Interval Arithmetic for the Design of Electrical Engineering Systems

Benoit DELINCHANT
(e-mail : benoit.delinchant@inpg.fr)
Conception et Dimensionnement Intégrés
Laboratoire Electrotechnique de Grenoble
INPG/UJF-CNRS.UMR 5529
LEG/ENSIEG – BP 46,
38402 Saint Martin d'Hères, France

Abstract **- Electrical engineering system design requires more and more light models like analytical ones. Using interval arithmetic calculation on algebraic formulas brings several benefits like stability analysis, global optimization or tolerance design. Software based on these techniques have been developed to bring these functionalities to designer hands, especially for preliminary stages of design.**

Interval Arithmetic basis

Needs analytical formulas:

- Algebraic operators : +, –, *, /
- Standard functions : power, trigonometric, ln, exp, …

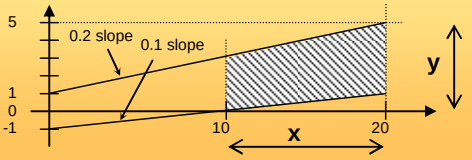
Algebraic operators:

- Addition : z=x+y, is defined by z=[x_inf+y_inf; x_sup+y_sup]
- substraction : z=x-y, is defined by z=[x_inf-y_sup; x_sup-y_inf]
- Multiplication : z=x*y, is defined by z=[min(A,B,C,D); max(A,B,C,D)]

where A=x_inf*y_inf, B=x_inf*y_sup, C=x_sup*y_inf, D=x_sup*y_sup

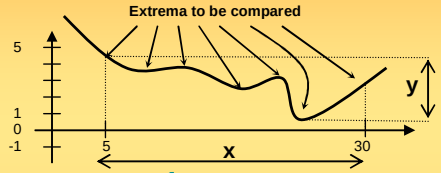
Example : y=a.x+b

- {a, b, x, y} are intervals : a=[0.1;0.2], b=[-1;1] and x=[10; 20]
- Then y = [0.1;0.2] x [10; 20] + [-1;1] = [1; 4] + [-1;1] = [0; 5] = y



Functions y = f(x)

- Decomposition into monotonous functions depending of its extrema.
- Then, bounds and each extrema are compared to extract resulting bounds



Example with standard functions y = sin(x)

- Algorithm :

If (x_rad ≥ π) : y=[-1, 1], end.
Moves x into]-π, π] with 2π translations
If (x contains π/2) : y_sup=1
If (x contains -π/2) : y_inf=-1
cuts into monotonous parts x ∪ {]-π, π/2],]-π/2, π/2],]π/2, π] }
Compares bounds value.

Component based software architecture

Generator :

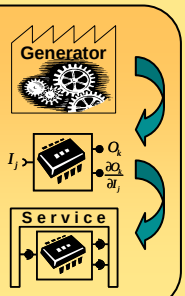
- analyzes algebraic equations of the device,
- produces interval arithmetic codes
- creates a software component

Component :

- Is able to compute itself autonomously,
- Is based on a norm.

Service :

- GUI to compute or plot intervals.
- Algorithm to produce a global optimization
- Algorithm for robust design with tolerances



Model stability checking

Rounding error :

- Intervals are computed with lower and upper rounding
- Propagation of 0 radius intervals gives model rounding error

Formulas stability :

- check the equations writing stability with rounding error
- Example : Needle like triangle area computation

$$\Delta_1 = \sqrt{s(s-a)(s-b)(s-c)} \quad with \quad s = \tfrac{1}{2}(a+b+c)$$

$$\Delta_2 = \tfrac{1}{4}\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))}$$

$\Delta_1$error = ±1.4 %

$\Delta_2$error = ±7.10^{-14} %

a=10^{14}
b=10^{14}
c=1.1

Toleranced design

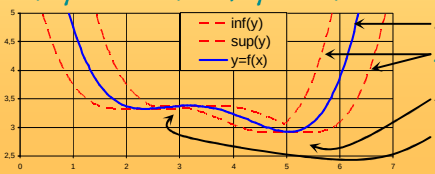
Constraining a bound or the radius :

- What is best inputs value which satisfy constraints and objective ?
- What is the more stable optimum ?

Model inputs / outputs

- Inputs : Middle value and Radius to avoid "improper intervals"
- Outputs : Lower and Upper bounds as well as Middle and Radius
- Jacobian : derivatives of each outputs depending on all inputs.

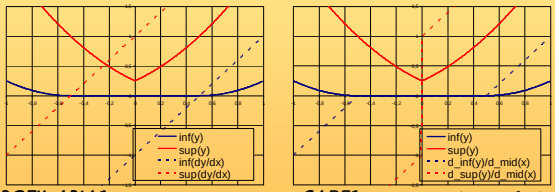
Example : toleranced optimization



Standard computing (radius=0)
Advanced Interval computing (radius=0.5)
Standard optimum (regardless optimum stability)
Radius minimal value (robust solution)

Sensitivity issues

- Interval arithmetic on derivative is different from derivative of the interval arithmetic.
- Comparison between two different implementations for $y=x^2$ with rad(x)=0.5 for mid(x) ∈ [-1, 1]



PROFIL / BIAS (Programmer's Runtime Optimized Fast Interval Library / Basic Interval Arithmetic Subroutines) Institute for Reliable Computing - Germany

CADES (Component Architecture for the Design of Engineering Systems)

- Discontinuities may brought troubles during gradient based optimization.

Global Optimization

The true global optimum !

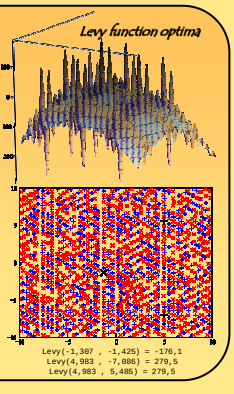
- Branch & Bound algorithm with constraints satisfaction checking.
  - Space is decomposed into parts with bounds = Intervals
  - If (0 ∈ [inf(gradient); sup(gradient)]) & (sup(constraints) < constraints_max) then decompose else reject part.
  - Until rad(parts) < tolerance

The true global optimum !

- Branch & Bound algorithm with constraints satisfaction checking.
- Each optimum can be found and then compared regarding stability

Perspectives

- Problem : Time explosion when variable number increase
- Solution : improve bounds approximation (affine arithmetic to avoid dependency loss)



Conclusions

An easy to use design framework has been improved with Interval Arithmetic capabilities to perform three complementary studies :

- Stability checking of formulas,
- Toleranced designs,
- Global optimization.

Interval analysis is a very interesting tool but is a complement to other since it has limitations :

- Lose variable dependencies, leading to bigger intervals than possible.
- May introduce sensitivity discontinuities.