High modes smoothing of time schemes for nonlinear parabolic PDEs

Matthieu Brachet, Jean-Paul Chehab

▶ **To cite this version:**

Matthieu Brachet, Jean-Paul Chehab. High modes smoothing of time schemes for nonlinear parabolic PDEs. 2024. hal-04700641

HAL Id: hal-04700641
https://hal.science/hal-04700641v1

Preprint submitted on 17 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High modes smoothing of time schemes for nonlinear parabolic PDEs

September 17, 2024

Matthieu Brachet[1] and Jean-Paul Chehab[2]
[1]Laboratoire de Mathématiques et Applications, UMR CNRS 7348
Université de Poitiers, Boulevard Marie et Pierre Curie - Téléport 2
86962 Chasseneuil, Futuroscope Cedex, France
matthieu.brachet@math.univ-poitiers.fr
[2]Laboratoire LAMFA (UMR CNRS 7352), Université de Picardie Jules Verne
33 rue Saint Leu, 80039 Amiens Cédex, France
Jean-Paul.Chehab@u-picardie.fr

Abstract

We here introduce a stabilization process applied to IMEX time schemes for the simulation of nonlinear parabolic PDEs. The new schemes we derived use a decomposition of the signal into low and high frequency components (to which different time schemes are applied) together with a proper damping of the high mode components. This approach allows to design new methods, with enhanced stability and limited perturbation of the constituency. We display numerical analysis of the schemes on the accuracy and the stability. The numerical illustrations we give concern Allen-Cahn and Swift-Hohenberg equations, and show the efficiency of our methods. The effect of the present number of frequencies, and of the stabilization is evaluated on both the energy decay and the dynamics of the model.

Keywords: Parabolic PDEs, stabilization, scales separations, Allen-Cahn equations, Swift-Hohenberg equations.

AMS subject classifications: 35K57, 65M12, 65M06, 65F15.

Introduction

The long time simulation of nonlinear parabolic equation is an important issue: these equations are often used to model natural phenomena, for instance in mathematical biology [24], in image processing [6, 20], in thermodynamic science [27, 28], or in material science [2, 11], just to give few examples. The classical numerical schemes that are implemented for the simulation of these equations must balance good stability and reasonable computational cost. These are antagonist constraints: indeed fully implicit schemes are often inconditionally stable but are

computationally expensive since they need to solve at each step a nonlinear system of equations; the fully explicit or semi-implicit iterations (such as IMEX) are fastly solved but can suffer from hard times step limitation. The limitation is governed by extreme situations related to the speed of propagation of the high frequency components of the solution.

As an illustration, consider Gradient flows such as Allen Cahn's systems:

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathcal{A}u + f(u) &= 0, \\ u_0(\boldsymbol{x}) &= u(\boldsymbol{x}, 0), \end{aligned} \tag{1}$$

where function f corresponds to the non linear term and F is its anti-derivative. The linear part is given by the elliptic operator $\mathcal{A}$. Equation (1) is considered with the space variable $\boldsymbol{x} \in \Omega$, where Ω is a regular subset of $\mathbb{R}^d$ ($d \in \{1, 2, 3\}$), and $t \geq 0$ is the time variable.
Due to non-linearity, this equation does not have any analytical solution. Moreover, it is challenging to solve it numerically keeping some desirable properties as, e.g., the decrease of the energy:

$$\mathcal{E}(t) = \int_\Omega \frac{1}{2} u(\boldsymbol{x}, t)\mathcal{A}u(\boldsymbol{x}, t) + F(u(\boldsymbol{x}, t))d\boldsymbol{x}. \tag{2}$$

This property is called energy stability or simply stability in this paper. We discretize the system in space (with finite element, finite volume or finite difference method),, and obtain

$$\begin{aligned} M\frac{du}{dt} + Au + f(u) &= 0 \\ u(0) &= u_0 \in \mathbb{R}^N, \end{aligned} \tag{3}$$

where A is the stiffness matrix, commonly symmetric and positive definite, and M is the mass matrix ; $M = \mathrm{Id}$ when finite differences are used, this will be the case in this paper. A classical time scheme consists to use an implicit method on the linear part and an explicit method on f:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k+1)} + f(u^{(k)}) = 0.$$

This corresponds to the IMEX methods class which is constraint by a stability condition

$$0 < \Delta t < \frac{2}{\|f'\|_\infty}. \tag{4}$$

It can be relaxed using a stabilization processes [6, 25, 26] by replacing $Au^{(k+1)}$ by $Au^{(k)} + \tau(u^{(k+1)} - u^{(k)})$. This allows to obtain an unconditionally stable integrator when the stabilization parameter $\tau > 0$ is large enough. However, when τ becomes large the dynamic is slowed down [5]. Other IMEX stable schemes, based on a total different approach, have been studied, see [29], and the references therein.

The parabolic regularization ensures the regularity of the solutions, or equivalently the fast convergence of its Fourier serie in a proper Hilbert space, usually based on the eigenfunctions of $\mathcal{A}$; the Fourier coefficients, that correspond to the high frequency components of the solution are then expected to be of small magnitude.

We propose then to apply the following compromise: stabilize only the high mode components by the above relaxation, this allows to stabilize the scheme without deteriorating its consistency. Notice that this type of approach has been considered, in specific cases in [1] in Finite Elements, however focusing here on the spectral case, we can display stability and error analysis in order to obtain explicit conditions.

The paper is organized as follow: first we establish the basic ideas on a linear equation, and displaying error analysis, we can quantify a compromise between the stability of the scheme and its accuracy. In the second part, to avoid to compute all the eigenvectors, we introduce an algorithm that uses only a low frequencies decomposition. The error is analyzed and the first experiments are done. Algorithm for non-linear parabolic equation is studied in the third section. We prove that energy decreases when the stabilization parameter is large enough and error estimates are given. Numerical experiments are done for Allen-Cahn and Swift-Hohenberg equations. Particularly, we analyze the energy decreasing and the pattern dynamics. The results are compared with other Convex-Splitting and stabilized schemes. Finally, the last part is devoted to the concluding remarks.

1 Basic stabilization strategies

1.1 The linear case

Consider, for simplicity, the following linear differential system

$$\begin{aligned} \frac{du}{dt} + Au &= 0, \\ u(0) &= u_0 \in \mathbb{R}^N, \end{aligned} \tag{5}$$

where $A \in \mathcal{M}_N(\mathbb{R})$ is a Symmetric Positive matrix. Typically A can be built as a discretisation matrix of an elliptic operator. Given the time step $\Delta t > 0$, the most simple schemes to generate a sequence of time approximation $u^{(k)}$ to the solution u at times $t^{(k)} = k\Delta t$, $k \in \mathbb{N}$, are the Euler's schemes.

- Forward Euler's. The sequence $u^{(k)} \approx u(t^{(k)}) \in \mathbb{R}^N$ is defined by induction as:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k)} = 0, \tag{6}$$

  say $u^{(k+1)} = u^{(k)} - \Delta t A u^{(k)} = (\mathrm{Id}_N - \Delta t A)u^{(k)}$, Id_N being the $N \times N$ identity matrix. $u^{(k+1)}$ is given directly from $u^{(k)}$. This scheme is fast but suffers from a hard time step restriction to be stable:

$$0 < \Delta t < \frac{2}{\rho(A)},$$

  where $\rho(A)$ is the spectral radius of A.

- Backward Euler's. The sequence $u^{(k)} \approx u(t^{(k)}) \in \mathbb{R}^N$ s defined by induction as:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k+1)} = 0, \tag{7}$$

say $u^{(k+1)} = (\mathrm{Id}_N + \Delta t A)^{-1} u^{(k)}$.
Vector $u^{(k+1)}$ is then built as the solution of a linear system. Conversely to Forward Euler's this scheme can be costly (depending on the cost of the numerical solution of the linear system), however it is unconditionally stable since $\rho((\mathrm{Id}_N + \Delta t A)^{-1}) < 1$ for all $\Delta t > 0$. This condition is automatically satisfied because A is symmetric and positive.

A simple idea to build a fast and stable scheme is to stabilize Forward Euler's [4]. For a given tuning parameter $\tau > 0$, we consider the stabilized Forward Euler's scheme:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + \tau(u^{(k+1)} - u^{(k)}) + Au^{(k)} = 0. \tag{8}$$

The iteration matrix is then

$$\mathrm{Id}_N - \frac{\Delta t}{1 + \tau \Delta t} A.$$

We can establish the following stability property.

Proposition 1.1 *Scheme* (8) *is stable under one of the following conditions:*

- *if* $\tau \geq \dfrac{\rho(A)}{2}$, *the scheme is unconditionally stable;*
- *if* $\tau < \dfrac{\rho(A)}{2}$, *the scheme is stable under the condition*

$$0 < \Delta t < \frac{2}{\rho(A) - 2\tau}.$$

Proof. The eigenvalues of the iteration matrix are the numbers

$$\mu = 1 - \frac{\Delta t \lambda}{1 + \tau \Delta t},$$

where λ is an eigenvalue of A. The stability condition is $|\mu| < 1$, so

$$0 < \Delta t \text{ and } (\lambda - 2\tau)\Delta t < 2,$$

hence the result. ■

A generalization of the stabilized Euler's method can be obtained following the RSS scheme (Residual Smoothing Scheme [5]): it consists to approach $Au^{(k+1)}$ by $Au^{(k)} + B_\tau(u^{(k+1)} - u^{(k)})$ where B_τ is a pre-conditioner of A; one recovers backward Euler's scheme when $B_\tau = A$ and Forward Euler's one for $B_\tau = 0$; stabilized Euler's is obtained for $B_\tau = \tau \mathrm{Id}_N$. The scheme writes as

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + B_\tau(u^{(k+1)} - u^{(k)}) + Au^{(k)} = 0. \tag{9}$$

The stability conditions are given in the following proposition when $B_\tau = \tau B$.

Proposition 1.2 *Let $B \in \mathcal{M}_N(\mathbb{R})$, be a symmetric positive matrix such that there exist two strictly positive numbers α and β such that*

$$\alpha < Bu, u > \leq < Au, u > \leq < \beta < Bu, u > .$$

Then the scheme (9) with the choice $B_\tau = \tau B$, is stable under one of the following conditions:

- *if $\tau \geq \dfrac{\beta}{2}$ (9) is unconditionally stable;*
- *if $\tau < \dfrac{\beta}{2}$, (9) is stable if*

$$0 < \Delta t < \frac{2}{\beta - 2\tau}.$$

Proof. See [5]. ■

The RSS scheme allows to compute fastly the iterations as respected to Backward Euler's since the linear system to be solved can be simpler: indeed, in a number of situations fast solvers as well as the efficient methods of the sparse matrix linear algebra can be used, see [5, 6]. However, the key point of RSS is the choice of the preconditioning matrix B and of the tuning stabilization parameter τ, the stabilization matrix being defined as $B_\tau = \tau B$. In an ideal situation, the stabilization should only act on the high mode components of the solution: their speed of propagation determines the time step restriction. Also a strong stabilization of the low mode components of the solution can slow down the dynamics, see [1]. When considering two finite difference discretisations A and B of the same operator, typically $-\Delta$, the lower eigenvalues of B are very close to those of A while the high ones are underestimated as respected to the A's ones, see [19]. So, when $\tau \approx 1$, τB stabilizes the scheme without deteriorating the consistency. To this end we rewrite hereafter the schemes using frequency decomposition.

1.2 Frequency decomposition based method

The matrix $A \in \mathcal{M}_N(\mathbb{R})$ is symmetric positive then diagonalisable in a orthonormalized eigenvector basis, $(w_i)_{i=1}^N$. The associated eigenvalues $(\lambda_i)_{i=1}^N$ are ranged in the increasing order,

$$0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N = \rho(A).$$

Let P_m be the rectangular matrix $N \times m$ defined by $P_m = [w_1, \cdots, w_m]$, and Q_m be the $N \times (N - m)$ defined by $Q_m = [w_{m+1}, \cdots, w_N]$. We recall the following relations and we introduce some notations:

- $\Lambda_m := P_m^T A P_m = \mathrm{diag}(\lambda_1, \cdots, \lambda_m) \in \mathcal{M}_m(\mathbb{R})$ and $Q_m^T A Q_m = \mathrm{diag}(\lambda_{m+1}, \cdots, \lambda_N) \in \mathcal{M}_{N-m}(\mathbb{R})$,
- $P_m P_m^T \in \mathcal{M}_N(\mathbb{R})$ is an orthogonal projector onto $V_m = \mathrm{span}(w_1, \cdots, w_m)$. In the same idea, $Q_m Q_m^T = \mathrm{Id}_N - P_m P_m^T$ is an orthogonal projector onto $V_m^\perp = \mathrm{span}(w_{m+1}, \cdots, w_N)$,
- For any $u \in \mathbb{R}^N$ we write

$$u = y + z = P_m P_m^T u + (\mathrm{Id} - P_m P_m^T)u,$$

and we define $\hat{y} = P_m^T u \in \mathbb{R}^m$ and $\hat{z} = Q_m^T u \in \mathbb{R}^{N-m}$.

The coefficients of the vector $\hat{y}$ are the low-frequencies spectral coefficients of u while those of $\hat{z}$ are the high-frequencies spectral coefficients of u.

1.3 Two-level stabilized schemes

To stabilize the scheme by damping the high mode components only, we use the spectral eigen-decomposition and treat each block of component of u, low-frequencies, then high-frequencies, spectral coefficients with a different time scheme. At first, we stabilize the Forward Euler scheme by damping the high mode components.

Algorithm 1 : Explicit-stabilized

1: **for** $k = 0, 1, \cdots$ **do**
2: **Set** $\hat{y}^{(k)} = P_m^T u^{(k)}$ and $\hat{z}^{(k)} = Q_m^T u^{(k)}$.
3: **Set** $\hat{y}^{(k+1)} = (\mathrm{Id}_m - \Delta t P_m^T A P_m)\hat{y}^{(k)}$
4: **Solve** $\hat{z}^{(k+1)} = (\mathrm{Id}_{N-m} - \dfrac{\Delta t}{1+\tau\Delta t} Q_m^T A Q_m)\hat{z}^{(k)}$
5: **Set** $u^{(k+1)} = P_m \hat{y}^{(k+1)} + Q_m \hat{z}^{(k+1)}$
6: **end for**

This approach has been considered in [8] in the spectral case (Fourier, Chebyshev).

Proposition 1.3 *Scheme related to Algorithm 1 is stable under one of the following conditions:*

- *if* $\tau \geq \dfrac{\rho(A)}{2}$ *then the scheme is stable if*

$$0 < \Delta t < \frac{2}{\lambda_m};$$

- *if* $\tau < \dfrac{\rho(A)}{2}$ *then scheme is stable when*

$$0 < \Delta t < \min\left(\frac{2}{\lambda_m}, \frac{2}{\rho(A) - 2\tau}\right).$$

Proof. The results follows from a simple computation. ■

A second possibility consists in applying the Backward Euler's scheme to the low-mode components $\hat{y}$ (leading to the solution of a low dimensional linear system) and to stabilize the $\hat{z}$ as above:

Algorithm 2 :Implicit-stabilized

1: **for** $k = 0, 1, \cdots$ **do**
2: **Set** $\hat{y}^{(k)} = P_m^T u^{(k)}$ and $\hat{z}^{(k)} = Q_m^T u^{(k)}$.
3: **Solve** $(\mathrm{Id}_m + \Delta t P_m^T A P_m)\hat{y}^{(k+1)} = \hat{y}^{(k)}$
4: **Solve** $z^{(k+1)} = (\mathrm{Id}_{N-m} - \dfrac{\Delta t}{1+\tau\Delta t} Q_m^T A Q_m) z^{(k)}$
5: **Set** $u^{(k+1)} = P_m \hat{y}^{(k+1)} + Q_m \hat{z}^{(k+1)}$
6: **end for**

This was considered in [1] with a multi-grid approach.

