Which Method for Solution of the System of Interval Equations Should we Choose?

A. Pownuk[1], J. Quezada[1], I. Skalna[2],
M.V. Rama Rao[3], A. Belina[4]

[1] The University of Texas at El Paso, El Paso, Texas, USA
[2] AGH University of Science and Technology, Krakow, Poland
[3] Vasavi College of Engineering, Hyderabad, India
[4] Silesian University of Technology, Gliwice, Poland

21th Joint UTEP/NMSU Workshop on Mathematics, Computer Science, and Computational Sciences

Outline

Solution of PDE

Parameter dependent Boundary Value Problem

$$A(p)u = f(p), u \in V(p), p \in P$$

Exact solution

$$\underline{u} = \inf_{p \in P} u(p), \overline{u} = \sup_{p \in P} u(p)$$

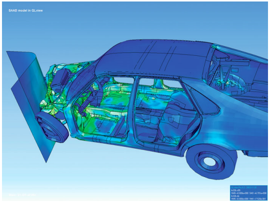
$$u(x,p) \in [\underline{u}(x), \overline{u}(x)]$$

Approximate solution

$$\underline{u}_h = \inf_{p \in P} u_h(p), \overline{u}_h = \sup_{p \in P} u_h(p)$$

$$u_h(x,p) \in [\underline{u}_h(x), \overline{u}_h(x)]$$

Mathematical Models in Engineering



High dimension $n > 10000$.
Linear and nonlinear equations.
Multiphysics (solid mechanics, fluid mechanics etc.)
Ordinary and partial differential equations, variational equations, variational inequalities, numerical methods, programming, visualizations, parallel computing etc.

Two point boundary value problem

Sample problem

$$\begin{cases} -(a(x)u'(x)) = f(x) \\ u(0) = 0, u(1) = 0 \end{cases}$$

and $u_h(x)$ is finite element approximation given by a weak formulation

$$\int_0^1 a(x)u_h'(x)v'(x)dx = \int_0^1 f(x)v(x)dx, \forall v \in V_h^{(0)}$$

or

$$a(u_h, v) = l(v), \forall v \in V_h^{(0)} \subset H_0^1$$

where $u_h(x) = \sum_{i=1}^{n} u_i\varphi_i(x)$ and $\varphi_i(x_j) = \delta_{ij}$.

The Finite Element Method

Approximate solution $\int_0^1 a(x)u_h'(x)v'(x)dx = \int_0^1 f(x)v(x)dx.$

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{n}\int_0^1 a(x)\varphi_i(x)\varphi_j(x)dxu_i - \int_0^1 f(x)\varphi_j(x)dx\right)v_j = 0$$

Final system of equations (for one element) $Ku = q$ where

$$K_{i,j} = \int_0^1 a(x)\varphi_i(x)\varphi_j(x)dx,\; q_i = \int_0^1 f(x)\varphi_i(x)dx$$

Calculations of the local stiffness matrices can be done in parallel.

Global Stiffness Matrix

Global stiffness matrix

$$\sum_{p=1}^{n}\left(\sum_{q=1}^{n}\sum_{e=1}^{n_e}\sum_{i=1}^{n_u^e}\sum_{j=1}^{n_u^e}U_{j,p}^{e}\int_{\Omega_e}a(x)\frac{\partial\varphi_i^e(x)}{\partial x}\frac{\partial\varphi_j^e(x)}{\partial x}dxU_{i,q}^{e}u_q - \right.$$

$$\left.\sum_{q=1}^{n}\sum_{e=1}^{n_e}\sum_{i=1}^{n_u^e}\sum_{j=1}^{n_u^e}U_{j,p}^{e}\int_{\Omega_e}f(x)\varphi_i^e(x)\varphi_j^e(x)dx\right)v_p = 0$$

Final system of equations

$$K(p)u = Q(p) \Rightarrow F(u,p) = 0$$

Computations of the global stiffness matrix can be done in parallel.

Solution Set

Nonlinear equation $F(u,p)=0$ for $p \in P$.

$$F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$$

Implicit function $u = u(p) \Leftrightarrow F(u,p)=0$

$$u(P) = \{u : F(u,p)=0, p \in P\}$$

Interval solution

$$\underline{u}_i = \min\{u : F(u,p)=0, p \in P\}$$

$$\overline{u}_i = \max\{u : F(u,p)=0, p \in P\}$$

Interval Methods

A. Neumaier, Interval Methods for Systems of Equations (Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1991.

Z. Kulpa, A. Pownuk, and I. Skalna, Analysis of linear mechanical structures with uncertainties by means of interval methods, Computer Assisted Mechanics and Engineering Sciences, 5, 443-477, 1998.

