone of the structure can be understood as a function $\mathscr{D}(u)$ of the design parameters u and the question which is addressed to in this Chapter is to set up a numerical algorithm allowing to identify an optimal design $u_* \in U_{ad}$ which minimizes the mapping $u \in U_{ad} \mapsto \mathscr{D}(u)$ or the mass under the constraint $\mathscr{D}(u) \leq d_{max}$.

After a short review on the gradient based optimization algorithms in Section 4.1, we will see that the hardest question is to set up a procedure for the computation of the derivative of the damage with respect to u. To this purpose, we recall in Section 4.2 an adjoint state method allowing to calculate the gradient of a criterion $J(u)$ written as the integral

$$J(u) = \int_0^T j(X_u(t))\,dt \quad \text{controlled by a system of differential equations} \tag{4.1}$$
$$\frac{dX_u}{dt} = f(X_u, u, t)$$

depending on the parameters u in an admissible set U_{ad}. Then, on the basis of the results obtained in the Chapters 2 and 3 we specify in Section 4.3 what is said in Section 4.2 to the parametric optimization of structures under fatigue criterion. We especially show that the integration of the adjoint equation can be performed with the help of the forced response method introduced in Chapter 3. *We will see that this permits to limit the*

[1] Satisfying the conditions given in the footnote 14 page 115.

volume of the data which are to be stored between the integration steps of the state and the adjoint equations, making the method ready to process FEM models.

The Chapter is organized as follows:

Contents

4.1. Optimization survival kit

Given a functional (ie. a numerical mapping) J defined on a set U, this Section deals with the question of identifying an element $u_* \in U$ such that

$$J(u_*) = \inf_{u \in U} J(u) \tag{4.2}$$

We are more precisely intending to introduce several gradient based algorithms for the computation of u_*. As this question goes well beyond the scope of a Section in a course devoted to structure optimization, we restrict ourselves to provide some indications on the ways to solve an optimization problem by gradient methods. We hope that they will convince the reader to spend both time and programming efforts in defining and implementing the adjoint state to the minimization problem defined in the introduction to this Chapter.

Preliminaries. We introduce in these preliminaries some conditions relating both to the function J and the design space U insuring well-posedness of the optimization problem (4.2). In this spirit, we see that the existence of a minimizer for the functional J is a consequence of *the continuity of* J and *compactness of design space* U, Propositions 4.1 and 4.2 provide existence results for the minimizers but, since they are demonstrated by reduction to absurd, they do not give any indication on the way to calculate these minimizers. More constructive results (which can be translated into algorithms) can be obtained under more restrictive assumptions about J and U: The functional J must be at least Lipschitz continuous and convexness hypotheses must be added on U or J when the design space is not of finite dimension (see for instance CEA [**8**]).

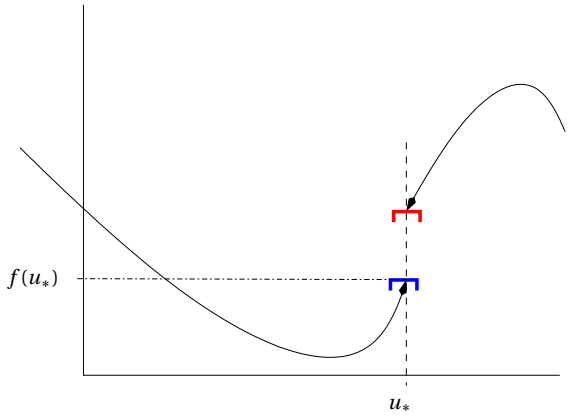


Fig. 4.1. **Example of lower semi-continuous functional**. We see on this picture that the reciprocal image of the set $\{y \in \mathbb{R}\,;\, y \geq f(u_*) - \varepsilon\}$ contains an open interval centered on u_*. It should not be believed that a mapping which is lower semi-continuous at a point necessarily has a limit from the left of this point: the reader may check that the mapping f defined on $[-1,1]$ by $f(u) = \sin\frac{1}{u}$ if $u \neq 0$ and $f(0) = -1$ is lower semi-continuous at 0 while the limit $\lim_{u\to 0^-} f(u)$ doesn't exist.

DEFINITION 4.1 (Lower semi-continuous functional) A functional J defined on a topological space E is, see figure (Fig.4.1), said to be lower semi-continuous[2] at $u_* \in E$ if for any $\varepsilon > 0$, there exists a neighborhood $\mathcal{U}_{u_*}$ of u_* such that $J(u) \geq J(u_*) - \varepsilon$ for all $u \in \mathcal{U}_{u_*}$. The functional J is said to be lower semi-continuous in E if it is lower semi-continuous at any point $u \in E$.

PROPOSITION 4.1 *A lower semi-continuous functional J defined on a compact topological space U attains its greatest lower bound. In other words there is $u_* \in U$ satisfying the equation* (4.2).

PROOF. Let $m = \inf_{u\in U} J(u) \in \mathbb{R} \cup \{-\infty\}$ be the greatest lower bound of J over U and *assume for contradiction that $J(u) > m$ for all $u \in U$*. Then by semi-continuity of J, for any $u \in U$ and $\varepsilon_u > 0$ such that $\varepsilon_u < J(u) - m$, we can find an open neighborhood $\mathcal{V}_u$ of u over which J is bounded below by $m_u = J(u) - \varepsilon_u > m$.

Varying $u \in U$, we define an open covering $(\mathcal{V}_u)_{u\in U}$ of U. Using the compactness of U, there is a finite sequence $(u_k)_{k=1}^n$ such that

$$U = \bigcup_{k=1}^{n} \mathcal{V}_{u_k}$$

As J is bounded below by m_{u_k} on each $\mathcal{V}_{u_k}$, it is bounded below by $m' = \min_{1\leq k\leq n} m_{u_k}$ on U. The fact that $m' > m$ contradicts the definition of m. □

[2] In the same manner, J is said to be upper semi-continuous at u_* if $-f$ is lower semi-continuous at u_*. One can check that a functional J is continuous at u_* if and only if it is lower and upper semi-continuous at u_*.

REMARK 4.1 When U is a subset of a finite dimensional vector space E endowed with a classical norm, the compactness condition on U simply means that U is a closed and bounded subset of E.

EXAMPLE 4.1 Due to non compactness of $]-1,1[$, the mapping $u \in]-1,1[\mapsto u^3$ doesn't reach its greatest lower bound.

We give in the following Proposition a condition allowing to generalize the Proposition 4.1 to the minimization of a functional J on non compact subsets of a normed space.

PROPOSITION 4.2 *A functional J defined on a normed vector space E is said to be coercive if there is a constant $c > 0$ such that*

$$\lim_{\|u\| \to \infty} \frac{J(u)}{\|u\|} \geq c \tag{4.3}$$

If we assume that E is a finite dimensional space and J is lower semi-continuous coercive, then J attains its greatest lower bound on any closed subset U of E.

PROOF. Let $u \in U$ such that $J(u) = a > -\infty$ be given, *if we proof that coerciveness of J entails that the set $U_a = \{u \in F\ ;\ J(u) \leq a\}$ is a bounded subset of E then*, by lower semi-continuity of J, the set U_a is bounded and closed. *As E is assumed to be a finite dimensional*[3] space, U_a is compact and J attains its greatest lower bound m on U_a. But $m \leq a$ and we have actually proofed that J attains its greatest lower bound on U. *To complete the proof it remains to show that U_a is a bounded subset of E.* If U_a were unbounded we could find a sequence $(u_n)_n$ such that $\lim_{n\to\infty} \|u_n\| = +\infty$ and $J(u_n) \leq a$; as this inequality entails $\lim_{n\to\infty} \frac{J(u_n)}{\|u_n\|} = 0$ it would contradict the coerciveness of J. □

Now we see how the convexity conditions permit to proof an uniqueness result and an existence result when the dimension of the design spaces is not finite[4].

DEFINITIONS 4.2 (Convexity) 1^o/ *A subset U of a vector space is said to be convex* if the conditions $u, v \in U$ entail $\lambda u + (1-\lambda) v \in U$ for any $\lambda \in [0,1]$.

2^o/ *A numerical mapping J defined on a convex subset U of a vector space E is said to be convex* (see figure (Fig. 4.2)) if

$$J(\lambda u + (1-\lambda) v) \leq \lambda J(u) + (1-\lambda) J(v) \quad \text{for any } \lambda \in [0,1] \tag{4.4}$$

The mapping J is said to be strictly convex if the previous inequality is strict for $u \neq v$ and $0 < \lambda < 1$.

REMARK 4.2 If J is assumed to be strictly convex then it achieves its greatest lower bound on at most one point u_* in its domain of definition.

[3] As bounded subsets of reflexive Banach spaces are weakly pre-compacts, such a result can be generalized to infinite dimensional spaces under the hypothesis J weakly semi-continuous.

[4] Such a case occurs when the design space is constituted of the virtual displacements of a mechanical problem defined by its strain energy.

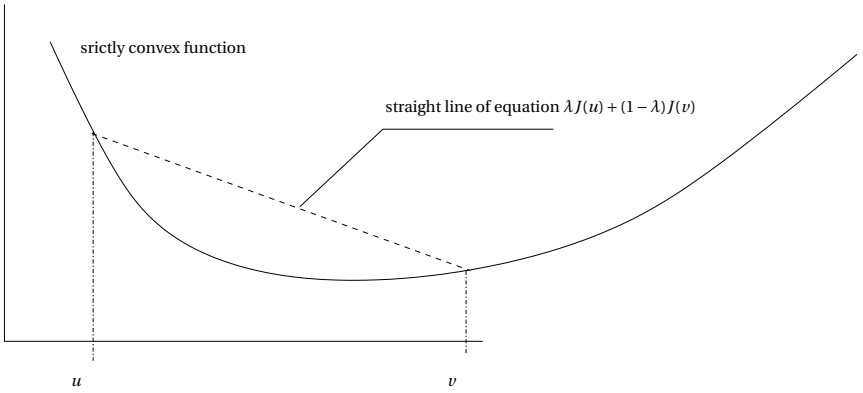


Fig. 4.2. **Graph of a convex function**.

Assuming that J is differentiable, we show in the Proposition bellow that a convexity condition imposed to the domain U or to the mapping J allows to reduce the optimization problem (4.2) to the resolution a variational inequality called *Euler inequality* associated with the optimization problem.

PROPOSITION 4.3 1^o/ *If we assume that U is convex and J differentiable then a solution $u_* \in U$ of* (4.2) *satisfies the following variational inequality*

$$J'(u_*)(u-u_*) \geq 0 \text{ for any } u \in U \tag{4.5}$$

2^o/ *If moreover J is convex, the converse is true (ie. if u_* is solution of the variational inequality* (4.5) *then it satisfies* (4.2)*).*

PROOF. 1^o/ For any $u \in U$ we have (by hypothesis)

$$J(u_* + \lambda(u-u_*)) - J(u_*) \geq 0 \quad \text{for any } 0 < \lambda < 1$$

The formula (4.5) is obtained in dividing this inequality by $\lambda > 0$ and taking the limit of the obtained result when λ goes to 0.

2^o/ If J is convex, we have

$$J(u) - J(u_*) \geq \frac{1}{\lambda} J((1-\lambda)u_* + \lambda u) \quad \text{for } 0 < \lambda < 1$$

passing to the limit when λ goes to 0, the right hand member of this inequality goes to

$$J'(u_*)(u-u_*)$$

and we see that the condition(4.5) entails $J(u) - J(u_*) \geq 0$ for any $u \in U$. □

REMARKS 4.3 1^o/ If a solution u_* of the variational inequality (4.5) is in the interior of U then $J'(u_*) \in E^*$ is identically 0 and we find the classical extremality condition;

2^o/ if we assume moreover that J is twice differentiable then *the condition $J''(u_*)(u,u) \geq C > 0$ for any u in the unit ball of E entails that $J(u_*)$ is, see figure (Fig. 4.3), a local minimum of J.*

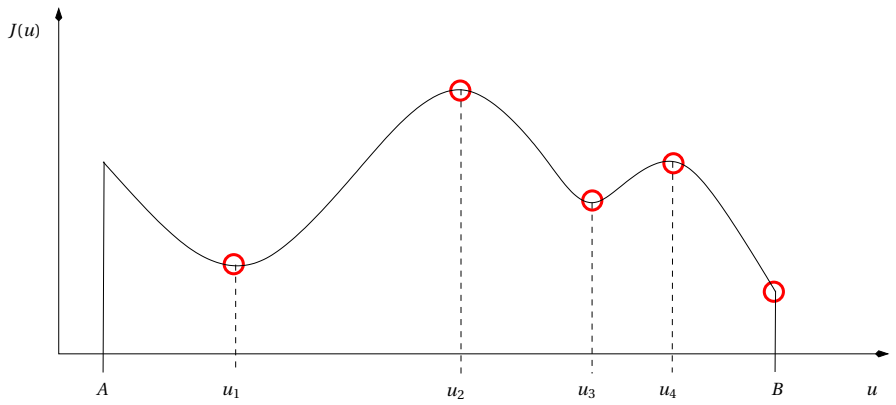


Fig. 4.3. **Solutions of the variational inequality** (4.5). On this picture, the mapping $u \mapsto J(u)$ is non convex, defined on the interval $[A,B]$. The points $(u_i)_{i=1}^4$ and B are solutions of (4.5), J reaches a local minimum at u_1 and u_3; these points can be characterized by the second statement of the Remark 4.3.

PROOF. The Remark $1^o/$ is obvious. To proof the second statement, assume that $J'(u_*) = 0$ and that $J''(u_*)(u,u) \geq C > 0$. Using a Taylor expansion of J about u_* we have

$$J(u) - J(u_*) = J''(u_*)(u-u_*, u-u_*) + \|u-u_*\|^2 \varepsilon(u-u_*)$$
$$\geq \|u-u_*\|^2 (C + \varepsilon(u-u_*))$$

for any u in a neighborhood of u_*. As $\varepsilon(u)$ goes to 0 when u goes to u_*, we see that $J(u_*) \leq J(u)$ for any u in a ball, of sufficiently small radius, centered on u_*. □

To conclude these preliminaries, we summarize the results given in Propositions 4.1 and 4.2 in the following Proposition which has the merit to be valid in infinite dimension. *Such a result is often used to proof existence results for linear or non-linear PDE. From a mechanical point of view, J is a total energy defined on the space of the virtual displacements and the Euler's inequality* (4.4) *is the principle of the virtual works.*

PROPOSITION 4.4 *Let J be a convex, lower semi-continuous functional defined on a closed convex subset U of a reflexive Banach space E, if we assume moreover that*

$1^o/$ *U is bounded in E*
$2^o/$ *or J is coercive when U is not bounded*

there is at least an element $u_* \in U$ satisfying (4.2)*; this minimizer being unique if J is strictly convex.*

SKETCH OF PROOF. If we endow E with the $\sigma(E,E^*)$-weak topology[5], and if we notice that

- strongly closed convex subsets of E are also weakly closed,
- strongly lower semi-continuous convex functions are weakly lower semi-continuous,

this Proposition is a straightforward generalization of the Propositions 4.1 and 4.2. □

The following Example is a generalization (or a proof) of the Lax-Milgram lemma stated in footnote 27 page 130[6].

EXAMPLE 4.2 Let $(u,v) \in E \times E \mapsto a(u,v) \in \mathbb{R}$ be a continuous bi-linear form defined a reflexive Banach space E and $f \in E^*$ be a continuous linear form defined on E. If a is coercive in the sense that there is a constant $c > 0$ such that

$$a(u,u) \geq c\|u\|^2 \quad \text{for any } u \in E$$

the mapping

$$u \in E \mapsto J(u) = \frac{1}{2}a(u,u) - f(u) \in \mathbb{R}$$

is continuous, strictly convex and coercive. Assuming that U is a closed convex subset of E, there is an unique element $u_* \in U$ such that

$$J(u_*) = \inf_{u \in U} J(u) \tag{4.6}$$

which is characterized by the variational inequality

$$\frac{1}{2}\left[a(u_*, u-u_*) + a(u-u_*, u_*)\right] \geq f(u-u_*) \quad \forall u \in U \tag{4.7}$$

PROOF. If we proof that J is strictly convex, this example is a consequence of the Propositions 4.4 and 4.3. To this end, let $(u,v) \in E \times E$ be given; as the relationship $a(v-u, v-u) \geq 0$ entails $a(u,v) + a(v,u) \leq a(u,u) + a(v,v)$ we see that

$$\begin{aligned} a(\lambda u + (1-\lambda)v, \lambda u + (1-\lambda)v) =& \lambda^2 a(u,u) + (1-\lambda)^2 a(v,v) \\ &+ \lambda(1-\lambda)\left[a(u,v) + a(v,u)\right] \\ \leq& \lambda a(u,u) + (1-\lambda)a(v,v) \end{aligned}$$

which shows that J is convex. To proof that J is strictly convex, assume that

$$J(\lambda u + (1-\lambda)v) = \lambda J(u) + (1-\lambda)J(v)$$

then

$$\lambda(1-\lambda)\left(a(u, v-u) - a(v, v-u)\right) = 0$$

If $0 < \lambda < 1$ we must have

$$0 = a(u, v-u) - a(v, v-u) = a(v-u, v-u)$$

[5] Let's recall that the weak topology on a normed space E is the coarest topology on E making continuous the linear forms which are continuous in the sense of the norm. This topology has the advantage to maximize the number of compact subsets of E, in counterpart it reduces the number of open sets and therefore the number of lower semi-continuous mappings. Note on the other hand that the weak topology is metrizable if and only if E is a finite dimensional vector space.

[6] Another proof of this lemma which doesn't refer to the weak topology is given in CIARLET [**9**].

and $u = v$ by coerciveness of a. □

Unconstrained minimization problems. Assume that J is a differentiable functional defined on a Hilbert space[7] E. Instead of using the Remarks 4.3 to reduce the resolution (4.2) to that of the equation $J'(u) = 0$ (complemented if needed by the Hessian condition of Remark 4.3-2^o/) we prefer solve the optimization problem by an iterative scheme. Basically, starting from a given point u_0, we will build stepwise a sequence[8] $(u_k)_k \subset E$ such that

$$J(u_{k+1}) \leq J(u_k)$$

and the questions we will have to face are the following:

- proof that the sequence $(u_k)_k$ converges to a point $u_* \in E$,
- and that $J(u_*)$ is at least a local minimum[9] of J.

The algorithm consists to write down u_{k+1} as $u_{k+1} := u_k + t_k d_k$ where t_k is a positive real number and d_k is a direction in E such that

$$< \nabla J(u_k), d_k >< 0$$

As under this conditions we have

$$J(u_k + t d_k) < J(u_k)$$

for t sufficiently small, the step size t_k can be chosen in order to minimize, at least locally, the mapping

$$t \in \mathbb{R}^+ \mapsto J(u_k + t d_k) \in \mathbb{R}$$

The basic principles of this algorithm are summarized in the algorithm 4.1.

Algorithm 4.1: Basic principles of a descent algorithm

input : Starting point u_0 and $k = 0$
outputs: Minimizer $u_* := u_k$ of J
while $\|\nabla J(u_k)\| \geq \varepsilon$ **do**
  1^o/ Compute a descent direction d_k for J at u_k.
  2^o/ Choose a step size δ_k minimizing the mapping

$$]0, +\infty[\ni t \mapsto J(u_k + t d_k) \tag{4.8}$$

  at least in a neighborhood of $t = 0$;
  3^o/ Set $u_{k+1} \leftarrow u_k + \delta_k d_k$ and $k := k + 1$
end

We highlight hereafter some algorithms allowing to compute a descent direction and to perform the unidimensional optimization. The reader is referred to specialized books such as " Numerical Optimization" [**4**] to have an exhaustive view on the topic.

[7] In this case, the derivative $J'(u)$ of J, which is in the dual space of E, can be identified with a vector $\nabla J(u) \in E$ and the value of $J'(u)$ on a vector $h \in E$ is the scalar product $< \nabla J(u), h >$.

[8] Called minimizing sequence for J.

[9] We say that u_* is a local minimum if there is a neighborhood $\mathcal{V} \subset U$ such that $J(v) \geq J(u_*)$ for any $v \in \mathcal{V}$.

1^o/ Among all the ways to define a direction of descent for the functional J, let's consider the following:

- The direction

$$d_1 = -\frac{\nabla J(u)}{\|\nabla J(u)\|} \tag{4.9}$$

which minimizes the functional

$$v \in E \mapsto < \nabla J(u), v > \in \mathbb{R}$$

on the unit ball of E, is a natural descent direction for J at u and the optimization algorithm 4.1, obtained in choosing d_1 as descent direction is called *steepest descent algorithm*.

- If we assume that J is twice continuously differentiable and that the bi-linear form

$$(v, w) \in E \times E \mapsto D^2 J(u)(v, w) \in \mathbb{R}$$

is coercive, then
  - denoting $[D^2 J(u)]^{-1}$ the continuous linear operator defined by the variational equation[10]

$$v = [D^2 J(u)]^{-1} h \quad \Leftrightarrow \quad D^2 J(u)\,(v, w) = < h, w > \forall\, w \in E$$

  - the vector

$$d_2 = -[D^2 J(u)]^{-1} \nabla J(u) \tag{4.10}$$

is well defined and is a descent direction for J at u. In this case, the optimization algorithm 4.1 is called *Newton's algorithm*.

The mapping

$$v \mapsto \hat{J}(v) = J(u) + < \nabla J(u), v > + \frac{1}{2} < D^2 J(u).v, v >$$

which is convex quadratic, reaches its minimum at $v = d_2$. As $\hat{J}$ is a second order approximation of J in a neighborhood of u, we can expect that $u + d_2$ is a good approximation of the minimizer u_* of J. This is illustrated in the figure (Fig. 4.4).

2^o/ The line search step (step 2^o/ in algorithm 4.1) searches along the direction d_k a new iterate with a lower value of the functional J. The distance δ_k to move along d_k is found by approximately minimizing the univariate mapping (4.8); we summarize in the algorithm 4.2 a backtracking method, which is illustrated in the figure (Fig. 4.5).

Algorithm 4.2: Example of backtracking algorithm for line search

- Given $c_1 < 0.5$ and $\beta \in]0, 1[$
- set $\delta := \delta_{max}$
- **while** *conditions* (4.11) *and* (4.12) *aren't satisfied* **do**
  - set $\delta := \beta\delta$

[10] In finite dimensional spaces, it is the inverse of the Hessian matrix $H_{ij} = \partial_{ij} J(u)$, which is assumed to be definite positive.

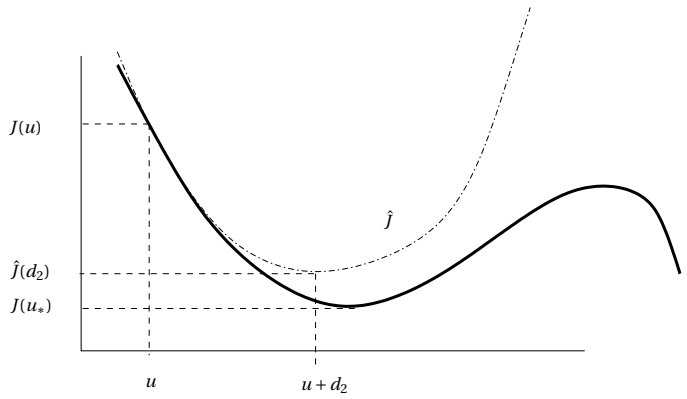


Fig. 4.4. **Second order approximation of** J. The functional J (in bold-line) and its second order approximation $\hat{J}$, in dashed line. The Newton step d_2 is the increment which must be added to u to obtain a minimizer for $\hat{J}$.

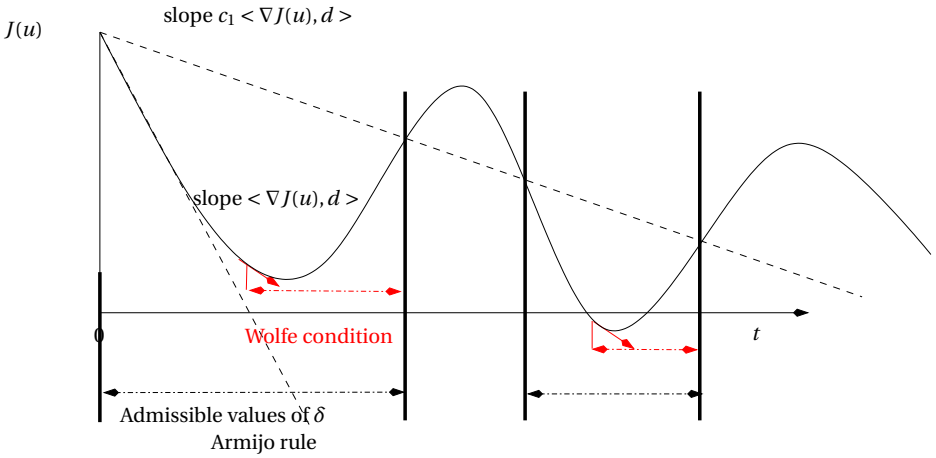


Fig. 4.5. **Backtracking method for the line search step**. The Armijo condition

$$J(u+\delta d) \leq J(u) + \delta c_1 < \nabla J(u), d > \tag{4.11}$$

insures that the steps length δ_k decreases J sufficiently. It can be supplemented by the the Wolfe rule

$$< \nabla J(u+\delta d), d > \geq c_2 < \nabla J(u), d > \tag{4.12}$$

which forces δ_k to be close to a critical point of the functional (4.8). We proof in Lemma 4.1 that, for an appropriate choice of the constants c_1 and c_2, the conditions (4.11) and (4.12) are satisfied on a non-empty interval.

LEMMA 4.1 *If J is assumed to be bounded below and if $0 < c_1 < c_2 < 1$ there is a non empty interval of step lengths which satisfies the conditions* (4.11) *and* (4.12).

PROOF. Let φ be the mapping $t \mapsto J(u_k + td_k)$, since $\varphi'(0) =< \nabla J(u_k), d_k >$ and φ is bounded below, the line $J(u_k) + tc_1 < \nabla J(u_k), d_k >$ must intersect the

graph of φ at least once. Let t_1 be the smallest intersecting value, we have

$$J(u_k + t_1 d_k) = J(u_k) + t_1 c_1 < \nabla J(u_k), d_k > \tag{4.13}$$

and the sufficient decrease condition (4.11) holds for all steep lengths less than t_1. Applying the mean value theorem, there is $t_2 \in]0, t_1[$ such that

$$J(u_k + t_1 d_k) = J(u_k) + t_1 < \nabla J(u_k + t_2 d_k), d_k > \tag{4.14}$$

Combining the formulas (4.13) and (4.14) we get

$$c_1 < \nabla J(u_k), d_k >=< \nabla J(u_k + t_2 d_k), d_k >> c_2 < \nabla J(u_k), d_k >$$

if $0 < c_2 < c_1$. By smoothness assumption on J, the inequality (4.12) holds in an interval centered on t_2. □

The following Proposition is a global convergence result for the steepest descent method.

PROPOSITION 4.5 *Assume that E is a finite dimensional vector space and that the set*

$$S = \{v \in E\,;\, J(v) \leq J(u_0)\}$$

is closed and bounded in E. If we assume moreover that

- *J is continuously differentialble on S*
- *and that the line search is exact*

then any cluster point $\bar{u}$ of the sequence $(u_k)_k$ produced by the steepest descent algorithm is a critical point of J (ie. satisfies $\nabla J(\bar{u}) = 0$).

PROOF. As S is a closed and bounded in the finite dimensional vector space E, it is a compact subset of E and, by the Heine-Borel property, the sequence $(u_k)_k$ produced by the descent algorithm 4.1 has at least a cluster point $\bar{u} \in S$. We can then define a sub-sequence $\left(u_j\right)_j$ of $(u_k)_k$ which converges to $\bar{u}$. As $\left(u_j\right)_j$ is a minimizing sub-sequence of $(u_k)_k$, we have moreover

$$\inf_k J(u_k) = \inf_j J(u_j) = J(\bar{u}) := J_*$$

Assume for contradiction that $\bar{d} := -\nabla J(\bar{u}) \neq 0$, we can find $\delta > 0$ such that

$$\Delta J := J_* - J(\bar{u} + \delta \bar{d}) > 0$$

Setting $d_j := \nabla J(u_j)$, we have (by continuity of the mapping $v \mapsto \nabla J(u)$)

$$\lim_{j\to\infty} d_j = \bar{d} \quad \text{and} \quad \lim_{j\to\infty} u_j + \delta d_j = \bar{u} + \delta \bar{d}$$

so that the inequality

$$J\left(u_j + \delta d_j\right) \leq J(\bar{u} + \delta \bar{d}) + \frac{\Delta J}{2} = J_* - \frac{\Delta J}{2} \tag{4.15}$$

takes place for j sufficiently large. As, on the other hand, the line search is exact we must have

$$J_* < J\left(u_j + \delta_j d_j\right) \leq J\left(u_j + \delta d_j\right) \tag{4.16}$$

for any j. Combining the inequalities (4.15) and (4.16), we see that the condition $\nabla J(\bar{u}) \neq 0$ entails $J_* < J_* - \frac{\Delta J}{2}$ and contradicts $\Delta J > 0$. □

REMARK 4.4 If J is assumed to be coercive and strictly convex, the above proof can be generalized to infinite-dimensional Hilbert spaces and shows actually that the steepest descent method (with exact line search) converges to the minimizer of J.

To obtain a convergence result for the descent algorithm 4.1, we must have well-chosen the descent direction d_k and the step-size δ_k in the line search step. The following Theorem, due to Zoutendijk, highlights the fact that *if the line search satisfies the conditions* (4.11) *and* (4.12) *and if the descent direction d_k is not too close to an orthogonal direction to the gradient, the algorithm 4.1* converges to a critical point of J.