Proposition 1.4 *Scheme related to algorithm 2 is stable under one of the following conditions:*

- *if* $\tau \geq \frac{\rho(A)}{2}$ *then scheme is unconditionally stable;*
- *if* $\tau < \frac{\rho(A)}{2}$ *then it is stable under condition*

$$0 < \Delta t < \frac{2}{\rho(A) - 2\tau}.$$

Proof. The results come from direct computations. ■

We now give a matrix interpretation which enables us to write the scheme into a RSS framework. The Explicit-stabilized scheme can be rewritten as

$$\begin{pmatrix} \hat{y}^{k+1} - \hat{y}^k \\ \hat{z}^{k+1} - \hat{z}^k \end{pmatrix} + \tau\Delta t \begin{pmatrix} 0 & 0 \\ 0 & \mathrm{Id}_{N-m} \end{pmatrix} \begin{pmatrix} \hat{y}^{k+1} - \hat{y}^k \\ \hat{z}^{k+1} - \hat{z}^k \end{pmatrix} = -\Delta t \begin{pmatrix} P_m^T A P_m & 0 \\ 0 & Q_m^T A Q_m \end{pmatrix} \begin{pmatrix} \hat{y}^k \\ \hat{z}^k \end{pmatrix}$$

It is a RSS scheme with stabilization matrix: $\begin{pmatrix} 0 & 0 \\ 0 & \mathrm{Id}_{N-m} \end{pmatrix}$ in A eigenvector basis. Indeed, it writes in to the canonical basis as:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + B_\tau(u^{(k+1)} - u^{(k)}) + Au^{(k)} = 0,$$

where $B_\tau = \begin{pmatrix} P_m & Q_m \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & \tau\mathrm{Id}_{N-m} \end{pmatrix} \begin{pmatrix} P_m^T \\ Q_m^T \end{pmatrix}$ is the stabilization matrix in the nodal basis.
Scheme Implicit-stabilized writes as:

$$\begin{pmatrix} \hat{y}^{k+1} - \hat{y}^k \\ \hat{z}^{k+1} - \hat{z}^k \end{pmatrix} + \Delta t \begin{pmatrix} P_m^T A P_m & 0 \\ 0 & \tau\mathrm{Id}_{N-m} \end{pmatrix} \begin{pmatrix} \hat{y}^{k+1} - \hat{y}^k \\ \hat{z}^{k+1} - \hat{z}^k \end{pmatrix} = -\Delta t \begin{pmatrix} P_m^T A P_m & 0 \\ 0 & Q_m^T A Q_m \end{pmatrix} \begin{pmatrix} \hat{y}^k \\ \hat{z}^k \end{pmatrix}.$$

Proceeding as above, we identify this scheme to a RSS one with the stabilization matrix $\begin{pmatrix} P_m^T A P_m & 0 \\ 0 & \tau\mathrm{Id}_{N-m} \end{pmatrix}$ in A eigenvector basis and $B_\tau = \begin{pmatrix} P_m & Q_m \end{pmatrix} \begin{pmatrix} P_m^T A P_m & 0 \\ 0 & \tau\mathrm{Id}_{N-m} \end{pmatrix} \begin{pmatrix} P_m^T \\ Q_m^T \end{pmatrix}$ in the nodal basis.

We here applied the frequency splitting to a first order scheme Implicit/Explicit scheme, it can be of course applied to higher order ones of BDF type. In the stabilization processes introduced here, it is necessary to compute all the eigen-elements of A. This is costly. To avoid to compute it, we consider a variant which require the computation of only low frequencies basis P_m. It can be done thanks to the algorithm detailed in the Appendix. 5.1.

2 High mode stabilization for linear equation

2.1 High mode stabilization strategy

In a previous section, we introduced two stabilized schemes to solve (5). Each of them assumed all the eigen-elements of A are available. We adapt the same approach but by assuming that

only the first m eigen-elements are available. The matrices P_m and Λ_m are known. The first algorithm is deduced by assuming the linear part is considered explicitly for high modes. We add a forcing term $f : t \mapsto f(t) \in \mathbb{R}^N$ to simulate the non-linear part. Thus, the equation to solve becomes

$$\frac{du}{dt} + Au = f.$$

We consider now the following algorithm, which corresponds to an Euler scheme with high-modes stabilization.

Algorithm 3 :Implicit-explicit/stabilized

1: **Set** $\hat{y}^{(0)} = P_m^T u^{(0)}$
2: **for** $k = 0, 1, \cdots$ **do**
3: **Solve** $\dfrac{\hat{y}^{(k+1)} - \hat{y}^{(k)}}{\Delta t} + \Lambda_m \hat{y}^{(k+1)} = \hat{f}^{(k+1)}$
4: **Set** $\hat{y}^{(k+1)} = P^T P_m \hat{y}^{(k+1)}$
5: **Solve** $\dfrac{\hat{u}^{(k+1)} - \hat{u}^{(k)}}{\Delta t} + \tau(\hat{u}^{(k+1)} - \hat{u}^{(k)}) = \tau(\hat{y}^{(k+1)} - \hat{y}^{(k)}) - \Lambda \hat{u}^{(k)} + \hat{f}^{(k+1)}$
6: **end for**

If the linear part is considered implicitly, the scheme becomes:

Algorithm 4 :Implicit-Implicit/stabilized

1: **Set** $\hat{y}^{(0)} = P_m^T u^{(0)}$
2: **for** $k = 0, 1, \cdots$ **do**
3: **Solve** $\dfrac{\hat{y}^{(k+1)} - \hat{y}^{(k)}}{\Delta t} + \Lambda_m \hat{y}^{(k+1)} = \hat{f}^{(k+1)}$
4: **Set** $\hat{y}^{(k+1)} = P^T P_m \hat{y}^{(k+1)}$
5: **Solve** $\dfrac{\hat{u}^{(k+1)} - \hat{u}^{(k)}}{\Delta t} + \tau(\hat{u}^{(k+1)} - \hat{u}^{(k)}) + \Lambda \hat{u}^{(k+1)} = \tau(\hat{y}^{(k+1)} - \hat{y}^{(k)}) + \hat{f}^{(k+1)}$
6: **end for**

For the two algorithms, the line 5 corresponds to a stabilization process acting only on high frequencies components. The term $\tau(\hat{y}^{(k+1)} - \hat{y}^{(k)})$ excludes low frequency components.

2.2 Error estimates

We now analyze the effect of the new stabilization strategy, and we compare each scheme to the Backward Euler's scheme written in Fourier basis:

Algorithm 5 : Backward Euler's

1: **Set** $\hat{v}^{(0)} = P^T u^{(0)}$
2: **for** $k = 0, 1, \cdots$ **do**
3: **Solve** $\dfrac{\hat{v}^{(k+1)} - \hat{v}^{(k)}}{\Delta t} + \Lambda \hat{v}^{(k+1)} = \hat{f}^{(k+1)}$
4: **end for**

where $P = [w_1, \cdots, w_N]$ is a $N \times N$ matrix and $P^T AP = \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_N)$.
The sequence generated by algorithm 5 satisfied the following lemma.

Lemma 2.1 *The sequence $(\hat{v}^{(k)})$ generated by Backward Euler's algorithm 5 satisfies*

$$\hat{v}_j^{(k)} = \frac{1}{(1+\Delta t\lambda_j)^k}\hat{v}_j^{(0)} + \Delta t \sum_{i=1}^{k} \frac{1}{(1+\Delta t\lambda_j)^{k+1-i}} \hat{f}_j^{(i)},$$

and

$$\begin{aligned} \hat{v}_j^{(k+1)} - \hat{v}_j^{(k)} &= -\Delta t\lambda_j \hat{v}_j^{(k+1)} + \Delta t \hat{f}_j^{(k+1)}, \\ &= -\frac{\Delta t\lambda_j}{(1+\Delta t\lambda_j)^{k+1}}\hat{v}_j^{(0)} + \Delta t\hat{f}_j^{(k+1)} - \frac{\lambda_j \Delta t^2}{1+\lambda_j\Delta t}\sum_{i=0}^{k} \frac{1}{(1+\lambda_j\Delta t)^i}\hat{f}_j^{(k+1-i)}. \end{aligned}$$

Proof. It follows from a simple and classical computation. ■

The difference between one of the stabilization scheme and the Backward Euler's scheme is given by the following results.

Proposition 2.2 *Assume $v^{(0)} = u^{(0)}$. Let $\hat{e}^{(k)} = \hat{u}^{(k)} - \hat{v}^{(k)}$ be the error between a stabilized scheme and the Backward Euler's scheme in the eigenvector basis of A and $\hat{e}_j^{(k)}$ the j-th value.*

- *If $(\hat{u}^{(k)})$ is generated by Algorithm 3*

$$\hat{e}_j^{(k)} = \begin{cases} \dfrac{\Delta t\lambda_j}{1+\Delta t\tau}\displaystyle\sum_{i=0}^{k-1}\left(\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\right)^{k-i-1}(\hat{v}_j^{(i+1)} - \hat{v}_j^{(i)}) & \text{if } j \le m, \\ \dfrac{\Delta t(\lambda_j-\tau)}{1+\Delta t\tau}\displaystyle\sum_{i=0}^{k-1}\left(\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\right)^{k-i-1}(\hat{v}_j^{(i+1)} - \hat{v}_j^{(i)}) & \text{else.} \end{cases}$$

- *If $(\hat{u}^{(k)})$ is generated by Algorithm 4*

$$\hat{e}_j^{(k)} = \begin{cases} 0 & \text{if } j \le m, \\ \dfrac{\tau\Delta t}{1+\Delta t(\tau+\lambda_j)}\displaystyle\sum_{i=0}^{k-1}\left(\frac{1+\tau\Delta t}{1+\Delta t(\tau+\lambda_j)}\right)^{k-i-1}\left(\hat{v}_j^{(i+1)} - \hat{v}_j^{(i)}\right) & \text{else.} \end{cases}$$

Proof.

- We first establish the result for Implicit-Explicit/stabilized Algorithm 3. Considering the scheme in the eigenvector basis of A, we have, for all $k \ge 0$,

$$\begin{aligned} \frac{\hat{u}^{(k+1)} - \hat{u}^{(k)}}{\Delta t} + \tau(\hat{u}^{(k+1)} - \hat{u}^{(k)}) + \Lambda\hat{u}^{(k)} &= \tau(\hat{y}^{(k+1)} - \hat{y}^{(k)}) + \hat{f}^{(k+1)}, \\ \frac{\hat{v}^{(k+1)} - \hat{v}^{(k)}}{\Delta t} + \tau(\hat{v}^{(k+1)} - \hat{v}^{(k)}) + \Lambda\hat{v}^{(k)} + \Lambda(\hat{v}^{(k+1)} - \hat{v}^{(k)}) &= \tau(\hat{v}^{(k+1)} - \hat{v}^{(k)}) + \hat{f}^{(k+1)}. \end{aligned}$$

Hence, taking the difference, we obtain

$$\frac{\hat{e}^{(k+1)}-\hat{e}^{(k)}}{\Delta t}+\tau(\hat{e}^{(k+1)}-\hat{e}^{(k)})+\Lambda\hat{e}^{(k)}=\tau\left((\hat{y}^{(k+1)}-\hat{y}^{(k)})-(\hat{v}^{(k+1)}-\hat{v}^{(k)})\right)+\Lambda(\hat{v}^{(k+1)}-\hat{v}^{(k)}).$$

For each component $j=1,\cdots,N$, we get

$$\frac{\hat{e}_j^{(k+1)}-\hat{e}_j^{(k)}}{\Delta t}+\tau(\hat{e}_j^{(k+1)}-\hat{e}_j^{(k)})+\lambda_j\hat{e}_j^{(k)}=\tau\left((\hat{y}_j^{(k+1)}-\hat{y}_j^{(k)})-(\hat{v}_j^{(k+1)}-\hat{v}_j^{(k)})\right)+\lambda_j(\hat{v}_j^{(k+1)}-\hat{v}_j^{(k)}).$$

For all $k\geq 0$, we remark $\hat{y}_j^{(k)}=\begin{cases}\hat{v}_j^{(k)} & j=1,\cdots,m,\\ 0 & j=m+1,\cdots,N.\end{cases}$

Thus, if $j\leq m$, we obtain

$$\hat{e}_j^{(k+1)}=\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\hat{e}_j^{(k)}+\frac{\Delta t\lambda_j}{1+\Delta t\tau}(\hat{v}_j^{(k+1)}-\hat{v}_j^{(k)}).$$

By induction, we deduce

$$\hat{e}_j^{(k)}=\left(\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\right)^k\hat{e}^{(0)}+\frac{\Delta t\lambda_j}{1+\Delta t\tau}\sum_{i=0}^{k-1}\left(\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\right)^{k-i-1}(\hat{v}_j^{(i+1)}-\hat{v}_j^{(i)}).$$

By construction $\hat{e}_j^{(0)}=0$ and we have

$$\hat{e}_j^{(k)}=\frac{\Delta t\lambda_j}{1+\Delta t\tau}\sum_{i=0}^{k-1}\left(\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\right)^{k-i-1}(\hat{v}_j^{(i+1)}-\hat{v}_j^{(i)})$$

for all integer $k\geq 0$.
Now, if $j\geq m+1$ we have $\hat{y}_j^{(k)}=0$ and

$$\hat{e}_j^{(k+1)}=\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\hat{e}_j^{(k)}+\frac{\Delta t(\lambda_j-\tau)}{1+\Delta t\tau}(\hat{v}_j^{(k+1)}-\hat{v}_j^{(k)}).$$

Knowing $\hat{e}_j^{(0)}=0$, we deduce for all $k\geq 0$

$$\hat{e}_j^{(k)}=\frac{\Delta t(\lambda_j-\tau)}{1+\Delta t\tau}\sum_{i=0}^{k-1}\left(\frac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}\right)^{k-i-1}(\hat{v}_j^{(i+1)}-\hat{v}_j^{(i)}).$$

- The second expression is derived in a similar way. We start from

$$\frac{\hat{u}^{(k+1)}-\hat{u}^{(k)}}{\Delta t}+\tau(\hat{u}^{(k+1)}-\hat{u}^{(k)})+\Lambda\hat{u}^{(k+1)}=\tau(\hat{y}^{(k+1)}-\hat{y}^{(k)})+\hat{f}^{(k+1)},$$
$$\frac{\hat{v}^{(k+1)}-\hat{v}^{(k)}}{\Delta t}+\tau(\hat{v}^{(k+1)}-\hat{v}^{(k)})+\Lambda\hat{v}^{(k+1)}=\tau(\hat{v}^{(k+1)}-\hat{v}^{(k)})+\hat{f}^{(k+1)},$$

and we write

$$\frac{\hat{e}^{(k+1)}-\hat{e}^{(k)}}{\Delta t}+\tau(\hat{e}^{(k+1)}-\hat{e}^{(k)})+\Lambda\hat{e}^{(k+1)}=\tau((\hat{y}^{(k+1)}-\hat{y}^{(k)})-(\hat{v}^{(k+1)}-\hat{v}^{(k)})).$$

At each component $j=1,\cdots,N$, we get

$$\frac{\hat{e}_j^{(k+1)}-\hat{e}_j^{(k)}}{\Delta t}+\tau(\hat{e}_j^{(k+1)}-\hat{e}_j^{(k)})+\lambda_j\hat{e}_j^{(k+1)}=\tau((\hat{y}_j^{(k+1)}-\hat{y}_j^{(k)})-(\hat{v}_j^{(k+1)}-\hat{v}_j^{(k)})).$$

As in the previous case, we get

$$\frac{\hat{e}_j^{(k+1)}-\hat{e}_j^{(k)}}{\Delta t}+\tau(\hat{e}_j^{(k+1)}-\hat{e}_j^{(k)})+\lambda_j\hat{e}_j^{(k+1)}=\begin{cases}0 & j=1,\cdots,m,\\ -\tau(\hat{v}^{(k+1)}-\hat{v}^{(k)}) & j=m+1,\cdots,N.\end{cases}$$