V. Kreinovich, A.V.Lakeyev, and S.I. Noskov. Optimal solution of interval linear systems is intractable (NP-hard). Interval Computations, 1993, 1, 6-14.

Optimization methods

Interval solution

$$\underline{u}_i = \min\{u(p) : p \in P\} = \min\{u : F(u,p) = 0, p \in P\}$$

$$\overline{u}_i = \max\{u(p) : p \in P\} = \max\{u : F(u,p) = 0, p \in P\}$$

$$\underline{u}_i = \begin{cases} \min u_i \\ F(u,p) = 0 \\ p \in P \end{cases}, \overline{u}_i = \begin{cases} \max u_i \\ F(u,p) = 0 \\ p \in P \end{cases}$$

KKT Conditions

Nonlinear optimization problem for $f(x) = x_i$

$$\begin{cases} \min_x f(x) \\ h(x) = 0 \\ g(x) \geq 0 \end{cases}$$

Lagrange function $L(x, \lambda, \mu) = f(x) + \lambda^T h(x) - \mu^T g(x)$
Optimality conditions can be solved by the Newton method.

$$\begin{cases} \nabla_x L = 0 \\ \nabla_\lambda L = 0 \\ \mu_i \geq 0 \\ \mu_i g_i(x) = 0 \\ h(x) = 0 \\ g(x) \geq 0 \end{cases}$$

KKT Conditions - Newton Step

$$F'(X)\Delta X = -F(X)$$

$$F'(X) = \begin{bmatrix} \left(\nabla_x^2 f(x) + \nabla_x^2 h(x) y\right)_{n\times n} & \nabla_x h(x)_{n\times m} & -I_{n\times n} \\ \left(\nabla_x h(x)\right)^T{}_{m\times n} & 0_{n\times m} & 0_{m\times n} \\ Z_{n\times n} & 0_{n\times m} & X_{m\times n} \end{bmatrix}$$

$$\Delta X = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}, X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$F(X) = -\begin{bmatrix} \nabla_x f(x) + \nabla_x h^T(x) y - z \\ h(x) \\ XYe - \mu_k e \end{bmatrix}$$

Steepest Descent Method

In order to find maximum/minimum of the function u it is possible to apply the steepest descent algorithm.

1. Given x_0, set $k = 0$.
2. $d^k = -\nabla f(x_k)$. If $d^k = 0$ then stop.
3. Solve $min_\alpha f(x_k + \alpha d^k)$ for the step size α_k. If we know second derivative H then $\alpha_k = \frac{d_k^T d_k}{d_k^T H(x_k) d_k}$.
4. Set $x_{k+1} = x_k + \alpha_k d_k$, update $k = k + 1$. Go to step 1.

I. Skalna and A. Pownuk, Global optimization method for computing interval hull solution for parametric linear systems, International Journal of Reliability and Safety, 3, 1/2/3, 235-245, 2009.

The Gradient

After discretization

$$Ku = q$$

Calculation of the gradient

$$Kv = \frac{\partial}{\partial p_k} q - \frac{\partial}{\partial p_k} Ku$$

where $v = \frac{\partial}{\partial p_k} u$.

Gradient Method and Sensitivity Analysis

A. Pownuk, Numerical solutions of fuzzy partial differential equation and its application in computational mechanics, in: M. Nikravesh, L. Zadeh and V. Korotkikh, (eds.), Fuzzy Partial Differential Equations and Relational Equations: Reservoir Characterization and Modeling, Physica-Verlag, 308-347, 2004.

Postprocessing of the interval solution.

$$\varepsilon = Cu$$

$$\sigma = D\varepsilon$$

Linearization

$$\Delta f(x) = f(x + \Delta x) - f(x) \approx f'(x)\Delta x$$

Derivative can be calculated numerically.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

The method can be used together with incremental formulation of the Finite Element Method.

$$K(p)\Delta u = \Delta Q(p)$$

Monte Carlo Simulation/Search Method

Monte Carlo Method (inner approximation of the solution set)

$$u(P) \approx Hull(\{u : K(p)u = Q(p), p \in \{random\ values\ from\ P\}\})$$

Search Method. $P \approx \{special\ \ points\}$

$$u(P) \approx Hull(\{u : K(p)u = Q(p), p \in \{special\ \ points\}\})$$

Vertex Method

$$u(P) \approx Hull(\{u : K(p)u = Q(p), p \in \{set\ \ of\ \ vertices\}\})$$

Cauchy Based Monte Carlo Simulation

$$\rho_\Delta(x) = \frac{\Delta}{\pi} \cdot \frac{1}{1 + x^2/\Delta^2}.$$

when $\Delta x_i \sim \rho_{\Delta_i}(x)$ are indep., then

$\Delta y = \sum\limits_{i=1}^{n} c_i \cdot \Delta x_i \sim \rho_\Delta(x)$, with $\Delta = \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.