PROPOSITION 4.6 (Theorem of Zoutendijk) *Suppose that*

- *the functional J is bounded bellow and continuously diffrentiable on an open set N containing the level set S defined in Proposition 4.5.*
- *and that the gradient $u \mapsto \nabla J(u)$ is Lipschitz continuous on N.*

Assume moreover that the step-size δ_k defined in step 2^o / of algorithm 4.1 satisfies the conditions (4.11) *and* (4.12) *then, if we define the angle θ_k between the descent direction d_k and the gradient $\nabla J(u_k)$ by*

$$\cos\theta_k = -\frac{< \nabla J(u_k), d_k >}{\|\nabla J(u_k)\| \, . \, \|d_k\|}$$

the series

$$\sum_{k=0}^{+\infty} cos^2\theta_k \, \|\nabla J(u_k)\| \tag{4.17}$$

is convergent.

Before going on to the proof of this Proposition, let's see how it can be used to establish a convergence result for the algorithm 4.1. As the series (4.17) is convergent we must have

$$\lim_{k\to\infty} cos^2\theta_k \, \|\nabla J(u_k)\| = 0 \tag{4.18}$$

If the descent direction is chosen in order to bound above $|\theta_k|$ by an angle $\theta < \frac{\pi}{2}$ then $cos^2\theta_k$ is bounded below by some positive constant and the formula (4.18) entails that $\lim_{k\to\infty} \|\nabla J(u_k)\| = 0$.

For instance:

- we have $\theta_k = 0$ for *the steepest descent method so that the sequence $(u_k)_k$ produced by the algorithm 4.1 converges to a critical point of J.* Note that the additional hypothesis, ∇J Lipschitz continuous, leads to a stronger result that the one which is obtained in Proposition 4.5 where, due to lack of regularity, the minimizing sequence can oscillate between two cluster points of the minimizing sequence $(u_k)_k$.

- If we assume that the twice derivative of J is coercive then

$$\left\|D^2J(u)\right\|.\left\|[D^2J(u)]^{-1}\right\|\leq M$$

for some constant M and $\cos\theta_k\geq\frac{1}{M}$ *in the Newton algorithm, which thus converges to a local minimizer of* J.

PROOF OF PROPOSITION 4.6. From the inequality (4.12) and by definition

$$u_{k+1}:=u_k+\delta_k d_k$$

we have

$$\langle\nabla J(u_{k+1})-\nabla J(u_k),d_k\rangle\geq(c_2-1)\langle\nabla J(u_k),d_k\rangle$$

while the Lipschitz condition $\|\nabla J(u_{k+})-\nabla J(u_k)\|\leq L\|u_k-u_{k+1}\|$ implies

$$\langle\nabla J(u_{k+1})-\nabla J(u_k),d_k\rangle\leq\delta_k L\|d_k\|^2$$

Combining these two inequalities we obtain

$$\delta_k\geq\frac{c_2-1}{L}\frac{\langle\nabla J(u_k),d_k\rangle}{\|d_k\|^2}$$

Substituting this inequality into the Armijo condition (4.11) we get

$$\begin{aligned}J(u_{k+1})&\leq J(u_k)+c_1\frac{c_2-1}{L}\frac{\langle\nabla J(u_k),d_k\rangle^2}{\|d_k\|^2}\\&\leq J(u_k)-c\cos^2\theta_k\|\nabla J(u_k)\|\quad\text{where}\quad c=c_1\frac{1-c_2}{L}>0\end{aligned}$$

and

$$J(u_0)-J(u_{k+1})\geq c\sum_{j=0}^{k}\cos^2\theta_j\left\|\nabla J(u_j)\right\|$$

Since J is bounded below, $J(u_0)-J(u_{k+1})$ must be lower than some positive constant for any k and this means the series (4.17) is convergent. □

We conclude this paragraph in explaining how to modify the descent algorithm 4.1 *to solve the constrained optimization problem*

$$J(u_*)=\inf_{u\in U}J(u)$$

where U *is a closed convex subset of a Hilbert space* E.

As for any $v\in E$ there is an unique element $P_U(v)\in U$ such that[11]

$$\|P_U(v)-v\|\leq\|u-v\|\quad\text{for all } u\in U$$

we can then modify the algorithm 4.1 by setting

$$u\leftarrow P_U(u+\delta d)$$

[11]Noticing that the mapping $u\in U\mapsto\|u-v\|^2$ is strongly convex and coercive, this result is a consequence of the Proposition 4.4. Using the Proposition 4.3 we can moreover see that $P_U(v)$ is characterized by the variational inequality

$$<v-P_U(v),u-P_U(v)>\leq 0\quad\text{for all } u\in U$$

A geometrical proof (which doesn't appeal to the properties of the weak topology) of this result is given in BREZIS [**6**] or CIARLET [**9**].

in the step 3^o/ if $u \neq P_U(u+\delta d)$. Indeed, if the minimum of J is reached on the boundary of U, the gradient $\nabla J(u_*)$ can be non zero.

EXAMPLE 4.3 Assume that $E = \mathbb{R}^n$ and that U is the box

$$U = \prod_{i=1}^{n} [a_i, b_i]$$

the projection operator P_U is the operator which associates to $x = (x_1, \cdots, x_n) \in \mathbb{R}^n$ the point $P_U(x) = (P_U(x)_1, \cdots, P_U(x)_n) \in U$ defined component by component as follows:

$$P_U(x)_i = \max\{b_i, \min(x_i, a_i)\}$$

Algorithm 4.3: Descent algorithm for constrained optimization problem.

input : Starting points $u := u_0 \in U$
outputs: Minimizer $u_* := u \in U$ of J
while $\|\nabla J(u_k)\| \neq 0$ **do**
  1^o/ Compute a descent direction d_k for J at u_k.
  2^o/ Line search choose a step size δ_k which minimizes the mapping $t \mapsto J(u_k + t d_k)$, at least in a neighborhood of $t = 0$;
  3^o/ **if** $u_k - P_U(u_k + \delta_k d_k) \neq 0$ **then**
    - set $u_{k+1} \leftarrow P_U(u_k + \delta_k d_k)$
    - set $k := k + 1$

  else
    exit while loop
  end
end

Optimization with inequalities constraints. Let $(g_i)_{i=1}^n$ be n numerical mappings defined on a Hilbert space E. This paragraph deals with the optimization problem (4.2) when U is of the special form

$$U = \{v \in E \text{ such that } g_i(v) \leq 0 \text{ for } 1 \leq i \leq n\} \tag{4.19}$$

and has a non-empty interior. This problem is much more complicated than its unconstrained counterpart and we merely give some indications on the classic ways to solve it, referring to specialized texts such as CIARLET [**9**] or CEA [**8**] for a complete proof of the stated results[12].

Assume that u_* is a local minimizer of J over U and that we can find a direction $d \neq 0$ in E such that

$$U_\delta = \{u_* + t d \in U \text{ for } t \in [0, \delta[\}$$

is non empty for some positive constant δ. The restriction of J to U_δ is then an univariate mapping $\hat{J}$ which is defined on $[0, \delta[$ and achieves a local minimum at $t = 0$, so that $\hat{J}'(0)$ can't be negative. Since $\hat{J}'(0) = < \nabla J(u_*), d >$ we must have

$$< \nabla J(u_*), d > \geq 0$$

[12] The literature is actually abundant and the objectives of this sub-section are to provide the reader with a first glance on the saillant results on the topic.

Introducing the set of feasible directions $F_U(v)$ as in Definition 4.3 we can generalize the variational inequality (4.5) to *"the not necessarily convex sets"* as follows:

If $u_* \in U$ is a local munimum of J over U then	
(4.20)	$< J(u_*), d >\geq 0$ for any $d \in F_U(u_*)$

DEFINITION 4.3 (Feasible directions) Let U be a subset of E, we say that a *direction $d \in E$ is, see figure (Fig. 4.6), a feasible direction at $v \in U$* if there is $\delta > 0$ such that

$$v + td \in U \quad \text{for all } t \in]0, \delta[$$

And *we will denote by $F_U(v)$ the subset of E made up of the directions which are feasible at v.*

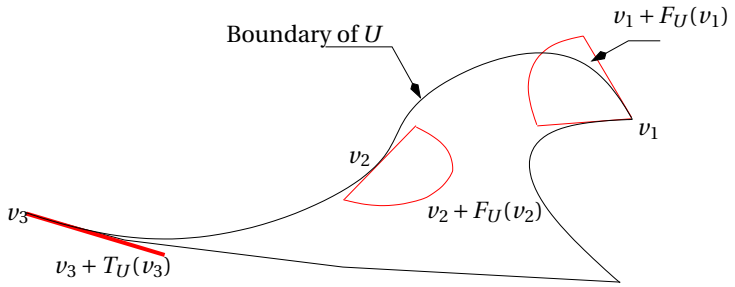


Fig. 4.6. **Cone of feasible directions**. We see on this picture that if v is in the interior of U any direction of E is a feasible direction. When v is on the boundary of U the cones of the feasible directions at the points v_1 and v_2 are surrounded by red lines. For $v = v_3$ the cone of feasible directions reduces to 0 and we represent on this picture the tangent cone to U such as defined in figure (Fig. 4.7).

We can check that:

- The set $F_U(v)$ is a cone in E, it need not be closed or convex;
- if U is convex then $F_U(v)$ consists of the vectors of the form $\alpha(v-u)$ for $v \in U$ for $\alpha > 0$;
- but the condition (4.20) can be vacuous because there may be no feasible directions other than 0.

To circumvent the drawbacks of the feasible cone, we introduce in Definition 4.4 the notion of *tangent cone, which is basically made up of all the feasible directions and their limits*; we will obtain in this manner a closed cone for which the property (4.20) remains valid.

DEFINITION 4.4 (Tangent cone) We say that a vector $d \in E$ is tangent to U at $v \in U$ if, see figure (Fig. 4.7), there is a sequence $(d_n)_n$ converging to d and a decreasing sequence of positive real numbers $(\varepsilon_n)_n$ which converges to 0, such that

$$v + \varepsilon_n d_n \in U \quad \text{for any } n \tag{4.21}$$

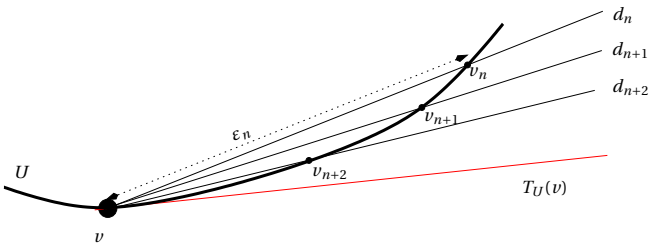


Fig. 4.7. **Tangent cone to U at v.** In this case, U is the bold-line curve so that the cone of feasible directions reduces to 0. In this spirit, the notion of cone tangent enriches, as it is explained in the Proposition 4.7, the cone feasible directions. Note moreover that we can characterize a tangent direction $d \in T_U(v)$ as follows: there is a sequence $(v_n)_n \subset U$ with $\lim_n v_n = v$ and sequence $(\alpha_n)_n$ of positive numbers such that $\lim_n \alpha_n = 0$ and $\lim_n \frac{v_n - v}{\alpha_n} = d$.

The cone $T_U(v)$ made up of all the tangents to U at v and is called tangent cone to U at v.

PROPOSITION 4.7 (Characterizations of the tangent cone) *Let U be a non empty subset of E and $v \in U$ be given then:*

$1^o/$ *A non zero vector $d \in E$ is in the tangent cone $T_U(v)$ to U at v if and only if there is a sequence $(v_n)_n \subset U$ such that*

$$v_n \neq v \text{ for all } n \quad \text{and} \quad \lim_n v_n = v$$
$$\lim_{n\to\infty} \frac{v_n - v}{\|v_n - v\|} = \frac{d}{\|d\|} \tag{4.22}$$

$2^o/$ *The tangent cone $T_U(v)$ to U at v is closed and we have $cl(F_U(v)) \subset T_U(v)$.*

$3^o/$ *If U is convex then $T_U(v)$ is convex and $cl(F_U(v)) = T_U(v)$.*

PROOF. Claim $1^o/$ Let $d \neq 0$ be in $T_v(U)$, according to the Definition 4.4 there are two sequences $(d_n)_n \subset E$ and $(\varepsilon_n)_n$ satisfying $\lim_{n\to\infty} d_n = d$ and $\lim_{n\to\infty} \varepsilon_n = 0$ and such that $v + \varepsilon_n d_n \in U$ for any n. Setting $v_n = v + \varepsilon_n d_n$ we have

$$\lim_{n\to\infty} v_n = v \quad \text{and} \quad \lim_{n\to\infty} \frac{v_n - v}{\|v_n - v\|} = \lim_{n\to\infty} \frac{d_n}{\|d_n\|} = \frac{d}{\|d\|}$$

Conversely, assume that there is a sequence $(v_n)_n \subset U$ satisfying the conditions (4.22). Setting $\varepsilon_n = \|v_n - v\|$ and $d_n = \frac{v_n - v}{\varepsilon_n}$ we have $\lim_{n\to\infty} d_n = \frac{d}{\|d\|}$ and

$$v + \varepsilon_n d_n = v_n \in U \quad \text{for any } n$$

Claim $2^o/$ Let $(d_n)_n$ be a sequence in $T_U(v)$ which converges to d in E, we proof that d is actually in $T_U(v)$. Using claim $1^o/$, for each n there is a sequence $\left(v_n^k\right)_k$ such that

$$v_n^k = v + \frac{d_n}{\|d_n\|} \left\| v_n^k - v \right\| + \left\| v_n^k - v \right\| \delta_n^k$$

with $\lim_k \delta_n^k = 0$. Let $(\varepsilon_n)_n$ be a sequence converging to 0, we can find an increasing mapping $n \mapsto k(n)$ such that $\left\| \delta_n^{k(n)} \right\| \leq \varepsilon_n$ and writing $v_n^{k(n)}$ as follows

$$v_n^{k(n)} = v + \frac{d}{\|d\|} \left\| v_n^{k(n)} - v \right\| + \left\| v_n^{k(n)} - v \right\| \delta'_n$$
$$\text{with } \delta'_n = \left(\delta_n^{k(n)} - \left(\frac{d_n}{\|d_n\|} - \frac{d}{\|d\|} \right) \right)$$

we see that $\lim_n \delta'_n = 0$ so that the sequence $\left(v_n^{k(n)} \right)_n$ satisfies the conditions (4.22) and d is in the cone $T_U(v)$. The inclusion $cl\,(F_U(v)) \subset T_U(v)$ is now a straightforward consequence of the inclusion $F_U(v) \subset T_U(v)$.

Claim 3^o/ If U is assumed to be convex then all the feasible directions at $v \in U$ are of the form $a(v-u)$, where $u \in U$ and $a > 0$. This proof that $F_U(v)$ is convex and, by definition (4.21), that $T_U(v) \subset cl\,(F_U(v))$. □

The following Lemma generalizes to non-convex domains the results stated in Proposition 4.3; *in this case the minimizer is however local.*

LEMMA 4.2 *Assume that J is continuously differentiable at $v \in U$ and define the set*

$$D(u) = \{d \in E\,;\, \langle \nabla J(u), d \rangle \geq 0\}$$

The local minimizers of J are characterized as follows:

1^o/ *If $u_* \in U$ is local minimizer for J over U then $T_{u_*}(U) \subset D(u_*)$, ie. $\langle \nabla J(u_*), d \rangle \geq 0$ for any $d \in T_{u_*}(U)$.*

2^o/ *When J is convex, assume conversely that $T_U(u) \subset D(u)$ and there is $\varepsilon > 0$ such that*

$$d = u - v \in C(u) \quad \forall\, v \in U \cap B(u, \varepsilon) \tag{4.23}$$
$$\text{where } B(u,\varepsilon) = \{v \in E\,;\, \|v-u\| < \varepsilon\}$$

then u is a local minimizer for J.

PROOF. Assume that u_* is a local minimizer of J over U, let $d \in T_U(u_*)$ be given and let $(u_k)_k$ be a sequence in U satisfying the conditions (4.22); using a Taylor expansion J about u_* we have

$$0 \leq J(u_k) - J(u_*) = < \nabla J(u_*), u_k - u_* > + \varepsilon(u_k - u_*)$$
$$\text{where } \lim_{k \to \infty} \frac{\varepsilon(u_k - u_*)}{\|u_k - u_*\|} = 0$$

for k large enough. Dividing this inequality by $\|u_k - u_*\| \neq 0$ we get

$$0 \leq \langle \nabla J(u_*), \frac{u_k - u_*}{\|u_k - u_*\|} \rangle + \frac{\varepsilon(u_k - u_*)}{\|u_k - u_*\|}$$

and by definition of the sequence u_k this proof that $< \nabla J(u_*), d > \geq 0$.

Since in claim 2^o/ the functional J is assumed to be convex, the inequality (4.82) page 190 shows that

$$J(v) \geq J(u) + < \nabla J(u), u - v > \quad \forall\, v \in E$$

thus the condition (4.23) entails $J(v) \geq J(u)$ for all $v \in U \cap B(u,\varepsilon)$, and this means that u is a local minimizer for J. □

We introduce the notion of qualification of constraints which ensure that the tangent cone $T_U(v)$ to U at v is computable when U is defined by the inequalities (4.19). This will lead us to translate the geometrical optimality conditions stated in the previous Lemma into analytical equations, *referred to as KKT conditions*.

DEFINITION 4.5 (Constraints Qualification) Assume that U is of the form (4.19) and define the *index set* $I(v)$ *of the active constraints at point* $v \in U$ as

$$I(v) = \left\{ i \in \{1, \cdots, n\} \quad \text{such that } g_i(v) = 0 \right\}$$

we say that *the constraints* $\left(g_i\right)_{i=1}^n$ *are qualified at* v if[13]

$$T_U(v) = \left\{ d \in E \,;\, \langle \nabla g_i(v), d \rangle \leq 0 \quad \forall\, i \in I(v) \right\}$$

EXAMPLE 4.4 Let U be the set

$$U = \left\{ (x_1, x_2) \in \mathbb{R}^2 \,;\, x_2 \geq x_1^3 \text{ and } x_2 \geq 0 \right\}$$

We can see on a picture that $T_U(0) = \{(x,0) \,;\, x \geq 0\}$ while, setting

$$g_1(x_1,x_2) = x_1^3 - x_2 \quad g_2(x_1,x_2) = -x_2$$

we have

$$\nabla g_1(0) = (0,-1) \quad \nabla g_2(0) = (0,1)$$

and

$$< d, \nabla g_1(0) > \leq 0 \quad < d, \nabla g_2(0) > \leq 0$$

if and only if $d_2 = 0$ and $d_1 \in \mathbb{R}$. *In this case, the constraints are not qualified at* $x = 0$; this proof that the notion qualification is a property related to the parameterization and not to the geometry of the domain.

We state in the following Proposition a necessary condition for the existence of a solution for the constrained optimization problem (4.2) when U if of the form (4.19).

PROPOSITION 4.8 (Karush-Kuhn-Tucker Conditions) *Assume that the design space* U *is of the form* (4.19) *and that* J *and* g_i *are continuously differentiable. Let* u_* *be a local minimizer of* J *over* U*, if the constraints are qualified at* u_* *there exist* n *positive real numbers* $(\lambda_i(u_*))_{i=1}^n$ *(referred to as Lagrange multipliers) such that*

$$\begin{aligned} &\nabla J(u_*) + \sum_{i=1}^{n} \lambda_i(u_*) \nabla g_i(u_*) = 0 \\ &\lambda_i(u_*) g_i(u_*) = 0 \quad \textit{for } 1 \leq i \leq n \end{aligned} \tag{4.24}$$

SKETCH OF PROOF. Accounting for the Definition 4.5, the Lemma 4.2 shows that if u_* is a local minimizer of J over U then, for any $d \in E$, the condition $\langle \nabla g_i(u_*), d \rangle \leq 0$

[13] We can check that if $d \in T_U(v)$ then $< \nabla g_i(v), d > \leq 0$ for all $i \in I(v)$, but the converse is not necessarily true.

for $i \in I(u_*)$ entails that $\langle \nabla J(u_*), d\rangle \geq 0$ and (4.24) is a consequence of the Lemma 4.3[14] with $b = \nabla J(u_*)$ and $a_i = \nabla g_i(u_*)$ for $i \in I(u_*)$.

LEMMA 4.3 (Farkas Lemma) *Let I a finite indexing set, $(a_i)_{i\in I} \subset E$ and $b \in E$ be given. The inclusion $\{d \in E\,;\, \langle a_i, d\rangle \leq 0 \quad \forall i \in I\} \subset \{d \in\,;\, < b, d > \geq 0\}$ is satisfied if and only if there exist $\lambda_i \geq 0,\ i \in I$ such that $b = -\sum_{i\in I} \lambda_i a_i$.*

□

REMARK 4.5 It is difficult to verify in practice the condition of qualification of the constraints, we thus prefer use the following condition, which is more restrictive. *The constraints $g_i(u) \leq 0$ are qualified at a point $u \in U$ if there is a vector $\delta \in E$ such that*

$$\begin{aligned} &< \nabla g_i(u), \delta >= 0 \quad \textit{if } g_i \textit{ is affine} \\ &< \nabla g_i(u), \delta >< 0 \quad \textit{in the other cases} \end{aligned} \tag{4.25}$$

for any $i \in I(u)$.

PROOF. Let $d \in E$ such that $< \nabla g_i(u), d >\leq 0$ for any $i \in I(u)$. We have to proof that d lies in $T_U(v)$. Let be given $\eta > 0$ and $(\varepsilon_n)_n$ a deceasing sequence of real numbers which converges toward 0, we will proof that the following sequence

$$v_n = u + \varepsilon_n(d + \eta\delta)$$

is actually in U so that the vector $d + \eta\delta$ belongs to $T_U(u)$ for any $\eta > 0$. Then using the fact that $T_U(u)$ is closed we conclude that $T_U(u) \ni d = \lim_{\eta\to 0^+} \big(d + \eta\delta\big)$.

To complete the proof it remains to show that the conditions (4.25) entail that $v_n \in U$ for n large enough. The task is achieved in distinguishing the following cases:

1^o/ If $i \notin I(u)$, we have $g_i(u) < 0$ and the continuity of g_i at u allows to conclude that $g_i(v_n) < 0$ for n large enough.

2^o/ If $g_i(u) = 0$ the conditions (4.25) show that

- If g_i is affine

  $$g_i(v_n) = g_i(u) + \varepsilon_n < \nabla g_i(u), d + \eta\delta >= \varepsilon_n < \nabla g_i(u), d >\leq 0 \quad \text{for all } n$$

- else, using a Taylor expansion of g_i about u, we have

  $$g_i(v_n) = \varepsilon_n < \nabla g_i(u), d + \eta\delta > + O(\varepsilon_n) \leq 0 \quad \text{for } n \text{ large enough}$$

so that $v_n \in U$ for n sufficiently large.

□

In the following we explicitly solve the Kuhn-Tucker equations to find the local minima of a functional.

[14] Proofed in CIARLET [**9**] page 208.

EXAMPLE 4.5 Find a minimizer of the functional $(x,y) \mapsto J(x,y) = x^2 + y^2$ under the constraint $x + y \geq 1$. Setting $g(x,y) = 1 - x - y$ the KKT conditions (4.24) are written down as the system of equations

$$\begin{aligned} &2x = \lambda \\ &2y = \lambda \\ &\lambda(1 - x - y) = 0 \end{aligned} \tag{4.26}$$

with the additional condition $\lambda \geq 0$. We can easily see that

$$x = y = \lambda = 0 \quad \text{and} \quad \lambda = 1\,,\ x = y = \frac{1}{2}$$

are solutions of (4.26) but, as $x = y = 0$ is not a feasible, $x = y = \frac{1}{2}$ is the only solution of the optimization problem.

In the rest of this sub-section we propose three ways to solve the KKT equations; each of them consist in obtaining the solution of the constrained optimization problem as the limit of a sequence of solutions of unconstrained optimzation problems.

Lagrange duality method. Assume that U is of the form (4.19) and define the Lagrangian

$$(v,\lambda) \in E \times \mathbb{R}^n \mapsto L(v,\lambda) = J(v) + \sum_{i=1}^{n} \lambda_i g_i(v) \tag{4.27}$$

We have

$$\sup_{\lambda \geq 0} L(v,\lambda) = \begin{cases} J(v) & \text{if } v \in U \\ +\infty & \text{else} \end{cases}$$

and the optimization problem (4.2) is equivalent to the following problem, referred to as *primal problem*.

$$L_*(u_*) = \inf_{v \in E} L_*(v) \quad \text{where} \quad L_* \text{ is the mapping } v \in E \mapsto \sup_{\lambda \geq 0} L(v,\lambda) \tag{4.28}$$

But we can also consider the optimization problem

$$L^*(\lambda^*) = \sup_{\lambda \geq 0} L^*(\lambda) \quad \text{where} \quad L^* \text{ is the mapping } \lambda \in \mathbb{R}^n_+ \mapsto \inf_{v \in E} L(v,\lambda) \tag{4.29}$$

referred to as *dual problem*. Noticing that the dual mapping L^*, defined on the convex set $U^* = \left\{\lambda \in \mathbb{R}^n_+ \,;\, \inf_{v \in \Omega} L(v,\lambda) > -\infty\right\}$, is concave[15], *the dual problem can be solved with the help of the projected gradient algorithm 4.4 referred, in this particular case, to as Uzawa algorithm for the resolution of the primal problem.*

[15] Indeed, let λ_1, λ_2 be given in U^*, we have

$$L(v, \mu\lambda_1 + (1-\mu)\lambda_2) = \mu L(v,\lambda_1) + (1-\mu) L(v,\lambda_2) \quad \text{for any } \mu \in [0,1]$$

Taking the infimum over $v \in E$ on the both sides of this equation, we obtain

$$\begin{aligned} \inf_{v \in E} L(v, \mu\lambda_1 + (1-\mu)\lambda_2) &= \inf_{v \in E} \left(\mu L(v,\lambda_1) + (1-\mu) L(v,\lambda_2)\right) \\ &\geq \mu \inf_{v \in E} L(v,\lambda_1) + (1-\mu) \inf_{v \in E} L(v,\lambda_2) \end{aligned}$$

Algorithm 4.4: Projected gradient for the maximization of the dual mapping.

input : Starting point $u_0 \in E$ and $\lambda^0 \in \mathbb{R}^n_+$
outputs: Minimizer $u_* := u_k$ of J
while $\left\|\lambda^k - \lambda^{k+1}\right\| \geq \varepsilon$ **do**
  $1^o/$ find u_k minimizing $L(v, \lambda^k)$ over Ω;
  $2^o/$ **for** $i = 1 : n$ **do**
    $\lambda_i^{k+1} \leftarrow \max\left(\lambda_i^k + \rho g_i(u_k), 0\right)$;
  end
  $3^o/$ Set $k := k+1$
end

This means that will have defined an algorithm for solving the primal problem (4.28) if we can define some conditions on J and g_i insuring that the duality gap

$$\inf_{v\in E} L_*(v) - \sup_{\lambda\geq 0} L^*(\lambda) \tag{4.30}$$

is zero. For this purpose let's introduce the following definition

DEFINITION 4.6 (Saddle point of a Lagrangian) A point $(u_*, \lambda^*) \in E \times \mathbb{R}^n_+$ is said to be *a saddle point of the Lagrangian* L if

$$L(u_*, \lambda) \leq L(u_*, \lambda^*) \leq L(v, \lambda^*) \quad \forall v \in E \quad \forall \lambda \in \mathbb{R}^n_+ \tag{4.31}$$

PROPOSITION 4.9 *If a point* (u_*, λ^*) *is a saddle point of the Lagrangian* (4.27) *the duality gap* (4.30) *is zero. So that* u_* *(resp.* λ^**) is a solution of the primal problem (rep. of the dual problem).*

PROOF. If (u_*, λ^*) is a saddle point of L then the inequalities (4.31) show that

$$\inf_{v\in E} L(v, \lambda^*) \geq L(u_*, \lambda^*) \geq L(u_*, \lambda) \quad \forall \lambda \geq 0$$

so that

$$\sup_{\lambda\geq 0} L^*(\lambda) \geq \sup_{\lambda\geq 0} L(u_*, \lambda) \geq \inf_{v\in E} L_*(v)$$

As the converse inequality always holds[16], we have actually proofed that the duality gap (4.30) is zero. □

REMARK 4.6 Assume that J and the mappings $\left(g_i\right)_{i=1}^n$ are convex. Let (u_*, λ^*) be a point in $E \times \mathbb{R}^n_+$ satisfying the conditions (4.24) of Proposition 4.8 and assume the constraints are qualified at u_* then (u_*, λ^*) is a saddle point of the Lagrangian (4.27).

[16] Let $L : E \times F \to \mathbb{R}$ be a numerical mapping defined on a Cartesian product $E \times F$ then, defining the mapping $G(x) = \inf_{y\in F} L(x, y)$, we have

$$G(x) \leq L(x, y) \quad \forall x \forall y$$

so that $\sup_{x\in E} G(x) \leq \sup_{x\in E} L(x, y)$ for all $y \in F$ and at last

$$\sup_{x\in E}\left(\inf_{y\in F} L(x, y)\right) \leq \inf_{y\in F}\left(\sup_{x\in E} L(x, y)\right)$$

This inequality is referred to as the weak min-max inequality.

PROOF. As the functional J and the constraints g_i are assumed to be convex, the condition

$$\nabla J(u_*) + \sum_{i=1}^{n} \lambda_i^* \nabla g_i(u_*) = 0$$

entails $L(u_*, \lambda^*) \leq L(v, \lambda^*)$ for all $v \in \Omega$. As $g_i(u_*) \leq 0$ for all i, the conditions $\lambda_i^* g_i(u_*) = 0$ entail

$$L(u_*, \lambda) = J(u_*) + \sum_{i=1}^{n} \lambda_i g_i(u_*) \leq J(u_*) = L(u_*, \lambda^*) \quad \forall \lambda \geq 0$$

so that (u_*, λ^*) is a saddle point of (4.27). □

EXAMPLE 4.6 We consider the minimization of $J(v) = \frac{1}{2} < [A]v, v > - < b, v >$ under the constraints $v \in \mathbb{R}^m$ such that $[B]v - c \leq 0$, where

- $[A]$ is a $m \times m$ symmetric definite positive matrix, $b \in \mathbb{R}^m$
- $[B]$ is a $m \times n$ matrix and c is a vector of $\mathbb{R}^n$.