Then, for $j=1,\cdots,m$, we have

$$\hat{e}_j^{(k+1)}=\frac{1+\Delta t\tau}{1+\Delta t(\tau+\lambda_j)}\hat{e}_j^{(k)}=\left(\frac{1+\Delta t\tau}{1+\Delta t(\tau+\lambda_j)}\right)^{k+1}\hat{e}_j^{(0)}=0.$$

For $j=m+1,\cdots,N$, we have

$$\hat{e}_j^{(k+1)}=\frac{1+\Delta t\tau}{1+\Delta t(\tau+\lambda_j)}\hat{e}_j^{(k)}-\frac{\tau\Delta t}{1+\Delta t(\tau+\lambda_j)}(\hat{v}_j^{(k+1)}-\hat{v}_j^{(k)}),$$

Since $\hat{e}_j^{(0)}=0$, we deduce, for all $k\geq 0$

$$\hat{e}_j^{(k)}=\frac{\tau\Delta t}{1+\Delta t(\tau+\lambda_j)}\sum_{i=0}^{k-1}\left(\frac{1+\tau\Delta t}{1+\Delta t(\tau+\lambda_j)}\right)^{k-i-1}(\hat{v}_j^{(i+1)}-\hat{v}_j^{(i)}).$$

■

We can now establish the folowing result:

Theorem 2.3 *For all $1\leq j\leq N$, let $\hat{f}_j^{\infty}=\max_{k\geq 0}|\hat{f}_j^{(k)}|$. The following estimates are satisfied:*

- *For Algorithm 3, assuming $\tau\geq\dfrac{\rho(A)}{2}$, we have*

$$\|e^{(k)}\|_2^2\leq 4\Delta t^2\left(\sum_{j=1}^{m}\left(\lambda_j|\hat{v}_j^{(0)}|+\hat{f}_j^{\infty}\right)^2+\sum_{j=m+1}^{N}\left(1-\frac{\tau}{\lambda_j}\right)^2\left(\lambda_j|\hat{v}_j^{(0)}|+\hat{f}_j^{\infty}\right)^2\right),$$

- *For Algorithm 4, we have*

$$\|e^{(k)}\|_2^2\leq 4\tau^2\Delta t^2\sum_{j=m+1}^{N}\left(|\hat{v}_j^{(0)}|+\frac{\hat{f}_j^{\infty}}{\lambda_j}\right)^2.$$

Proof. In each case, the following equality is useful:

$$\|e^{(k)}\|_2^2 = \frac{1}{N}\sum_{j=1}^{N}|\hat{e}_j^{(k)}|^2 = \frac{1}{N}\left(\sum_{j=1}^{m}|\hat{e}_j^{(k)}|^2 + \sum_{j=m+1}^{N}|\hat{e}_j^{(k)}|^2\right). \tag{10}$$

In the following, we use the error $\hat{e}_j^{(k)}$ given by Proposition 2.2.

- Start with Algorithm 3. Let $\theta_j = \dfrac{1+\Delta t(\tau-\lambda_j)}{1+\Delta t\tau}$ and $\gamma_j = \dfrac{1}{1+\Delta t\lambda_j}$ for $1 \le j \le N$. Let start by estimate the following sum using Lemma 2.1:

$$\begin{aligned}\left|\sum_{i=0}^{k-1}\theta_j^{k-i-1}(\hat{v}_j^{(i+1)} - \hat{v}_j^{(i)})\right| &\le \Delta t\lambda_j\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}|\hat{v}_j^{(i+1)}| + \Delta t\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}|\hat{f}_j^{(i+1)}| \\ &\le \Delta t\lambda_j\left(\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}\gamma_j^{i+1}\right)|\hat{v}_j^{(0)}| + \cdots \\ &\qquad + \Delta t^2\lambda_j\left(\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}\sum_{\ell=1}^{i+1}\gamma_j^{i+1-\ell}\right)\hat{f}_j^{\infty} + \cdots \\ &\qquad + \Delta t\left(\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}\right)\hat{f}_j^{\infty}.\end{aligned}$$

  Note that the first two terms are zero when $\lambda_j = 0$. So, without any loss of generality, we assume that this is not the case here. Now, we estimate each of the three previous terms. At first, and because $\tau \ge \dfrac{\rho(A)}{2}$, we have $|\theta_j| \in [0,1]$ then

$$\begin{aligned}\Delta t\lambda_j\left(\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}\gamma_j^{i+1}\right)|\hat{v}_j^{(0)}| &\le \Delta t\lambda_j\left(\sum_{i=0}^{k-1}\gamma_j^{i+1}\right)|\hat{v}_j^{(0)}|, \\ &\le \Delta t\lambda_j\left(\sum_{i=0}^{\infty}\gamma_j^{i} - 1\right)|\hat{v}_j^{(0)}|, \\ &\le |\hat{v}_j^{(0)}|.\end{aligned}$$

  Considering geometric sequences, we have

$$\begin{aligned}\Delta t^2\lambda_j\left(\sum_{i=0}^{k-1}|\theta_j|^{k-i-1}\sum_{\ell=1}^{i+1}\gamma_j^{i+1-\ell}\right)\hat{f}_j^{\infty} &\le \Delta t^2\lambda_j\frac{1}{1-\theta_j}\frac{1}{\Delta t\lambda_j}\hat{f}_j^{\infty}, \\ &\le \frac{\Delta t}{1-\theta_j}\hat{f}_j^{\infty}, \\ &\le \frac{1+\tau\Delta t}{\lambda_j}\hat{f}_j^{\infty}.\end{aligned}$$

The third term gives

$$\Delta t \left(\sum_{i=0}^{k-1} |\theta_j|^{k-i-1} \right) \hat{f}_j^{\infty} \leq \frac{1+\tau\Delta t}{\lambda_j} \hat{f}_j^{\infty}.$$

This inequalities give us the following formula

$$\left| \sum_{i=0}^{k-1} \theta_j^{k-i-1} (\hat{v}_j^{(i+1)} - \hat{v}_j^{(i)}) \right| \leq 2(1+\tau\Delta t) \left(|\hat{v}_j^{(0)}| + \frac{\hat{f}_j^{\infty}}{\lambda_j} \right).$$

Thanks to the Proposition 2.2, we deduce

$$|\hat{e}_j^{(k)}| \leq \begin{cases} 2\Delta t \lambda_j \left(|\hat{v}_j^{(0)}| + \dfrac{\hat{f}_j^{\infty}}{\lambda_j} \right) & j \leq m, \\ 2\Delta t |\lambda_j - \tau| \left(|\hat{v}_j^{(0)}| + \dfrac{\hat{f}_j^{\infty}}{\lambda_j} \right) & \text{else.} \end{cases}$$

This gives the conclusion.

- Now, consider Algorithm 4 which is similar to the previous one. For all $1 \leq j \leq N$ and $k \geq 0$ we define $\omega_j = \dfrac{1+\tau\Delta t}{1+\Delta t(\tau+\lambda_j)}$. Note that $\omega_j \in [0,1]$. If $j \leq m$, we have $\hat{e}_j^{(k)} = 0$. Hence, we obtain

$$\left| \sum_{i=0}^{k-1} \omega_j^{k-i-1} \left(\hat{v}_j^{(i+1)} - \hat{v}_j^{(i)} \right) \right| \leq (1+\Delta t(\tau+\lambda_j)) \left(|\hat{v}_j^{(0)}| + \frac{\hat{f}_j^{\infty}}{\lambda_j} \right),$$

which gives

$$|\hat{e}_j^{(k)}| \leq \begin{cases} 0 & j \leq m, \\ 2\Delta t \tau \left(|\hat{v}_j^{(0)}| + \dfrac{\hat{f}_j^{\infty}}{\lambda_j} \right) & \text{else.} \end{cases}$$

The conclusion is deduced from (10).

■

Remark 2.4 *As it will be observed in forthcoming simulations, the bounds given for Algorithm 3 overestimates $\|\hat{e}^{(k)}\|_2^2$. Indeed, in order to guarantee the stability we choose at best $\tau = \dfrac{\rho(A)}{2} = \dfrac{\lambda_N}{2}$. Then, the estimate gives:*

$$\begin{aligned} \|\hat{e}^{(k)}\|_2^2 &\leq 4\Delta t^2 \left(\sum_{j=1}^{m} \left(\lambda_j |\hat{v}_j^{(0)}| + \hat{f}_j^{\infty} \right)^2 + \sum_{j=m+1}^{N} \left(1 - \frac{\tau}{\lambda_j} \right)^2 \left(\lambda_j |\hat{v}_j^{(0)}| + \hat{f}_j^{\infty} \right)^2 \right), \\ &\leq 4\tau^2 \Delta t^2 \sum_{j=1}^{N} \left(|\hat{v}_j^{(0)}| + \frac{\hat{f}_j^{\infty}}{\lambda_j} \right)^2. \end{aligned}$$

For the Algorithm 4, inequality is similar but considering all frequencies in the sum instead of only high frequencies. However, the approach is maintained to fix the ideas that will be used for nonlinear equations.

In practice, it is useful to know λ_m (and thus m) to ensure the difference $e^{(k)}$ between the solution generated by a stabilized scheme and Backward Euler's is smaller than a fixed threshold. This is the purpose of the following result.

Corollary 2.5 *We consider Algorithm 4. Let C be a given real positive constant. Assume that*

i) $\displaystyle\sum_{j=m+1}^{N} |\hat{v}_j^{(0)}|^2 \leq \frac{C^2}{16\tau^2},$

ii) $\displaystyle\lambda_{m+1} \geq \frac{4\tau}{C}\sqrt{\sum_{j=m+1}^{N} |\hat{f}_j^{\infty}|^2},$

then $\|\hat{e}^{(k)}\|_2 \leq C\Delta t$ for all $k \geq 0$.

Proof. We have $\|\hat{e}^{(k)}\|_2^2 \leq 4\tau^2\Delta t^2 \left(\displaystyle\sum_{j=m+1}^{N} |\hat{v}_j^{(0)}|^2 + \frac{(\hat{f}_j^{\infty})^2}{\lambda_{m+1}^2}\right)$ which is smaller than $C^2\Delta t^2$ if and only if

$$\left(C^2 - 8\tau^2 \sum_{j=m+1}^{N} |\hat{v}_j^{(0)}|^2\right)\lambda_{m+1}^2 - 8\tau^2 \sum_{j=m+1}^{N} |\hat{v}_j^{(0)}|^2 (\hat{f}_j^{\infty})^2 \geq 0.$$

This condition is satisfied. Indeed, we have

$$\begin{aligned}
&\left(C^2 - 8\tau^2 \sum_{j=m+1}^{N} |\hat{v}_j^{(0)}|^2\right)\lambda_{m+1}^2 - 8\tau^2 \sum_{j=m+1}^{N} |\hat{v}_j^{(0)}|^2 (\hat{f}_j^{\infty})^2, \\
&\qquad \geq \frac{C^2}{2}\lambda_{m+1}^2 - 8\tau^2 \sum_{j=m+1}^{N} (\hat{f}_j^{\infty})^2 \text{ using i)}, \\
&\qquad \geq \frac{C^2}{2}\left(\lambda_{m+1}^2 - \frac{16\tau^2}{C^2} \sum_{j=m+1}^{N} (\hat{f}_j^{\infty})^2\right), \\
&\qquad \geq 0 \text{ using ii)}.
\end{aligned}$$

■

2.3 Space discretization

Before presenting the first numerical results, we introduce the space discretization that will be used in the simulations done. In the latter, we focus on a finite difference scheme with

Neumann boundary condition. Let the regular 1D grid designed with the nodes $x_j = j\Delta x$ where $1 \leq j \leq N$. The integer $N \in \mathbb{N}^*$ corresponds to the number of grid points. The step is $\Delta x = \dfrac{L}{N-1}$ where L is the domain length.
We use a second order discretization for the discrete Laplacian operator:

$$\partial_x^2 u(x_j) \approx \frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{\Delta x^2}.$$

If associated to Neumann boundary condition, it is related to the matrix

$$A_{1\mathrm{D}} = \frac{1}{\Delta x^2} \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & (0) & \\ & \ddots & \ddots & \ddots & & \\ & (0) & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix} \in \mathcal{M}_N(\mathbb{R}). \tag{11}$$

Let $U = [u(x_1), \cdots, u(x_N)]^T$ then we have $(A_{1\mathrm{D}}U)_j \approx -\partial_x^2 U(x_j)$ and $(A_{1\mathrm{D}}^2 U)_j \approx \partial_x^4 U(x_j)$, adding appropriate boundary conditions. For two- and three-dimensional differential equations considered, the discrete Laplacian is obtained by using Kronecker's products (symbolized with $\otimes$). In a square grid with $N \times N$ regularly spaced points, the matrix becomes

$$A_{2\mathrm{D}} = A_{1\mathrm{D}} \otimes \mathrm{Id}_N + \mathrm{Id}_N \otimes A_{1\mathrm{D}}.$$

In a cubic grid with $N \times N \times N$ points, it is

$$A_{3\mathrm{D}} = A_{1\mathrm{D}} \otimes \mathrm{Id}_N \otimes \mathrm{Id}_N + \mathrm{Id}_N \otimes A_{1\mathrm{D}} \otimes \mathrm{Id}_N + \mathrm{Id}_N \otimes \mathrm{Id}_N \otimes A_{1\mathrm{D}}.$$

Remark 2.6 *Linear systems based on this discretization can be solved efficiently thanks discrete cosinus transform (see [6] and reference therein).*

2.4 Numerical results for the linear equation

In this section, we compare different simulations for the one dimensional heat equation (5) on the domain $\Omega = [0, 1]$. Solutions are computed by using the stabilized Algorithms 3 and 4. The difference between IMEX and stabilized solutions are compared to the estimates obtained in Theorem 2.3.
The initial state u_0 and the right hand side function f are given by the following examples. To simplify the analysis and for a sake of clarity, in each example we have $\displaystyle\int_0^1 u_0(x)dx = \int_0^1 f(x,t)dx = 0$ and thus $\hat{u}_1^{(k)} = \hat{f}_1^{(k)} = 0$ for all $t \geq 0$ and $k \in \mathbb{N}$. If this assumption is not numerically exactly satisfied, a mass conservation projector is added in algorithms to avoid the accumulation of rounding errors that would reduce the accuracy (see [6], RSS modified with projection approach, for more details).

Example 2.7 *: Gaussian function. Initial function is*

$$u_0(x) = \exp\left(-100(x-0.5)^2\right), \tag{12}$$

which is a regular function and $f(x,t) = \sin(10\pi x)\sin(5t)$.

Example 2.8 *: regularized triangular function with forcing function. The forcing function* f *is such that the exact solution is*

$$u(x,t) = 10\sum_{\ell=1}^{100}\left(\frac{\sin(j)}{\pi\ell}\right)^2 \cos(2\pi(x-1/2)\ell)\exp(\sin(t)). \tag{13}$$

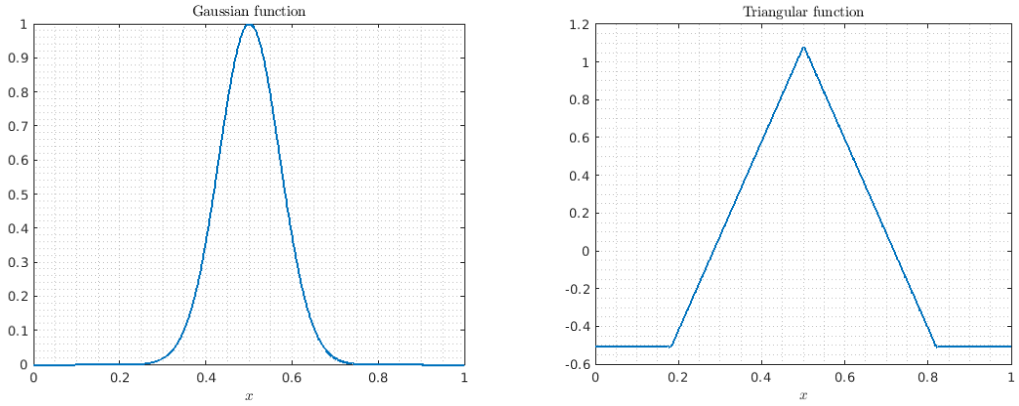

Figure 1: Initial functions u_0 considered to solve equation (5). Left plot: function (12); Right plot : function (13).