Thus, we simulate $\Delta x_i^{(k)} \sim \rho_{\Delta_i}(x)$; then,

$\Delta y^{(k)} \stackrel{\text{def}}{=} \widetilde{y} - f(\widetilde{x}_1 - \Delta x_1^{(k)}, \ldots) \sim \rho_\Delta(x)$.

Maximum Likelihood method can estimate Δ:

$\prod\limits_{k=1}^{N} \rho_\Delta(\Delta y^{(k)}) \to \max$, so $\sum\limits_{k=1}^{N} \frac{1}{1 + (\Delta y^{(k)})^2/\Delta^2} = \frac{N}{2}$.

To find Δ from this equation, we can use, e.g., the bisection method for $\underline{\Delta} = 0$ and $\overline{\Delta} = \max\limits_{1 \le k \le N} |\Delta y^{(k)}|$.

Theory of perturbations

J. Skrzypczyk1, A. Belina, FEM ANALYSIS OF UNCERTAIN SYSTEMS WITH SMALL GP-FUZZY TRIANGULAR PERTURBATIONS, Proceedings of the 13th International Conference on New Trends in Statics and Dynamics of Buildings October 15-16, 2015 Bratislava, Slovakia Faculty of Civil Engineering STU Bratislava Slovak Society of Mechanics SAS

$$A = A_0 + \varepsilon^1 A_1 + \varepsilon^2 A_2 + \ldots$$

J.D. Cole, Perturbation methods in applied mathematics, Bialsdell, 1968.

Interval Boundary Element Method

T. Burczynski, J. Skrzypczyk, Fuzzy aspects of the boundary element method, Engineering Analysis with Boundary Elements, Vol.19, No.3, pp. 209216, 1997

$$cu = \int_{\partial\Omega} \left(G \frac{\partial u}{\partial n} - \frac{\partial G}{\partial n} u \right) dS$$

Element by Element Method

Muhanna, R. L. and R. L. Mullen. Uncertainty in Mechanics ProblemsInterval-Based Approach, Journal of Engineering Mechanics 127(6), 557-566, 2001.

$$\Pi^* = \frac{1}{2}\{\boldsymbol{U}\}^T[\boldsymbol{K}]\{\boldsymbol{U}\} - \{\boldsymbol{U}\}^T\{\boldsymbol{P}\} + \lambda_1^T([C]\{U\}\text{-}\{V\}) + \lambda_2^T([B_1]\{U\}\text{-}\{\boldsymbol{\kappa}\}) \quad (40)$$

Invoking the stationarity of Π^*, that is $\delta\Pi^* = 0$, and considering Eq. (40), we obtain

$$\left(\begin{pmatrix} \boldsymbol{0} & C^T & B_1^T & 0 \\ C & 0 & 0 & 0 \\ B_1 & 0 & 0 & -I \\ 0 & 0 & -I & 0 \end{pmatrix} + \begin{bmatrix} A \\ 0 \\ 0 \\ 0 \end{bmatrix} [\boldsymbol{D}][A \quad 0 \quad 0 \quad 0] \right) \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{\lambda_1} \\ \boldsymbol{\lambda_2} \\ \kappa \end{pmatrix} = \begin{pmatrix} \boldsymbol{P_c} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{Bmatrix} M \\ 0 \\ 0 \\ 0 \end{Bmatrix} \{\boldsymbol{\delta}\} \quad (41)$$

Parametric Linear System

I. Skalna, A method for outer interval solution of systems of linear equations depending linearly on interval parameters, Reliable Computing, 12, 2, 107-120, 2006.