As the functional J is convex coercive, and the set $U = \left\{ v \in \mathbb{R}^m \,;\, [B]v - c \leq 0 \right\}$ is convex, the optimization problem

$$J(u_*) = \inf_{v \in U} J(v)$$

has an unique solution which is the first argument of a saddle point of the Lagrangian

$$L(v, \lambda) = \frac{1}{2} < [A]v, v > - < b - [B]^t \lambda, v > + < c, \lambda >$$

Using the algorithm (4.4), such a saddle point (u_*, λ^*) can be obtained as the limit of the sequences

$$u_k = [A]^{-1} \left(b - [B]^t \lambda^k \right) \quad \lambda^{k+1} = \max \left[\lambda^k + \rho \left([B] u_k - c \right), 0 \right]$$

where ρ is a given positive constant[17]. In this case, the opposite of the dual mapping L^* defined in formula (4.29) is

$$\frac{1}{2} \langle [B][A]^{-1}[B]^t \lambda, \lambda \rangle - \langle [B][A]^{-1} b - c, \lambda \rangle + \frac{1}{2} \langle [A]^{-1} b, b \rangle$$

This shows *that if* $[B]$ *is of rank* m the matrix $[B][A]^{-1}[B]^t$ is definite positive, so that L^* is strictly concave and *the dual problem as only one solution* λ^*. Notice that we have actually solved the system of non-linear equations

$$[A]u + [B]^t \lambda = b \quad \text{with} \quad [B]u - c \leq 0$$

$$\lambda_i = \begin{cases} 0 & \text{if } \sum_{j=1}^{m} B_{ij} u_j < c_i \\ \geq 0 & \text{if } \sum_{j=1}^{m} B_{ij} u_j = c_i \end{cases}$$

Sequential Quadratic Programming Method. The SQP algorithm 4.5 is a second order method providing a sequence $\left((v_k, \lambda^k) \right)_k$ approximating both the solutions of the primal and dual problem. We simply indicate that this algorithm is an implementation of the Newton's method for the resolution of the Kuhn-Tucker equations (4.24) written

[17] It can be shown, see CIARLET[**9**], that the algorithm 4.4 converges if $0 < \rho < \frac{2\sigma_1(A)}{\|B\|^2}$, where $\sigma_1(A)$ is the smallest eigenvalue of A.

Algorithm 4.5: Main steps of the SQP algorithm

input : Starting point $u_0 \in E$ and $\lambda^0 \in \mathbb{R}^n_+$.

outputs: Minimizer $u_* := u_k$ of J satisfying the constraints $g_i(u) \leq 0$.

while $\|u_k - u_{k+1}\| \geq \varepsilon$ **do**

$1^o/$ solve the quadratic sub-problem

$$\hat{J}(d_*) = \min_{d \in \hat{U}} \hat{J}(d) \tag{4.32}$$

where

- $\hat{J}$ is the mapping

$$d \in E \mapsto \hat{J}(d) = \langle \left[\nabla^2 J(u_k) + \lambda^k \nabla^2 g(u_k)\right] d, d\rangle + \langle \nabla J(u_k), d\rangle \tag{4.33}$$

- and $\hat{U}$ is defined by

$$\hat{U} = \left\{d \in E\,;\, < \nabla g_i(u_k), d > + g_i(u_k) \leq 0 \text{ for } 1 \leq i \leq n\right\} \tag{4.34}$$

$2^o/$ set $u_{k+1} := u_k + d_*$ and $\lambda^{k+1} = \lambda^k + \lambda^*$, where λ^* is the Lagrange multiplier associated with the constrained optimization problem (4.32).

$3^o/$ Set $k := k+1$

end

in a variational form[18]. We refer to BONNANS [**3**], BONNANS Et al. [**4**], and IZMAILOV Et al. [**17**] for the convergence analysis, which is quadratic when the initial point u_0 is located in a neighborhood of a local optimum.

Due to its super-linear convergence, the SQP algorithm is very popular (see for instance the website http://www.klaus-schittkowski.de/) and the following improvements are made in its operational versions:

- *to avoid computation of second order derivatives in formula* (4.33), the Hessian matrix $\left[\nabla^2 J(u_k) + \lambda^k \nabla^2 g(u_k)\right]$ is replaced by a symmetric definite positive matrix $[H^k]$ and the sequence $\left([H^k]\right)_k$ is tuned to converge to $\left[\nabla^2 J(u_*) + \lambda^* \nabla^2 g(u_*)\right]$;
- *step* $2^o/$ *in algorithm* (4.5) *is often replaced by a line search phase* and u_{k+1} is defined by $u_{k+1} := u_k + \alpha_k d^*$ where α_k is chosen so that

$$\psi_k(u_k + \alpha_k d^*) < \psi_k(u_k)$$

where ψ_k is a suitable merit function allowing to bring ASAP the sequence $(u_k)_k$ in a neighborhood of a local minimizer of J regardless the starting

[18] Considering the mapping

$$x = (u, \lambda) \in E \times \mathbb{R}^n \mapsto \left(\nabla J(u) + \sum_{i=1}^n \lambda_i \nabla g_i(u), -g_1(u), \cdots - g_n(u)\right) \in E \times \mathbb{R}^n$$

we can check that the equations (4.24) can be written as $< F(x_*), x - x_* > \geq 0$ for all $x \in E \times \mathbb{R}^n_+$ or as

$$F(x) + \partial I_{E \times \mathbb{R}^n_+}(x) \ni 0 \quad \text{where } I_{E \times \mathbb{R}^n_+} \text{ is the characteristic function of } E \times \mathbb{R}^n_+ \tag{4.35}$$

In this context, the algorithm (4.5) consists to solve the variational inequality (4.35) by the following approximation sequence

$$F(x_k) + DF(x_k)(x_{k+1} - x_k) + \partial I_{E \times \mathbb{R}^n_+}(x_{k+1}) \ni 0$$

which is a generalized version the Newton-Raphson algorithm.

point u_0; a popular choice for the merit function is

$$\psi_k(u) = J(u) + \omega_k \sum_{i=1}^{n} \left(g_i^+(u)\right)^2$$

where $g_i^+(u) := \max(g_i(u), 0)$.

Penalty Method. The penalty method consists to replace the constraint optimization problem by a sequence of unconstrained problems whose solutions are expected to converge to the solution of the original problem. The unconstrained problem is obtained by adding to the objective $u \mapsto J(u)$ a penalty function $u \mapsto \psi(u) \in \mathbb{R}_+$ measuring the degree of violation of the constraints $g_i(u) \leq 0$. This leads to solve the constrained optimization problem by the algorithm 4.6, which can be set up with the help of two kinds of penalties:

$1^o/$ In the *exterior penalty method*, the penalty is of the form $\psi(u) = \sum_{k=1}^{n} \left(g_k^+(u)\right)^2$ and allows constraints violation;

$2^o/$ we can also introduce the penalty with the help of a *barrier function*, which is defined by $\psi(u) = \sum_{k=1}^{n} \ln(-g_k(u))$ and make sense only if $u \in Int(U)$. In this case, the initial point u_0 must satisfy $g_k(u_0) < 0$ and the barrier function prohibit the violation of constraints.

Algorithm 4.6: Penalty method

input : Starting point $u_0 \in E$.

outputs: Minimizer $u_* := u_k$ of J satisfying the constraints $g_i(u) \leq 0$.

Let be given $(\omega_k)_k$ a monotone sequence of positive numbers, and assume that

- $\lim_{k\to\infty} \omega_k = +\infty$ in case of exterior penalty;
- $\lim_{k\to\infty} \omega_k = 0$ for the barrier function.

while $\|\nabla J_k(u_k)\| \geq \varepsilon$ **do**

$1^o/$ set $J_{k+1}(u) := J(u) + \omega_{k+1}\psi(u)$

$2^o/$ start from the initial condition $u = u_k$ to compute an approximate solution u_{k+1} for the unconstrained optimization problem

$$J_{k+1}(u_*) = \inf_{u\in E} J_{k+1}(u)$$

$3^o/$ Set $k := k+1$

end

We can check that under suitable conditions on the criteria J, the sequence $(u_k)_k$ produced by the algorithm 4.6 is bounded and any accumulation point $\bar{u}$ of this sequence is solution of the constrained optimization problem. Considering if necessary a sub-sequence $(u_h)_h$ converging to $\bar{u}$, the limit

$$\lim_{h\to\infty} \omega_h g_i^+(u_h) \quad (\text{resp. } \lim_{h\to\infty} \omega_h \ln(g_i(u_h)))$$

exists and is the Lagrange multiplier λ_i associated with the KKT conditions (4.24).

Form a practical point of view, an exterior penalty function is used to find the minimizers of J located on the boundary of the domain U while a barrier function is preferred when the minimizers are in the interior of U.

Conclusions & software survey. In the previous subsections we have outlined some existence results for a given optimization problem and we have reviewed some algorithms allowing to reach the optima. These methods are implemented within several contexts of the structure analysis:

- From a theoretical point of view, optimization is usually used to discretize PDE or to proof well-posedness of a mechanical problem. *In this case the design space E is a functional space*, often an infinite dimensional vector space[19], *constituted of the virtual displacements (resp.virtual velocities)* modeling the cinematic of the considered mechanical system. The criterion J is an energy and in this case, *we essentially seek to establish coerciveness and convexity of J*. We provide in Exercises 4.4 and 4.5 page 190 two illustrative examples borrowed from the theories of beams and unilateral contact.
- In the engineering field, optimization allows to improve the design of a mechanical structure. In this context, see Example 4.7, the designer has to
  - 1^o/ parametrize a relevant numerical model of the studied structure (FEM model parametrized for instance by thicknesses, sections, inertia, masses etc.);
  - 2^o/ define an optimization criterion J (such as mass, stiffness, damage etc.) to qualify the designing goal;
  - 3^o/ identify the areas U where the product or process parameters can be run safely. This set, referred to as design space, will be assumed to be a multi-dimensional set defined by box constraints or by inequalities connecting together the model parameters defined in step 1^o/.

  Thought in this way, an engineering optimization problem can be written in the form (4.2) page 140 and numerically solved with the help of the algorithms introduced in the previous sub-sections. *In this case the problem is posed on a finite dimensional vector space but convexity of J can hardly be expected.* This means that the previously defined optimization methods will provide a *stationary point of the criterion and at the best a local optimum*, usually located on the boundary of the design space.

EXAMPLE 4.7 The designer may consider maximizing the stiffness of the electic post shown in figure (Fig. 4.8); for this purpose

- he assumes that the structure consists in an assembly of N beams whose cross-sections parameters d_i can take values between two bounds a_i and b_i,
- and looks for cross-sections diameters d_i that minimize the compliance[20] $\mathscr{C}$ of the post submitted to its one weight and given external forces $\vec{F}$.

In this case, $\mathscr{C}$ depends on the variables d_i $(1 \le i \le N)$ and, considering a FEM model of the post, it is the scalar product

$$\mathscr{C}(d_1, \cdots, d_N) = < u(d_1, \cdots, d_N)\ ,\ f >$$

[19] In case of FEM interpolation, the design space E is made up of piecewise polynomial functions and is a finite dimensional vector space.

[20] It is a choice to reduce the question of "stiffness increase" to the minimization of a compliance. The engineer might as well thought to minimize the dynamic compliance in a given frequency range etc.

where f and u are respectively the nodal forces and the nodal displacements computed in solving equation

$$[K(d_1,\cdots,d_N)]u = f \text{ where } [K] \text{ is the stiffness matrix of the post}$$

We can check that, so formalized, the optimization problem

$$\mathscr{C}(d_*) = \min_{a_i \le d_i \le b_i} \mathscr{C}(d)$$

can be numerically solved by the descent algorithm 4.3. The derivatives of $\mathscr{C}$ with respect to the design variables $d_1,\cdots,d_N$ are indeed defined as

$$\partial_{d_j}\mathscr{C} = < \partial_{d_j} f \ , \ [K]^{-1} f > - < [\partial_{d_j} K][K]^{-1} f \ , \ [K]^{-1} f >$$

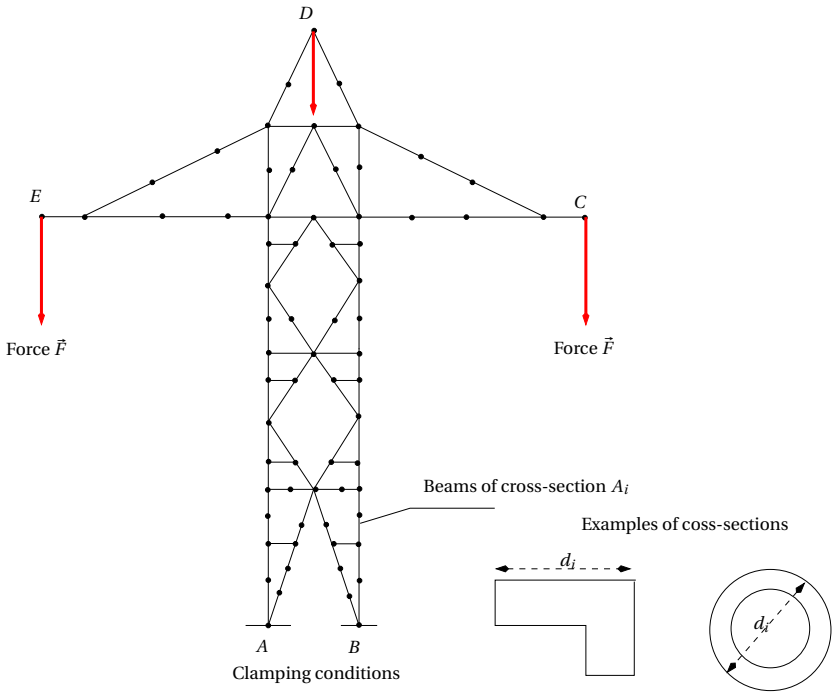


Fig. 4.8. **Example of beam structure**. In this case the electric post is made up of 90 beam elements, it is loaded by given forces $\vec{F}$ at points C, D, E, which are accounting for the weight of electrical cables.

Note that a SQP or a penalty algorithm (algorithms 4.5 and 4.6) allows in the same manner to minimize for instance the weight M of the post under constraint $\mathscr{C} \le \varepsilon$, where ε is a given constant. In this case, the design space is the set

$$U = \prod_{i=1}^{N} [a_i, b_i] \cap \left\{(d_1,\cdots,d_N) \in \mathbb{R}^N \, ; \, \mathscr{C}(d) \le \varepsilon\right\}$$

and *the designer must first of all verify that this set not empty (it is not a trivial task).*

Several optimization software are available, they are adapted to the specificities of the treated problem and we refer to LEYFFER et al. [**23**] for the optimization's software survey provided in table 4.1. In any cases the user must provide programs allowing to compute both the criterion, the constraints and their derivatives with respect to

the design parameters. It should be noticed that contrary to the example given in Example 4.7, there are situations where these derivatives can't be defined explicitly. In these cases computation of derivatives is carried out in solving an additional equation, referred to as adjoint equation[21]. The question of structural optimization under fatigue criterion introduced in Chapter 1 (problem (1.13) page 27) enters into this category of optimization problem and the rest of this chapter aims at writting down the adjoint equation and stetting up an algorithm for the resolution of this equation.

4.2. Adjoint State equation

We first define the adjoint equation associated with the criterion

$$J(u) = \int_0^T j(X_u(t))\,dt \quad \text{where } t \mapsto X_u(t) \quad \text{satifies the diffrerential equation:}$$
$$\frac{dX_u}{dt} = f(X_u, u, t) \text{ for } t \in [0,T]\ ;\ X_u(0) = X_0 \tag{4.37}$$

More precisely, we are intending

$1^o/$ to poof that if f satisfies the conditions of the Lemma 4.4, complemented by the Remark 4.7 then the mapping

$$u \in U \mapsto J(u) \in \mathbb{R}$$

is differentiable;

$2^o/$ and *to provide an algorithm to compute its derivative*.

This will set up the formal framework allowing to write down in the next Section the optimization problem posed in figure (Fig. 1.16) page 26.

The results obtained in this Section are summarized in the flowing Proposition:

PROPOSITION 4.10 $1^o/$ *Let* $f : (X,u,t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \mapsto f(X,u,t) \in \mathbb{R}^n$ *be a mapping continuously differentiable with respect to the variables* X, t *and differentiable with*

[21] Assume that we are looking for the computation of the derivatives of the mapping

$$u \mapsto J(u) := j(X_u), \text{ where } X_u \text{ is solution of an equation } F(X,u) = 0$$

To fix the ideas, j is assumed to be a functional defined on $\mathbb{R}^n$ and F is a mapping defined on $\mathbb{R}^n \times \mathbb{R}^m$ taking its values in $\mathbb{R}^n$. Using the chain rule

- we get $\partial_{u_i} J(u) = < \nabla j(X_u), \partial_{u_i} X(u) >$,
- and we can define the derivative $\partial_{u_i} X(u)$ as the solution of the equation

$$[D_X F(X(u),u)]\partial_{u_i} X(u) + \partial_{u_i} F(X(u),u) = 0$$

So that $\partial_{u_i} J(u) = - < [D_X F(X(u),u)]^{-T} \nabla j(X_u), \partial_{u_i} F(X(u),u) >$. Introducing the adjoint equation

$$[D_X F(X(u),u)]^T \lambda = -\nabla j(X_u) \tag{4.36}$$

posed on $\mathbb{R}^n$, we can write down the derivative $\partial_{u_i} J(u)$ as $< \lambda, \partial_{u_i} F(X(u),u) >$ where λ is solution of (4.36). Practical interest of this way of saying comes from the fact that building and resolution of state and adjoint equations can be performed in the same computational pass.

Name	Description	Interfaces	Language
ALGENCAN	Fortran code for general nonlinear programming that does not use matrix manipulations at all and, so, is able to solve extremely large problems with moderate computer time. The general algorithm is of Augmented Lagrangian type and the sub-problems are solved using GENCAN. GENCAN (included in ALGENCAN) is a Fortran code for minimizing a smooth function with a potentially large number of variables and box-constraints.	AMPL, C/C++, CUTEr, Java, MATLAB, Octave, Python, R	Fortran 77
CVXOPT	Free software package for convex optimization based on the Python programming language.	Python	Python
GALAHAD	Library particularly addressed to quadratic programming problems, containing both interior point and active set algorithms, as well as tools for preprocessing problems prior to solution. It also contains an updated version of the venerable nonlinear programming package, LANCELOT.	Library of Fortran 90 packages for largescale nonlinear optimization.	Fortran 90
IPOPT	Interior Point OPTimizer, pronounced eye-pea-Opt is a software package for large-scale nonlinear optimization. It is designed to find (local) solutions of mathematical optimization problems.	AMPL, CUTEr, C, C++, Fortran 77.	C++
KNITRO	Implements four state-of-the-art interior-point and active-set methods for solving continuous, nonlinear optimization problems.	AMPL, AIMMS, GAMS, MPL, MATLAB, R,C, C++, Java, Python or Fortran.	C++
NLPQL	Non-Linear Programming by Quadratic Lagrangian, is a sequential quadratic programming (SQP) method which solves problems with smooth continuously differentiable objective function and constraints. The algorithm uses a quadratic approximation of the Lagrangian function and a linearization of the constraints.	MATLAB,C, C++, Python or Fortran.	FORTRAN 77
NPSOL	Package for solving constrained optimization problems (nonlinear programs). It employs a dense SQP algorithm and is especially effective for nonlinear problems whose functions and gradients are expensive to evaluate. The functions should be smooth but need not be convex. An augmented Lagrangian merit function ensures convergence from an arbitrary point.	MATLAB, C or Fortran.	FORTRAN 77
SQPlab	The SQPlab (pronounce S-Q-P-lab) software is a modest Matlab implementation of the SQP algorithm for solving constrained optimization problems. The functions defining the problem can be nonlinear and nonconvex, but must be differentiable.	MATLAB.	MATLAB

Tab. 4.1. Short description of some non-linear optimization software;
a few of them are freely available on the Internet.

respect to the variable u in a neighborhood of $u_0 \in \mathbb{R}^m$. Then there is $T > 0$ such that the differential equation

$$\frac{dX}{dt} = f(X,u,t) \quad \text{with the initial condition } X(0) = X_0 \tag{4.38}$$

has an unique solution X_u on $[0,T[$ which is differentiable with respect to u at u_0.

2^o/ Moreover, if $X \in \mathbb{R}^n \mapsto j(X) \in \mathbb{R}$ is differentiable on an open subset $\mathcal{O}$ of $\mathbb{R}^n$ and if the set

$$\left\{ t \in [0,T[\ ;\ X_\varphi(t) \in \complement\mathcal{O} \right\}$$

is a null part of $[0,T]$ then the mapping

$$u \mapsto J(u) = \int_0^T j(X_u(t))\,dt \tag{4.39}$$

is differentiable with respect to u at u_0 and its gradient $\nabla J(u_0)$ can be computed as follows:

i) denoting by $t \in]0,T] \mapsto \Lambda(t) \in \mathbb{R}^n$ the solution of the differential equation, referred to as adjoint equation

$$\boxed{\begin{array}{l} \dfrac{d\Lambda}{dt} + D_X f(X_{u_0}(t),u_0,t)^t \Lambda = -\nabla j(X_{u_0}(t)) \\ \text{with the ending condition } \lambda(T) = 0 \end{array}} \tag{4.40}$$

integrated backward in time,

ii) the gradient of J at u_0 is defined by the formula

$$\boxed{\nabla J(u_0) = \int_0^T D_u f(X_{u_0}(t),u_0,t)^t \Lambda(t)\,dt} \tag{4.41}$$

PROOF. We start the proof with the following technical Lemma.

LEMMA 4.4 *Assume that the mapping f satisfies the condition 1^o/ of Proposition 4.10. Then, for any $X_0 \in \mathbb{R}^n$ there is $\eta_0 > 0$ such that the differential equation*

$$\frac{dX}{dt} = f(X,u,t) \quad X(t_0) = X_0 \tag{4.42}$$

has an unique solution $t \in [t_0, t_0+\eta_0[\mapsto X_u(t)$. This solution is differentiable with respect to u in any direction Ψ at u_0 and its derivative[22] $\partial_u X_u$ is solution of the differential equation

$$\begin{array}{l} \dfrac{d\delta}{dt} = D_X f(X_{u_0}(t),u_0,t).\delta + D_u f(X_{u_0}(t),u_0,t).\Psi \text{ on } [0,\eta_0[\\ \text{with the initial condition } \delta(t_0) = 0 \end{array} \tag{4.43}$$

PROOF OF LEMMA 4.4. Denoting by $C_0^1([0,T],\mathbb{R}^n)$ the space of the continuously differentiable mappings $V : [0,1] \to \mathbb{R}^n$ such that $V(0) = 0$, we will see that this result is

[22] Since it is a directional derivative and not a derivation the notation is abusive.

a consequence of the implicit function theorem[23] applied to the mapping F defined as follows

$$
\begin{aligned}
&(v,\eta,\tau) \in C_0^1\left([0,T],\mathbb{R}^n\right) \times \mathbb{R} \times \mathbb{R} \mapsto F(V,\eta,\tau) \in C^0\left([0,1],\mathbb{R}^n\right) \\
&F(V,\eta,\tau) := V' - \eta\, f(X_0 + V, u_0 + \tau\Psi, t_0 + \eta\,.)
\end{aligned} \tag{4.45}
$$

where u_0 (resp. Ψ) is an arbitrary point (resp. direction) in $\mathbb{R}^m$.

As we have the following results :

$1^o/$ F is continuously differentiable and satisfies the equation $F(0,0,0) = 0$;
$2^o/$ its derivative $D_1F(0,0,0)$, which is the linear operator

$$V \in C_0^1\left([0,T],\mathbb{R}^n\right) \mapsto V' \in C^0([0,1],\mathbb{R}^n)$$

satisfies the following properties:

- $D_1F(0,0,0)$ is continuous: it is an immediate consequence of the inequality $\left\|V'\right\|_{C^0} \leq \|V\|_{C^1}$, which takes place for any $V \in C^1([0,T],\mathbb{R}^n)$;
- $D_1F(0,0,0)$ is bijective because for any $f \in C^0([0,T],\mathbb{R}^n)$, the mapping $g(t) = \int_0^t f(s)ds$ is in $C_0^1([0,T],\mathbb{R}^n)$ and verifies $g'(t) = f(t)$ for any $t \in [0,1]$.

the hypotheses of the implicit function theorem 4.1 are satisfied; we can thus define a neighborhood $\mathcal{U}_0 \times]-\eta_1,\eta_1[\times]\tau_0,\tau_0[$ of $0 \in C_0^1([0,T],\mathbb{R}^n) \times \mathbb{R} \times \mathbb{R}$ and a mapping

$$(\eta,\tau) \in]-\eta_1,\eta_1[\times]-\tau_0,\tau_0[\mapsto V(\eta,\tau) \in C_0^1\left([0,T],\mathbb{R}^n\right)$$

such that[24]:

$$F(V(\eta,\tau),\eta,\tau) = 0 \quad \text{for any} \quad (\eta,\tau) \in]-\eta_1,\eta_1[\times]-\tau_0,\tau_0[$$

Let $\eta_0 \in \left]0,\eta_1\right[$ be given, the mapping

$$t \in [t_0, t_0+\eta_0] \mapsto X_\tau(t) := X_0 + V(\eta_0,\tau)\left(\frac{t-t_0}{\eta_0}\right) \in \mathbb{R}^n \tag{4.46}$$

satisfies

$$
\begin{aligned}
&\frac{dX_\tau}{dt}(t) = \frac{1}{\eta_0} V'(\eta_0,\tau)\left(\frac{t-t_0}{\eta_0}\right) = f(X_\tau, u_0 + \tau\Psi, t) \\
&X_\tau(t_0) = X_0
\end{aligned}
$$

[23]The proof of the following theorem can be found in any course of analyze, see for instance SCHWARTZ [**37**]. Note that this theorem, while fundamental in analysis, is written in many texts with useless hypotheses on the spaces Y and Z. It is not necessary to assume that these spaces are Banach!

THEOREM 4.1 (Implicit function theorem) *Let X be a Banach space, Y, Z be normed spaces and let $F : X \times Y \to Z$ be differentialble mapping defined in a neighborhood $\mathcal{U} \times \mathcal{V}$ of a point $(x_0, y_0) \in X \times Y$; assume that*

i) x_0 is solution of the equation $F(x_0, y_0) = 0$
ii) and that the derivative $D_1F(x_0, y_0)$ is a bijective continuous linear mapping between X and Z.

Then there are two neighborhoods $\mathcal{U}_0 \subset \mathcal{U}$ and $\mathcal{V}_0 \subset \mathcal{V}$ of x_0 and y_0 respectively such that for any $y \in \mathcal{V}_0$ the equation $F(x,y) = 0$ has one and only one solution $x(y)$ in $\mathcal{U}_0$. The mapping $y \in \mathcal{V}_0 \mapsto x(y) \in \mathcal{U}_0$ is moreover differentiable with respect to $y \in \mathcal{U}_0$ and its derivative is defined by

$$Dx(y) = -[D_1f]^{-1}(x(y),y) \circ [D_2f](x(y),y) \text{ for all } y \in \mathcal{V}_0 \tag{4.44}$$

[24]Note that this equation takes place in $C^0([0,1],\mathbb{R}^n)$.

and is solution of the differential equation (4.42) for any parameter u of the form

$$u = u_0 + \tau\,\Psi \text{ when } \tau \text{ ranges in the interval }]-\tau_0, \tau_0[$$

The mapping $\tau \mapsto V(\eta_0, \tau)$ is moreover differentibale with respect to τ and the formula (4.44) shows that its derivative satisfies the differential equation[25]

$$\begin{aligned}(\partial_\tau V)' = \eta\, D_X f(x_0 + V(\eta,\tau), u_0 + \tau\Psi, t_0 + \eta.).\partial_\tau V \\ +\eta\, D_\varphi f(x_0 + V(\eta,\tau), u_0 + \tau\Psi, t_0 + \eta.).\Psi\end{aligned}$$

This proofs that X_τ defined in (4.46) is differentiable with respect to τ and that its derivative is solution of the differential equation (4.43).

Applying the Taylor formula to the mapping $\tau \in]-\tau_0, \tau_0[\mapsto V(\eta_0,\tau) \in C_0^1([0,1],\mathbb{R}^n)$ we have on the other hand

$$\begin{aligned}&V(\eta_0,\tau) = V(\eta_0,0) + \tau\ (\partial_\tau V)(\eta_0,0) + |\tau|\,\varepsilon(\tau) \\ &\text{with} \quad \lim_{\tau\to 0} \|\varepsilon(\tau)\|_{C^1} = 0\end{aligned} \tag{4.47}$$

□

REMARKS 4.7 1^o/ Other existence results for the differential equation (4.42) might be obtained under weaker regularity conditions on the mapping $X \mapsto f(X,u,t)$. But as we wish the solution $X_u(t)$ to be diffrentiable with respect to u, we must impose some regularity conditions on f, and this justifies the use of the implicit fuction theorem to proof the Lemma 4.4.

2^o/ Lemma 4.4, which is a local existence result, allows to show that the equation (4.38) has a solution on a maximal time interval $[0, T_{max}[$ and that the condition $T_{max} < +\infty$ entails $\lim_{t\to T_{max}} \|X(t)\| = +\infty$. To see this:

- *i*) use the Lemma 4.4 to extend to $[0, \tilde{T} + \eta_0[$ a solution of the equation (4.38) which is defined on a time interval $[0, \tilde{T}]$;
- *ii*) call T_{max} the upper bound of the positives numbers $\tilde{T}$ for which the previous operation can be performed; note that the condition $\lim_{t\to T_{max}} \|X(t)\| < +\infty$ entails $T_{max} = +\infty$; indeed, assuming the opposite would mean that the solution $t \mapsto X(t)$ could be extended beyond T_{max} (as asserted in Lemma 4.4) and would contradict the definition of T_{max}.

We will subsequently assume that $T < T_{max}$ and that the solution X of (4.42) is defined on a closed interval $[0, T]$.