In Figure 2, we plot the absolute value of vector $\hat{v}^{(0)} = P^T v^{(0)}$ when discretized with $N = 1024$ points. It corresponds to the initial value written in eigen-vector basis $(w_1, w_2, \cdots, w_N)$.

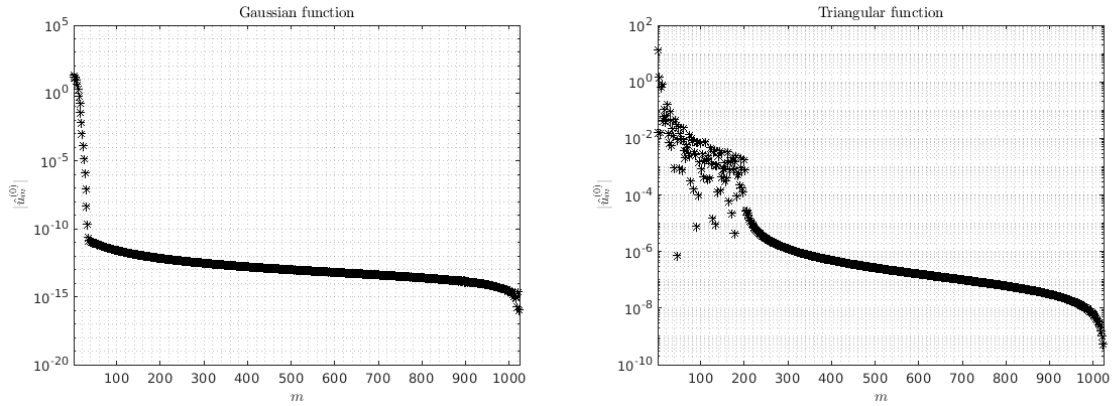

Figure 2: Vector $|P^T u^{(0)}|$ where $u^{(0)}$ is written in the eigenvector basis and discretized with $N = 1024$ points. Left plot : function (12); Right plot: function (13).

First, considering the Gaussian function (12), half of the values are 0 (up to the machine precision) due to symmetry properties. More precisely, only 16 values are larger than 10^{-10}. Similarly, half of the values are larger than 10^{-10} for the triangular function (13) but values decrease significantly from the 200-th.

The vector $\hat{v}^{(0)}$ has an important role in the errors estimates and thus in errors measured in simulations. To illustrate this, we analyze $\max_k \|v^{(k)} - u^{(k)}\|_2$ where $v^{(k)}$ is computed with Backward Euler's scheme (Algorithm 5) and $u^{(k)}$ is computed with stabilized Algorithms 3 and 4. Results are plotted in Figures 3 and 4 with $\Delta t = 0.01$ and $N = 1024$ for different values m. The final time is $T = 5$.

Due to stability property, results with Algorithm 3 are not plotted if $\tau \leq \dfrac{\rho(A)}{2} \approx 2.0831 \times 10^6$.

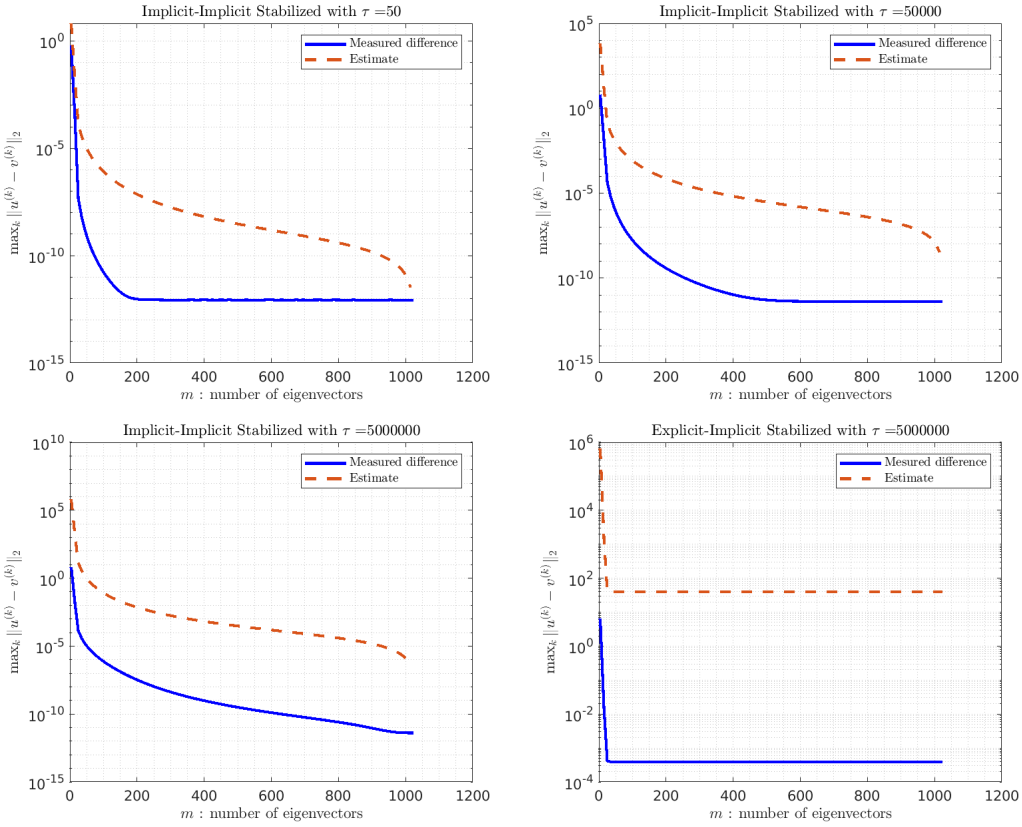

Figure 3: Example 2.7. Maximum difference $\max_k \|v^{(k)} - u^{(k)}\|_2$ where $v^{(k)}$ is computed with Alg. 5 and $u^{(k)}$ computed with Alg. 4 (plots 1-3) and Alg. 3 (plot 4). The final time is $T = 5$. The numerical parameters are $N = 1024$ and $\Delta t = 0.01$.

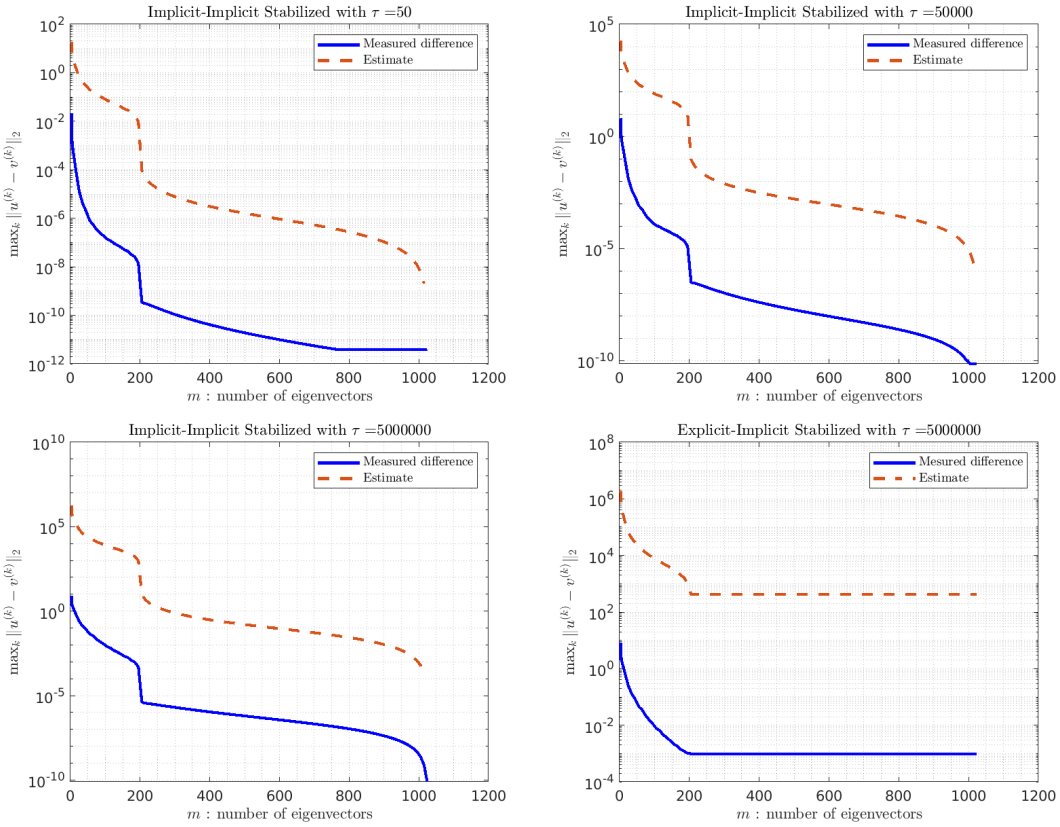

Figure 4: Example 2.8. Maximum difference $\max_k \|v^{(k)} - u^{(k)}\|_2$ where $v^{(k)}$ is computed with Alg. 5 and $u^{(k)}$ computed with Alg. 4 (plots 1-3) and Alg. 3 (plot 4). The final time is $T = 5$. The numerical parameters are $N = 1024$ and $\Delta t = 0.01$.

In our experiments, $\max_k \|u^{(k)} - v^{(k)}\|_2$ is overestimated by the formulas in Theorem 2.3. However, in all cases values $\max_k \|v^{(k)} - u^{(k)}\|_2$ are related to the eigenvector decomposition $P^T v^{(0)}$. Thus, the estimate gives an idea of the difference. Furthermore Implicit-Implicit stabilized scheme give the same results than Backward Euler's scheme as soon as the eigenvector decomposition is sufficient (i.e. m large enough) and thus $\max_k \|u^{(k)} - v^{(k)}\|_2 = 0$.
We do not find this property with Implicit-Explicit stabilization due to the scheme's construction. Even if the full eigenvector basis is considered, the scheme does not give the same solution than those given by Backward Euler's scheme. It was predictable thanks to Proposition 2.2 in which the error is never 0.

3 High mode stabilization for nonlinear Equation

In this section, we consider the nonlinear equation

$$\frac{du}{dt} + Au + f(u) = 0, \quad t > 0, \tag{14}$$

$$u(0) = u_0 \in \mathbb{R}^N, \tag{15}$$

where f is a given function, F is an anti-derivative of f (i.e. $F' = f$) and A is a semi-definite symmetric positive matrix. Denoting $\mathbf{1} = [1, \cdots, 1]^T \in \mathbb{R}^N$ and $< \cdot, \cdot >$ the Euclidian scalar product in $\mathbb{R}^N$, the energy

$$E(u) = \frac{1}{2} < Au, u > + < F(u), \mathbf{1} >$$

is decreasing in the sense that $\dfrac{dE(u)}{dt} \leq 0$ if u satisfies (14). The aim of the stabilization is to obtain an equivalent of this property, when discretizing in time, with weak restrictions on the time step Δt.

3.1 Stabilization strategy

We adapt the stabilization technique introduced in [1] when using a bi-grid approach with finite elements to the spectral case. The key idea is to concentrate the computational effort on the low mode components, by solving an unconditionally energy decreasing scheme. Conversely, we use a stabilized IMEX scheme on the high mode ones. It can be seen as a predictor-corrector scheme in which the prediction is realized on a small dimensional space (that of low modes), the correction being applied on the whole space by combining an IMEX scheme and a high mode smoothing (see also [7]).
We keep the notation used in the linear case and adapt the stabilization strategy to (14). We propose the following algorithm:

Algorithm 6 :Implicit-stabilized

1: **Set** $\hat{y}^{(0)} = P_m^T u^{(0)}$
2: **for** $k = 0, 1, \cdots$ **do**
3: **Solve** $\dfrac{\hat{y}^{(k+1)} - \hat{y}^{(k)}}{\Delta t} + P_m^T A P_m \hat{y}^{(k+1)} + P_m^T \left(f_i(P_m \hat{y}^{(k+1)}) + f_e(P_m \hat{y}^{(k)}) \right) = 0$
4: **Set** $y^{(k+1)} = P_m \hat{y}^{(k+1)}$
5: **Solve** $\dfrac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k+1)} + \tau(u^{(k+1)} - u^{(k)}) + f(u^{(k)}) = \tau(y^{(k+1)} - y^{(k)})$
6: **end for**

Functions f_i and f_e are the derivatives of F_i and F_e. Let $u \in \mathbb{R}^N$, we have $F(u) = F_e(u) + F_i(u)$ and $f(u) = f_e(u) + f_i(u)$. F_i denotes the convex part of F and F_e corresponds to the

contracting part. Thus, the inequalities $< f_i(u), u > \geq 0$ and $< f_e(u), u > \leq 0$ are satisfied for all $u \in \mathbb{R}^N$. For all u and v in $\mathbb{R}^N$, we deduce

$$< F_i(u) - F_i(v), \mathbf{1} > \leq < f_i(u), u - v > \tag{16}$$
$$< F_e(u) - F_e(v), \mathbf{1} > \leq < f_e(v), u - v > . \tag{17}$$

Non-linear equation given at line 3 of Algorithm 6 is solved by using the Newton-Raphson method. The computational cost is small because $\hat{y}^{k+1} \in \mathbb{R}^m$ while $u^{k+1} \in \mathbb{R}^N$ with $m \ll N$. It corresponds to low frequencies resolution when applying a Convex Splitting (unconditionally stable) while Line 5 corresponds to an IMEX stabilization acting only on high frequencies.

3.2 Energy decay and error estimates

The obtained scheme is energy stable in the sense of the following proposition.

Proposition 3.1 *Assume $\mathbf{1} \in Span(w_1, \cdots, w_m)$. Let f be $\mathcal{C}^1(\mathbb{R})$ and F be an anti-derivative of f. Assume that $L = \|f'\|_\infty < \infty$ and F be a real function bounded from below. Under one of the following assumptions :*

- $\tau \geq L$;
- $\tau < L$ *and* $\Delta t \leq \dfrac{2}{L - \tau}$;

Algorithm 6 satisfies

$$E(u^{(k+1)}) - E(u^{(k)}) \leq \frac{\tau \Delta t}{2}(E_{LF}(y^{(0)}) - E_{\min}),$$

where $E_{LF}(y) = \dfrac{1}{2} < Ay, y > + < P_m P_m^T F(y), \mathbf{1} >$ is the low frequencies energy and $E_{\min} \in \mathbb{R}$ is the minimal possible energy.