Table I. Algorithm

$$R := \mathrm{mid}(A([p]))^{-1};$$

$$\tilde{x} := R \cdot \mathrm{mid}(b([p]));$$

$$[Z]_i = \sum_{j=1}^{n} R_{ij} \left(\omega(0,\, j) - \sum_{k=1}^{n} \tilde{x}_k \omega(j,\, k) \right)^{\mathrm{T}} [p]$$

$$[D]_{ij} := \left(\sum_{\nu=1}^{n} R_{i\nu} \omega(\nu,\, j) \right)^{\mathrm{T}} [p];$$

$$\mathrm{outer} := \tilde{x} + [-1,\, 1] \langle [D] \rangle^{-1} |[Z]|$$

The use of diagonal matrix

A. Neumaier and A. Pownuk, Linear Systems with Large Uncertainties, with Applications to Truss Structures, Journal of Reliable Computing, 13(2), 149-172, 2007.

$$K = A^T * D * A$$

Element by element method

M. V. Rama Rao, R. L. Muhanna, and R. L. Mullen. Interval Finite Element Analysis of Thin Plates 7th International Workshop on Reliable Engineering Computing, At Ruhr University Bochum, Germany, 2016

$$[\boldsymbol{K}] = \begin{bmatrix} \boldsymbol{K}_1^{(e)} & & & \\ & \boldsymbol{K}_2^{(e)} & & \\ & & \boldsymbol{K}_3^{(e)} & \\ & & & \boldsymbol{...} \end{bmatrix} = \begin{bmatrix} A_1^{(e)} & A_2^{(e)} & A_3^{(e)} & ... \end{bmatrix} \begin{bmatrix} diag(\Lambda_1 \boldsymbol{\alpha}_1) & & & \\ & diag(\Lambda_2 \boldsymbol{\alpha}_2) & & \\ & & diag(\Lambda_3 \boldsymbol{\alpha}_3) & \\ & & & ... \end{bmatrix} \begin{Bmatrix} A_1^{T(e)} \\ A_2^{T(e)} \\ A_3^{T(e)} \\ ... \end{Bmatrix} \quad (34)$$

This can be denoted as

$$[\boldsymbol{K}] = [A][\boldsymbol{D}][A]^T \quad (35)$$

Comparison between the diffrent methods

$$Comp.Complexity(Method1) < Comp.Complexity(Method2)$$

$$Accuracy(Method1) < Accuracy(Method2)$$

Accuracy include also information about guaranteed accuracy.

$$PossibleApplications(Method1) < PossibleApplications(Method2)$$

$$Scalability(Method1) < Scalability(Method2)$$

Scalability include information about parallelization.

How to find the best method?

Example:
method 1: linearization
method 2: Monte Carlo

The problem is small

$$EasyToImplement(Method1) < EasyToImplement(Method2)$$

$$Accuracy(Method1) < Accuracy(Method2)$$

Better method is the method 2, i.e. the Monte Carlo method.

What to do in the conflict situations?

Example:
method 1: linearization
method 2: interval methods

$$Comp.Complexity(Method1) < Comp.Complexity(Method2)$$

$$Accuracy(Method1) > Accuracy(Method2)$$

If the main requremant is guaranteed solution,
then we can use the interval methods.

What to do in the conflict situations?

Example:
method 1: linearization
method 2: interval methods

$$Comp.Complexity(Method1) < Comp.Complexity(Method2)$$

$$Accuracy(Method1) > Accuracy(Method2)$$

If the problem is very large or nonlinear,
then it is not possible to apply the interval methods
and it is necessary to use linearization.

What to do in the conflict situations? Linear model

Example:
method 1: m_1
method 2: m_2
Total score

$$\mu_1 = \sum_i w_i f_i(m_1)$$

$$\mu_2 = \sum_i w_i f_i(m_2)$$

If $\mu_1 > \mu_2$ then we need to pick the method 1.
If $\mu_1 < \mu_2$ then we need to pick the method 2.

Example:
method 1: m_1
method 2: m_2
Total score

$$\mu_1 = \Phi(f_1(m_1), f_2(m_1), ..., f_k(m_1))$$

$$\mu_2 = \Phi(f_1(m_1), f_2(m_1), ..., f_k(m_1))$$

If $\mu_1 > \mu_2$ then we need to pick the method 1.
If $\mu_1 < \mu_2$ then we need to pick the method 2.
or more generally

$$\Omega(f_1(m_1), ..., f_k(m_1), f_1(m_2), ..., f_k(m_2))) > 0$$

Conclusions

- Interval equations can be solved by using many diffrent methods.
- Every method has some advantages and disadvantages.
- In order to choose the optimal method it is necessary to consider many diffrent features of every computational method.