[25] One can check that the derivative of F with respect to $V \in C_0^1([0,T],\mathbb{R}^n)$ is the linear mapping

$$H \in C_0^1([0,T],\mathbb{R}^n) \mapsto H' - \eta D_X f(X_0 + v, ...).H \in C^0([0,T],\mathbb{R}^n)$$

so that its inverse is the linear mapping which associate to any $W \in C^0([0,T],\mathbb{R}^n)$ the solution $H \in C_0^1([0,T],\mathbb{R}^n)$ of the differential equation

$$\frac{dH}{dt} = \eta D_X f(X_0 + v, ...).H + W$$

3^o/ We can proof in the same way that the derivative in the direction Ψ of the solution X_u of the differential equation (4.38) is defined at a point u_0 by[26]

$$\begin{aligned} &\frac{d\delta}{dt} = D_X f(X_{u_0}(t), u_0, t).\delta + D_u f(X_{u_0}(t), u_0, t).\Psi \quad \text{on} \quad [0, T_{max}[\\ &\text{with the initial condition } \delta(0) = 0 \end{aligned} \tag{4.48}$$

To complete the proof of Proposition 4.10, we have to compute the limit

$$\lim_{\tau\to 0} \frac{1}{\tau} \int_0^T \left[j(X_{u_0+\tau\Psi}(t)) - j(X_{u_0}(t)) \right] dt \tag{4.49}$$

where $X_{u_0+\tau\Psi}(t)$ is the solution of the differential equation (4.38) for the particular values $u = u_0 + \tau\Psi$.

To this end, we will assume that

- the set $\mathscr{F}$ of the points $X \in \mathbb{R}^n$ where j is not differentiable is a closed subset of $\mathbb{R}^n$, then the following formula makes sense for each t such that $X_{u_0}(t) \in \complement\mathscr{F}$

$$\begin{aligned} j(X_{u_0+\tau\Psi}(t)) - j(X_{u_0}(t)) = \langle \nabla j(X_{u_0}(t)) \,,\, X_{u_0+\tau\Psi}(t) - X_{u_0}(t) \rangle + \\ + \left\| X_{u_0+\tau\Psi}(t) - X_{u_0}(t) \right\| o\left(X_{\varphi_0+\tau\Psi}(t) - X_{\varphi_0}(t) \right) \\ \text{with } \lim_{X\to 0} o(X) = 0 \end{aligned} \tag{4.50}$$

- and that *for each $u \in U$, the set*

$$\{ t \in [0, T] \,;\, X_u(t) \in \mathscr{F} \}$$

is a null set. Then the formula (4.50) takes place almost everywhere in $[0, T]$ and the integral

$$\delta J(u) = \int_0^T \left[j(X_{u+\tau\Psi}(t)) - j(X_u(t)) \right] dt$$

can be obtained by integrating on $[0, T]$ the right hand member of the equation (4.50).

Let δ be the solution of the differential equation (4.48); taking into account the definition of v introduced in the proof of Lemma 4.4, the formula (4.47) can be rewritten as

$$\begin{aligned} &X_{u_0+\tau\Psi}(t) - X_{u_0}(t) = \tau\delta(t) + |\tau|\varepsilon(\tau, t) \quad \text{for any } t \in [0, T] \\ &\text{with } \lim_{\tau\to 0} \varepsilon(\tau, t) = 0 \text{ uniformly in } t \in [0, T] \end{aligned}$$

The formula (4.50) shows then that

$$\begin{aligned} \delta J(u_0) = \tau \int_0^T \langle \nabla j(X_{u_0}(t)) \,,\, \delta(t) \rangle \, dt + |\tau| \int_0^T \langle \nabla j(X_{u_0}(t)) \,,\, \varepsilon(\tau, t) \rangle \, dt \\ + |\tau| \int_0^T \| \delta(t) + \varepsilon(\tau, t) \| \, o\left(\tau\delta(t) + |\tau|\varepsilon(\tau, t) \right) dt \end{aligned}$$

[26] To do this, we must assume in the Lemma 4.4 that the initial condition X_0 depends on u and replace the initial condition $\delta(t_0) = 0$ of the equation (4.43) by $\delta(t_0) = D_u X_0.\Psi$.

As $t \mapsto \varepsilon(\tau, t)$ converges uniformly to 0, we have[27]:

$$\lim_{\tau \to 0} \left| \int_0^T \langle \nabla j(X_{u_0}(t)), \varepsilon(\tau, t) \rangle \, dt \right| \leq \left(\int_0^T \left\| \nabla j(X_{u_0}(t)) \right\| dt \right) \lim_{\tau \to 0} sup_{t \in [0,T]} \|\varepsilon(\tau, t)\| = 0$$

Since the mapping $t \mapsto \delta(t) + \varepsilon(\tau, t)$ is of class C^1 on $[0, T]$, the previous inequality and the formula (4.47) show that the limit (4.49) exists; this means that J is differentiable in the direction Ψ at u_0 and that its derivative is

$$D_\Psi J(u_0) = \int_0^T \langle \nabla j(X_{u_0}(t)), \delta(t) \rangle \, dt \tag{4.51}$$

The objective is now to rewrite this derivative as the following scalar product:

$$D_\Psi J(u_0) = \langle \nabla J(u_0), \Psi \rangle$$

To this end, we will assume that f is continuously differentiable with respect of φ and we introduce an *adjoint variable*[28]

$$t \in [0, T] \mapsto \Lambda(t) \in \mathbb{R}^m$$

Whose definition will be specified as needed. Computing the scalar product of the equation (4.43) by Λ and integrating the obtained result on $[0, T]$ we have

$$\int_0^T \langle \frac{d\delta}{dt}, \Lambda \rangle = \int_0^T \langle \delta, D_X f(X_{u_0}, u_0, t)^t \Lambda \rangle + \int_0^T \langle \Psi, D_u f(X_{u_0}, u_0, t)^t \Lambda \rangle \tag{4.52}$$

integrating by parts the left hand side of this equation, we get:

$$\int_0^T \langle \frac{d\delta}{dt}, \Lambda \rangle = \langle \frac{d\delta}{dt}, \Lambda \rangle_0^T - \int_0^T \langle \delta, \frac{d\Lambda}{dt} \rangle$$

Now, if Λ is defined as the solution of the linear differential equation

$$\frac{d\Lambda}{dt} = -D_X f(X_{u_0}(t), u_0, t)^t \Lambda - \nabla j(X_{u_0}(t))$$
$$\text{with the ending condition } \Lambda(T) = 0$$

we deduce from (4.52) that

$$\int_0^T \langle \Psi, D_u f(X_{u_0}, u_0, t)^t \Lambda(t) \rangle \, dt = \int_0^T \langle \nabla j(X_{u_0}(t)), \delta(t) \rangle \, dt$$

Taking into account of (4.51), this shows that $\nabla J(u_0)$ can be defined as

$$\nabla J(u_0) = \int_0^T D_\varphi f(X_{u_0}(t), u_0, t)^t \Lambda(t) \, dt$$

□

[27] Because the assumptions made about the functions j and X_{φ_0} entail that $t \mapsto \nabla j(X_{\varphi_0}(t))$ is bounded with respect to the norm $\|.\|_\infty$.

[28] Also called Lagrange multiplier.

REMARKS 4.8 1^o/ When the integrand $j(X,u)$ of the criterion (4.39) depends explicitly on the variable u, the gradient of J is defined by

$$\nabla J(u_0) = \int_0^T D_u j(X_{u_0}(t), u_0)\, dt + \int_0^T D_u f(X_{u_0}(t), u_0, t)^t \Lambda(t)\, dt \tag{4.53}$$

where Λ is the solution of the adjoint equation (4.40) with the right hand member

$$-D_X j(X_{u_0}(t), u_0) \tag{4.54}$$

2^o/ The adjoint equation (4.40) is a linear differential equation excited by the term $\nabla j(X_{u_0}(t))$ which is discontinuous on a null subset of $[0,T]$ but remains bounded in the $\|\,.\,\|_\infty$ norm, thus the adjoint equation has a solution in $W^{1,\infty}([0,T],\mathbb{R}^n)$ and the integral (4.41), which defines the gradient $\nabla J(u_0)$, always makes sense.

3^o/ There is no evidence that the gradient $\nabla J(u_0)$ remains bounded when the horizon T goes to $+\infty$ because even if the solution $X_{u_0}(t)$ of the state equation is periodic, the term $\nabla j(X_{u_0}(t))$ can excite the poles[29] of the linear operator $\Lambda \mapsto D_X f\left(X_{u_0}, u_0, t\right)\Lambda$ which can be purely imaginary if the state equation is undamped, *this means that the adjoint equation can be unstable*.

4^o/ When the state equation derives from a second-order linear system such as (3.3) page 96, the adjoint equation is

$$\frac{d}{dt}\left\{\begin{array}{c}\Lambda_1\\ \Lambda_2\end{array}\right\} = \begin{bmatrix} 0 & [K][M]^{-1} \\ -I & [W][M]^{-1}\end{bmatrix}\left\{\begin{array}{c}\Lambda_1\\ \Lambda_2\end{array}\right\} - \left\{\begin{array}{c} D_x j(x_{u_0}, y_{u_0})\\ D_y j(x_{u_0}, y_{u_0})\end{array}\right\}$$

and the change of variables $\Lambda_i \mapsto \Lambda_i = [M]^{\frac{1}{2}}[Q]\hat{\Lambda}_i$ (similar to the one introduced in Section 3.1) diagonalizes this equation under the form

$$\frac{d}{dt}\left\{\begin{array}{c}\hat{\Lambda}_1\\ \hat{\Lambda}_2\end{array}\right\} = \begin{bmatrix} 0 & \ulcorner k_{ii}\lrcorner \\ -Id_{ii} & \ulcorner c_{ii}\lrcorner\end{bmatrix}\left\{\begin{array}{c}\hat{\Lambda}_1\\ \hat{\Lambda}_2\end{array}\right\} - \left\{\begin{array}{c} [Q]^t[M]^{-\frac{1}{2}} D_x j(x_{u_0}, y_{u_0})\\ [Q]^t[M]^{-\frac{1}{2}} D_y j(x_{u_0}, y_{u_0})\end{array}\right\} \tag{4.55}$$

Algorithmic implementation. We have thus to do the following operations to compute a descent direction of the optimization problem (4.37):

1^o/ solve the state equation (4.38) to compute $X_{u_0}(t)$ for $0 \le t \le T$ and the value $J(u_0)$ of the criterion;

2^o/ solve backward in time the adjoint equation (4.40) with the results of the previous step to calculate $\Lambda(t)$ $(0 \le t \le T)$; *the algorithm* 4.7 *summarizes the computations which are to be performed to integrate the state equation and its adjoint by a finite difference method*;

3^o/ define the descent direction by computing the integral (4.41) with the help of the stored data $X_{u_0}(t)$ and $\Lambda(t)$.

[29] And this happens at each point of discontinuity of the excitation $\nabla j(X_{u_0}(t))$.

Algorithm 4.7: *Simultaneous integration of the sate and adjoint equations by a backward Euler method (unconditionally stable).*

input : Integration step size h; we assume that $h = \frac{T}{N_{samp}}$

output : Tables $X = \left(X_i\right)_{i=0}^{N_{samp}}$ and $\lambda = \left(\lambda_i\right)_{i=0}^{N_{samp}}$ containing state and adjoint state sampling.

begin

1) **Integration of the state equation**

initialisation :For $i = 0$, $X_i \leftarrow X_0$, where X_0 is the initial condition for the state equation

for $i = 1$ **to** N_{samp} **do**

- **Solve the equation** $Z - hf(Z, hi) = X_{i-1}$ with the help of a fixed point algorithm: if the mapping $Z \mapsto f(Z, hi)$ is Lipschitz, one can choose h small enough so that $Z \mapsto hf(Z, hi) - X_{i-1}$ is contracting.
- $Z_0 \leftarrow X_{i-1}$ and $Z \leftarrow 0$

  while $\|Z - Z_0\| \geq \varepsilon$ **do**
  - $Z \leftarrow X_{i-1} + hf(Z_0, hi)$
  - $Z_0 \leftarrow Z$

  end
- $X_i \leftarrow Z$

end

2) **Integration of the adjoint equation**

initialisation : $\Lambda_{N_{samp}} \leftarrow 0$

for $i = N_{samp} - 1$ **to** 0 **do**

- **Compute the derivative** $D_X f(X_i, ih)$ **and the gradient** $\nabla j(X_i)$
- **Solve the linear equation**

$$\Lambda_i - hD_X f(X_i, hi)^t \Lambda_i = \Lambda_{i+1} + h\nabla j(X_i)$$

  to compute Λ_i from Λ_{i+1}.

end

end

When the state equation is obtained from a FEM model, the amount of data that must be processed to compute a descent direction may seems unacceptable. However, when we deal with a system of linear equations with constant coefficients, the computations can be simplified in integrating the state and the adjoint equations with the help of the forced response method introduced in Chapter 3. The Remarks 4.9 explain how to handle the task.

REMARKS 4.9 1^o/ In order to integrate the quation (4.55), which is set backward in time, by the forced response method introduced in the Section 3.1 we must reformulate this system of equations in the increasing direction of time. To this end, we make the change of unknowns

$$\hat{\Lambda}_j \mapsto \bar{\Lambda}_j \ (j = 1, 2) \quad \text{defined by} \quad \bar{\Lambda}_j(t) := \hat{\Lambda}_j(T - t) \text{ for } t \in [0, T]$$

to rewrite (4.55) as

$$\frac{d}{dt}\left\{\begin{array}{c} \bar{\lambda}_1 \\ \bar{\lambda}_2 \end{array}\right\} = \begin{bmatrix} 0 & -\ulcorner k_{ii} \urcorner \\ Id_{ii} & -\ulcorner c_{ii} \urcorner \end{bmatrix} \left\{\begin{array}{c} \bar{\lambda}_1 \\ \bar{\lambda}_2 \end{array}\right\} + \left\{\begin{array}{c} [Q]^t [M]^{-\frac{1}{2}} D_x j(\bar{x}_{u_0}, \bar{y}_{u_0}) \\ [Q]^t [M]^{-\frac{1}{2}} D_y j(\bar{x}_{u_0}, \bar{y}_{u_0}) \end{array}\right\} \tag{4.56}$$

and *integrate this system of equations between* 0 *and* T*, with the initial condition* $\bar{\lambda}_j(0) = 0$.

Next, an appropriate renumbering of the equations allows to reduce (4.56) to a system of uncoupled oscillators

$$\frac{d}{dt}\left\{\begin{array}{c}\bar{\lambda}_1\\ \bar{\lambda}_2\end{array}\right\}=\left[\begin{array}{cc}0 & -\omega^2\\ 1 & -c\end{array}\right]\left\{\begin{array}{c}\bar{\lambda}_1\\ \bar{\lambda}_2\end{array}\right\}+\left\{\begin{array}{c}\bar{g}_1(t)\\ \bar{g}_2(t)\end{array}\right\}$$

which, modulo a transposition, was studied in the Section (3.1) page 98. For sub-critical damping[30], denoting $\delta=\frac{1}{2}\sqrt{4\omega^2-c^2}$, *this equation can be solved by the convolution product*

$$\overline{\Lambda}(t)=\int_0^t [N(t-s)]\,\bar{g}(s)\,ds$$

where the kernel $t\mapsto[N(t)]$ *is now the following* 2×2 *matrix*:

$$\boxed{[N(t)]=e^{-\frac{ct}{2}}\left[\begin{array}{cc}\frac{c}{2\delta}\sin(\delta t)+\cos(\delta t) & -\frac{\omega^2}{\delta}\sin(\delta t)\\ \frac{1}{\delta}\sin(\delta t) & \cos(\delta t)-\frac{c}{2\delta}\sin(\delta t)\end{array}\right]} \tag{4.57}$$

2^o/ In contrast with what happens for the integration of the state equation, the terms $\bar{g}_i(s)$ (for $i=1,2$) are generally both non-zero.

3^o/ The algorithm 4.8 summarizes the computation steps to perform the simultaneous integration of the state and the adjoint equations by the forced response method. *The advantage is to limit the volume of the data which are to be stored*; we only store[31] 5 tables of dimensions $k\times N_{samp}$ where k is the dimension of the reduced state equation and N_{samp} is the number of samples of the excitations.

A first illustration. We conclude this Section by treating the following example, which is intended to illustrate the implementation of the previously introduced method on a simple case.

EXAMPLE 4.8 We are intending to identify the coefficients ω and c of the second order equation

$$\ddot{\xi}+\omega^2\xi+c\dot{\xi}=\cos(\alpha_0 t)\quad\text{with the initial conditions}\quad\xi(0)=\dot{\xi}(0)=0$$

which minimizes the criterion

$$J(\omega,c)=\int_0^T|\sin(\alpha_1 t)-\xi(t)|\,dt$$

where ω and α_1 are two given angular velocities.

i) The second order equation is first written under the form of the first order system

$$\frac{d}{dt}\left\{\begin{array}{c}\xi_1\\ \xi_2\end{array}\right\}=\left[\begin{array}{cc}0 & 1\\ -\omega^2 & -c\end{array}\right]\left\{\begin{array}{c}\xi_1\\ \xi_2\end{array}\right\}+\left\{\begin{array}{c}0\\ \cos(\alpha_0 t)\end{array}\right\} \tag{4.63}$$

[30] We leave to the reader to deal with the cases of critical and over-critical damping.

[31] To be compared with a table of dimension $2m\times N_{samp}$ where m is the dimension of the state equation for the classical algorithm 4.7.

Algorithm 4.8: *Simultaneous integration of the state and adjoint equations by the forced response method.*

inputs :

- Sampling of excitations $\left(f_i\right)_{i=1}^{N_{samp}}$ and sampling frequency F_{samp}.
- Matrices $[M]^{-\frac{1}{2}}$ et $[M]^{\frac{1}{2}}$.
- Truncated modal base $[\hat{Q}_k]$ (with k modes) of the structure.

outputs :

- Criterion $J(u_0) = \int_0^T j(x(t), \dot{x}(t))\,dt$.
- Sampled solution $\left(\Lambda_i\right)_{i=1}^{N_{samp}}$ of the adjoint equation.

begin

1) **Pass the forces in the modal base**, $\hat{f}_i$ is the k-dimensional vector

$$\left(\hat{f}_i\right)_{i=1}^{N_{samp}} = [\hat{Q}_k]^t [M]^{-\frac{1}{2}} \left(f_i\right)_{i=1}^{N_{samp}} \tag{4.58}$$

2) **Computation of k convolution products**

$$\boxed{\left(\hat{x}_i\right)_{i=1}^{N_{samp}} = G_1 * \left(\hat{f}_i\right)_{i=1}^{N_{samp}} \quad \text{and} \quad \left(\hat{y}_i\right)_{i=1}^{N_{samp}} = G_2 * \left(\hat{f}_i\right)_{i=1}^{N_{samp}}} \tag{4.59}$$

3) **Computation of the criterion and the right hand member of the adjoint equation**
for $i = 1$ **to** N_{samp} **do**

- Go back to the original basis with the help of the matrices products

$$x_i = [M]^{-\frac{1}{2}} [\hat{Q}_k] \hat{x}_i \quad \text{and} \quad \dot{x}_i = [M]^{-\frac{1}{2}} [\hat{Q}_k] \hat{y}_i \tag{4.60}$$

- Sampling of the integrand $j(x_i, \dot{x}_i)$ and updating of the criterion $J(\varphi_0)$
- Compute the derivatives $D_x j(x_i, \dot{x}_i)$ and $D_{\dot{x}} j(x_i, \dot{x}_i)$
- Pass the derivatives of j in the modal base in applying the formula (4.58) to compute $\widehat{D_x j}$ and $\widehat{D_{\dot{x}} j}$.
- Store the results backward in time (ie. starting with the end)

$$\boxed{\begin{array}{l} \left(\overline{D_x j}\right)_{(N_{samp}+1-i)} \leftarrow \left(\widehat{D_x j}\right)_i \\ \left(\overline{D_{\dot{x}} j}\right)_{(N_{samp}+1-i)} \leftarrow \left(\widehat{D_{\dot{x}} j}\right)_i \end{array}} \tag{4.61}$$

end

4) **Compute the convolution products** with kernels (4.57). Note that the kernels (4.59) used to integrate the state equation are entries of the matrix $[N]$.

$$\left(\overline{\Lambda}_i\right)_{i=1}^{N_{samp}} = [N] * \left(\begin{array}{c} \overline{\nabla j_{x\,i}} \\ \overline{\nabla j_{\dot{x}\,i}} \end{array} \right)_{i=1}^{N_{samp}}$$

5) **Go back to sampling in ascending time** by a formula analogue to (4.61)

6) **Go back to the initial basis** by the base change

$$\Lambda_i = [M^{\frac{1}{2}}][\hat{Q}_k] \hat{\Lambda}_i \tag{4.62}$$

to compute the sampling $\left(\Lambda_i\right)_{i=1}^{N_{samp}}$ of the solution the adjoint equation.

end

parametrized by ω and c. In this case, the mapping f of the Proposition 4.10 is defined by

$$\left\{ \begin{array}{c} \xi_1 \\ \xi_2 \end{array} \right\} \in \mathbb{R}^2 \mapsto f(\xi_1, \xi_2, t) = \left\{ \begin{array}{c} \xi_1 \\ -\omega^2 \xi_1 - c\,\xi_2 + \cos(\alpha_0 t) \end{array} \right\} \in \mathbb{R}^2$$

and its derivative with respects of the parameters (ω, c), computed at (ω_0, c_0) is the linear mapping of matrix

$$D_{(\omega,c)}f = \begin{bmatrix} 0 & 0 \\ -2\omega_0\xi_1 & -\xi_2 \end{bmatrix}$$

ii) Thus the formula (4.41) allowing to compute the gradient of J at (ω_0, c_0), says that:

$$\begin{aligned} \partial_\omega J(\omega_0, c_0) &= -2\omega_0 \int_0^T \xi_1(t)\lambda_2(t)dt \\ \text{and } \partial_c J(\omega_0, c_0) &= -\int_0^T \xi_2(t)\lambda_2(t)dt \end{aligned} \tag{4.64}$$

iii) where $t \mapsto \Lambda(t) = (\lambda_1(t), \lambda_2(t))$ is the solution of the differential equation

$$\frac{d}{dt}\begin{Bmatrix} \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{bmatrix} 0 & \omega^2 \\ -1 & c \end{bmatrix}\begin{Bmatrix} \lambda_1 \\ \lambda_2 \end{Bmatrix} - \begin{Bmatrix} sign(\xi_1(t) - \sin\alpha_1 t) \\ 0 \end{Bmatrix} \tag{4.65}$$

Integrated backward in time from T to 0, with the initial condition

$$\lambda_1(T) = \lambda_2(T) = 0$$

The numerical application shown in figures (Fig. 4.9) and (Fig. 4.10) is carried out with $\omega = 2$ and $\alpha_1 = 4\,rd/s$. Once computed the derivatives (4.64), it only remains to use its " favorite optimizer " to compute the optimal values of the parameters ω and c.

4.3. Application to damage criterion

In this Section we specify what is said in the previous Section to the case where the state equation, depending on a design parameters u, is of the form (3.3) page 96 and where the optimization criterion is defined by the integral (3.2).

Let's start this Section by studying two particular cases which show how to use the results of Proposition 4.10, Theorem 2.3 page 72 and Remark 2.8 page 75 to define the algorithms allowing to calculate the gradient of the criterion (3.2) with respect to u and to set up the "steepest descent methods" to identify a set of parameters u_{opt} minimizing the damage (3.2).

One-dimensional examples. We study two examples which consist to optimize a one-dimensional spring under fatigue criterion. The design parameters of the spring are first its resonant frequency and its damping, next we optimize its stiffness and its mass.

EXAMPLE 4.9 In this first example we intend to find the natural frequency ω and the damping coefficient c which minimize the damage $\mathscr{D}(\omega, c)$ caused on the spring described in figure (Fig. 4.11) when it is submitted to traction-compression loading defined on a time horizon $[0, T]$ and the number of cycles to failure is given by a Stromeyer formula.

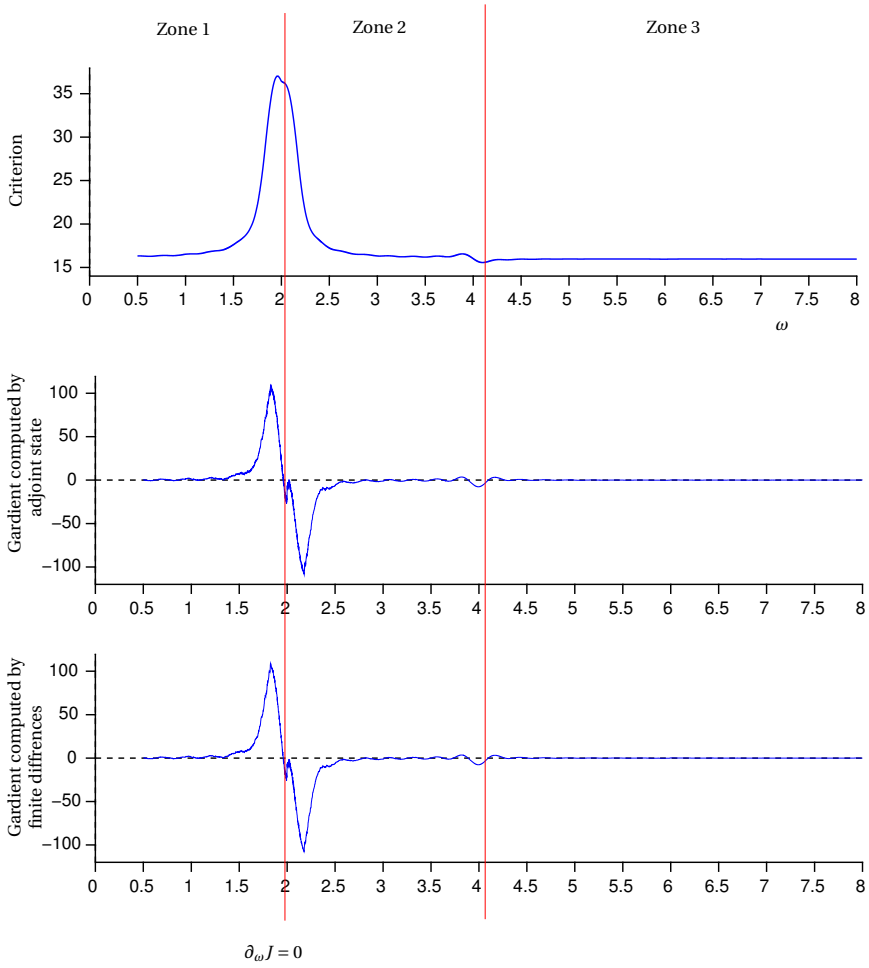


Fig. 4.9. **Comparison between the derivatives of $\omega_0 \mapsto \partial_{\omega_0} J$ obtained by integration of the adjoint equation and finite differences**. Computations are made in assuming that $\alpha_0 = 2$ and $\alpha_1 = 4$ on the time interval $[0, 8\pi]$, the adjoint state is integrated with the algorithm 4.8 and the convolution products are computed on 512 samples. Beyond the fact that this figure shows a good correlation level between the two computation methods of the gradient, it allows to verify that the criterion $\int_0^{8\pi} |\xi(t) - sin(4t)|\, dt$ is not a convex function of the parameter ω. Thus a steepest descent method initialized in the zone 1 leads to soften the structure to take away ω from the pulsation α_0 of the excitation, where the system enters into resonance. When it is initialized in the zone 2, it leads to stiffen the structure to bring ω on $\alpha_1 = 4$ which is the global optimum. At last, the criterion has a stationary point at phase reversal of the response which occurs at resonance passing of the undamped system($\omega = 2$).

i) The mechanical system is governed by the second-order differential equation

$$\begin{aligned} &\ddot{\xi} + \omega^2 \xi + c\dot{\xi} = F_0 \cos(\alpha t) \quad \text{for } t \in [0, T] \\ &\xi(0) = \dot{\xi}(0) = 0 \end{aligned} \tag{4.66}$$

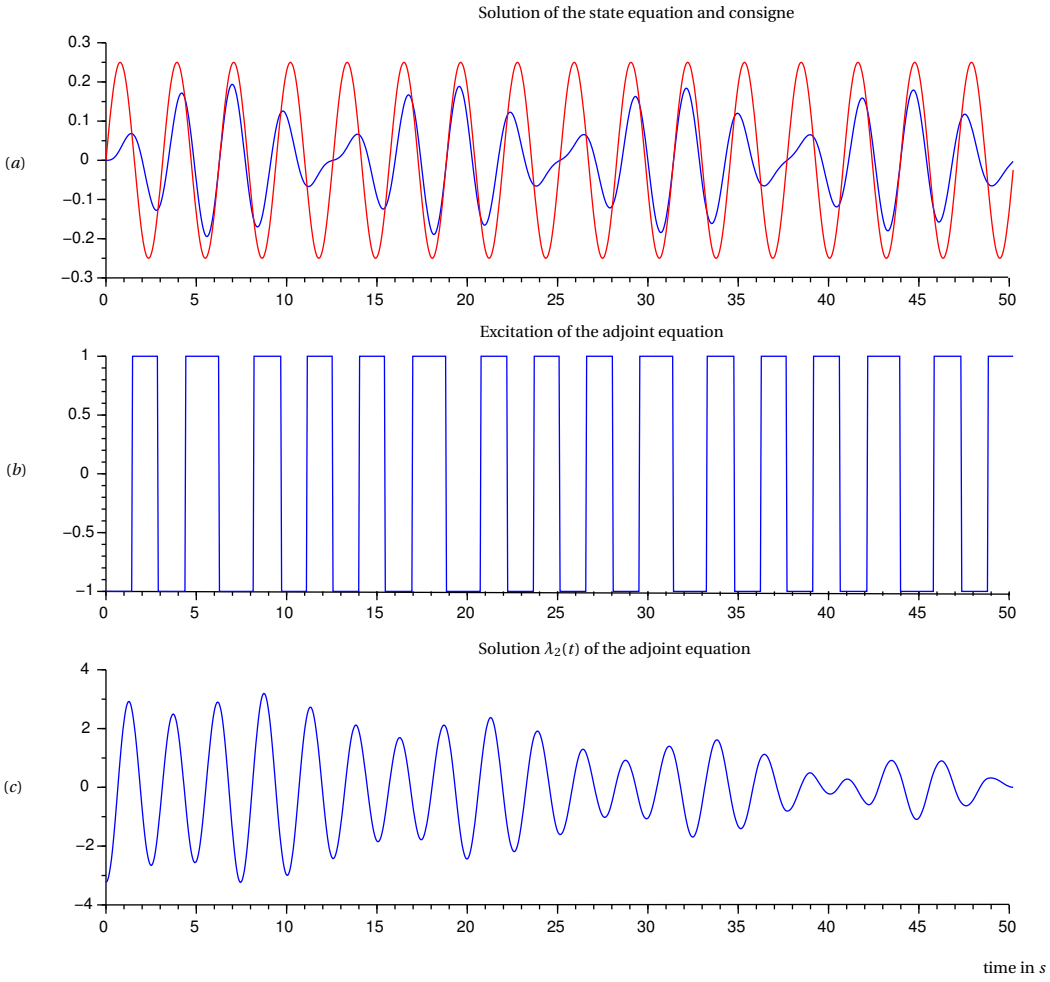


Fig. 4.10. **Details of the resolution of the adjoint equation**. Since in this case their is no bases change to do, the computations are made with a simplified version of the algorithm 4.8. Notice on the other hand that the excitation of the adjoint equation jumps a discontinuity every time the sign of $\xi_1(t) - \sin\alpha_1 t$ changes and the solution of adjoint system never reaches a stationary state. This justifies the interest of the forced response method introduced in the chapter 3.

ii) According to the formula (2.63) page 74, the damage is

$$\mathscr{D}(\omega, c) = \frac{1}{2b_s C_s} \int_0^T [\max\{(\sigma_0(T+t) - \sigma_d), 0\}]^{\frac{1}{b_s}-1} |\dot{\sigma}(t)|\, dt \tag{4.67}$$

where:

- $\sigma(t)$ is a stress defined (according to the displacement $\xi^{per}(t)$ and the geometrical characteristics of the spring) as

$$\sigma(t) = \frac{E\xi^{per}(t)}{L} \quad \text{where } E \text{ is the Young modulus of the material} \tag{4.68}$$

- $\sigma_0(t)$ is, see Theorem 2.3 page 72, the abscissa of the first extremum of the mapping $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(\sigma, t)$ defined by the variational inequality (2.39) page 61, parametrized in σ_a and integrated between 0 and $2T$;

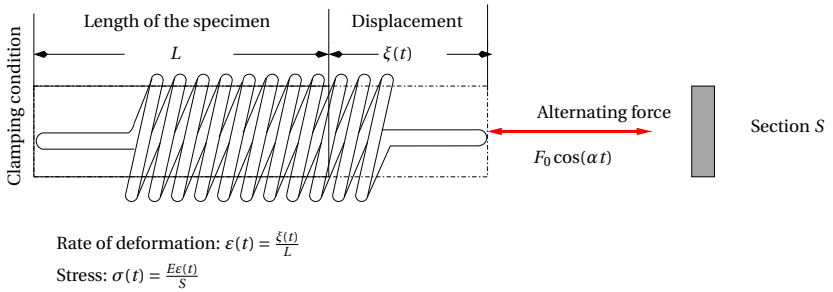


Fig. 4.11. **Mechanical parameters**. Assume that the specimen is submitted to tension-compression loaded by a force $f(t) = F_0\cos(\alpha t)$, we want define the mass-stiffness ratio $\omega = \sqrt{\frac{k}{m}}$ which minimizes the damage $\mathscr{D}(\omega)$ when the number of cycles to failure is given by a stromeyer formula (without accounting of mean stress effect).