Proof. Consider the convex and contracting parts of the low frequency energy:

$$E_i(\hat{y}) = \frac{1}{2} < Ay, y > + < P_m P_m^T F_i(y), \mathbf{1} >,$$
$$E_e(y) = < P_m P_m^T F_e(y), \mathbf{1} > .$$

For all ϕ and η in $\mathbb{R}^N$,

$$\begin{aligned}
E_i(\phi) - E_i(\eta) &= \frac{1}{2} < A\phi, \phi > - \frac{1}{2} < A\eta, \eta > + < P_m P_m^T \left(F_i(\phi) - F_i(\eta)\right), \mathbf{1} >, \\
&= < A\phi, \phi - \eta > - \frac{1}{2} < A(\phi - \eta), \phi - \eta > + < P_m P_m^T \left(F_i(\phi) - F_i(\eta)\right), \mathbf{1} > \\
&= < A\phi, \phi - \eta > - \frac{1}{2} < A(\phi - \eta), \phi - \eta > + < F_i(\phi) - F_i(\eta), \mathbf{1} >, \\
&\qquad \text{because } \mathbf{1} \in \text{Span}(w_1, \cdots, w_m), \\
&\leq < A\phi, \phi - \eta > + < F_i(\phi) - F_i(\eta), \mathbf{1} > \\
&\leq < A\phi, \phi - \eta > + < f_i(\phi), \phi - \eta > \text{ thanks to (16).}
\end{aligned}$$

Furthermore, considering the contracting part, we have

$$\begin{aligned} E_e(\phi) - E_e(\eta) &=< P_m P_m^T (F_e(\phi) - F_e(\eta)), \mathbf{1} >, \\ &=< F_e(\phi) - F_e(\eta), \mathbf{1} >, \\ &\leq< f_e(\eta), \phi - \eta > \text{ thanks to (17).} \end{aligned}$$

The two previous inequities lead to the decrease of low frequency energy. Indeed, we have

$$\begin{aligned} E_{\mathrm{LF}}(y^{(k+1)}) - E_{\mathrm{LF}}(y^{(k)}) &= E_i(y^{(k+1)}) - E_i(y^{(k)}) + E_e(y^{(k+1)}) - E_e(y^{(k)}), \\ &\leq< Ay^{(k+1)} + f_i(y^{(k+1)}) + f_e(y^{(k)}), y^{(k+1)} - y^{(k)} > . \end{aligned}$$

By definition of the sequence $(y^{(k)})$, we deduce

$$E(y^{(k+1)}) - E(y^{(k)}) \leq -\frac{1}{\Delta t}\|y^{(k+1)} - y^{(k)}\|_2^2. \tag{18}$$

On the other hand, multiplying the line 5 of Algorithm 6 by $u^{(k+1)} - u^{(k)}$, we obtain

$$\begin{aligned} \left(\frac{1}{\Delta t} + \tau\right)\|u^{(k+1)} - u^{(k)}\|_2^2 + \frac{1}{2} < Au^{(k+1)}, u^{(k+1)} > -\frac{1}{2} < Au^{(k)}, u^{(k)} > + \cdots \\ \cdots + \frac{1}{2} < A(u^{(k+1)} - u^{(k)}), u^{(k+1)} - u^{(k)} > + < f(u^{(k)}), u^{(k+1)} - u^{(k)} >= \cdots \\ \cdots \tau < y^{(k+1)} - y^{(k)}, u^{(k+1)} - u^{(k)} > . \end{aligned}$$

Considering $\|f'\|_\infty$, $E(u^{(k+1)})$ and $E(u^{(k)})$, we deduce

$$\left(\frac{1}{\Delta t} + \tau - \frac{L}{2}\right)\|u^{(k+1)} - u^{(k)}\|_2^2 + E(u^{(k+1)}) - E(u^{(k)}) \leq \tau < y^{(k+1)} - y^{(k)}, u^{(k+1)} - u^{(k)} > .$$

The Holder's inequality leads to

$$\left(\frac{1}{\Delta t} + \tau - \frac{L}{2}\right)\|u^{(k+1)} - u^{(k)}\|_2^2 + E(u^{(k+1)}) - E(u^{(k)}) \leq \frac{\tau}{2}\|y^{(k+1)} - y^{(k)}\|_2^2 + \frac{\tau}{2}\|u^{(k+1)} - u^{(k)}\|_2^2.$$

With the previous inequality and the stability result (18), we deduce

$$\left(\frac{1}{\Delta t} + \frac{\tau - L}{2}\right)\|u^{(k+1)} - u^{(k)}\|_2^2 + E(u^{(k+1)}) - E(u^{(k)}) \leq \frac{\tau \Delta t}{2}\left(E(y^{(k)}) - E(y^{(k+1)})\right).$$

Under the assumption $\tau \geq L$, we have $\dfrac{1}{\Delta t} + \dfrac{\tau - L}{2} \geq 0$. It leads to

$$E(u^{(k+1)}) - E(u^{(k)}) \leq \frac{\tau \Delta t}{2}\left(E(y^{(k)}) - E(y^{(k+1)})\right).$$

The sequence $(E(y^{(k)}))_k$ is bounded from below by $E_{\min}$ and decreasing. Thus, we obtain the desired result :

$$E(u^{(k+1)}) - E(u^{(k)}) \leq \frac{\tau \Delta t}{2}(E(y^{(0)}) - E_{\min}).$$

Otherwise, under the assumption $\tau \leq L$, the value $\dfrac{1}{\Delta t} + \dfrac{\tau - L}{2}$ is positive only if $\Delta t \leq \dfrac{2}{L - \tau}$. The rest of the proof is unchanged. ■

Remark 3.2

- $E_{\min}$ *exists since* A *is positive and* $u \mapsto F(u)$ *is bounded from below.*
- *The key inequality is* (18)*. If we consider any energy decreasing scheme instead of the convex splitting, the stabilized scheme satisfies Proposition 3.1.*
- *Proposition 3.1 does not imply energy decreasing (i.e.* $E(u^{(k)+1}) \leq E(u^{(k)})$ *for all* $k \geq 0$*) but it leads to* $E(u^{(k+1)}) \leq E(u^{(k)}) + \mathcal{O}(\tau\Delta t)$*.*
- *Assumption* $\mathbf{1} \in Span(w_1, \cdots, w_m)$ *is not very restrictive in practice. It is typically satisfied with Neumann or Periodic boundary conditions and a finite difference method as in our expriments with Allen-Cahn and Swift-Hohenberg equations.*

Now, we analyze the error with the continuous solution of (3). We start with the following lemma.

Lemma 3.3 *Let* $u \in \mathcal{C}^2(\mathbb{R})$ *and* $R^{(k+1)} = \dfrac{u(t^{(k+1)}) - u(t^{(k)})}{\Delta t} - \dfrac{du}{dt}(t^{(k+1)})$ *for all* $k \in \mathbb{N}$*. Then we have*

$$2\|R^{(k+1)}\|_2^2 \leq \frac{\Delta t}{3} \int_{t^{(k)}}^{t^{(k+1)}} \left\| \frac{d^2u}{dt^2}(s) \right\|_2^2 ds$$

$$\|u(t^{(k+1)}) - u(t^{(k)})\|_2^2 \leq \Delta t^2 \left\| \frac{du}{dt}(t^{(k+1)}) \right\|_2^2 + \frac{\Delta t^3}{3} \int_{t^{(k)}}^{t^{(k+1)}} \left\| \frac{d^2u}{dt^2}(s) \right\|_2^2 ds.$$

Proof. The result follows from a consequence of Taylor's inequalities. ■

Let $u : t \mapsto u(t)$ be the solution of (3). Sequences $(u^{(k)})$ and $(y^{(k)})$ are generated by Algorithm 6. Consider the error at time $t^{(k)}$:

$$\mathfrak{e}^{(k)} = u(t^{(k)}) - u^{(k)}.$$

We can establish the following result:

Theorem 3.4 *Let* $T > 0$*. Assume* $u : t \mapsto u(t) \in \mathcal{C}^2([0, T))$ *and* $\tau \geq L = \|f'\|_\infty$*. If there exists* $C > 0$ *such that for all* $0 \leq k \leq \lfloor \dfrac{T}{\Delta t} \rfloor - 1$*:*

$$\|\mathfrak{z}^{(k+1)} - \mathfrak{z}^{(k)}\|_2^2 \leq \frac{C\Delta t^2}{\tau^2},$$

with $\mathfrak{z}^{(k)} = u(t^{(k)}) - y^{(k)}$*, then we have*

$$\|\mathfrak{e}^{(k)}\|_2^2 \leq \Delta t^2 \left(CT + \frac{2\Delta t}{3} \int_0^T \left\| \frac{d^2u}{dt^2}(s) \right\|_2^2 ds + T \max_{\sigma \in [0,T]} \left\| \frac{du}{dt}(\sigma) \right\|_2^2 \right) \exp\left(2T(1 + 2L)\right).$$

Proof. Sequence $(u^{(k)})$ is generated by

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + \tau(u^{(k+1)} - u^{(k)}) + Au^{(k+1)} + f(u^{(k)}) = \tau(y^{(k+1)} - y^{(k)}),$$

while $(u(t^{(k)}))$ satisfies

$$\frac{du}{dt}(t^{(k+1)}) + \tau(u(t^{(k+1)}) - u(t^{(k)})) + Au(t^{(k+1)}) + f(u(t^{(k+1)})) = \tau(u(t^{(k+1)}) - u(t^{(k)})).$$

Taking the difference, and considering the definition of $R^{(k+1)}$, we have

$$-R^{(k+1)} + \frac{\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}}{\Delta t} + \tau(\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}) + A\mathfrak{e}^{(k+1)} + f(u(t^{(k+1)})) - f(u^{(k)}) = \tau(\mathfrak{z}^{(k+1)} - \mathfrak{z}^{(k)}).$$

Rearranging these terms, we deduce

$$\left(\frac{1}{\Delta t} + \tau\right)(\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}) + A\mathfrak{e}^{(k+1)} = \tau(\mathfrak{z}^{(k+1)} - \mathfrak{z}^{(k)}) + R^{(k+1)} + f(u^{(k)}) - f(u(t^{(k+1)})).$$

Multiplying by $\mathfrak{e}^{(k+1)}$, and since $< A\mathfrak{e}^{(k+1)}, \mathfrak{e}^{(k+1)} > \geq 0$ and

$$< \mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}, \mathfrak{e}^{(k+1)} >= \frac{1}{2}\left(\|\mathfrak{e}^{(k+1)}\|_2^2 - \|\mathfrak{e}^{(k)}\|_2^2 + \|\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}\|_2^2\right),$$

we obtain the following inequality

$$\begin{aligned}(1+\tau\Delta t)\left(\|\mathfrak{e}^{(k+1)}\|_2^2 - \|\mathfrak{e}^{(k)}\|_2^2 + \|\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}\|_2^2\right) \leq\ & 2\Delta t < R^{(k+1)}, \mathfrak{e}^{(k+1)} > \\ & + 2\tau\Delta t < \mathfrak{z}^{(k+1)} - \mathfrak{z}^{(k)}, \mathfrak{e}^{(k+1)} > \\ & + 2\Delta t < f(u^{(k)}) - f(u(t^{(k+1)})), \mathfrak{e}^{(k+1)} > .\end{aligned} \tag{19}$$

Each term of the right hand side of the previous inequality is considered. We start by noticing that

$$\begin{aligned}2\Delta t < R^{(k+1)}, \mathfrak{e}^{(k+1)} > &\leq 2\Delta t\|R^{(k+1)}\|_2\|\mathfrak{e}^{(k+1}\|_2 \text{ using Cauchy-Schwarz inequality,} \\ &\leq \Delta t\|R^{(k+1)}\|_2^2 + \Delta t\|\mathfrak{e}^{(k+1)}\|_2^2 \text{ using Young inequality.}\end{aligned}$$

The second term is estimated by the same method:

$$2\tau\Delta t < \mathfrak{z}^{(k+1)} - \mathfrak{z}^{(k)}, \mathfrak{e}^{(k+1)} > \leq \Delta t\tau^2\|\mathfrak{z}^{(k+1)} - \mathfrak{z}^{(k)}\|_2^2 + \Delta t\|\mathfrak{e}^{(k+1)}\|_2^2.$$

We consider the third term :

$$\begin{aligned}2\Delta t < f(u^{(k)}) - f(u(t^{(k+1)})), \mathfrak{e}^{(k+1)} > &\leq 2\Delta t\|\mathfrak{e}^{(k+1)}\|_2 \cdot \|f(u^{(k)}) - f(u(t^{(k+1)}))\|_2, \\ &\leq 2\Delta t\|\mathfrak{e}^{(k+1)}\|_2\left(\|f(u^{(k+1)}) - f(u^{(k)})\|_2 + \|f(u^{(k+1)}) - f(u(t^{(k+1)}))\|_2\right) \\ &\leq 2L\Delta t\|\mathfrak{e}^{(k+1)}\|_2\left(\|u^{(k+1)} - u^{(k)}\|_2 + \|u^{(k+1)} - u(t^{(k+1)})\|_2\right) \text{ because } \|f'\|_\infty = L, \\ &\leq 2L\Delta t\|\mathfrak{e}^{(k+1)}\|_2\left(\|\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}\|_2 + \|u(t^{(k+1)}) - u(t^{(k)})\|_2 + \|\mathfrak{e}^{(k+1)}\|_2\right) \\ &\qquad \text{using triangular inequality,} \\ &\leq 4\Delta tL\|\mathfrak{e}^{(k+1)}\|_2^2 + \Delta tL\|\mathfrak{e}^{(k+1)} - \mathfrak{e}^{(k)}\|_2^2 + \Delta tL\|u(t^{(k+1)}) - u(t^{(k)})\|_2^2 \\ &\qquad \text{with Young inequality.}\end{aligned}$$

Combining the above inequalities into (19), we obtain

$$(1+\tau\Delta t)(\|\mathfrak{e}^{(k+1)}\|_2^2-\|\mathfrak{e}^{(k)}\|_2^2)+(1+\Delta t(\tau-L))\|\mathfrak{e}^{(k+1)}-\mathfrak{e}^{(k)}\|_2^2\leq 2\Delta t(1+2L)\|\mathfrak{e}^{(k+1)}\|_2^2 \\ +\Delta t\|R^{(k+1)}\|_2^2+\Delta t\tau^2\|\mathfrak{z}^{(k+1)}-\mathfrak{z}^{(k)}\|_2^2+\Delta t\|u(t^{(k+1)})-u(t^{(k)})\|_2^2.$$

However $\tau\geq L$, and there exist $C>0$ such that $\|\mathfrak{z}^{(k+1)}-\mathfrak{z}^{(k)}\|_2^2\leq\dfrac{C\Delta t^2}{\tau^2}$. Then, we have

$$(1+\tau\Delta t)(\|\mathfrak{e}^{(k+1)}\|_2^2-\|\mathfrak{e}^{(k)}\|_2^2)\leq 2\Delta t(1+2L)\|\mathfrak{e}^{(k+1)}\|_2^2+\Delta t\|R^{(k+1)}\|_2^2 \\ +C\Delta t^3+\Delta t\|u(t^{(k+1)})-u(t^{(k)})\|_2^2.$$

Using Lemma 3.3, we deduce

$$\|\mathfrak{e}^{(k+1)}\|_2^2-\|\mathfrak{e}^{(k)}\|_2^2\leq 2\Delta t(1+2L)\|\mathfrak{e}^{(k+1)}\|_2^2+\frac{\Delta t^3}{3}\int_{t^{(k)}}^{t^{(k+1)}}\left\|\frac{d^2u}{dt^2}(s)\right\|_2^2ds \\ +C\Delta t^3+\Delta t^3\left\|\frac{du}{dt}(t^{(k+1)})\right\|_2^2.$$

Summing up this inequality from $k=0,\cdots\lfloor\frac{T}{\Delta t}\rfloor-1$, and since $\|\mathfrak{e}^{(0)}\|_2=0$, we obtain

$$\|\mathfrak{e}^{(N+1)}\|_2^2\leq 2\Delta t(1+2L)\sum_{k=0}^{N}\|\mathfrak{e}^{(k+1)}\|_2^2+C(N+1)\Delta t^3+\cdots \\ \cdots+\frac{2\Delta t^3}{3}\int_0^T\left\|\frac{d^2u}{dt^2}(s)\right\|_2^2ds+\Delta t^3\sum_{k=0}^{N}\left\|\frac{du}{dt}(t^{(k+1)})\right\|_2^2.$$

Thus, we have

$$\|\mathfrak{e}^{(N+1)}\|_2^2\leq CT\Delta t^2+\frac{2\Delta t^3}{3}\int_0^T\left\|\frac{d^2u}{dt^2}(s)\right\|_2^2ds+T\Delta t^2\max_{0\leq k\leq\lfloor T/\Delta t\rfloor-1}\left\|\frac{du}{dt}(t^{(k+1)})\right\|_2^2+\cdots \\ \cdots+2\Delta t(1+2L)\sum_{k=0}^{N}\|\mathfrak{e}^{(k+1)}\|_2^2.$$

We conclude by using the discrete Gronwall lemma. ■

Remark 3.5 *The condition* $\|z^{(k+1)}-z^{(k)}\|_2=\mathcal{O}\left(\dfrac{\Delta t}{\tau}\right)$ *can give a criterion for choosing* τ*, and make it vary in time. Indeed, once* Δt *fixed, we can write*

$$\tau=\frac{C\Delta t}{\|z^{(k+1)}-z^{(k)}\|_2}\approx\frac{C}{\left\|\dfrac{\partial z}{\partial t}(t^{(k)})\right\|_2},$$

for a specific $C>0$*. This can be an argument to develop adaptive versions of the stabilization.*

3.3 Numerical simulations

In this section, we consider two partial differential equations : Allen-Cahn equation and Swift-Hohenberg equation., both associated with homogeneous Neumann boundary conditions (11). Discretization of the laplacean operator is adapted in two and three dimension using the Kronecker product, as presented in Section 2.4.
The purpose here is to analyze the effect of the Algorithm 6. It is compared to different stabilized and convex splitting schemes.