- b_s, C_s and σ_d (fatigue limit) are material constants which characterize the Wöhler's curve used to compute the number of cycles to failure according to the applied alternating stresses.

iii) The same arguments as those developed in the framework of the Example 4.8 show that *the gradient of* $(\omega, c) \mapsto \mathscr{D}(\omega, c)$ *must be defined* (formally for the time being) *by the formulas* (4.64) *where*, using the Remarks 2.8 page 75 and the formulas given in the Example 2.3-1, *the Lagrange multipliers* λ_1 *and* λ_2 *satisfy of the adjoint equation*

$$\frac{d}{dt}\left\{\begin{array}{c}\lambda_1\\ \lambda_2\end{array}\right\} = \left[\begin{array}{cc}0 & \omega^2\\ -1 & c\end{array}\right]\left\{\begin{array}{c}\lambda_1\\ \lambda_2\end{array}\right\} - \frac{E}{L}\left\{\begin{array}{c}\frac{1-b_s}{4b_s^2C_s}\dot{\sigma}\,[\max\{(\sigma_0(T+.)-\sigma_d),0\}]^{\frac{1}{b_s}-2}\\ \frac{1}{2b_sC_s}sign(\dot{\sigma})\,[\max\{(\sigma_0(T+.)-\sigma_d),0\}]^{\frac{1}{b_s}-1}\end{array}\right\} \tag{4.69}$$

defined backward in time from T to 0. Using the change of variables introduced in the Remark 4.9, *this system can be solved with the help of the forced response method*[32].

iv) *Noticing that the discontinuities of the right hand member of the equation* (4.69) *result of the discontinuities of*

- the $sign(\dot{\sigma})$ when $\dot{\xi}(t)$ vanishes and changes its sign;
- and of the mapping $t \mapsto \sigma_0(t)$ when $\sigma_0(t)$ is in the $RMS(\sigma, t)$ sequence[33] of $t \mapsto \sigma(t)$, and that these discontinuities actually take place only if $\sigma_0 > \sigma_d$, the results explained in comments of figure (Fig. 2.11) page 56 show they impact the right hand member of equation (4.69) at a finite number of times.

we see that the theoretical condition 2^o) *of Proposition* 4.10 *can be assumed to be satisfied and that the computations defined* in (4.69) and (4.64) *lead to the gradient of* $\mathscr{D}(\omega, c)$.

It remains at last to explain the computation sequence allowing to

[32] In contrast with what happens in the case of the Example 4.8, all the terms of the convolution kernel (4.57) are now taken into account to perform the integration.

[33] See Definition 2.7 page 53 and figures (Fig. 2.16) and (Fig. 2.20).

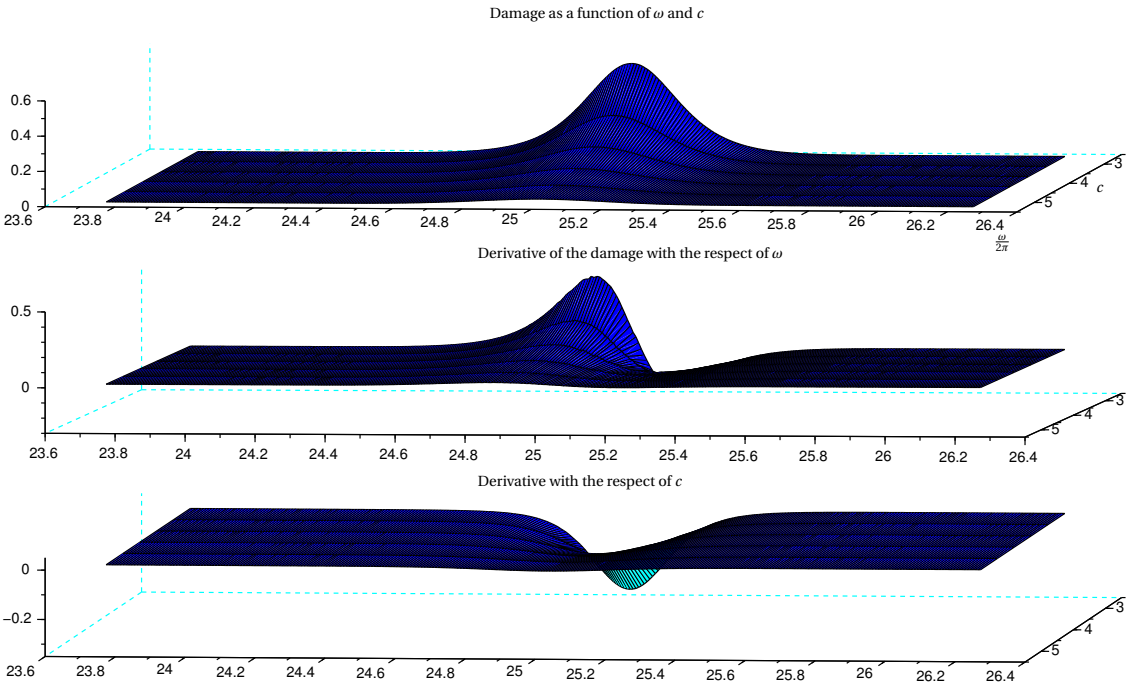


Fig. 4.12. *Computation of the damage and its partial derivatives with respect to ω and c.* The excitation is a pure sine which is set so that the state equation has a resonance at $25Hz$.

- solve the state equation (4.66) and it's adjoint (4.69),
- and to compute the integrals (4.67) and (4.64), to obtain the damage and its partial derivatives with respect to ω and c.

In essence this leads to the algorithm 4.9, whose implementation is illustrated by the example given in figure (Fig. 4.12).

REMARK 4.10 Due to the nonlinearities introduced by the formulas (2.68) and (2.69) the spectrum of the excitation (4.70) of the adjoint equation is, see figure (Fig. 4.13), richer than the one of the state equation, *this leads to over-sample the state equation to accurately compute the Lagrange multiplier λ_2* by the convolution product (4.71).

EXAMPLE 4.10 Under the conditions of Example 4.9, we look for a stiffness k and a mass m of the spring shown in figure (Fig. 4.11) which minimize the mapping $(m,k) \mapsto \mathscr{D}(m,k)$ defined in (4.67). In this case, the state equation is

$$\begin{aligned} &\ddot{\xi} + \frac{k}{m}\xi + \frac{c'}{m}\dot{\xi} = \frac{F(t)}{m} \text{ for } t \in [0,T] \\ &\xi(0) = \dot{\xi}(0) = 0 \end{aligned} \tag{4.72}$$

The mapping f of the Proposition 4.10 is then defined by

$$\left\{ \begin{array}{c} \xi_1 \\ \xi_2 \end{array} \right\} \in \mathbb{R}^2 \mapsto f(\xi_1,\xi_2) = \left\{ \begin{array}{c} \xi_2 \\ -\frac{k}{m}\xi_1 - \frac{c'}{m}\xi_2 + \frac{F(t)}{m} \end{array} \right\} \in \mathbb{R}^2$$

Algorithm 4.9: *Application of the forced response method to the simultaneous integration of state and adjoint equations for an unidimensional damage problem.*

inputs :

- Sampling of the right hand member of the state equation in the table $\left(F(t_k)\right)_{k=1}^{N_{samp}}$.
- Natural frequency ω of the processed oscillator
- Table $\left(\sigma_{a_j}\right)_{j=1}^{M_{\sigma_{amax}}}$ of the sampling points of the function $\sigma_a \mapsto \mathscr{E}_{\sigma_a}(\sigma, t_i)$ for $(1 \leq j \leq N_{samp})$.
- Material parameters of the Wöhler's curves (see the Examples 2.2 page 74).

outputs :

- Damage $\mathscr{D}(\omega, c)$
- Derivatives $\nabla\mathscr{D}(\omega, c)$ of the damage computed at ω and c.

begin

1) **Solve the state equation by the forced response method** (algorithm 4.8)
   - In case of subcritical damping, set $\delta = \frac{1}{2}\sqrt{4\omega^2 - c^2}$ and $dt = T/N_{samp}$ (step size of the sampling)
   - Sample the damping $C = \left(dte^{-ct_k/2}\right)_{k=1}^{N_{samp}}$, the convolution kernels $si = \left(\sin(\delta t_k)\right)_{k=1}^{N_{samp}}$ and $co = \left(\cos(\delta t_k)\right)_{k=1}^{N_{samp}}$
   - Carry out the discrete convolution products
     - $\frac{1}{\delta}\left(C_k si_k\right)_{k=1}^{N_{samp}} * \left(F_k\right)_{k=1}^{N_{samp}}$ to get the sampling $\xi_1 = \left(\xi_1(t_k)\right)_{k=1}^{N_{samp}}$ of the displacement $\xi(t)$.
     - $\left(C_k(co_k - si_k/(2\delta)\right)_{k=1}^{N_{samp}} * \left(F_k\right)_{k=1}^{N_{samp}}$ for the sampling $\xi_2 = \left(\xi_2(t_k)\right)_{k=1}^{N_{samp}}$ of the velocity $\dot{\xi}(t)$.
2) **Use** $\left(\xi_1(t_k)\right)_{k=1}^{N_{samp}}$ and $\left(\xi_2(t_k)\right)_{k=1}^{N_{samp}}$ **to sample the stresses** $\sigma = \left(\sigma(\xi(t_k))\right)_{k=1}^{N_{samp}}$ **and their time derivatives** $\sigma_{dot} = \left(\dot{\sigma}(\xi(t_k))\right)_{k=1}^{N_{samp}}$; in the case of the Example 4.9, simply carry out the product of the table ξ_i $(i = 1,2)$ by $\frac{E}{L}$ but *mores complicated cases are dealt subsequently.*
3) **Use the algorithm** 3.3 with the data σ and σ_{dot} **to make the sampling** $\sigma_0 = \left(\sigma_0(t_k)\right)_{k=1}^{N_{samp}}$ of the mapping $t \mapsto \sigma_0(T + t)$.
4) Use the datat σ, σ_0 and σ_{dot} to sample
   - the integrand
   $$W = \left(w(\sigma(t_k), \sigma_0(t_k), \dot{\sigma}(t_k))|\dot{\sigma}(t_k)|\right)_{k=1}^{N_{samp}}$$
   of the damage defined the formula (2.62) page 72 **and compute** $\mathscr{D}(\omega, c)$ **by numerical integration**. In the case of the Example 4.9, this reduces to the computation of $\frac{1}{2C_s b_s}\left[\max\{(\sigma_0(t_k) - \sigma_d), 0\}\right]^{\frac{1}{b_s}-1}\left|\dot{\Sigma}_e(t_k)\right|$.
   - the derivatives (2.68) and (2.69) defined in the Remark 2.8 page 75, to make **the following tables**, which **contain the sampling of the right hand member of the adjoint equation**.

$$D_1 J = \left(\frac{\partial j}{\partial v}(\sigma(t_k), \sigma_0(t_k), \sigma_{dot}(t_k))\right)_{k=1}^{N_{samp}} \text{ and } D_2 J = \left(\frac{\partial j}{\partial \dot{v}}(\sigma(t_k), \sigma_0(t_k), \sigma_{dot}(t_k))\right)_{k=1}^{N_{samp}} \tag{4.70}$$

   In the case of the Example 4.9, this reduces to the computation of

$$\frac{1 - b_s}{4b_s^2 C_s}\dot{\sigma}(t_k)\left[\max\{(\sigma_0(t_k) - \sigma_d), 0\}\right]^{\frac{1}{b_s}-2} \text{ and } \frac{1}{2b_s C_s} sign(\sigma_{dot}(t_k)\left[\max\{(\sigma_0(t_k - \sigma_d), 0\}\right]^{\frac{1}{b_s}-1}$$

5) **Solve the adjoint state equation** (simplified version of the algorithm 4.8)
   - Make the change of variable of the Remark 4.9 on the tables $D_1 J$ and $D_2 J$ to define the tables $\widetilde{D_i J}$ $(i = 1,2)$.
   - Sample the Lagrange multipliers $\widehat{\lambda_2}$ with the help of the convolution product

$$\widetilde{\lambda_2} = \left(C_k si_k/\delta\right)_{k=1}^{N_{samp}} * \widetilde{D_1 J} + \left(C_k(co_k - csi_k/(2\delta))\right)_{k=1}^{N_{samp}} * \widetilde{D_2 J} \tag{4.71}$$

   - Apply the reciprocal change of variables to return to $\lambda_2(t_k)$, sampled in the increasing time.
6) **Compute the partial derivatives** $\partial_\omega \mathscr{D}$ and $\partial_c \mathscr{D}$ **by carrying out numerical the integration** on the tables
$$-\omega\left(\lambda_2(t_k)\xi_1(t_k)\right)_{k=1}^{N_{samp}} \quad \text{and} \quad -\left(\lambda_2(t_k)\xi_2(t_k)\right)_{k=1}^{N_{samp}}$$

end

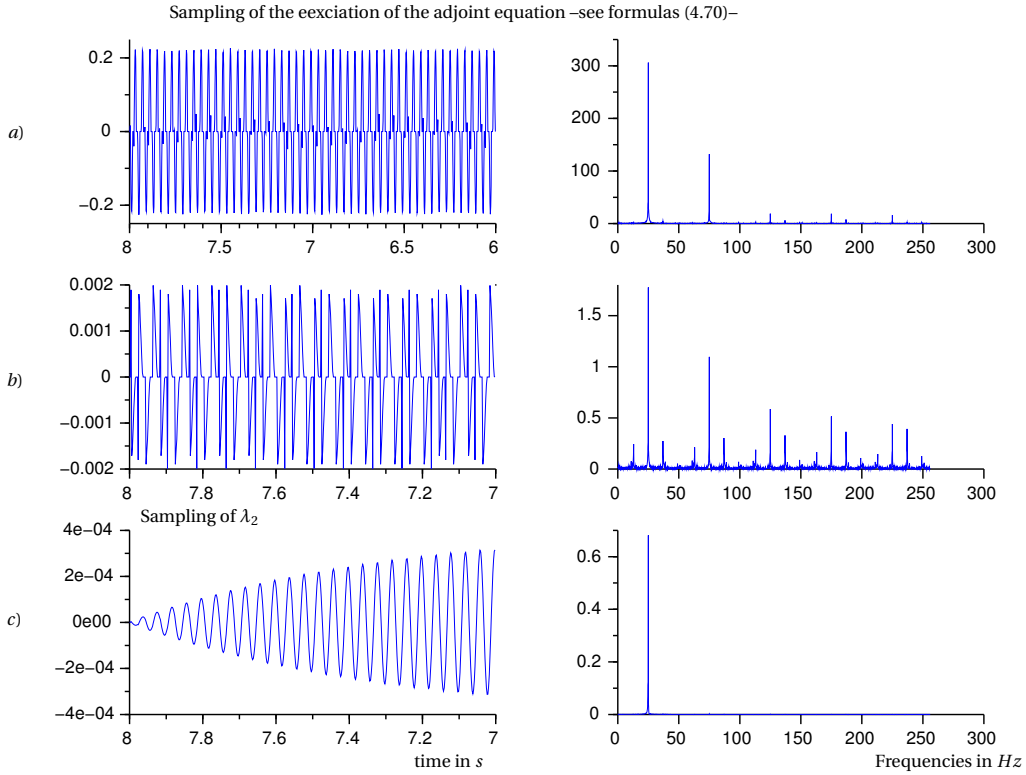


Fig. 4.13. **Detail of the computations for the resolution of the adjoint equation in the case of the Example 4.9**. We see that the spectrum of the excitation of the adjoint equation is richer than that of the state equation, which contains only one frequency.

and the derivative $D_{(k,m)}f$ at a point (k_0, m_0) is

$$D_{(k,m)}f = \begin{bmatrix} 0 & 0 \\ -\frac{1}{m_0}\xi_1 & \frac{k_0}{m_0^2}\xi_1 + \frac{c'}{m_0^2}\xi_2 - \frac{F(t)}{m_0^2} \end{bmatrix}$$

this shows that the gradient of $(k, m) \mapsto \mathscr{D}(k, m)$ is given by

$$\begin{aligned} \partial_k \mathscr{D}(k_0, m_0) &= -\frac{1}{m_0}\int_0^T \xi_1(t)\lambda_2(t)\,dt \\ \partial_m \mathscr{D}(k_0, m_0) &= \frac{1}{m_0^2}\int_0^T \left[k_0\xi_1(t) + c'\xi_2(t) - F(t)\right]\lambda_2(t)\,dt \end{aligned} \tag{4.73}$$

where (λ_1, λ_2) are the solutions of an adjoint system analogous to (4.69) (in which it suffices to set $\omega = \sqrt{\frac{k}{m}}$ et $c = \frac{c'}{m}$) note, on the other hand, that $\partial_m \mathscr{D}(k_0, m_0)$ can be simplified as

$$\partial_m \mathscr{D}(k_0, m_0) = \frac{1}{m_0}\int_0^T \dot{\xi}_2(t)\lambda_2(t)\,dt$$

A numerical application is proposed in the figures (Fig. 4.15), (Fig. 4.16) and (Fig. 4.17), when the mission profile $t \mapsto F(t)$ is the signal shown in the figure (Fig. 4.14). This numerical application shows that

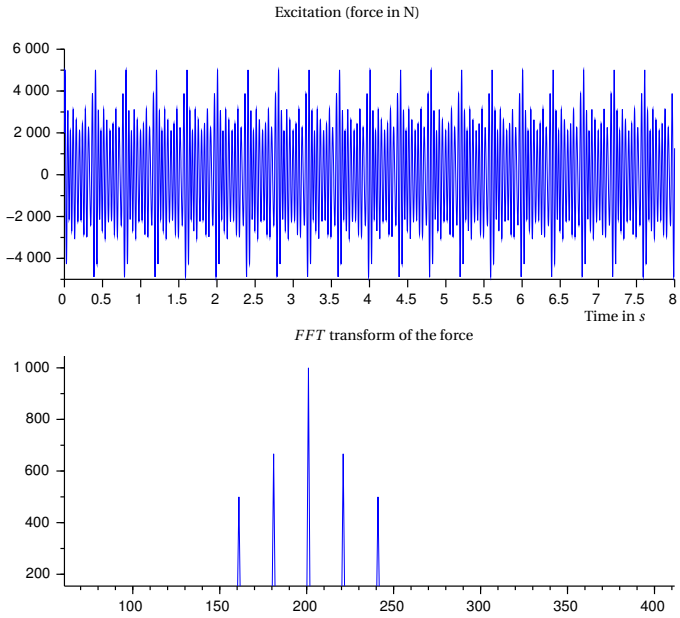


Fig. 4.14. **Excitation force in case of the Example 4.10**. In this case is the force is $F(t) = f_0\cos(\omega_0 t) + \frac{2}{3}f_0(\sin(1.1\omega_0 t) + \sin(0.9\omega_0 t)) + \frac{1}{2}f_0(\sin(1.2\omega_0 t) + \sin(0.8\omega_0 t))$, where ω_0 and f_0 are given constants; they are sampled on 8 seconds.

- in this case, the damage is not a convex function of the parameters m and k;
- and the global optimum would be reached for an "ideal" structure which would be at the same time very stiff and lightweight, however, local optima can be found *but they depend upon the mission profile*!

REMARK 4.11 If the spring is modeled as a tensile-compression bar parametrized by the length L and the section S, the stiffness and the mass are $k = \frac{ES}{L}$ et $m = \rho SL$. The state equation (4.72) can be written as

$$\ddot{\xi} + \frac{E}{\rho L^2}\xi + \frac{c'}{\rho SL}\dot{\xi} = \frac{F(t)}{\rho SL} \tag{4.74}$$

Formulas (4.67) and (4.68) show that damage caused by the loading $F(t)$ defined on the time horizon $[0, T]$ is

$$\mathscr{D}(S,L) = \frac{1}{4b_sC_s}\left(\frac{E}{L}\right)^{\frac{1}{b_s}}\int_0^T \left[\max\{(\sigma_0(t+T) - \sigma'_d(L)), 0\}\right]^{\frac{1}{b_s}-1}\left|\dot{\xi}(t)\right| dt$$

where $\sigma'_d(L)$ is defined on the basis of the fatigue limit σ_d by

$$\sigma'_d(L) = \frac{\sigma_d L}{E}$$

and explicitly depends upon the design parameter L. The term $\partial_L\mathscr{D}$ obtained form (4.73) by *the formula* $\partial_L\mathscr{D} = \rho S\partial_m\mathscr{D} - \frac{ES}{L^2}\partial_k\mathscr{D}$ *must be completed according to* the Remark 4.8-1 page 172.

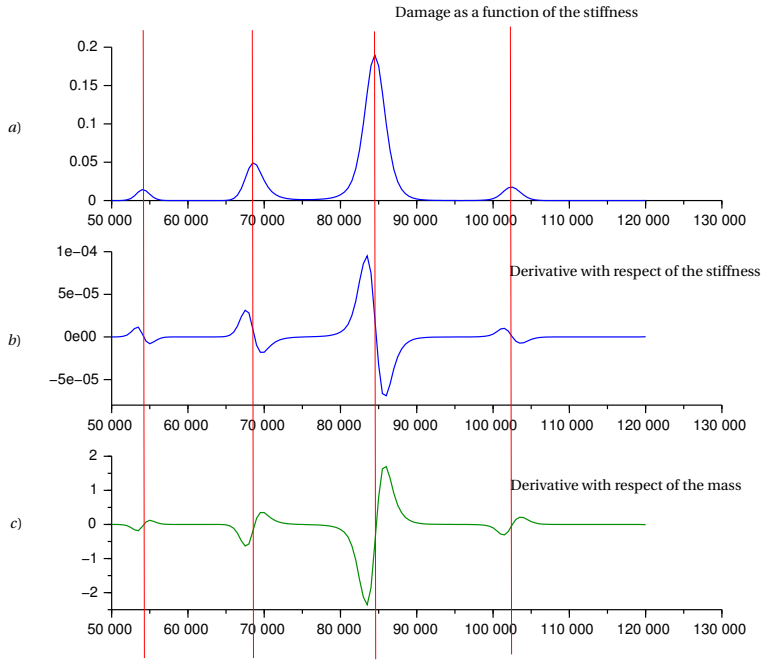


Fig. 4.15. **Damage and its derivatives computed for a given mass**. In this case, we fix the mass at a given value m_0 and we plot on the figures *a*), *b*) and *c*) the mappings $k \mapsto \mathscr{D}(k, m_0)$, $\partial_k D(k, m_0)$ and $\partial_m D(k, m_0)$.

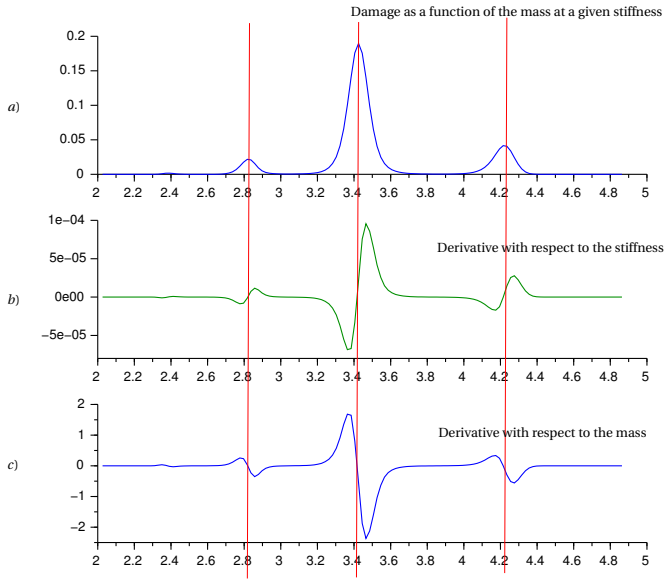


Fig. 4.16. **Damage and its derivatives computed for a given stiffness**. In this case, we fix the stiffness at a given value k_0 and we plot on the figures *a*), *b*) and *c*) the mappings $m \mapsto \mathscr{D}(k_0, m)$, $\partial_k D(k_0, m)$ and $\partial_m D(k_0, m)$.

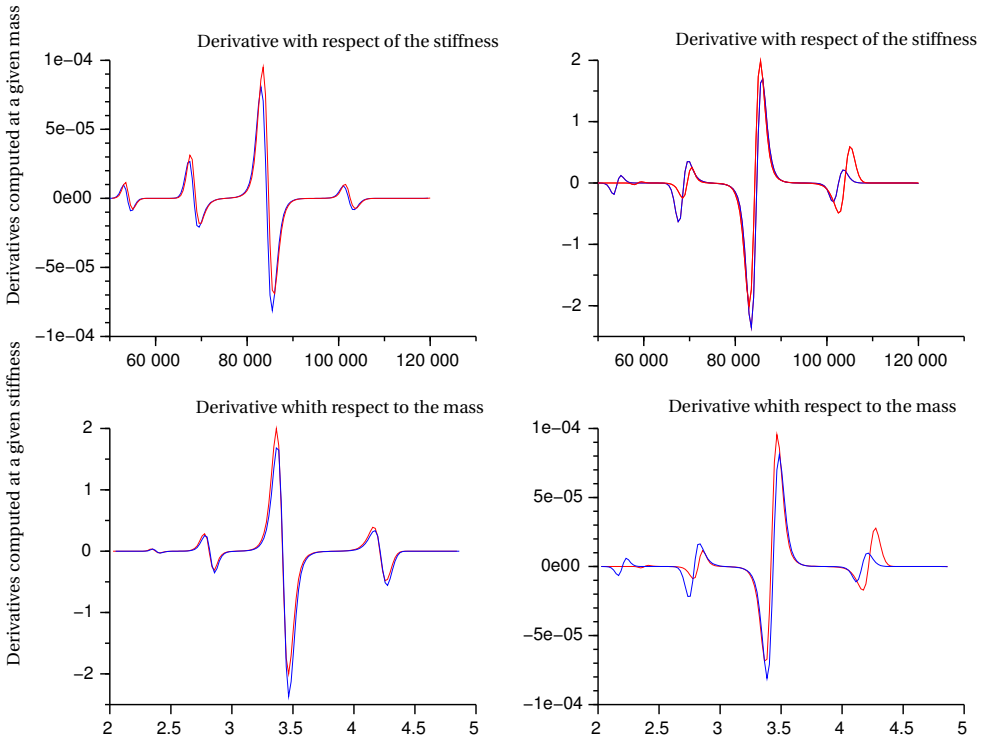


Fig. 4.17. **Comparison between the gradients obtained by integration of the adjoint equation and finite differences**. The curves in red correspond to the computations by finite differences and the curves in blue are the results obtained by resolution of the adjoint equation.

Multidimensional case. In the general case, a structural optimization problem under fatigue life criterion can be written as

Find a set of parameters $u_{opt} \in U_{ad}$ which minimize the damage

$$u \in U_{ad} \mapsto \mathscr{D}(u) = \int_0^T w\left(\Sigma_e(t), \Sigma_0(t+T), \dot{\Sigma}_e(t)\right) \left|\dot{\Sigma}_e(t)\right| dt \in \mathbb{R}^+ \tag{4.75.a}$$

where $\Sigma_e(t) := \Sigma_e(x_u(t))$ is an "equivalent stress" depending on the first component of the variable $X_u := (x_u, \dot{x}_u)$[34], itself solution of the equation

$$\frac{dX_u}{dt} = f(X_u, u, t) \text{ on } [0,T] \quad \text{with the initial condition } X_u(0) = 0 \tag{4.75.b}$$

where:

- see Theorem 2.3 page 72, the mapping $(v_1, v_2, v_3) \mapsto w(v_1, v_2, v_3)$ is defined (as a function of the inverse of the number cycles to failure given in the Wöhler abacuses) by the formula (2.62);
- the system (4.75.b) is obtained in reducing the second order system $[M_u]\,\ddot{x} + [K_u]\,x + [W_u]\,\dot{x} = F(t)$ to a first order one by the change of variables introduced in Section 3.1 page 104.

[34] Here x and $\dot{x}$ are considered as independent variables.

If we denote by j the mapping

$$X = (x, \dot{x}) \mapsto j(X) = w\left(\Sigma_e(x), \Sigma_0, \dot{\Sigma}_e\right)\left|\dot{\Sigma}_e\right| \quad \text{where } \dot{\Sigma}_e = (\nabla\Sigma_e(x)\,;\,\dot{x})$$

Proposition 4.10 shows that the i-th component of the gradient (with the respect of the parameters u) of the damage, computed at $u_0 \in \Phi_{ad}$ is given by

$$\begin{aligned}\nabla\mathscr{D}(u_0)_i = \int_0^T \Big(&\partial_{u_i}\left[M_{u_0}\right]^{-1} F - \partial_{u_i}\left(\left[M_{u_0}\right]^{-1}\left[K_{u_0}\right]\right) x \\ &-\partial_{u_i}\left(\left[M_{u_0}\right]^{-1}\left[W_{u_0}\right]\right)\dot{x};\Lambda_2\Big)\,dt\end{aligned} \tag{4.76}$$

where:

- u_i denote generically the i-th component $u \in U_{ad}$;
- the Lagrange multipliers Λ_1 and Λ_2 are solutions of the adjoint system

$$\frac{d}{dt}\left\{\begin{array}{c}\Lambda_1\\ \Lambda_2\end{array}\right\} = \left[\begin{array}{cc}[0] & [K]\,[M]^{-1}\\ -[Id] & [W]\,[M]^{-1}\end{array}\right]\left\{\begin{array}{c}\Lambda_1\\ \Lambda_2\end{array}\right\} - \left\{\begin{array}{c}[M]^{-1}\nabla_x j(\Sigma_e,\dot{\Sigma}_e)\\ [M]^{-1}\nabla_{\dot{x}} j(\Sigma_e,\dot{\Sigma}_e)\end{array}\right\} \tag{4.77}$$

  integrated backward in time from T to 0, with the ending condition $\Lambda_i(T) = 0$.

Remark 4.8-4 shows that this system may be diagonalized under the form (4.55) by the change of basese $\Lambda_i = [M]^{\frac{1}{2}}[\hat{Q}]\hat{\lambda}_i$ and not by $[M]^{-\frac{1}{2}}[\hat{Q}]$, which is the change of bases which diagonalizes the state equation.

Using the notations of Remarks 2.8 page 75, the right hand member of the adjoint equation (4.77), which is formally written as

$$\begin{aligned}\nabla_x j =&\nabla\Sigma_e(x)\partial_v j(\Sigma_e(x),\Sigma_0,\dot{\Sigma}_e)\\ &+\left(D^2\Sigma_e(x).\dot{x}\right)\partial_{\dot{v}} j(\Sigma_e(x),\Sigma_0,\dot{\Sigma}_e)\\ \nabla_{\dot{x}} j =& \nabla\Sigma_e(x)\partial_{\dot{v}} j(\Sigma_e(x),\Sigma_0,\dot{\Sigma}_e)\end{aligned} \tag{4.78}$$

can be made explicit by the formulas (2.68) and (2.69) of the Remark 2.8-2) page 75 in which we have set $v = \Sigma_e$ and $\dot{v} = \dot{\Sigma}_e = (\nabla\Sigma_e\,;\,\dot{x})$.

REMARK 4.12 The formula (4.76) requires the computation of the partial derivatives

$$\partial_{u_i}[M_u]^{-1},\ \partial_{u_i}\left([M_u]^{-1}[K_u]\right) \text{ etc.}$$

with the respect to those of the matrices $[M_u]$, $[K_u]$, $\cdots$, which are assumed to be defined elsewhere.

- As, except to the simplifications due to commutativity, the differentiation of a product of matrices follows the same rules as the differentiation of an ordinary product and we have

$$\partial_{u_i}\left([M_{u_0}]^{-1}[K_{u_0}]\right) = \left(\partial_{u_i}[M_{u_0}]^{-1}\right)[K_{u_0}] + [M_{u_0}]^{-1}\left(\partial_{u_i}[K_{u_0}]\right) \tag{4.79}$$

- Then, differentiating the equation $[M_{u_0}^{-1}][M_{u_0}] = [Id]$ we obtain

$$\partial_{u_i}[M_{u_0}]^{-1} = -[M_{u_0}]^{-1}[\partial_{u_i} M_u][M_{u_0}]^{-1} \tag{4.80}$$

- Computation of the derivative of $[M_{u_0}]^{-1}[W_{u_0}]$ can be a more complicated exercise:
  - when the damping is proportional to the mass matrix, the matrix $[M_u]^{-1}[W_u]$ doesn't depend on u and its derivative is 0;
  - if the damping is proportional to the stiffness matrix, the job is already done;
  - but *if the damping is a fraction of the critical damping per mode formula* (3.28) *page 114 shows that we have to differentiate the square root of a matrix to do the computation*; and this leads to solve the Lyapounov equation

$$[Q][P]^{\frac{1}{2}} + [P]^{\frac{1}{2}}[Q] = \partial_{u_i}[P] \tag{4.81}$$

where the unknown matrix $[Q]$ is the wished derivative $\partial_{u_i}[P]^{\frac{1}{2}}$. Since the condition $[P]$ "definite positive" is a necessary and sufficient condition for the existence of a positive definite symmetric solution for the equation (4.81), the *damping matrix is differentiable only if the stiffness matrix is invertible*.

At this stage of the presentation, all arguments are in place to formalize in the following subsection the sequence of computations needed to solve the state and the adjoint equations of a structural optimization problem.

Algorithmic implementation. All the methods allowing to compute (with the help of the Proposition 4.10 page 165) the gradient of the damage $\mathscr{D}$ caused on a structure are summarized in the algorithm 4.8, in which it remains to modify, as explained in algorithm 4.10, the steps 5) and 6) to be able to *compute the derivatives $\partial_{u_i}\mathscr{D}$ without having to store the sampling of the solution of the adjoint equation*.

REMARKS 4.13 1) The most expensive step of algorithm 4.10 is that of damage computation, because it requires the sampling of the mapping $t \in [0,2T] \mapsto \Sigma_0(t)$ and, in fine, the computation (at each instant of the time sampling) of the graph of the mapping

$$\sigma_a \mapsto \mathscr{E}_{\sigma_a}(\Sigma_e(t), t)$$

defined page 61. Thus, *once the damage calculation has been carried out, the algorithm* 4.10 provides, for a few supplementary matrix manipulations, *the sensitivity analysis of the said damage to the design parameters of the structure*.

2) Assume that damping is a fraction of the critical damping per mode allows to *make the optimization process damping independent*; thus as unpleasant that it is, the computation of the derivative of $M_{u_0}^{-1}W_{u_0}$ is necessary in order that the optimization leads to modify design parameters such as mass or stiffness, and not, by inadvertence, *the damping, which is not a relevant design parameter in structural mechanics*.

3) Some structure sofware make use of *"mass-lumping" technique* to diagonalize the mass matrix. If this technique allows to simplify the integration of the state and adjoint equations, it leads to a mass matrix which is "not differentiable" with respect to the design parameters and *must be surrendered in the framework of structural optimization*.

Algorithm 4.10: *Complements of algorithm 4.8 to compute the partial derivatives of the damage.*

inputs :

- Derivatives $\partial_{u_i}[K_u]$, $\partial_{u_i}[M_u]$, $\partial_{u_i}[W_u]$ of the stiffness, mass and damping matrices with respect of the design parameters.

outputs :

- Value $\mathscr{D}(u_0)$ of the damage
- Gradient $\nabla\mathscr{D}(u_0)$ of the damage computed at u_0.

begin

1) **Use the steps** 1) **to** 4) **of the algorithm** 4.8 to compute $\left(\overline{\Lambda}_i\right)_{i=1}^{N_{samp}}$
2) **Compute the derivatives of** $[M_u]^{-1}F$, $[M_u]^{-1}[K_u]$ and $[M_u]^{-1}[W_u]$ **in function of the derivatives** $\partial_{u_i}[M_u]$ etc.
3) **Sample the integrand**

$$\left(\partial_{u_k}[M_u]^{-1}F(t) - \partial_{u_k}\left([M_u]^{-1}[K_u]\right)x(t) - \partial_{u_k}\left([M_u]^{-1}[W_u]\right)\dot{x}(t)\,;\,\Lambda(t)\right)$$

and perform the numerical integration to calculate the derivative $\partial_{u_k}\mathscr{D}$
for $i = 1$ **to** N_{samp} **do**

- Compute the product $\partial_{u_k}[M_u]^{-1}F_i$
- Pass modal displacements and velocities in the initial basis **by the matrix product** (4.60)
- Pass the Lagrange multipliers $\overline{\Lambda}_{N_ech-i+1}$ in the initial basis **by the change of bases** (4.62) to obtain the component Λ_i
- Compute the scalar product

$$\left(\partial_{u_k}[M_u]^{-1}F_i - \partial_{u_k}\left([M_u]^{-1}[K_u]\right)x_i - \partial_{u_k}\left([M_u]^{-1}[W_u]\right)\dot{x}_i\,;\,\Lambda_i\right)$$

and update the numerical integration to compute $\partial_{u_k}\mathscr{D}$.

end

end

Application to shape optimization of a torsional beam. At last, we will use the algorithm 4.10 to calculate the derivative of the damage with respect to the additional inertias of the beam studied in Section 3.2. This example has been encoded in Scilab language and the source code is reproduced in Annex B.2 page 216. The torques used to calculate the damage and its derivative with respect of the total inertia I_{ad} are shown in figure (Fig. 4.16), they simultaneously excite the four main eigen-modes of the beam. The damage and its derivative are computed at points B and C of the beam, see figure (Fig. 3.13) page 120. The derivative of the damage is calculated by integration of the adjoint sate equation and compared with a finite difference computation; the results are plotted in figure (Fig. 4.19).

We can assume that the damage of the beam at point C mainly depends on the excitation at $35\,Hz$; this hypothesis could be confirmed in calculating the sensitivity of the damage with respect of the diameter of the notch and this could be carried out with the help of the algorithm 4.10.

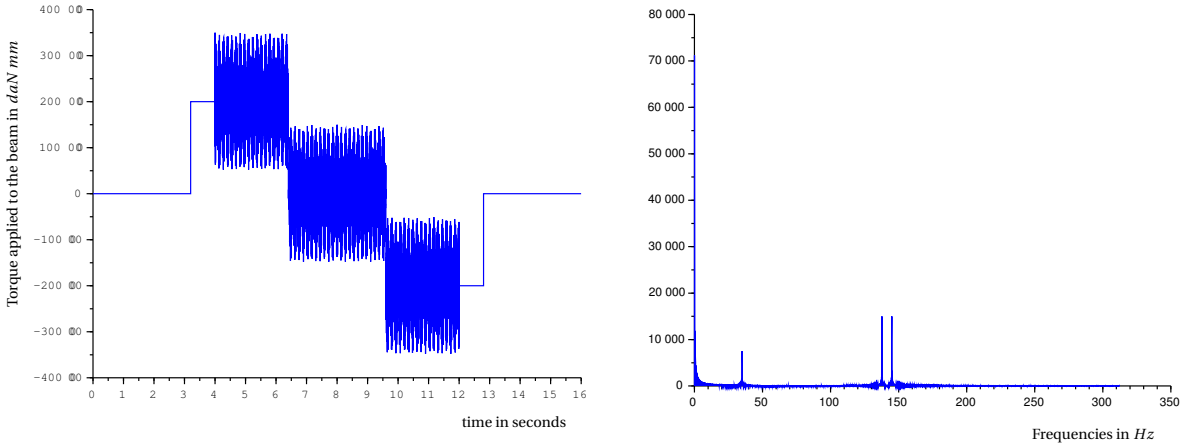


Fig. 4.18. **Torques applied to the ends A and E of the beam**. They contain the frequencies 35, 137 and 145 which permit to excite the first eigen-modes of the beam, depicted in figure (Fig. 3.8) page 111.

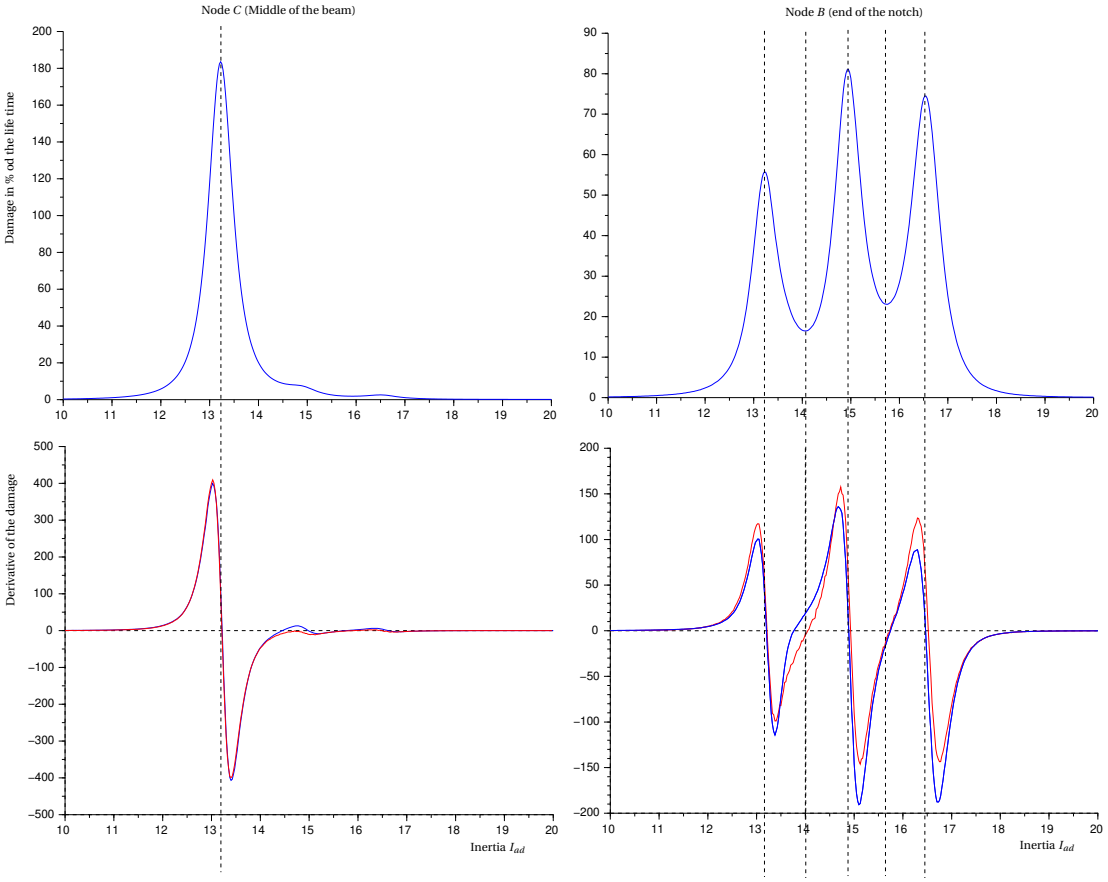


Fig. 4.19. **Comparison between the gradients obtained by integration of the adjoint equation and finite differences**. The curves in red correspond to the computations by finite differences while the curves in blue are the results obtained in solving the adjoint equation.

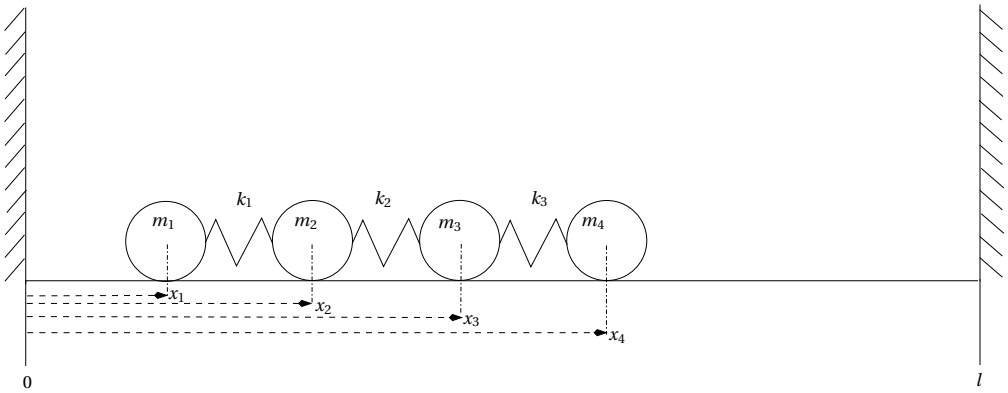


Fig. 4.20. **Contact problem between sliding masses**. We are considering a mechanical system made up of 4 point masses m_i (connected together by springs k_i of lengths l_i) constrained to slide along the horizontal axis between two vertical walls. Denoting by x_i the position of the point mass m_i, the objective is to write down an algorithm for the integration of the motion equations of this mechanical system which takes care of the non-penetration conditions $0 \le x_1 \le x_2 \le \cdots \le x_4 \le l$.

4.4. Exercises and complements

EXERCICE 4.1 Solve the variational inequality (4.5) page 143 when J is the mapping

$$u \in [u_1, u_2] \mapsto u^2 \quad \text{or} \quad u^3$$

EXERCICE 4.2 Proof that a differentiable numerical mapping J defined on a normed space E is convex if and only if

$$J(v) - J(u) \ge J'(u)(v-u) \quad \text{for all } u, v \in E \tag{4.82}$$

and give a geometrical interpretation of this inequality.

EXERCICE 4.3 Maximize the functional $(x_1, x_2) \mapsto 4x_1 + 3x_2$ under the constraints $x_1, x_1 \ge 0$ and $2x_1 + x_2 \le 10$.

EXERCICE 4.4 Use the Proposition 4.4 page 144 to proof an existence result for the bending Timoschenko equations of beams; this is a helpful exercise to do the homework proposed page 130.

EXERCICE 4.5 Use the implicit Euler method and the results obtained in Example 4.6 page 160 to write down an algorithm for the numerical integration of the motion equations of the mechanical system depicted in figure (Fig. 4.20). Give a mechanical interpretation of the Lagrange multipliers associated with the optimization problem (under inequality constraints) solved at each time step of the Euler algorithm. Perform numerical simulations (with different stiffnesses and masses) at a given initial velocity and mechanically explain the obtained restitution coefficients.

EXERCICE 4.6 Compute the derivatives of the mapping

$$(a, b) \mapsto J(a, b) := \int_0^T x(t)\, dt$$

controlled by the differential equation $\dot{x} = bx$ satisfying the initial condition $x(0) = a$.

Solutions & homeworks.

Solution of exercise 4.1. In case of mapping $u \mapsto J(u) = u^2$ we have to find u_* in the interval $[u_1, u_2]$ such that

$$u_*(u - u_*) \geq 0 \quad \forall u \in [u_1, u_2] \tag{4.83}$$

This means that the condition $u_* \neq 0$ entails $u_* = u_1$ or $u_* = u_2$. More precisely if $u_1 > 0$ we must have $u_* = u_1$, while $u_* = u_2$ if $u_2 < 0$. If $u_1 < 0 < u_2$, the solution u_* of (4.83) can't be u_1 nor u_2 and we necessarily have $u_* = 0$.

We have to solve the variational inequality

$$u_*^2(u - u_*) \geq 0 \quad \forall u \in [u_1, u_2] \tag{4.84}$$

for the mapping $u \mapsto J(u) = u^3$. In this case, it is immediate to see that $u_* = u_1$ is always solution (4.84) but if $u_1 < 0 < u_2$ this inequality has also the solution $u_* = 0$ which is not a minimizer for J.

Solution of exercise 4.2. If J is assumed to be convex we have

$$J(u + \lambda(v - u)) - J(u) \leq \lambda(J(v) - J(u))$$

for any $0 \leq \lambda \leq 1$. Dividing this inequality by $\lambda \neq 0$ and taking the limit of the obtained result when λ goes to 0 we get

$$J'(u).(v - u) = \lim_{\lambda \to 0^+} \frac{J(u + \lambda(v - u)) - J(u)}{\lambda} \leq J(v) - J(u)$$

Conversely, assume that J satisfies (4.82) then we have

$$J(v) \geq J(v + \lambda(u - v)) - \lambda J'(v + \lambda(u - v)).(u - v)$$

$$J(u) \geq J(v + \lambda(u - v)) + (1 - \lambda)J'(v + \lambda(u - v))(u - v)$$

for any real number λ. Assuming that $0 \leq \lambda \leq 1$, we can multiply the first inequality by $(1 - \lambda)$, the second one by λ (which are positive numbers) and add the obtained results to get

$$(1 - \lambda)J(v) + \lambda J(u) \geq J(v + \lambda(u - v))$$

which is the convexity inequality for the mapping J. The geometric interpretation of (4.82) is given in figure (Fig. 4.21).

Homework.

1^o/ Proof that a twice differentiable numerical mapping J defined on a normed space E is convex if an only if

$$J''(u)(v - u, v - u) \geq 0 \quad \forall u, v \in E \tag{4.85}$$

2^o/ Proof that J is strictly convex if and only if the inequalities (4.82) or (4.85) are strict when $u \neq v$.

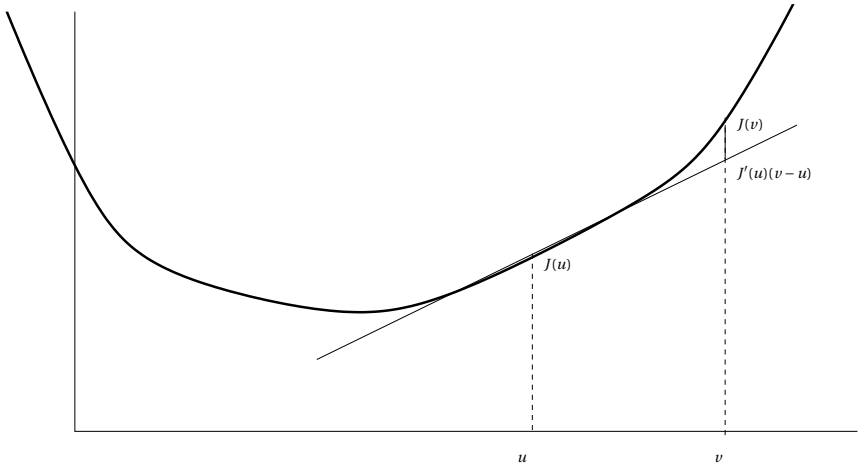


Fig. 4.21. **Geometric interpretation of the inequality** (4.82). The function is located above it's tangent plane.

Solution of exercise 4.3. Let's Introduce the functional $J(x_1,x_2) := -4x_1 - 3x_2$. The question can be reformulated as finding $(x_1^*, x_2^*) \in \mathbb{R}^2$ such that:

$$J(x_1^*, x_2^*) = \min_{\substack{-x_1 \le 0 \\ -x_2 \le 0 \\ 2x_1 + x_2 - 10 \le 0}} j(x_1,x_2) \tag{4.86}$$

Using Proposition 4.8 page 156, we see that if (x_1^*, x_2^*) is solution of (4.86) there are 3 positive constants λ_1, λ_2 and λ_3 satisfying the equations

$$\begin{aligned} -4 - \lambda_1 + 2\lambda_3 &= 0 \\ -3 - \lambda_2 + \lambda_3 &= 0 \end{aligned} \tag{4.87}$$

with the complementary conditions

$$\lambda_1 x_1^* = \lambda_2 x_2^* = \lambda_3 \left(2x_1^* + x_2^* - 10\right) = 0 \tag{4.88}$$

As the conditions $x_1^* > 0$ and $x_2^* > 0$ entail $\lambda_1 = \lambda_2 = 0$, they contradict (4.87) and we must have $x_1^* = 0$ or $x_2^* = 0$.

1^o/ The condition $x_2^* \neq 0$ entails $\lambda_2 = 0$ (by the second equation of (4.88)) and $\lambda_3 = 3$ (by the second equation of (4.87)). Using the third equation of (4.88) we conclude that $x_2^* = 10$.

2^o/ We can see in the same manner that the condition $x_1^* \neq 0$ entails $x_1^* = 5$.

In this particular case, the KKT conditions (4.24) provide the candidates $(0,10)$ and $(5,0)$ for the resolution of the minimization problem (4.87). As $J(0,10) < J(5,0)$, we must choose $(x_1^*, x_2^*) = (0,10)$.

This exercise was intending to remind you that Kuhn-Tucker conditions are necessary (but not sufficient) conditions for a point to be a minimizer.

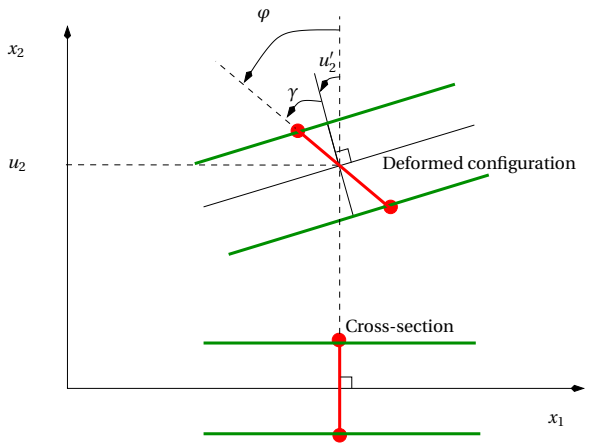


Fig. 4.22. **Mechanical meaning of the Timoschenko's parameters**. In their deformed configuration, the cross-sections of the beam remain plane but they are not necessarily orthogonal to the deformed configuration of the neutral line so that the angle γ measure a shear strain due to the normal loading.

Solution of exercise 4.4. Let's consider an isotropic rectilinear beam of length L and define a Cartesian frame (x_1, x_2, x_3) such that x_1 becomes the axis of the beam, whereas x_2, x_3 are assumed to be principal axes of the cross-sections. In the Timoschenko beam theory we assume, see figure (Fig. 4.22), that the displacements are written as

$$u_1 = -x_2\varphi(x_1), \quad u_2 := u(x_1) \quad \text{and } u_3 = 0$$

where φ is the rotation of the cross-section about the axis x_3. As the rotation φ can be different from the rotation u' of the neutral axis, *the difference*

$$\gamma = u' - \varphi$$

defines an additional rotation due to the shear deformation. The strain energy of the beam is defined by[35]

$$E_S(u,\varphi) := \frac{1}{2}\int_0^L EI\left(\varphi'\right)^2 + \kappa GA\left(u'-\varphi\right)^2 dx_1 \tag{4.89}$$

where

- A is the cross-sectional area,
- I is the cross-sectional second moment of area about x_3 axis,
- κ is a shear correction factor which depends on cross-sectional shape,
- and $G = \frac{E}{2(1+\nu)}$ is the shear modulus of the material.

Assuming that the beam is submitted to

$1^o/$ the distributed forces $f(x_1)$ (along the x_2 axis) and torques $C(x_1)$ (about the x_3 axis)
$2^o/$ and to the punctual forces f_0 (resp. f_1) and torques C_0 (resp. C_1) at the ends

[35] To simplify the notations, we drop the dependance in x_1 of the functions $f, C, u, v, \varphi, \psi$ etc. when they are written under the integral sign.

we can define the virtual work produced by this loading system as the following linear form

$$V(u,\varphi) := \int_0^L f u + C\varphi \, dx_1 + f_0 u(0) + f_1 u(L) + C_0\varphi(0) + C_1\varphi_1(L) \tag{4.90}$$

Applying the principle of minimum total potential energy, the deflection (u_*, φ_*) *of the beam is a minimizer of the functional*

$$(u,\varphi) \mapsto J(u,\varphi) = E_S(u,\varphi) - V(u,\varphi) \tag{4.91}$$

defined on an appropriate vector space H. It is immediate to see that J is convex; assuming for the time being that H is defined so that J is moreover differentiable and coercive, the Proposition 4.4 says that a minimizer (u_*, φ_*) of J satisfies

$$J'(u_*,\varphi_*)(v,\psi) = 0 \quad \text{for any } (v,\psi) \in H \tag{4.92}$$

The derivative of J at a point (u,φ) is the linear mapping

$$(v,\psi) \in H \mapsto \int_0^L EI\varphi'\psi' + \kappa GA\big(u'-\varphi\big)\big(v'-\psi\big)\, dx_1 - \int_0^L f v + C\psi \, dx_1 - f_0 v(0) - f_1 v(L) - C_0\psi(0) - C_1\psi(L) \tag{4.93}$$

Using an integration by parts we see that the right end member of the previous formula can be rewritten as

$$\begin{aligned} &-\int_0^L \big(EI\varphi'' - \kappa GA\big(u'-\varphi\big) + C\big)\psi \, dx_1 \\ &-\int_0^L \Big(\kappa GA\big(u'-\varphi\big)' + f\Big) v \, dx_1 \\ &+ \big(EI\varphi'(L) - C_1\big)\psi(L) - \big(EI\varphi'(0) + C_0\big)\psi(0) \\ &+ \big(\kappa GA\big(u'(L) - \varphi(L)\big) - f_1\big) v(L) - \big(\kappa GA\big(u'(0) - \varphi(0)\big) + f_0\big) v(0) \end{aligned}$$

and the formula (4.92) shows that *a minimizer* (u_*, φ_*) *of the functional* J *is solution in the sense of distributions of the following Timoschenko beam equations*

$$\begin{aligned} EI\varphi_*'' - \kappa GA\big(u_*' - \varphi_*\big) + C = 0 \\ \kappa GA\big(u_*' - \varphi_*\big)' + f = 0 \end{aligned} \tag{4.94}$$

The boundary conditions are introduced by appropriately defining the space H of the admissible displacements: For instance

- *the beam is said to be clamped* at the end $x_1 = 0$ or $x_1 = L$ if the displacement $u(x_1)$ and the rotation $\varphi(x_1)$ are both zero,
- *it is simply supported* a x_1 if the displacement $u(x_1)$ is zero while the rotation $\varphi(x_1)$ is left arbitrary,
- *the end* x_1 *is said to be free* if $u(x_1)$ and $\varphi(x_1)$ are arbitrary.