3.3.1 Allen-Cahn equation

The Allen-Cahn equation was initially introduced to describe the process of phase separation in iron alloys [3]. It writes as

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta u + f(u) = 0, \quad x \in \Omega; \;\; t \geq 0, \\ \dfrac{\partial u}{\partial n} = 0, \qquad\qquad t \geq 0, \\ u(0,x,y) = u_0(x,y), \qquad (x,y) \in \Omega. \end{cases} \tag{20}$$

The linear part of the equation is obtained by discretizing $-\Delta$. Function f is the potential that will be taken as the Landau-Lipschitz one $f(u) = F'(u)$ where $F(u) = \dfrac{1}{4\varepsilon^2}(u^2-1)^2$. The transition phase parameter is $\varepsilon > 0$. Typically, it is chosen close to 0 which makes the equation difficult to solve (numerically).
The two important following properties are satisfied by the solution of (20), and numerical scheme should reproduce them, at least numerically :

- Energy decreasing: as (20) is a gradient flow of energy, the energy

$$\mathcal{E}(u) = \int_\Omega \left(\frac{1}{2}\|\nabla u\|^2 + F(u) \right) d\boldsymbol{x},$$

  is time decreasing. The discrete energy is $E(u) = \dfrac{1}{2} < Au, u > + < F(u), \mathbf{1} >$, where A corresponds to the discrete Laplacian.

- The maximum principle: function u is bounded in time. For all $t \geq 0$, we have the inequality $\|u(t,\cdot)\|_{L^\infty(\Omega)} \leq u_\infty$.

Before considering time integrators, we define the convex and contracting parts of F by $f_i(u) = \dfrac{1}{\varepsilon^2}u^3$ and $f_e(u) = -\dfrac{1}{\varepsilon^2}u$. For solving numerically the equation (20), we chose to use Algorithm 6 for which the error estimates require to have $\|f'\|_\infty < \infty$. It is a common constraint, not too restrictive because the solution is expected to be bounded. For this reason, as in [26],

we replace F by the function F_M:

$$F_M(u) = \begin{cases} \frac{1}{\varepsilon^2}\left(\frac{3M^2-1}{2}u^2 + 2M^3u + \frac{1}{4}(3M^4+1)\right) & \text{if } u \leq -M \\ \frac{1}{4\varepsilon^2}(u^2-1)^2 & \text{if } u \in [-M, M] \\ \frac{1}{\varepsilon^2}\left(\frac{3M^2-1}{2}u^2 - 2M^3u + \frac{1}{4}(3M^4+1)\right) & \text{if } u \geq M. \end{cases}$$

We denote $f_M = F'_M$. In our case, the choice $M = 2$ is satisfactory. Thanks to the proposition 3.1, the energy is mainly decreasing as long as $\tau \geq \|f'_M\|_\infty = \dfrac{3M^2-1}{\varepsilon^2}$. Two experiments are done to analyze the properties of the new stabilized Algorithm 6. The results are compared to those obtained with the two following schemes:

- Stabilized scheme [9, 26]:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k+1)} + f_M(u^{(k)}) + \tau(u^{(k+1)} - u^{(k)}) = 0, \tag{21}$$

  which is first order accurate and linear. It is energy decreasing if $\tau \geq \|f'_M\|_\infty = \dfrac{3M^2-1}{\varepsilon^2}$. However, the dynamic of the solution could be slowed down when τ is large because the stabilization affects all frequencies.

- Convex splitting time scheme [12]:

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k+1)} + f_i(u^{(k+1)}) + f_e(u^{(k)}) = 0, \tag{22}$$

  which is first order accurate. The non-linear equation is solved with a Newton-Raphson method at each time iterate.

Example 3.6 *We consider the Allen-Cahn model in the domain $\Omega = [-1, 1]^2$ with the transition phase parameter $\varepsilon = 0.02$. The initial state is suggested by [21]:*

$$\begin{aligned} u_0(x, y) = &- \tanh\left(\frac{(x-0.3)^2 + y^2 - 0.2^2}{\varepsilon}\right) \tanh\left(\frac{(x+0.3)^2 + y^2 - 0.2^2}{\varepsilon}\right) \\ &\times \tanh\left(\frac{x^2 + (y-0.3)^2 - 0.2^2}{\varepsilon}\right) \tanh\left(\frac{x^2 + (y+0.3)^2 - 0.2^2}{\varepsilon}\right). \end{aligned}$$

It corresponds to four smoothed circles.

Using schemes previously introduced, we compute the numerical solutions. These solutions are plotted at time $t = 0.005$, $t = 0.025$ and $t = 0.05$ in Figure 5. Simulations are computed on a 128×128 grid, with the time step $\Delta t = 10^{-3}$. To ensure stability, scheme (21) and Algorithm 6 are used with $\tau = 30000$.
It is remarkable that only a small number of m eigenvectors (relative to the size of the grid) is

sufficient to calculate the solution accurately. The solutions are close to those calculated with (22). However, if m is too small, the accuracy is poor.

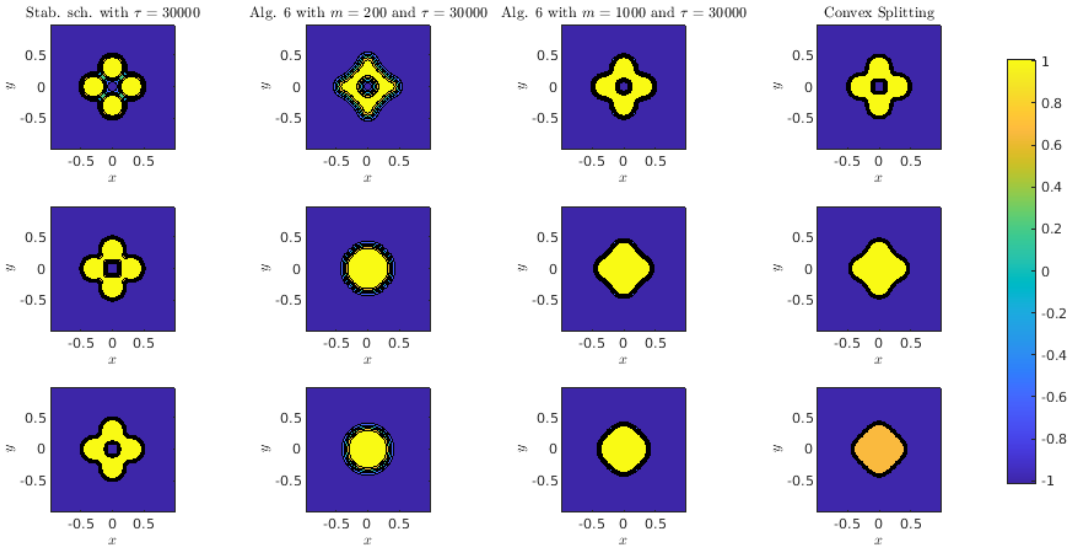

Figure 5: Example 3.6. Comparison of solutions computed with scheme (22) (first column), Alg. 6 (columns 2 and 3), stabilized scheme (21) (last column). The grid is 128×128, the time step is $\Delta t = 10^{-3}$. Row correspond to $t \in \{0.005, 0.025, 0.05\}$.

Figure 6 shows the energy history for each time integrator. The Algorithm 6, with $m = 1000$, is the closest to the convex splitting scheme. This was expected from the construction of the scheme. Conversely, the energy is far from that expected when $m = 200$. The energy decreases in the simulations realized for all the time integrators.

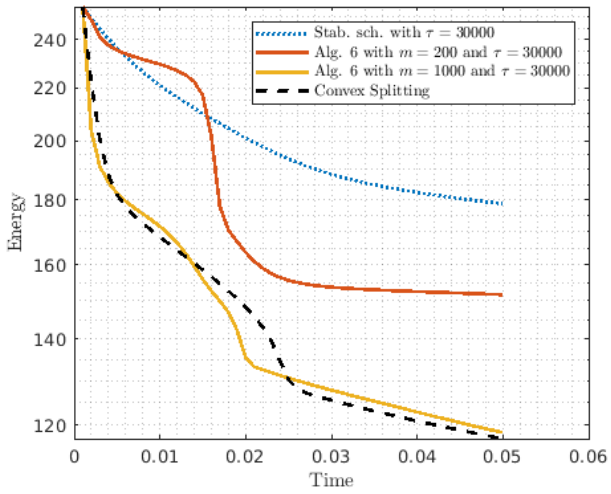

Figure 6: Example 3.6. History of energy with the scheme (22), Algorithm 6 and stabilized Scheme (21). The grid is 128×128, the time step is $\Delta t = 10^{-3}$.

Example 3.7 *The second experiment consists in solving the three dimensional Allen-Cahn model in* $\Omega = [0,1]^3$. *As in the previous example, the transition phase parameter is* $\varepsilon = 0.02$. *Simulation are done starting with a random initial state* $u_0 \in [-1,1]$.

Figure 7 corresponds to the solution computed with Algorithm 6 at different time. The grid is $25 \times 25 \times 25$ and the time step is $\Delta t = 10^{-2}$. The stabilization parameters are $\tau = 30000$ and $m = 2000$.

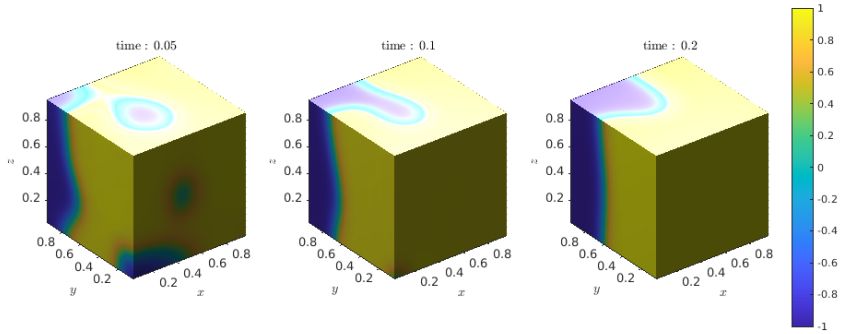

Figure 7: Example 3.7. Solutions computed with Algorithm 6. The grid is $25 \times 25 \times 25$, the time step is $\Delta t = 10^{-2}$, $\tau = 30000$ and $m = 2000$.

In Figure 8, we plot the history of the u extreme's and of energy. Maximum and minimum

of u are close to ± 1 but u is sometimes outside of $[-1, 1]$. As predicted by Proposition 3.1, the energy is decreasing.

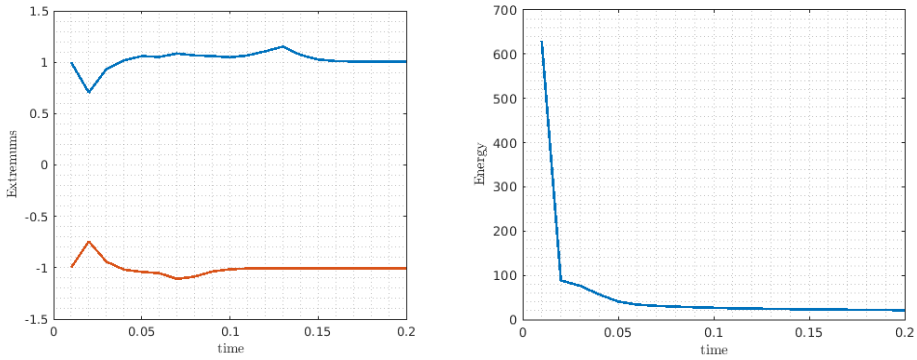

Figure 8: Example 3.7. History of energy with algorithm 6 (columns 2 and 3). The grid is $25 \times 25 \times 25$, the time step is $\Delta t = 10^{-2}$, $\tau = 30000$ and $m = 2000$.

3.3.2 Swift-Hohenberg equation

To study the pattern formation, the Swift-Hohenberg equation is commonly considered. Conversely to Allen-Cahn equation (20), linear part includes a bi-Laplacian and is given by $(1+\Delta)^2$. The model is

$$\begin{cases} \dfrac{\partial u}{\partial t} + (1+\Delta)^2 u + f(u) = 0, \quad x \in \Omega; \;\; t \geq 0 \\ \dfrac{\partial u}{\partial n} = 0 = \dfrac{\partial^3 u}{\partial n^3}, \qquad\qquad t \geq 0 \\ u(0, x, y) = u_0(x, y), \qquad\qquad (x, y) \in \Omega, \end{cases} \tag{23}$$

where $f(u) = u^3 - gu^2 - \varepsilon u$ is the derivative of $F(u) = \dfrac{1}{4}u^4 - \dfrac{g}{3}u^3 - \dfrac{\varepsilon}{2}u^2$. Parameters ε and g are positive values related to the physical context. The decreasing energy associated to (23) is

$$\mathcal{E}(u) = \int_\Omega \left(\frac{1}{2} u (1+\Delta)^2 u + F(u) \right) d\boldsymbol{x}. \tag{24}$$

As in [17], we consider a modified version of F to ensure $\|f'\|_\infty < \infty$:

$$F_M(u) = \begin{cases} \left(\dfrac{3M}{2} \left(\dfrac{2g}{3} + M \right) - \dfrac{\varepsilon}{2} \right) u^2 + M^2 (g + 2M) u + M^3 \left(\dfrac{g}{3} + \dfrac{3M}{4} \right) & \text{if } u \leq -M, \\ \dfrac{1}{4}u^4 - \dfrac{g}{3}u^3 - \dfrac{\varepsilon}{2}u^2 & \text{if } u \in \left[-M, \dfrac{2g}{3} + M \right], \\ \left(\dfrac{3M}{2} \left(\dfrac{2g}{3} + M \right) - \dfrac{\varepsilon}{2} \right) u^2 - \left(\dfrac{g}{3} + 2M \right) \left(\dfrac{2g}{3} + M \right)^2 u & \text{if } u \geq \dfrac{2g}{3} + M, \end{cases}$$

where $M > 0$ is a truncation parameter. Thus, assumption $\|f'_M\|_\infty = 3M \left(\dfrac{2g}{3} + M \right) - \varepsilon < \infty$ is satisfied with $f_M = F'_M$. Furthermore, the convex splitting is used without truncated function

F_M. Convex and contracting parts are given by $f_i(u) = u^3 - gu^2 + \frac{g^2}{3}u$ and $f_e(u) = -\left(\frac{g^2}{3} + \varepsilon\right)u$ (see [18] and reference therein).
In this example, A is the discretization of the linear part $(1+\Delta)^2$. The properties of the scheme detailed in Algorithm 6 are compared to those of the two following schemes.