REMARK 4.14 We can readily check that the rotation φ and the displacement u satisfy the boundary conditions

$$\varphi'(L) = \frac{C_1}{EI} \quad \left(\text{resp. } \varphi'(0) = -\frac{C_0}{EI}\right)$$

if the beam is simply supported or free at the end $x_1 = L$ (resp. $x_1 = L$), while we must add the condition

$$u'(L) - \varphi(L) = \frac{f_1}{\kappa GA} \quad \left(\text{resp. } u'(0) - \varphi(0) = -\frac{f_0}{\kappa GA}\right)$$

if the end in question is free.

To conclude, it remains to specify a space H of tests functions satisfying the previously introduced boundary conditions and provide this space with a norm $\|.\|_H$ ensuring differentiability and coerciveness of J.

To this end we will first define H as a subspace of $H^1([0,L]) \times H^1([0,L])$ and endow it with the "scalar product"

$$\big((u,\varphi),(v,\psi)\big) \in H \mapsto < u, v > + < \varphi, \psi > \in \mathbb{R}$$

where

$$< u, v >= \int_0^L uv\,dx_1 + \int_0^L u'v'\,dx_1$$

is the standard scalar product on $H^1([0,L])$. To simplify we will use the following notations:

$$\|u\|_0 := \left(\int_0^L u(x)^2 dx\right)^{\frac{1}{2}} \quad |u|_1 := \left(\int_0^L \big(u'(x)\big)^2 dx\right)^{\frac{1}{2}} \quad \text{and} \quad \|u\|_1 := \big(\|u\|_0^2 + |u|_1^2\big)^{\frac{1}{2}}$$

and remind that the mappings

$$u \in L^2([0,L]) \mapsto \|u\|_0 \quad \text{and} \quad u \in H^1([0,L]) \mapsto \|u\|_1$$

are norms on the spaces $L^2([0,L])$ and $H^1([0,L])$, providing these spaces with structures of Hilbert spaces. This enables us to endow H with the norm

$$(u,\varphi) \in H \mapsto \|(u,\varphi)\|_H := \big(\|u\|_1^2 + \|\varphi\|_1^2\big)^{\frac{1}{2}}$$

These preliminaries being laid down, we are able to proof that:

1^o/ *The total energy* $(u,\varphi) \mapsto J(u,\varphi)$ *is continuously differentiable.* Using the Holder's inequality[36] we can indeed see that the formal derivative (4.93) is well defined and is a continuous function[37] of $\big((u,\varphi),(v,\psi)\big) \in H \times H$.

[36] If u and v are square integrable functions defined on $[0,L]$ then the product uv is integrable and the following inequality holds

$$\int_0^L |uv| dx_1 \le \|u\|_0 \|v\|_0$$

[37] Introducing the bi-linear form

$$\big((u,\varphi),(v,\psi)\big) \in H \times H \mapsto B\big((u,\varphi),(v,\psi)\big) := \int_0^L EI\varphi'\psi' + \kappa GA(u'-\varphi)(v'-\psi)dx_1 \tag{4.95}$$

the Holder's inequality allows to proof that there are two positive constants c_1 and c_2 such that:

$$\big|B\big((u,\varphi),(v,\psi)\big)\big| \le c_1\|(u,\varphi)\|_H\|(v,\psi)\|_H \quad \text{and} \quad \big|V(v,\psi)\big| \le c_2\|(v,\psi)\|_H \tag{4.96}$$

Inequalities (4.96) show actually that B and V are respectively bounded bilinear and linear forms; as such they are continuous and even c^∞.

2^o/ *If the beam is clamped at $x_1 = 0$ for instance the functional J is coercive*, or in other words, there is a positive constant c such that

$$\lim_{\|(u,\varphi)\|_H \to \infty} \frac{J(u,\varphi)}{\|(u,\varphi)\|_H} \geq c \tag{4.97}$$

PROOF. By definition (4.91) of J we have

$$\begin{aligned} J(u,\varphi) &\geq E_S(u,\varphi) - |V(u,\varphi)| \\ &\geq E_S(u,\varphi) - c_2\|(u,\varphi)\|_H \quad \text{by the second inequality of (4.96)} \end{aligned}$$

so that if we can define the space H in order to bound below the strain energy as follows

$$E_S(u,\varphi) \geq c_3\|(u,\varphi)\|_H^2 \tag{4.98}$$

where c_3 is a positive constant, the inequality (4.97) clearly holds. □

PROOF OF INEQUALITY (4.98) FOR CLAMPED BEAM. The proof of this inequality makes use of the following result:

LEMMA 4.5 (Poincaré-Friedrichs inequality) *Assume that $v \in H^1([0,L])$ satisfies the boundary condition $v(0) = 0$ then there is positive constant c_4 such that*

$$\|v\|_0 \leq c_4|v|_1 \tag{4.99}$$

Now, the strain energy (4.89) can be written down as

$$2E_S(u,\varphi) = EI|\varphi|_1^2 + \kappa GA\left(|u|_1^2 + \|\varphi\|_0^2\right) - 2\kappa GA\int_0^L u'\varphi dx_1$$

Let μ be a positive constant, we have[38]

$$2\int_0^L u_2'\varphi dx_1 \leq \mu\int_0^L \varphi^2 dx_1 + \frac{1}{\mu}\int_0^L \left(u_2'\right)^2 dx_1$$

using the inequality (4.99) with $v = \varphi$ we deduce that

$$\begin{aligned} 2E_S(u,\varphi) &\geq \kappa GA\left(1 - \frac{1}{\mu}\right)|u|_1^2 + EI|\varphi|_1^2 + \kappa GA(1-\mu)\|\varphi\|_0^2 \\ &\geq \kappa GA\left(1 - \frac{1}{\mu}\right)|u|_1^2 + \frac{EI}{2}|\varphi|_1^2 + \left(\frac{EIc_1^2}{2} + \kappa GA(1-\mu)\right)\|\varphi\|_0^2 \end{aligned}$$

If we set $\mu = \frac{EIc_1^2}{2\kappa GA} + 1$ we get

$$2E_S(u,\varphi) \geq \min\left(\frac{EIc_4^2}{2\kappa GA + EIc_4^2}, \frac{EI}{2}\right)\left(|u|_1^2 + |\varphi|_1^2\right)$$

Then, using once more the inequality (4.99) we obtain

$$E_S(u,\varphi) \geq c_3\left\|(u,\varphi)\right\|_H^2$$

[38] For any couple of numbers x, y, we have the inequality

$$2xy \leq x^2 + y^2$$

form which we deduce

$$2ab = 2(\sqrt{\mu}a)\left(\frac{b}{\sqrt{\mu}}\right) \leq \mu a^2 + \frac{1}{\mu}b^2$$

if μ is a given positive number and a, b is a couple of real numbers.

with

$$c_3 = \frac{1}{2}\min\left(\frac{EIc_4^2}{4\kappa GA + 2EIc_4^2}, \frac{EI}{4}\right)\min\left(1, c_4^2\right) > 0$$

□

PROOF OF LEMMA 4.5. Assume first that v is continuously differentiable. Using the condition $v(0) = 0$, we can see that $v(x_1)$ is defined by the formula

$$v(x_1) = \int_0^{x_1} v'(x)dx$$

so that

$$\begin{aligned} v(x_1)^2 &\leq \left(\int_0^{x_1} |v'(x)|dx\right)^2 \\ &\leq \int_0^{x_1} \left(v'(x)\right)^2 dx \int_0^{x_1} dx \quad \text{(by the Holder's inequality)} \\ &\leq x_1 |v|_1^2 \end{aligned}$$

for any $x_1 \in [0, L]$. Integrating the previous inequality between 0 and L, we get

$$\|v\|_0^2 = \int_0^L v^2 dx_1 \leq |v|_1^2 \int_0^L x_1 dx_1 = \frac{L^2}{2}|v|_1^2$$

and we have proofed the formula (4.99) with $c_4 = \frac{L}{\sqrt{2}}$ for continuously differentiable functions. Extension of formula (4.99) to the space $H^1([0, L])$ is obtained by a density argument (of $C^1([0, L])$ in $H^1([0, L])$). □

REMARK 4.15 The condition $u(0) = \varphi(0) = 0$ is essential for establishing the coerciveness of J, one can easily check that J is not coercive on $H^1[0, L] \times H^1[0, L]$.

Homework.

1^o/ Implement a numerical method for the resolution of the mechanical problem depicted in figure (Fig.4.23).

- Use the Proposition 4.3 to proof an existence result (Hint: check that the set

  $$K := \left\{(u, \varphi) \in H\,;\, u(l_1) \geq 0 \text{ and } u(l_2) \leq 0\right\}$$

  is a closed and convex subset of H);
- write down a FEM interpolation of the bending beam equation, formulate the previously defined optimization problem under the form of a saddle point problem and give an interpretation of the Lagrange multipliers (Hint: you can have a look at "http://iut.univ-lemans.fr/ydlogi/index.html" for the computation of the elementary matrices);
- implement the Uzawa algorithm to compute numerically the saddle point, plot the deformed configuration of beam and compute the reaction forces F_1 and F_2.
- how to linearize the problem? (Hint: use a penalty method to compute the reaction forces and define the linear equations according to the direction of the loading forces);

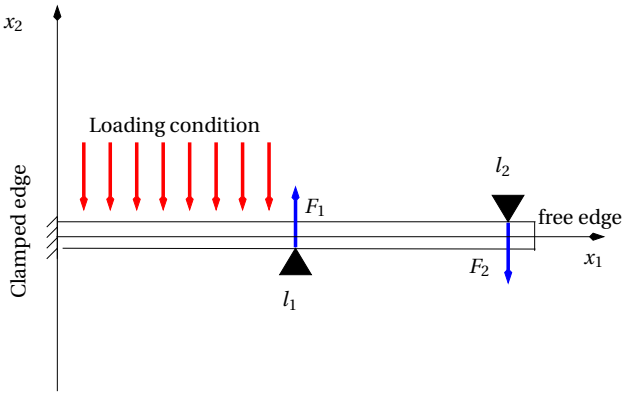


Fig. 4.23. **Beam bending under unilateral constraints**. We are considering a beam of neutral axis x_1, clamped at the end $x_1 = 0$ and free at $x_1 = L$. We assume that two mountings are installed around the beam to impose the displacements $x_2(l_1) \geq 0$ and $x_2(l_2) \leq 0$. The beam is at last loaded by a distributed system of forces $f(x_1)$ directed along the axis x_2 and non zero for $0 \leq x_1 \leq l_1$.

- how to use the previously defined mathematical machinery to numerically solve the associated dynamical equations, assume for instance that the forces are of the form $\sin(\omega t) f(x_1)$? (Hint: use for instance an implicit Euler method);
- can you define a linear problem having the same solution as this dynamical problem?

2^o/ Implement the steepest and the Newton algorithms to identify the coefficients of the Stromeyer's formula in Exercise 1.1 page 28.

Solution of exercise 4.5. Setting $x = (x_1, \cdots, x_4)^t$, and introducing potential energy[39]

$$x \mapsto \begin{cases} \frac{1}{2} \sum_{i=1}^{3} k_i (x_{i+1} - x_i - l_i)^2 & \text{if } 0 \leq x_1 \leq \cdots \leq x_4 < L \\ +\infty & \text{else} \end{cases}$$

the motion equations of the mechanical system defined in figure (Fig. 4.20) can be written as

$$[M]\ddot{x} + [K]x \in f_0 + \partial \psi_U(x)$$

$$\text{with the initial conditions } \begin{cases} x_1(0) = x_0,\ x_2(0) = x_0 + l_1, \cdots, \\ \qquad x_4(0) = x_0 + l_1 + l_2 + l_3 \\ \dot{x}(0) = y_0 \text{ are given initial velocities} \end{cases} \tag{4.100}$$

where

- ψ_U is the characteristic function of the convex $U = \left\{x \in \mathbb{R}^4;\ 0 \leq x_1 \leq \cdots \leq x_4 < L\right\}$, it is defined by

$$\psi_U(X) = \begin{cases} 0 & \text{if } x \in U \\ +\infty & \text{else} \end{cases}$$

[39] Which account for the non-penetrating conditions $0 \leq x_1 \leq \cdots \leq x_4 < L$.

- $[M]$, $[K]$ and f_0 are respectively defined by

$$[M] = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \quad [K] = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 \\ 0 & 0 & -k_3 & k_3 \end{bmatrix}$$

and

$$f_0 = \begin{pmatrix} -k_1 l_1 \\ k_1 l_1 - k_2 l_2 \\ k_2 l_2 - k_3 l_3 \\ k_3 l_3 \end{pmatrix}$$

Denoting h the step size of the time discretization, an iteration of the implicit Euler method consists, knowing x_t and y_t, to calculate the positions x_{t+h} and the velocities y_{t+h} in solving the variational inequation

$$([M] + h^2[K])x_{t+h} \in h^2 f_0 + [M](hy_t + x_t) + \partial\psi_U(x_{t+h})$$

and in setting $y_{t+h} := \frac{x_{t+h}-x_t}{h}$. Using the definition (2.38) page 61 of a subdifferential and setting $z_t := h^2 f_0 + [M](hy_t + x_t)$, we see that this inequation rewrite as

$$\psi_U(x) - \psi_U(x_{t+h}) \geq \langle([M] + h^2[K])x_{t+h} - z_t, x - x_{t+h}\rangle \quad \text{for any } x \in \mathbb{R}^4$$

or, by definition of ψ_U, as $x_{t+h} \in U$ satisfies

$$\langle([M] + h^2[K])x_{t+h} - z_t, x_{t+h} - x\rangle \leq 0 \quad \forall x \in U$$

which is, see Proposition 4.3 page 143, the Euler-Lagrange inequation associated with the constrained optimization problem

$$\begin{aligned} &J(x_{t+h}) = \min_{x \in U} J(x) \\ &\text{where } J \text{ is the functional } x \mapsto \frac{1}{2}\langle([M] + h^2[K])x, x\rangle - < z_t, x > \end{aligned} \tag{4.101}$$

Now, introducing the matrix

$$[B] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we see that the convex U is the subset

$$U = \left\{x \in \mathbb{R}^4 \,;\, [B]x \leq (0,0,0,0,l)^t\right\}$$

and the problem (4.101) rewrites, in the form defined in the Example 4.6 page 160, as

$$J(x_{t+h}) = \min_{[B]x \leq q} J(x) \quad \text{where} \quad q := (0,0,0,0,l)^t$$

As such, it can be solved in using the Uzawa algorithm 4.4 page 159 to compute x_{t+h} as the first argument x_* of the saddle point (x_*, λ^*) of the Lagrangian

$$(x,\lambda) \in \mathbb{R}^4 \times (\mathbb{R}_+)^5 \mapsto L(x,\lambda) = \frac{1}{2}\langle([M] + h^2[k])x, x\rangle - \langle z_t - [B]^t\lambda, x\rangle + < q, \lambda > \tag{4.102}$$

The attentive reader will have noticed that the term $\frac{1}{h^2}[B]^t\lambda^*$ is a contact force between the particles which forbids interpenetrations.

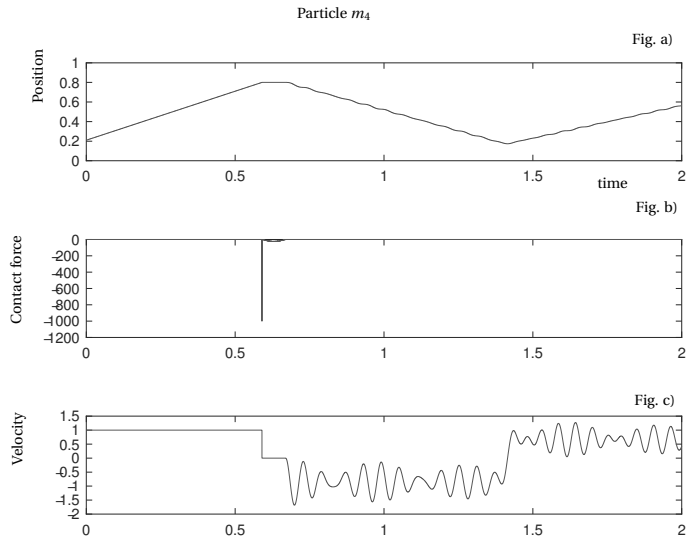


Fig. 4.24. **Motion of particle** m_4. We represent in figure Fig.a) the positions $t \mapsto x_4(t)$ of the particle m_4. We see that the motion takes place at a constant average velocity between the walls of abscissas 0 and l and that each contact phase (between m_1 and m_4 and the walls) contribute to reduce the speed. This is explained in Figures Fig.b) and Fig.c) which show that the contact takes place into two steps. The first one consists to instantaneously stop the particle when it reaches the wall of abscissa l; this phase is characterized by a strong discontinuity of the contact force Λ_4 plotted in figure Fig.b). In the second stage, the pushing masses $m_1, \cdots, m_3$ deform the spring k_1 and convert their kinetic energies into a deformation energies. The deformation energy accumulated in the spring k_4 is at last reconverted in kinetic energy of the center of gravity which allows to continue the motion in the opposite direction.

We propose page 205 a Matlab program which implements this algorithm for the transient integration of the equations (4.100). This program was run to perform a simulation of the mechanical system plotted in figure (Fig. 4.20) with the following data

- masses: $m_1 = m_4 = 0.1kg$, $m_2 = m_3 = 0.25kg$,
- stiffnesses:

$$k_1 = k_3 = 1.0e+03N/m,\ k_2 = 1.0e+04N/m$$

  of lengths $l_1 = l_3 = 0.05m$, $l_2 = 0.1m$,
- initial conditions: $x_0 = 0.01$ (position of particle m_1) initial velocity $1m/s$ on each particle,
- distance between walls: $l = 0.8m$.

Results of the simulation are plotted and analyzed in figures (Fig. 4.24), (Fig. 4.25) and (Fig. 4.24).

REMARK 4.16 With these numerical data only the contacts between the particles m_4 (resp. m_1) and the can take place, so that the simulation results can easily be analyzed and interpreted. We see that the contact force Λ_4, which is $\frac{m_4 v}{h}$, diverges when the time

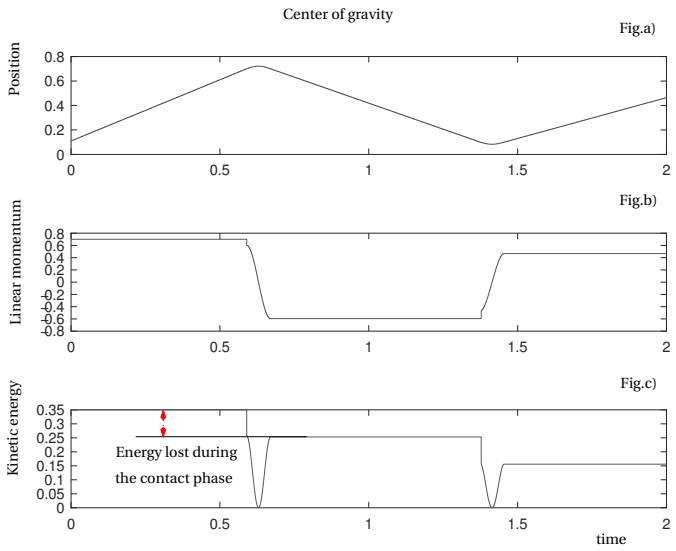


Fig. 4.25. **Motion of the COG of the mechanical system**. We see here that the motion of the center of gravity takes place at a constant velocity between two impacts of the particles m_4 and m_1, and that the velocity changes its sign at each impact. Denoting by m_t the total mass of the mechanical system, we can check that the restitution coefficient is $-\left(1-\frac{m_1}{m_t}\right)$ and corresponds to the loss of linear momentum resulting of the stopping of the particles m_4 and m_1 during the impacts. We thus retrieve the classic results of the theory of inelastic collisions while restitution of energy is not instantaneous but depends on the time spent by the pushing masses to compress the springs k_1 and k_2; this loading time depends for instance on the momentums of the particles m_2 and m_3. We let the reader check that this coefficient is an upper limit of the possible restitution coefficients because, choosing $k_2 >> k_1 = k_3$, the deformation energy is actually stored in the contact springs k_1 and k_3.

step size h goes to 0. There is nothing surprising about this, because the involved mechanical phenomena introduce discontinuities of velocities and then of accelerations; equation (4.100) must thus be understood in the sense of distributions and not in the classical meaning of the term. In this spirit, we will not draw any conclusions about the first peak of forces depicted in figure (Fig. 4.25-b) and it will be better to understand this term as a distribution rather than as a function.

Homework.

1^o/ Is the problem (4.100) well-posed? (hint assume that $m_2 = m_3 = m_4 = 0$ and $k_1 = k_2 = k_3 = 0$, verify that the mapping

$$t \mapsto x_1(t) = \begin{cases} x_0 + t v_0 & \text{if } x_1(t) \le l \\ l - t c_1 v_0 & \text{for } l \ge x_1(t) \ge 0 \text{ where } c \text{ is a given positive constant} \\ c_2 v_0 & \text{for } 0 \le x_1(t) \le l \end{cases}$$

satisfies the equation (4.100). How to generalize this example when m_1 and m_2 are both non-zeros?)

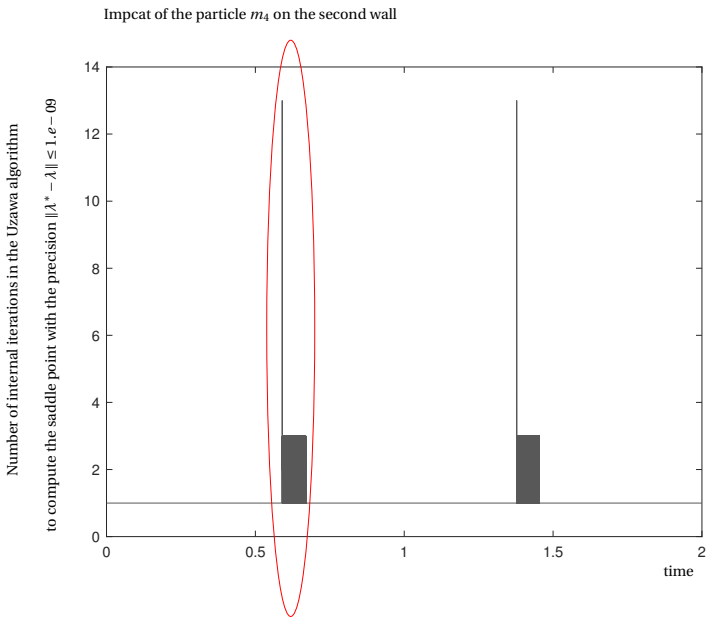


Fig. 4.26. **Performance of the iterative method**. We show on this picture the number of iterations of the Uzawa algorithm necessary to the computation of the saddle point of the Lagrangian (4.102). We see that the algorithm reduces to the resolution of a linear system as long as there is no contact. A non-constant step size projection algorithm should improve (ie. shorten) the Lagrange multiplier updating process during the contact phases.

2^o/ Run the program in assuming that

- $k_1 = k_2 = k_3 = 0$ with the initial velocities $(1,1,1,1)^t$, $(1,0,0,-1)^t$ and $(2,0,0,-1)^t$
- the numerical values of the stiffnesses and masses permit multiple simultanous contacts between the particles.

explain the obtained results (by examining the contact forces and the energy transfers). The reader wishing a better understanding of the mathematical theory of shocks can for instance find his inspiration in the pioneering work of MOREAU [**28**] [**29**].

3^o/ How to take into account energy dissipation by local lamination? (hint : assume for instance that the Rayleigh dissipation function is defined by $\sum_{i=1}^{3} c_i|\dot{x}_i - \dot{x}_{i+1}|$, where $(c_i)_{i=1}^3$ are positive constants).

Solution of exercise 4.6. We can easily check that the $x(t) = ae^{bt}$ is solution of the equation $\dot{x} = bx$ with the initial condition $x(0) = a$ so that

$$J(a,b) := \int_0^T x(t)\,dt = \frac{a}{b}(e^{bT} - 1)$$

This formula leads to compute the derivatives as:

$$\partial_a J = \frac{1}{b}(e^{bT} - 1) \quad \partial_b J = \frac{a[(Tb-1)e^{bT} + 1]}{b^2} \tag{4.103}$$

Let's check that these results are obtained with the help the procedure defined in Proposition 4.10 page 165. within this framework we have $f(x,b,t) := bx$ and $j(x) := x$

so that

$$D_x f.\lambda = b\lambda \quad \text{and} \quad \nabla j(x) = 1$$

and the adjoint equation (4.40) is rewritten as

$$\dot{\lambda} + b\lambda = -1 \quad \lambda(T) = 0$$

and has the solution $\lambda(t) = e^{b(T-t)} - \frac{1}{b}$. Using formula (4.41) this leads to compute the derivative of J with respect to b as the integral

$$\begin{aligned}\partial_b J &= \int_0^T \partial_b f(x,b)\lambda = \int_0^T x(t)\lambda(t)dt \\ &= a\int_0^T \left(e^{b(T-t)} - \frac{1}{b}\right) e^{bt} dt\end{aligned}$$

and we find the second formula in (4.103). Note, on the other hand, that $\lambda(0) = \frac{1}{b}(e^{bT}-1)$ is $\partial_a J$.

Homework. Within the framework of Proposition 4.10 proof that

- the criteria J is a differentiable function of the initial conditions $X(0) = X_0$ of the state equation,
- and that the gradient of J is defined by $\nabla J(X_0) = \Lambda(0)$ where $t \mapsto \Lambda(t)$ is the solution of the adjoint equation (4.40).