- The non-iterative and unconditionally energy stable method [17] :

$$\frac{u^{(*)} - u^{(k)}}{\Delta t} + Au^{(*)} + \tau(u^{(*)} - u^{(k)}) + f_M(u^{(k)}) = 0 \tag{25}$$

$$\frac{u^{(**)} - \left(-u^{(k)}/2 + 3u^{(*)}/2\right)}{\Delta t/2} + Au^{(**)} + \tau(u^{(**)} - u^{(*)}) + f_M(u^{(*)}) = 0 \tag{26}$$

$$\frac{u^{(k+1)} - \left(-u^{(k)}/2 + 5u^{(*)}/2 - u^{(**)}\right)}{\Delta t/2} + Au^{(k+1)} + \tau(u^{(k+1)} - u^{(**)}) + f_M(u^{(**)}) = 0. \tag{27}$$

  The scheme was initially introduced as a second order version of the stabilized scheme for Cahn-Hilliard equation (see [16]). It corresponds to a more accurate version of the previous stabilized scheme (21). It is unconditionally energy stable if $\tau \geq 3M\left(\frac{2g}{3} + M\right) - \varepsilon$.

- The convex splitting scheme used in [18] is given by

$$\frac{u^{(k+1)} - u^{(k)}}{\Delta t} + Au^{(k+1)} + f_i(u^{(k+1)}) + f_e(u^{(k)}) = 0. \tag{28}$$

  It is first order accurate. As seen for Allen-Cahn equation, a nonlinear equation is solved at each time step. This is done thanks a quasi-Newton Algorithm.

Three experiments are considered to analyze schemes introduced to solve equation (23).

Example 3.8 *We consider equation* (23) *with* $\Omega = [0, 40]^2$. *Parameters are* $g = 0$ *and* $\varepsilon = 2$. *The initial condition is*

$$u_0(x, y) = \begin{cases} 1 & \text{if } \sin\left(\frac{2\pi y}{10}\right) + 15 < x < \cos\left(\frac{2\pi y}{10}\right) + 25, \\ -1 & \text{otherwise.} \end{cases}$$

In Figure 9, we plot the solution at time $t = 40$ obtained with 128×128 grid points and $\Delta t = 10^{-3}$ obtained with scheme (28). It is considered as a reference solution and it is in accordance with the results available in the literature [17]. We plot the solution computed with algorithm 6 compared to the solution computed with the stabilized scheme (25-27) plotted on the last column. Different values of τ and m are considered. As expected, with the smallest time step $\Delta t = 0.1$, all solutions are acceptable in Figure 10. Energy histories, plotted in Figure 11, are close to the reference energy (black dashed line) even though the decay is slower as τ increases.

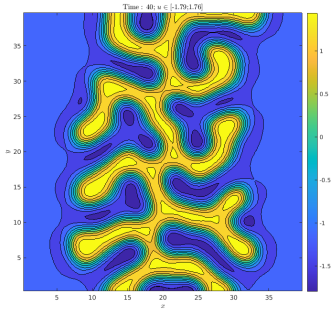

Figure 9: Example 3.8. Reference solution at time $t = 40$ computed with scheme (28). The grid is 128×128 and the time step is $\Delta t = 10^{-3}$.

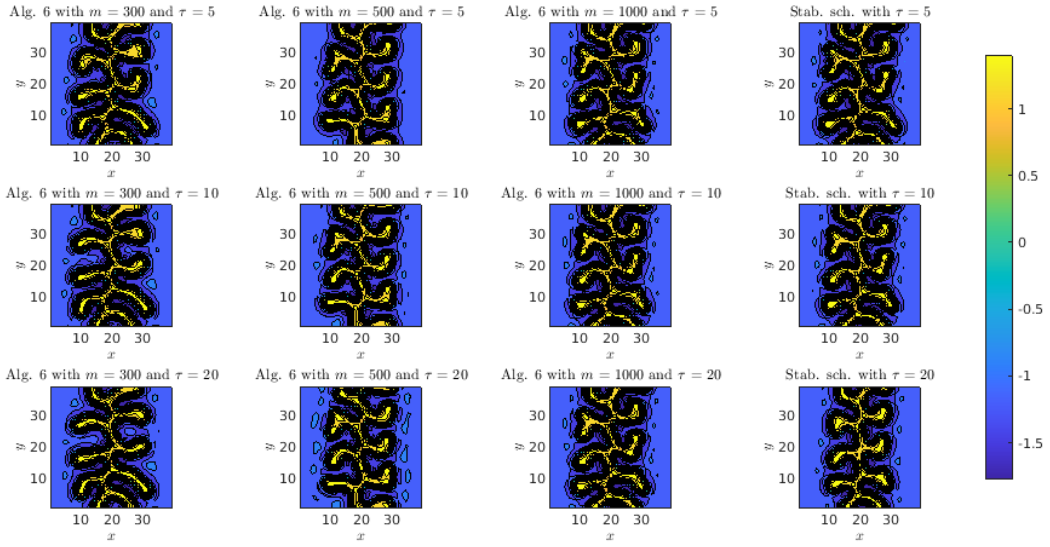

Figure 10: Example 3.8. Comparison of solutions computed with algorithm 6 and stabilized scheme (25-27). The grid is 64×64, the time step is $\Delta t = 0.1$. Row correspond to $\tau \in \{5, 10, 20\}$ and columns to $m \in \{300, 500, 1000\}$.

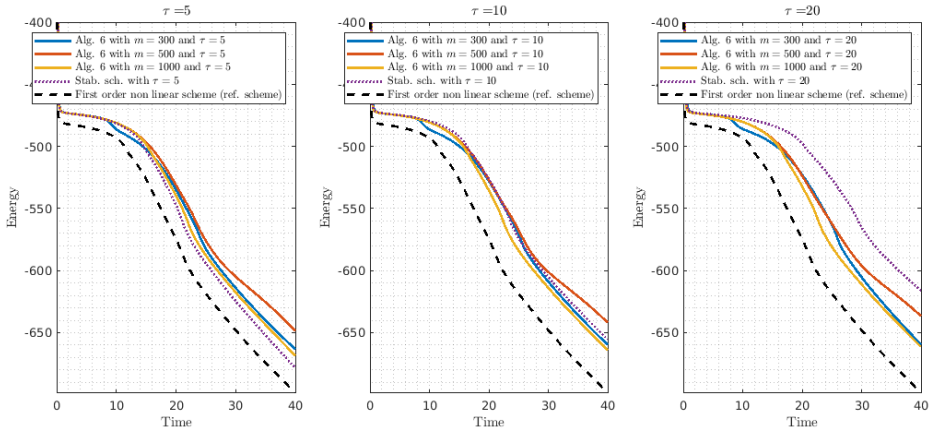

Figure 11: Example 3.8. Comparison of energies related to solutions computed with algorithm 6 and stabilized scheme (25-27). The grid is 64×64, the time step is $\Delta t = 0.1$. Rows correspond to $\tau \in \{5, 10, 20\}$ and columns to $m \in \{300, 500, 1000\}$.

Figure 12 shows the numerical solutions computed with the different time integrators using $\Delta t = 0.4$, for different value of m and τ. Algorithm 6 is less impacted by the increase of τ than stabilized scheme (25-26). This is observed even the scheme is second order accurate while Algorithm 6 is only first order accurate. Histories of energies are plotted in Figure 13. As mentioned previously, the decreasing of the energy is slowed down while τ increases, but the decreasing of the energy with Algorithm 6, is less slowed down than (25-26). Also, we remind that it would be desirable that m to be small to reduce the computational cost. However, the accuracy is reduced when the number of eigenvectors is too small, so quantifying a compromise is necessary.

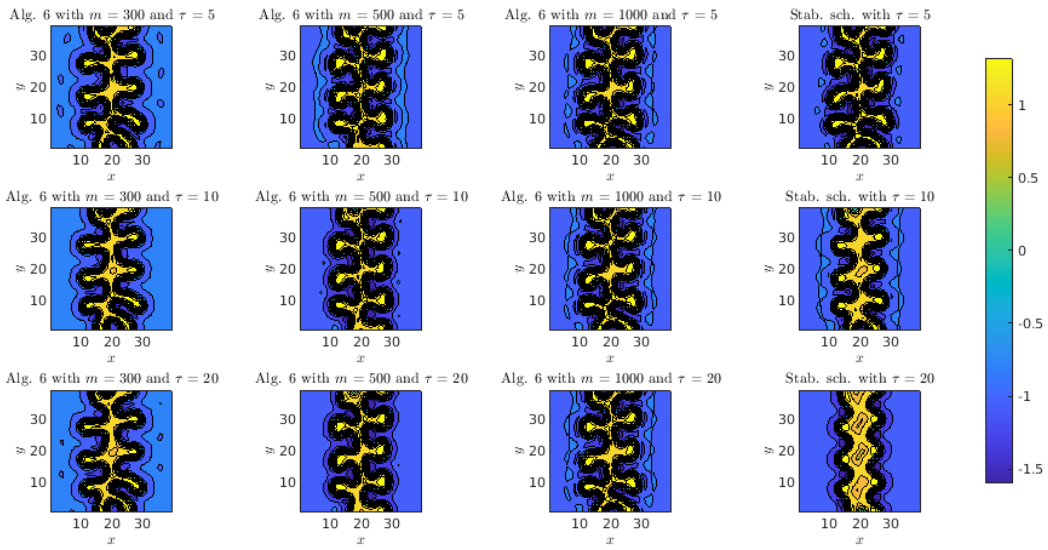

Figure 12: Example 3.8. Same than Figure 10 with $\Delta t = 0.4$.

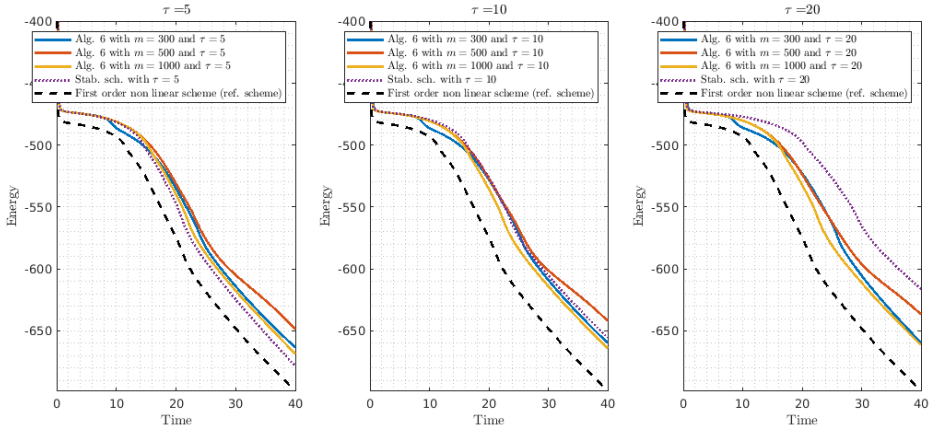

Figure 13: Example 3.8. Same than Figure 11 with $\Delta t = 0.4$.

Example 3.9 *The second test case analyzes the formation of patterns in the largest domain* $\Omega = [0, 128]^2$. *Schemes are initialized with a randomized state bounded in* $[-1, 1]$. *We consider* $\varepsilon = 0.25$. *Two behaviours are analyzed: with* $g = 0$ *or* $g = 1$ *(see [22] for comparison) .*

In all simulations we done, we took $\tau = 20$, which is enough to ensure unconditional stability. The time step is $\Delta t = 1$ and the grid is 128×128. The final time is $t = 100$. Results are compared to those obtained with the convex splitting (28) in the same context.

In Figure 14, we plot the results with $g = 0$. As expected, stripes appear in the domain Ω. the strips are not as visible with m small. Conversely, with $m = 2000$ (which corresponds

to 12% of all eigenvectors), the accuracy is satisfactory. This is also visible on the history of energy. The energy decay is slower at the beginning of the simulation but the energies remain very similar.

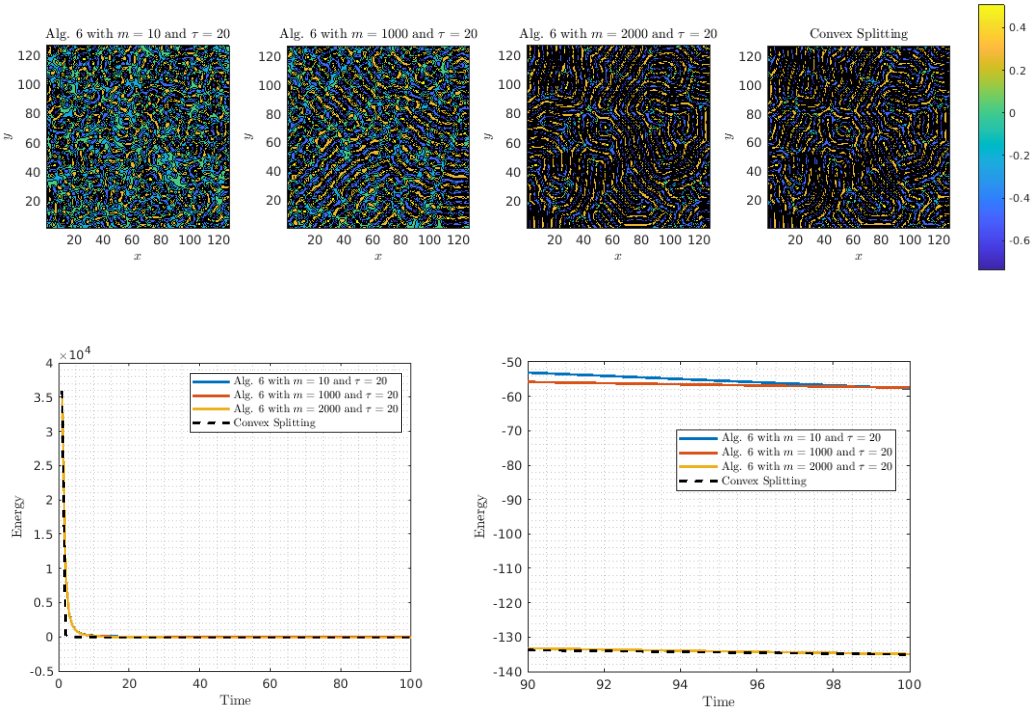

Figure 14: Example 3.9 with $g = 0$. Comparison of solutions computed with algorithm 6 and the convex splitting (28). The grid is 128×128, the time step is $\Delta t = 1$. The number of eigenvectors considered is $m = 10$, $m = 1000$ and $m = 2000$. Last column is obtained with the convex splitting.

Figure 15 corresponds to the results with $g = 1$. Conversely to the previous experiment, spots are expecting to appear but similarly the behaviour is more satisfactory if m increases. Again, with m large, the energy history is closer to the one obtain with the convex splitting (28).

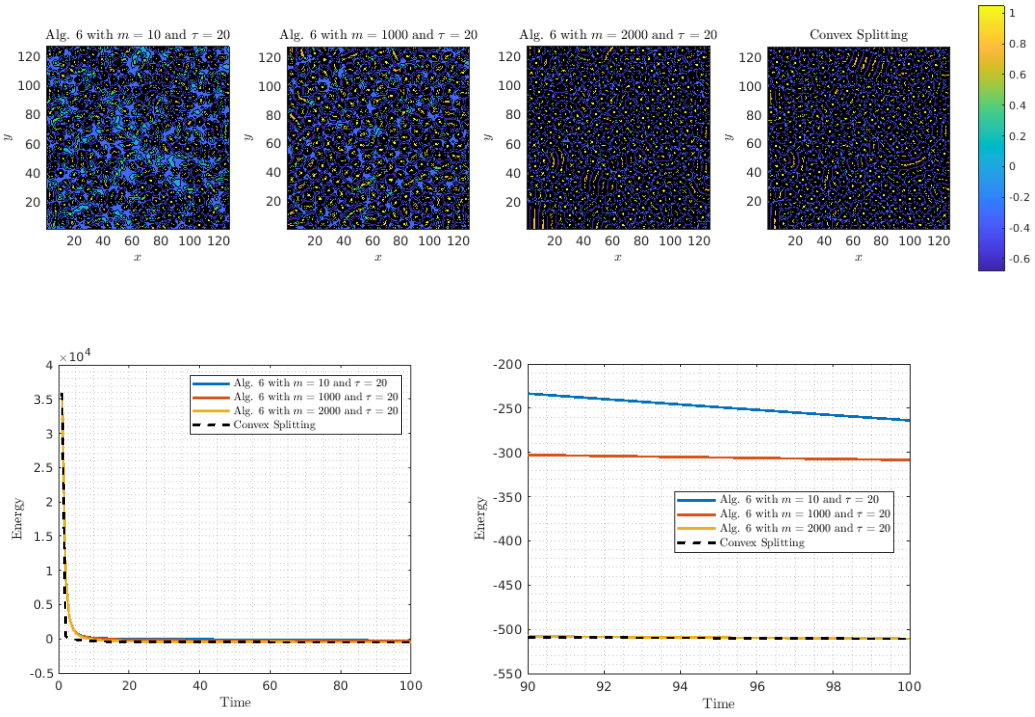

Figure 15: Example 3.9 with $g = 1$. Same than Figure 14.

Example 3.10 *This example is devoted to a three dimensional case. The domain is* $\Omega = [0, 10]^3$. *We consider* $\varepsilon = 0.25$, *and* $g \in \{0, 1\}$. *The initial state is composed with randomized values in* $[-1, 1]$.

In the experiment, the grid is $25 \times 25 \times 25$ and the time step is $\Delta t = 0.5$. The stabilization parameter is $\tau = 20$. The low frequency part is composed with $m = 2000$ eigenvectors which is small compared to the 15625 grid points.

The solution at time $t = 100$ is plotted in Figure 16. Left plot is obtained with $g = 0$. The final solution is bounded in $[-0.6374; 0.6476]$. The case $g = 1$ is shown on the right plot. Then u evolves in $[-0.6963; 1.4139]$.

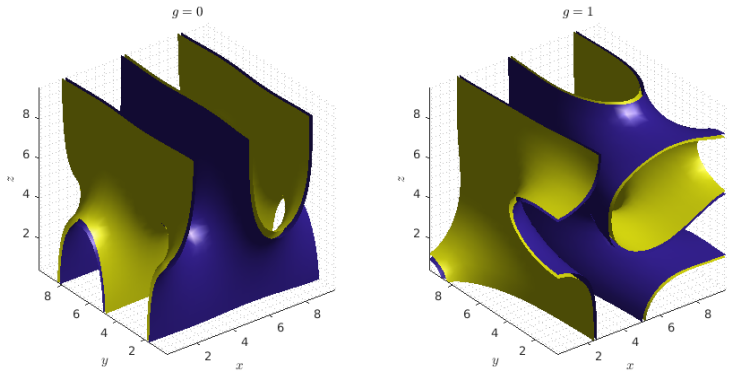

Figure 16: Example 3.10. Numerical parameters are $25 \times 25 \times 25$, $\Delta t = 0.5$, $m = 2000$ and $\tau = 20$. Solution is plotted at time $t = 100$ with $g = 0$ (left panel) and $g = 1$ (right panel).

In Figure 17, we represent the history of energy. It is decreasing as expected.

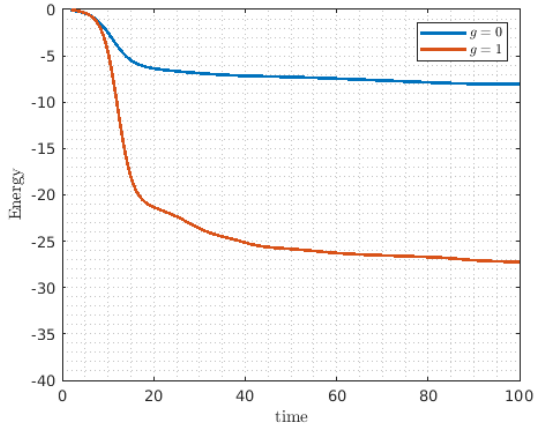

Figure 17: Example 3.10. Numerical parameters are $25 \times 25 \times 25$, $\Delta t = 0.5$, $m = 2000$ and $\tau = 20$. History of energy with $g \in \{0, 1\}$.

4 Concluding remarks

We here proposed and analyzed new times iteration algorithms derived from IMEX schemes, and which are based on proper eigendecomposition. This situation needs to be considered at first before adapting the procedure to other spatial discretizations, such as, e.g. orthogonal polynomials; it can be also applied to any kind of discretization if used together with a filtering pre-processing that allows an efficient separation between low and large frequency components. The numerical results we obtain are encouraging:

- they concern significative cases: Allen-Cahn and Swift-Hohenberg equations in two and three-dimensional domains. The Long-time simulations performed, and the resulting patterns are compared to reference schemes. The results are satisfactory, provided the eigenbasis is large enough. Whatever, the number of eigenvectors that seems to be required is always much more smaller than the dimension of the ambiant space.
- they validate the compromise between stability, reduced computational cost, and precision, allowed by the separation in frequency of the signal, and agree with the formal analysis of the time schemes.
- The stabilization process mainly affects the high frequencies, leading to less slow down of the dynamics as compared to fully stabilized schemes. Furthermore, the implicit equation is solved efficiently as it only involves low frequencies

This motivates to adapt our approach to other discretization methods in space, then to the simulation of a larger family of nonlinear parabolic PDEs.

5 Appendix

5.1 Practical numerical implementation: Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG)

The schemes detailed in this document are based on the calculation of the eigen-elements of a stiffness matrix A. They can be computed by solving an optimisation problem. This type of problem has been considered, for example, in [15, 23]. To avoid the computation of all the eigen-elements, we have introduced a scheme that uses only the first m eigen-elements of A. These can be computed using the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) algorithm.

LOBPCG is a matrix-free method aimed at finding the smallest eigenvalues and associated eigenvectors of a symmetric positive definite generalized eigenvalue problem

$$Ax = \lambda Bx,$$

where both A and B are hermitian matrices, B is definite and positive. The key idea is to minimize a Rayleigh ratio. Let us recall briefly the principle of the LOBPCG method detailed in [14].

We consider first the vector case. The vector associated to the smallest eigenvalue minimizes the Rayleigh quotient:

$$r(x) = \frac{< x, Ax >}{< x, Bx >}.$$

A way to compute it numerically is to apply a gradient-like method. The method generates a sequence $(x^{(k)})$ that converges to the eigenvector associated to the desired eigenvalue.
As we have $\nabla r(x) = \dfrac{2}{< x, Bx >}(Ax - r(x)Bx)$, we define the iterations of a gradient method by

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla r(x^{(k)}).$$

α_k minimises the Rayleigh quotient $r(x^{(k+1)})$. Thus, vector $x^{(k+1)}$ belongs in the subspace $\text{Span}(x^{(k)}, r^{(k)})$ where $r^{(k)} = Ax^{(k)} - r(x^{(k)})Bx^{(k)}$. In order to improve the convergence properties, in LOBPCG algorithm, the subspace is enhanced with $x^{(k-1)}$. Then, $x^{(k+1)}$ is computed thanks the Rayleigh–Ritz method such that:

$$x^{(k+1)} = \underset{x \in \text{Span}(x^{(k)}, r^{(k)}, x^{(k-1)})}{\arg\min} r(x).$$

Now, if K is a pre-conditioner of A, we can compute the preconditioned gradient

$$\nabla_K r(x) = K^{-1}\nabla r(x) = \frac{2}{< x, Bx >} K^{-1}(Ax - r(x)Bx).$$

and $x^{(k+1)}$ is computed in the subspace $\text{Span}(x^{(k)}, w^{(k)}, x^{(k-1)})$ where $Kw^{(k)} = r^{(k)}$. However, in practice, $x^{(k)}$ and $x^{(k-1)}$ become co-linear when the sequence converges. For this reason, subspace $\text{Span}(x^{(k)}, w^{(k)}, x^{(k-1)})$ is replaced by $\text{Span}(x^{(k)}, w^{(k)}, p^{(k)})$ where $p^{(k)}$ is computed in such a way the subspace is unchanged.

To compute the m smallest eigenvalues and associated eigenvectors, the vector $x^{(k)}$ is replaced by the block matrix $X^{(k)} = [x_1^{(k)}, x_2^{(k)}, \cdots, x_m^{(k)}]$ and $r^{(k)}$ by $R^{(k)} = [r_1^{(k)}, r_2^{(k)}, \cdots, r_m^{(k)}]$ where $r_i^{(k)} = Ax_i^{(k)} - r(x_i^{(k)})Bx_i^{(k)}$. The method writes as

Algorithm 7 :LOBPCG

1: **Let** $X^{(0)}$ and $P^{(0)} = (0)$ given block matrices
2: **for** $k = 0, 1, \cdots$ **do**
3: **Set** $R^{(k)} = AX^{(k)} - r(X^{(k)})BX^{(k)}$
4: **Solve** $KW^{(k)} = R^{(k)}$
5: **Set** $X^{(k+1)} = \underset{X \in \text{Span}(X^{(k)}, W^{(k)}, P^{(k)})}{\arg\min} r(X)$ thanks to Rayleigh–Ritz method.
6: $x_j^{(k+1)} = \sum_{j=1}^{m} \tau_j^{(k)} x_j^{(k)} + \alpha_j^{(k)} w_j^{(k)} + \gamma_j^{(k)} p_k^{(k)}$.
7: **Set** $p_j = \sum_{j=1}^{m} \alpha_j^{(i)} w_j^{(i)} + \gamma_j^{(i)} p_j^{(i)}$,
8: **end for**

The generated sequence $(X^{(k)})$ converges to the m eigenvectors that minimizes the Rayleigh ratio $r(x)$.

Remark 5.1

- *An efficient implementation of LOBPCG is given in [10]. This version is more stable than algorithm 7 but it is less natural.*
- *As mentioned in [13], algorithm 7 is available in various implementations. Without been exhaustive, we refer to the MATLAB implementation in* `https://github.com/lobpcg/blopex/` *and to the Python's library* `numpy`.

Acknowlegments : part of this work was done during visits of the second author at Laboratoire de Mathématiques et Application (UMR CNRS 7348) of the University of Poitiers, France.

This work was supported by the French National program LEFE (Les Enveloppes Fluides et l'Environnement).

References

[1] Hyam Abboud, Clara Al Kosseifi, and Jean-Paul Chehab. A stabilized bi-grid method for Allen–Cahn equation in finite elements. *Computational and Applied Mathematics*, 38(2):1–27, 2019.

[2] Samuel M Allen and John W Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta metallurgica*, 27(6):1085–1095, 1979.

[3] Samuel Miller Allen and John W Cahn. Ground state structures in ordered binary alloys with second neighbor interactions. *Acta Metallurgica*, 20(3):423–433, 1972.

[4] A Averbuch, A Cohen, and M Israeli. A stable and accurate explicit scheme for parabolic evolution equations. *URL http://ann. jussieu. fr/'cohen/par. ps. gz*, 1998.

[5] Matthieu Brachet and Jean-Paul Chehab. Stabilized times schemes for high accurate finite differences solutions of nonlinear parabolic equations. *Journal of Scientific Computing*, 69(3):946–982, 2016.

[6] Matthieu Brachet and Jean-Paul Chehab. Fast and stable schemes for phase fields models. *Computers & Mathematics with Applications*, 80(6):1683–1713, 2020.

[7] Jean-Paul Chehab. Damping, stabilization, and numerical filtering for the modeling and the simulation of time dependent PDEs. *Discrete & Continuous Dynamical Systems-S*, 14(8):2693, 2021.

[8] Bruno Costa, Lucia Dettori, D Gottlieb, and Roger Temam. Time marching multilevel techniques for evolutionary dissipative problems. *SIAM Journal on Scientific Computing*, 23(1):46–65, 2001.

[9] Amanda Emily Diegel. Numerical Analysis of Convex Splitting Schemes for Cahn-Hilliard and Coupled Cahn-Hilliard-Fluid-Flow Equations. 2015.

[10] Jed A Duersch, Meiyue Shao, Chao Yang, and Ming Gu. A robust and efficient implementation of LOBPCG. *SIAM Journal on Scientific Computing*, 40(5):C655–C676, 2018.

[11] Heike Emmerich. *The diffuse interface approach in materials science: thermodynamic concepts and applications of phase-field models*, volume 73. Springer Science & Business Media, 2003.

[12] DJ Eyre. Unconditionallly Stable One-step Scheme for Gradient Systems, June 1998, unpublished.

[13] Andrew Knyazev. Recent implementations, applications, and extensions of the Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG). *arXiv preprint arXiv:1708.08354*, 2017.

[14] Andrew V Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.

[15] Effrosini Kokiopoulou and Yousef Saad. Orthogonal neighborhood preserving projections. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[16] Hyun Geun Lee. Stability condition of the second-order SSP-IMEX-RK method for the Cahn–Hilliard equation. *Mathematics*, 8(1):11, 2019.

[17] Hyun Geun Lee. A non-iterative and unconditionally energy stable method for the Swift–Hohenberg equation with quadratic–cubic nonlinearity. *Applied Mathematics Letters*, 123:107579, 2022.

[18] Seunggyu Lee, Sungha Yoon, and Junseok Kim. Effective time step analysis of convex splitting schemes for the Swift–Hohenberg equation. *Journal of Computational and Applied Mathematics*, 419:114713, 2023.

[19] Sanjiva K Lele. Compact finite difference schemes with spectral-like resolution. *Journal of computational physics*, 103(1):16–42, 1992.

[20] Yibao Li, Darae Jeong, Jung-il Choi, Seunggyu Lee, and Junseok Kim. Fast local image inpainting based on the Allen–Cahn model. *Digital Signal Processing*, 37:65–74, 2015.

[21] Hong-lin Liao, Tao Tang, and Tao Zhou. On Energy Stable, Maximum-Principle Preserving, Second-Order BDF Scheme with Variable Steps for the Allen–Cahn Equation. *SIAM Journal on Numerical Analysis*, 58(4):2294–2314, 2020.

[22] Hailiang Liu and Peimeng Yin. High order unconditionally energy stable RKDG schemes for the Swift–Hohenberg equation. *Journal of Computational and Applied Mathematics*, 407:114015, 2022.

[23] Thanh T Ngo, Mohammed Bellalij, and Yousef Saad. The trace ratio optimization problem for dimensionality reduction. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2950–2971, 2010.

[24] Ömer Oruç. An efficient wavelet collocation method for nonlinear two-space dimensional Fisher–Kolmogorov–Petrovsky–Piscounov equation and two-space dimensional extended Fisher–Kolmogorov equation. *Engineering with Computers*, 36(3):839–856, 2020.

[25] Longzhao Qi and Yanren Hou. Error estimate of a stabilized second-order linear predictor–corrector scheme for the Swift–Hohenberg equation. *Applied Mathematics Letters*, 127:107836, 2022.

[26] Jie Shen and Xiaofeng Yang. Numerical approximations of Allen-Cahn and Cahn-Hilliard equations. *Discrete & Continuous Dynamical Systems*, 28(4):1669, 2010.

[27] Jian Su, Weiwei Fang, Qian Yu, and Yibao Li. Numerical simulation of Swift–Hohenberg equation by the fourth-order compact scheme. *Computational and Applied Mathematics*, 38:1–15, 2019.

[28] Ju Swift and Pierre C Hohenberg. Hydrodynamic fluctuations at the convective instability. *Physical Review A*, 15(1):319, 1977.

[29] Chenhui Zhang, Jie Ouyang, Cheng Wang, and Steven M Wise. Numerical comparison of modified-energy stable SAV-type schemes and classical BDF methods on benchmark problems for the functionalized Cahn-Hilliard equation. *Journal of Computational Physics*, 423:109772, 2020.