APPENDIX A

SOME ADDITIONAL PROGRAMS

A.1. Transient Integration algorithm for a contact Problem

The program listed below is the Matlab implementation of the algorithm defined in Exercise 4.5 pages 190 and 198.

```
clear all
close all
k_1=1.e+03;k_2=1.e+04;k_3=1.e+03; % Stiffnesses (N/m)
%k_1=1.e+02;k_2=0.0;k_3=1.e+02; % Stiffnesses (N/m)
% Stiffness matrix
K=[k_1,-k_1,0,0;-k_1,k_1+k_2,-k_2,0;...
0,-k_2,k_2+k_3,-k_3;0,0,-k_3,k_3];
%
m_1=0.10;m_2=0.25;m_3=0.25;m_4=0.10;% Masses (kg)
% Mass matrix
M=[m_1,0,0,0;0,m_2,0,0;...
0,0,m_3,0;0,0,0,m_4];
%
% Constraints matrix
B=[-1,0,0,0;1,-1,0,0;...
0,1,-1,0;0,0,1,-1;...
0,0,0,1]; % Note that rank(B)=4
%
% Lengths of the springs (m)
l_1=0.05;l_2=0.1;l_3=0.05;
%
F_0=[-k_1*l_1;k_1*l_1-k_2*l_2;...
k_2*l_2-k_3*l_3;k_3*l_3];
%
L=0.8;% Distance between the walls (m)
%
```

```
x_0=0.01; %
X=[x_0;x_0+l_1;...
x_0+l_1+l_2;x_0+l_1+l_2+l_3];% Initial positions
Y=[1.;1.;1.;1];% Initial velocities
% Discretization step-size
Dt=0.0001; A=M+(Dt**2)*K;
% Dt=0.0001/10; A=M+(Dt**2)*K;
%
T=2.0; % Integration time (s)
% T=10.0;
time=0:Dt:T;
%
sol(1:4,1)=X; % Positions
sol(5:8,1)=Y; % Velocities
sol(9:12,1)=[0;0;0;0]; % Contact forces
%
lambda=[0;0;0;0;0]; % Initial condition on the Lagrange multiplayer
rho=1.9*min(eig(A))/norm(B)^2;
N_iter(1)=1;
%
for i=2:size(time,2)
   Z=(Dt**2)*F_0+M*(Dt*sol(5:8,i-1)+sol(1:4,i-1));
   D_lambda=1.0;
   n_iter=0.;
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Iterative method to compute
   % the saddle point of the Lagrangian (4.102)
   % (see algorithm 4.4 page 159).
   % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   while D_lambda>=1.e-09 % Until convergence
      X=inv(A)*(Z-B'*lambda);% Compute the positions
      %
      lambda_1=max(lambda+rho*(B*X-[0;0;0;0;L]),0); % Update the Lagrange multiplier
      %
      % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % Note that rho has been defined according to the formula given in
      % footnote 17 page 160 so that the iterative method is convergent.
      % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %
      D_lambda=norm(lambda_1-lambda);
      lambda=lambda_1;
      n_iter=n_iter+1;
   end
   % Store in the tables N_iter and sol
   N_iter(i)=n_iter; % the number of iterations
   sol(1:4,i)=X; % the positions
   sol(5:8,i)=(X-sol(1:4,i-1))/Dt; % the velocities
   sol(9:12,i)=-B'*lambda/Dt/Dt; % and the contact forces
   % at time t+Dt.
end
%
% Plot some results
%
figure
subplot(311)
```

```
plot(time,sol(4,:));title('Position of the particle m_4')
subplot(313)
plot(time,sol(8,:));title('Velocity')
subplot(312)
plot(time,sol(12,:));;title('Contact force')
%
% Center of gravity
Mass=m_1+m_2+m_3+m_4;
CoG=(m_1*sol(1,:)+m_2*sol(2,:)+m_3*sol(3,:)+m_4*sol(4,:))/Mass; % Position
Ma=(m_1*sol(5,:)+m_2*sol(6,:)+m_3*sol(7,:)+m_4*sol(8,:));% Linear momentum
KE_CoG=Ma.*Ma/2/Mass; % Kinetic energy
figure
subplot(311);plot(time,CoG);title('Position')
subplot(312)
plot(time,Ma);title('Linear momentum')
subplot(313)
plot(time,KE_CoG);title('Kinetic energy')

% Numerical simulation
figure
plot(time,N_iter); title('Number of internal iteration in the Uzawa algorithm')
% Compute the restitution coefficient
Rest=min(Ma)/max(Ma)
```

APPENDIX B

IMPLEMENTATION OF SOME EXAMPLES

B.1. Damage computation for a beam

- Main program

Step 1) Initialization of the computations and definition & loading conditions

```
%Time horizon in seconds
T0=4.0;
% To frame the significant part of the excitation between 3T0/2 and 5T0/2
T=T0*2;
% Number of samples and sampling frequency.
Nb_samp=512*4*2;
Dt=T/Nb_samp;
time=0:Dt:T-Dt;
% Sampling of the torques applied on the ends of the beam
% in the standard case, it is an input of the program,
% which is given as tabulated data.
Torque=1.e+05;% Here a torque at 100 daN*m
% Quasi-static component of the torque
for i=1:Nb_samp
    t=time(i);
    if t<=T/5.
        e1(i)=0;
    else if t<=2*T/5
            e1(i)=Torque;
        else if t<=3*T/5
                e1(i)=0.0;
            else if t<=4*T/5
```

```
                    e1(i)=-Torque;
                else
                    e1(i)=0.0;
                endif
            endif
        endif
    endif
endfor
% Plused component of the torque
% the frequencies are closed to the first natural frequency of the beam
omega_e0=145*2*pi;% the 3^{i-th} mode
omega_e1=35*2*pi;% the  2^{i-ème} mode
% Objective : see the effect of excitation frequencies close to each other
omega_e2=0.95*omega_e0;
%
Torque_2=0.3*(cos(omega_e0*time(Nb_samp/4+1:3*Nb_samp/4))...
              +0.5*cos(omega_e1*time(Nb_samp/4+1:3*Nb_samp/4))...
              +cos(omega_e2*time(Nb_samp/4+1:3*Nb_samp/4))...
              );
%
% Frame the excitations to satisfy the conditions
% of footnote 14 page 115
e1(Nb_samp/4+1:3*Nb_samp/4)=e1(Nb_samp/4+1:3*Nb_samp/4)...
                            +Torque*Torque_2(1:Nb_samp/2);
%
figure(1);
subplot(2,1,1)
plot(time,e1/1000)% Torque in daN*m
xlabel("Time in second")
ylabel("Torque in daN*m")
title("Torques applied at the ending cross-sections")
% Post-process the fast-Fourier transforms
subplot(2,1,2)
F_samp=1/Dt/2;
dfreq=2*F_samp/Nb_samp;
FFT_signal=abs(fft(e1))(1:Nb_samp/2);
Freq=0:dfreq:F_samp;
plot(Freq(2:Nb_samp/2+1),FFT_signal/Nb_samp/1000)
xlabel("Frequencies in Hz")
ylabel("Torque in daN*m")
title("Fast fourier transform of the applied torques")
clear F_samp dfreq FFT_signal Freq;
%
% Mechanical data
Module_young=0.16500E+05;% Young modulus daN / mm^2
Poisson=0.3;
J=Module_young/(1+Poisson)/2;% Lamé coefficient
Diam=40.5;% Diameter mm
J=J*pi*Diam**4/32.;% Torsional stiffness module
Masse_vol=0.79e-09;% mass density  kg*e-04 / mm^3
section=pi*Diam**2/4;
masse_sect=Masse_vol*section
Inert=masse_sect*Diam**2/4/2; % Inertia of the cross-sections
I_ad=15.0; % Additional inertia in  kg*1e-04  mm^2
%
```

```
% Parameters of the Wohler's curve (a Stromeyer formula in this case)
b_s=0.42;
C_s=3.6E+09;
Sigma_d=80.0; % in MPa
%Sigma_d=Sigma_d*0.8;
Sigma_d=Sigma_d*1.2;
epsilon_sigma=1.0e-5;
%
% Geometric data of the beam
Nb_nodes=27;%  Number of nodes
delta_s=20.0;% Length of an element in  mm
C_1=0.4;% Diameter of notch (nominal)
%
% The table G is intended for defining the diameter of beam at each node
G=[1,1,1,1,0.9,0.8,0.7,0.6,0.5,C_1,C_1,C_1,C_1,...
   C_1,C_1,C_1,C_1,C_1,0.5,0.6,0.7,0.8,0.9,1,1,1,1];
% Plot the  profile of the beam
figure(2)
hold on
plot(Diam*G/2)
plot(-Diam*G/2)
xlabel('Node number')
ylabel('Diameter')
title("Profile of the beam")
%
% Damping model
% Type_damp=2;%  Damping proportional to the stiffness matrix
% coef_damp=1/100;
%
Type_damp=2; % Damping proportional to the critical damping per mode
coef_damp=2.5/100;
%
% Type_damp=3.0;% Damping proportional to the mass matrix
% coef_damp=10.0;
```

Step 2) Assembling of the state equation, which is modified during the optimization process if for instance C_1 or I_{ad} is an optimization parameter.

```
%
% Assembling of the stiffness matrix
% (this matrix will not be modified)
XK=assemb_K(Nb_nodes,G,delta_s,J);
%
% Remove the singularity of the stiffness
XK((Nb_nodes-1)/2,(Nb_nodes-1)/2)= ...
   1.01*XK((Nb_nodes-1)/2,(Nb_nodes-1)/2);
%
% Assembly of the mass matrix
XM=assemb_M(Nb_nodes,G,delta_s,Inert);
%
% Take into account the additional inertias
XM(4,4)=XM(4,4)+0.5*I_ad;
XM(10,10)=XM(10,10)+0.25*I_ad;
XM(18,18)=XM(18,18)+0.25*I_ad;
XM(24,24)=XM(24,24)+0.5*I_ad;
```

Step 3) Matrix manipulations to implement the second of the algorithm 3.5 page 106

```
%Computation of the square root of M^-1
M_s=real(sqrtm(inv(XM)));
%
% Computation of the modal basis
% Digonalization of the matrix [M]^-1/2 [K] [M]^-1/2
[R,omega]=eig(M_s*XK*M_s);% omega are the natural velocities
eig_0=M_s*R;% Eigen modes for the state equation
% Eigen-modes for the adjoint equation see formula (4.62)
eig_1=sqrtm(XM)*R;
%
puls=sqrt(abs(diag(omega)));
Freq_prop=puls/2/pi; % Natural frequencies
%
% Definition of the damping
if Type_damp==1
   % Case proportional to stiffness matrix
   damp=abs(diag(omega))*coef_damp;
   else if Type_damp==2
           % Case proportional to critical damping
           damp=puls*coef_damp;
        else
           % Constant damping in the modal basis
           damp=coef_damp*ones(size(diag(omega),1),1);
       endif
endif
```

Step 4) Sampling of the convolution kernels (according to sampling of the inputs data)

```
j1=1;
for j=1:Nb_nodes
    omega1=puls(j,1);
    c0=damp(j);
    % Only the modes whose frequencies are lower than F_samp/2 are
    % are taken into account; see table 3.3,
    % the contribution of the other modes can be neglected.
    if (4*omega1^2-c0^2)>=0
       if omega1/pi<=1./Dt
          Ind_mod(j1)=j;
          delta=sqrt(4*omega1^2-c0^2)/2;
          co=cos(delta*time);
          si=sin(delta*time);Diam_mil=Diam*(G(No_elt)+G(No_elt-1))/2.;
          % Damping of the current mode
          amor=exp(-c0*time/2)*Dt;
          % See the formulas (3.11)
          tab1(j1,:)=(si/delta).*amor;
          tab2(j1,:)=(co-c0*si/2/delta).*amor;
          j1=j1+1;
       endif
    else % Case of the rigid body mode see formula (3.14)
       tab1(j1,:)=time;
       tab2(j1,:)=ones(1:Nb_samp);
       j1=j1=1;
    endif
```

```
endfor
%
% Plot the eigen-modes taken into account to compute the damage
% Note that they are normalized with respect to the mass matrix
figure(3)
nb_plots=size(Ind_mod,2);
coor=delta_s*(0:1:Nb_nodes-1);
for i=1:nb_plots
    subplot(nb_plots,1,i)
    plot(coor,eig_0(:,Ind_mod(i)))
    j=Ind_mod(i);
    txt=['Mode N° ',num2str(j),' at ',num2str(puls(j,1)/2/pi), ' Hz'];
    title(txt)
endfor
```

Step 5) Integration of the state equations in the modal basis

```
f=zeros(Nb_nodes,1);
EIG=R'*M_s;
for j=1:Nb_samp
    % Loading of the beam from the imposed torques
    % At the right end (point E) torque is distributed over 3 nodes
    f(Nb_nodes)=e1(j)/4;
    f(Nb_nodes-1)=e1(j)/2;
    f(Nb_nodes-3)=e1(j)/4;
    % Idem on the left (point A), but with the opposite torque
    f(1)=-e1(j)/4;
    f(2)=-e1(j)/2;
    f(3)=-e1(j)/4;
    % Pass the loads in the modal basis
    f_m=EIG*f;
    % Right hand member of the uncoupled system
    for ind_mod=1:size(Ind_mod,2)
        ex(ind_mod,j)=f_m(Ind_mod(ind_mod));
    endfor
endfor
%
% Computation of the convolution products
for ind_mod=1:size(Ind_mod,2)
    sol(ind_mod,:)=conv(tab1(ind_mod,:),ex(ind_mod,:));% Displacements
    sol_p(ind_mod,:)=conv(tab2(ind_mod,:),ex(ind_mod,:));% Velocities
end;
```

Step 6) Post-processing and computation of the stresses at the middle of the elements

```
% Element to be post-processed in fatigue
No_elt=10;
% No_elt=14;
% Diameter of the middle of the element
%
Diam_mil=Diam*(G(No_elt)+G(No_elt-1))/2.;
Coeff=Diam_mil*Module_young/delta_s/(1+Poisson)/4;
%
% Return in the original basis and compute the stress
%
for no_samp=1:Nb_samp
```

```
    % Stresses and their time derivatives
    z=0;
    for ind_mod=1:size(Ind_mod,2)
        z=z+(eig_0(No_elt,Ind_mod(ind_mod)) ...
             -eig_0(No_elt-1,Ind_mod(ind_mod)) ...
             )*sol(ind_mod,no_samp);
    endfor
    % Sampling of Σ_e(t)
    sigma_e(no_samp)=Coeff*real(z);
    %
    z=0;
    for ind_mod=1:size(Ind_mod,2)
        z=z+(eig_0(No_elt,Ind_mod(ind_mod)) ...
             -eig_0(No_elt-1,Ind_mod(ind_mod)) ...
             )*sol_p(ind_mod,no_samp);
    endfor
    % Sampling of Σ̇_e(t)
    dot_sigma_e(no_samp)=Coeff*real(z);
endfor
%
% Plot fast Fourier transform of stresses
figure(4);
F_samp=1/Dt/2;
dfreq=2*F_samp/Nb_samp;
FFT_signal=abs(fft(sigma_e))(1:Nb_samp/2);
Freq=0:dfreq:F_samp;
plot(Freq(2:Nb_samp/2+1),FFT_signal/Nb_samp)
xlabel("Frequencies in Hz")
ylabel("Stress in MPa")
txt=["Fast fourier transform of the stress on element ",int2str(No_elt-1)];
title(txt)
clear F_samp dfreq FFT_signal Freq;
```

Step 7) Compute the damage; implementation of the algorithm 3.3.

```
% Sampling of the table sigma_a
% (at 1% of the maximal stress)
% see algorithm 3.2
sigma_max=1.1*max(abs(sigma_e));
delta_sigma=sigma_max/100;
sigma_a=[0,epsilon_sigma:delta_sigma:sigma_max];
[n_0,N_sigma]=size(sigma_a);
% Partial integration of the equation (2.39)
% Algorithm 3.1
P_1=0*eye(1,N_sigma);
for no_samp=1:Nb_samp-1
  [sigma,P_1]=integ_E_sigma(P_1,sigma_e(no_samp+1),sigma_a,N_sigma);
endfor
% At this stage the relay operators have the right initialization
% and the response of the Preisach operator is ‘‘periodic’’.
% We can start the cycles counting.
sigma_0(1)=sigma; % Sampling of Σ_0(t).
for no_samp=1:Nb_samp-1
    [sigma,P_1]=integ_E_sigma(P_1,sigma_e(no_samp+1),sigma_a,N_sigma);
    sigma_0(no_samp+1)=sigma;
```

```
endfor
%
for no_samp=1:Nb_samp
    [w_0,w_1]=w_strom(C_s,b_s,Sigma_d,sigma_0(no_samp));
    % see formula (2.63) page 74
    v_dom(no_samp)=w_0*abs(dot_sigma_e(no_samp));
    % Store backward the time  derivatives;
    % see the formulas (2.68) and (2.69) these data are used
    % for the integration of the adjoint equation
    d1_v_dom(Nb_samp-no_samp+1)=w_1*dot_sigma_e(no_samp);
    d2_v_dom(Nb_samp-no_samp+1)=w_0*sign(dot_sigma_e(no_samp));
endfor
% Computation of the damage by numerical integration (trapezes method).
damage=trapz(time(1:Nb_samp),v_dom)
```

- Auxiliary functions

  – Assembling of the mass matrix

```
function XM=assemb_M(Nb_nodes,G,delta_s,Inert)
    XM=zeros(Nb_nodes,Nb_nodes);
    for i=1:Nb_nodes-1
        % Elementary Mass matrix
        M_11=G(i+1)**4+3*G(i)*G(i+1)**3 ...
             +6*G(i)**2*G(i+1)**2+10*G(i)**3*G(i+1)+15*G(i)**4;
        M_12=5*G(i+1)**4+8*G(i)*G(i+1)**3 ...
             +9*G(i)**2*G(i+1)**2+8*G(i)**3*G(i+1)+5*G(i)**4;
        M_22=G(i)**4+3*G(i+1)*G(i)**3 ...
             +6*G(i+1)**2*G(i)**2+10*G(i+1)**3*G(i)+15*G(i+1)**4;
        M_i=delta_s*Inert*[M_11/105.0,M_12/210.0; ...
                           M_12/210.0,M_22/105.0];
        % Assembly to the global matrix
        XM(i,i)=XM(i,i)+M_i(1,1);
        XM(i,i+1)=XM(i,i+1)+M_i(1,2);
        XM(i+1,i)=XM(i+1,i)+M_i(2,1);
        XM(i+1,i+1)=XM(i+1,i+1)+M_i(2,2);
    endfor
endfunction
```

  – Assembling of the stiffness matrix

```
function XK=assemb_K(Nb_nodes,G,delta_s,J)
    XK=zeros(Nb_nodes,Nb_nodes);
    for i=1:Nb_nodes-1
        % Elementary stiffness
        K_i=J/delta_s*(G(i+1)**4+G(i)*G(i+1)**3+G(i)**2*G(i+1)**2 ...
                       +G(i)**3*G(i+1)+G(i)**4 ...
                       )/5.0*[1,-1;-1,1];
        % Assembly in the global matrix
        XK(i,i)=XK(i,i)+K_i(1,1);
        XK(i,i+1)=XK(i,i+1)+K_i(1,2);
        XK(i+1,i)=XK(i+1,i)+K_i(2,1);
        XK(i+1,i+1)=XK(i+1,i+1)+K_i(2,2);
    endfor
endfunction
```

– Computation of $\Sigma_0(t)$.

```
function [sigma_0,P_1]=integ_E_sigma(P_1,v,sigma_a,N_sigma);
%
% Partial integration of (2.39) for σ_a ∈ (σ_i)_{i=1}^{N_sigma}
%
    P_1=min(v+sigma_a,max(v-sigma_a,P_1));
    %
    % Research of the first extremum in the table P_1
    % (see algorithm (3.2) page 118 )
    %
    for i=1:N_sigma-2
        p=(P_1(i)-P_1(i+1))*(P_1(i+1)-P_1(i+2));
        if p<=0.0
            break;
        end
    end
    %
    % Computation of σ_0 at time t_{k+1}.
    sigma_0=sigma_a(i+1);
endfunction
```

– Computation of the partial derivatives $\partial_v j(v,\sigma_0,\dot{v})$ and $\partial_{\dot{v}} j(v,\sigma_0,\dot{v})$ see formulas (2.68) and (2.69) page 76

```
function [w_0,w_1]=w_strom(C_s,b_s,sigma_d,v_2)
    % Stromeyer's formula
    % (without accounting for mean stress).
    w_0=max(v_2-sigma_d,0)^(1/b_s-1)/b_s/C_s/2;
    w_1=(1-b_s)/(4*b_s**2*C_s)*max(v_2-sigma_d,0)^(1/b_s-2);
endfunction
```

B.2. Integration of the adjoint equation

The elements of program given below are intended for complementing the previous program to simultaneously integrate the state and the adjoint equations and compute the gradient of the damage with respect to the design parameters.

- Continuation of the main program

Step 8) Computation of the derivative of the mass matrix with respect to the additional inertias.

```
%
% Derivative of the mass matrix
% with respect to the additional inertias
DI_XM=zeros(Nb_nodes,Nb_nodes);
DI_XM(4,4)=0.5;
DI_XM(10,10)=0.25;
DI_XM(18,18)=0.25;
DI_XM(24,24)=0.5;
```

Step 9) Computation of the derivatives of $[M]^{-1}$, $[M]^{-1}[K]$ and $[M]^{-1}[W]$ with respect to the additional inertias

```
pkg load control;% to have access to the Lyapunov function
% Computation of the derivative of the matrices
%
% Derivative of  [M]^-1[K]
DI_M=-inv(XM)*DI_XM*inv(XM);
DI_K=DI_M*XK;
%
% Derivative of the damping
if Type_damp==1
   % Proportional to the stiffness matrix
   DI_W=coef_damp*DI_K;
   else if Type_damp==2
            % Proportional to critical damping per mode
            % apply the product rule and the chain rule
            % to M^-1W, where  W is defined by
            % the formula (3.28) page 114
            DI_M_s=lyap(M_s,DI_M,'c');% Solve a Lyapounov's equation
            % W1 is the square of the damping matrix in the modal basis
            W1=real(M_s*XK*M_s);
            W2=real(DI_M_s*sqrtm(W1)*sqrtm(XM));
            W3=real(M_s*sqrtm(W1)*lyap(sqrtm(XM),DI_XM,'c'));
            DW1=real(DI_M_s*XK*M_s + M_s*XK*DI_M_s);
            W4=M_s*real(lyap(real(sqrtm(W1)),DW1,'c'))*real(sqrtm(XM));
            DI_W=real(W2+W3+W4)*coef_damp;
     else
           % DI_W=0
     end
end
```

Step 10) Integration of the adjoint equation in its modal basis

```
% Computation of right hand member of the adjoint equation
%
E=zeros(Nb_nodes,1);
E(No_elt-1)=-Coeff; % The damage is computed on the element No_elt
E(No_elt)=Coeff;
E_m=eig_0'*E; % Switching to the base which diagonalizes the adjoint equation
%
% Sampling of the right hand member of the adjoint equation
for no_samp=1:Nb_samp
    for ind_mod=1:size(Ind_mod,1)
         ex_ad_1(ind_mod,no_samp)=d1_v_dom(no_samp)*E_m(Ind_mod(ind_mod));
         ex_ad_2(ind_mod,no_samp)=d2_v_dom(no_samp)*E_m(Ind_mod(ind_mod));
    endfor
endfor
%
% Integration of the adjoint equation by convolution
for ind_mod=1:size(Ind_mod,1)
    lambda_2(ind_mod,:)=conv(tab1(ind_mod,:),ex_ad_1(ind_mod,:)')...
                        +conv(tab2(ind_mod,:),ex_ad_2(ind_mod,:)');
endfor
```

Step 11) Back to the original basis and compute the gradient

```
F=zeros(Nb_nodes,1);
if Type_damp ==3
   for no_samp=1:Nb_samp
       X=zeros(Nb_nodes,1);% Displacements
       Lambda=zeros(Nb_nodes,1); % Lagrange multipliers
       for ind_mod=1:size(Ind_mod,1)
           X=X+eig_0(:,Ind_mod(ind_mod))*sol(ind_mod,no_samp);
           % Sampling in the direction of increasing time
           Lambda=Lambda+eig_1(:,Ind_mod(ind_mod))* ...
                        lambda_2(ind_mod,Nb_samp-no_samp+1);
        end
        % Distribution of the torques on 3 nodes at the right end
        F(Nb_nodes)=e1(no_samp)/4;
        F(Nb_nodes-1)=e1(no_samp)/2;
        F(Nb_nodes-2)=e1(no_samp)/4;
        % Idem on the left
        F(1)=-e1(no_samp)/4;
        F(2)=-e1(no_samp)/2;
        F(3)=-e1(no_samp)/4;
        int_Lambda(no_samp)=Lambda'*(DI_M*F-DI_K*X);
    end
else
     for no_samp=1:Nb_samp
         X=zeros(Nb_nodes,1);% Displacements
         V=zeros(Nb_nodes,1);% Velocities
         Lambda=0*eye(Nb_nodes,1);% Lagrange multipliers
         for ind_mod=1:size(Ind_mod,1)
             X=X+eig_0(:,Ind_mod(ind_mod))*sol(ind_mod,no_samp);
             V=V+eig_0(:,Ind_mod(ind_mod))*sol_p(ind_mod,no_samp);
             Lambda=Lambda+eig_1(:,Ind_mod(ind_mod))* ...
                          lambda_2(ind_mod,Nb_samp-no_samp+1);
         endfor
         % At the right end, distribution of torques on 3 nodes
         F(Nb_nodes)=e1(no_samp)/4;
         F(Nb_nodes-1)=e1(no_samp)/2;
         F(Nb_nodes-2)=e1(no_samp)/4;
         % idem on the left
         F(1)=-e1(no_samp)/4;
         F(2)=-e1(no_samp)/2;
         F(3)=-e1(no_samp)/4;
         int_Lambda(no_samp)=Lambda'*(DI_M*F-DI_K*X-DI_W*V);
      endfor
endif
%
% Numerical integration
diff_dom=trapz(time(1:Nb_samp),int_Lambda)
%
% Now use your favorite optimizer
% to find the optimal inertia!
%
```

BIBLIOGRAPHY

[1] ALLEN E.J., BAGLAMA J., BOYD S.K. (2000), *Numerical approximation of the product of the square root of a matrix with a vector*, Linear Algebra and its Applications, vol. 310, Issues 1–3, 1 May 2000, p 167-181.

[2] BELBAS S.A., MAYERGOYZ I.D. (2002), *Optimal control of dynamical systems with Preisach hysteresis*, Int. Journ. of Non-Linear Mechanics, vol.37 p1351 - 1361

[3] BONNANS J.F. (1994), *Local analysis of Newton-type methods for variational inequalities and nonlinear programming*, Applied Mathematics and Optimization, vol. 29, Issue 2, March 1994, p. 161-186.

[4] BONNANS J.F., GILBERT J., LEMARECHAL C., SAGASTIZABAL C.A. (2006), *Numerical Optimization, Theoretical and Practical Aspects, Second Edition*, ISBN:3-540-35445-X Springer-Verlag.

[5] BREZIS H. (1973), *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*, North-Holland Mathematics Studies (Vol. 5).

[6] BREZIS H. (2005), *Analyse fonctionnelle - Théorie et applications*, Dunod.

[7] BROKATE M., DREBLER K., KREJCI P. (1996), *Rainflow Counting and Energy Dissipation for Hysteresis Models in Elastoplasticity*, European journal of mechanics A. Solids 15.4 : p705-737.

[8] CEA J. (1978) *Lectures on Optimization – Theory and Algorithms*, Tata Institute of Fundamental Research, Bombay. ISBN 3-540-08850-4 Springer-Verlag Berlin, Heidelberg. New York

[9] CIARLET P.G. (1982) *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson

[10] CLARKE F.H. (1983), *Optimization and Nonsmooth Analysis*, Wiley-Interscience Publication.

[11] CROUZIEX M., MIGNOT A.L. (1997), *Analyse numérique des équations différentielles*, Masson.

[12] DUNFORD N., SCHWARTZ J. T. (1988) *Linear Operators, Part 1: General Theory*, Wiley, ISBN 978 – 0 – 471 – 60848 – 6.

[13] DUNFORD N., SCHWARTZ J. T. (1988) *Linear Operators, Part 2: Spectral Theory, Self Adjoint Operators in Hilbert Space*, Wiley, ISBN: 978 – 0 – 471 – 60847 – 9.

[14] GERADIN M., RIXEN D.J. (2015), *Mechanical Vibrations: Theory and Application to Structural Dynamics, 3rd Edition*, Wiley, ISBN: 978-1-118-90020-8.

[15] HENRI J. (1982), *Fast Fourier Transform and Convolution*, Springer Series in Information Sciences, ISBN: 978 – 3 – 642 – 81897 – 4.

[16] HIGHAM N.J. (1986), *Newton's Method for the Matrix Square Root*, Mathematics of Computation, Vol. 46, Numb. 174 April 1986 174 p 537-549

[17] IZMAILOV A.F., SOLODOV M.V. (2010), *Newton-Type Methods for Optimization and Variational Problems*, Springer, ISBN: 978-3-319-04246-6.

[18] JULISSON S. (2016) *Optimisation de formes de coques minces pour des géométries complexes*, Thèse Paris Saclay.

[19] KRASNOSEL'SKII M.A., POKORVSKII A.V. (1983) *Systems with hysteresis (translation from the Russian edition)*, Springer-Verlag 1989.

[20] KREJCI P. (1991), *Hysteresis memory preserving operators*, Applications of Mathematics, (Vol. 36, No. 4), p305-326.

[21] LALANNE C. (2002) *Mechanical Vibration and Shock, Fatigue Damage (Vol 4)*, Wiley.

[22] LEMAITRE J., CHABOCHE JL., BENALLAL A., DESMORAT R. (2009), *Mécanique des matériaux solides - 3ème édition*, Dunod.

[23] LEYFFER S., MAHAJAN A. (2010) *Nonlinear Constrained Optimization: Methods and Software*, Mathematics and Computer Science Division, Preprint ANL/MCS-P1729-0310.

[24] LIU Y.F., LI J., ZHANG Z.M., HU X.H., ZHANG W.J. (2015) *Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system*, Mech. Sci., 6, 15–28, 2015.

[25] LOURKIS I. A. (2005) *A Brief Description of the Levenberg-Marquardt Algorithm Implemened by levmar*, Institute of Computer Science, Foundation for Research and Technology - Hellas(FORTH), Vassilika Vouton, P.O. Box 1385, GR 711 10, Heraklion, Crete, GREECE.

[26] MATSISHI M., ENDO T. (1968), *Fatigue of metals subjected to varying stress*, Japan Soc Mech Engineering.

[27] MAYERGOYZ I.D. (2003), *Mathematical Models of Hysteresis and Their Applications*, Elsevier Academic Press an imprint of Elsevier.

[28] MOREAU JJ. (1966), *Quadratic Programming in Mechanics: Dynamics of One-Sided Constraints*, SIAM Journal on Control and Optimization, Society for Industrial and Applied Mathematics, 1966, 4 (1), pp.153 - 158. 10.1137/0304014. hal-01379713

[29] MOREAU JJ. (1983), *Standard inelastic shocks and the dynamics of unilateral constraints*, Unilateral Problems in Structural Analysis, 1983, 9783211818596. 10.1007/978-3-7091-2632-5-9. hal-01544442

[30] MORI M. (1974) *Approximation of Exponential Function of a Matrix by Continued Fraction Expansion*, Publ. RIMS, Kyoto Univ. (vol. 10), p 257-269

[31] de NAZELLE P. (2013) *Paramétrage de formes surfaciques pour l'optimisation*, Thèse Ecole centrale de Lyon.

[32] RABBE P., LIEURADE H.P., GALTIER A. (2000), *Essais de fatigue- Partie I*, Techniques de l'Ingénieur.

[33] ROSHANFAR M., SALIMI M.H. (2015), *Comparing of methods of cycle calculating and counting to the rain flow method*, EJAS Journal-2015-3-7, 291-296.

[34] SURESH S. (1998), *Fatigue of materials*, Cambridge University Press, 2nd Edition.

[35] THOM R. (1993), *Prédire n'est pas expliquer*, Flammarion

[36] TIKRI B., NADJITONON N., ROBERT J.L. (2009) *Nouvelle loi non linéaire d'endommagement par fatigue basée sur la courbe de Bastenaire*, Département Génie Mécanique et Productique, Laboratoire de Mécanique et Ingénieries (LaMI), IUT de Montluçon.

[37] SCHWARTZ L. (1967) *Cours d'Analyse Ecole Polytechnique* (tome 1), Hermann, Paris, 1967

[38] VARGA R. S.(1962) *Matrix iterative analysis*, Prentice-Hall, 1962.

[39] RUDIN W. (1997) *Analyse réelle et complexe* Masson

[40] VISINTIN A. (1993-2001), *Differential Models of Hysteresis*, Springer, Applied Mathemetical Sciences (Vol 11).

[41] WANG L. (2009), *Regulation of Hysteretic Systems with Preisach Representation*; Thesis in Electrical and Computer Engineering, University of Waterloo.

[42] ZAKERRZADEH M.R., SAYYAADI H., VAZIRI ZANJANI M. A. (2011), *Characterizing Hysteresis Nonlinearity Behavior of SMA Actuators by Krasnosel'skii-Pokrovskii Model*, Scientific & Academic Publishing, Applied Mathematics; (vol. 1) p 28-38.

[43] ZETTI A. (2005), *Sturm-Liouville Theory*, Mathematical Surveys and Monographs (n^o121),American Mathematical Soc., 2005.