A Posteriori Error Bounds for Two Point Boundary Value Problem with Uncertain Parameters

Andrew Pownuk
Jazmin Quezada

The University of Texas at El Paso

Sixteenth New Mexico Analysis Seminar,
Department of Mathematical Sciences,
New Mexico State University,
Las Cruces, May 21, 2017.

Outline

Errors in numerical calculations

Boundary value problem.

$$L(u) = f, u \in V$$

u - exact solution, u_h - approximate solution.
Approximation error $\|u - u_h\| = \|e\|$.
Parameter dependent boundary value problem.

$$L(u, p) = f, u \in V$$

$u(p)$ - parameter dependent exact solution,
$u_h(p)$ - parameter dependent approximate solution.
Maximal approximation error

$$\sup_{p \in P} \|u(p) - u_h(p)\|_E = \sup_{p \in P} \|e(p)\|_E = \overline{\|e\|}_E$$

.

Extreme values of the solution

Parameter dependent boundary value problem.

$$L(u, p) = f, u \in V$$

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

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

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

Interval parameters (worst case analysis)

Solution of the equation with interval parameters for given x can be defined as the following set:

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

$$= \diamond\{u(x, p_1, ..., p_m) : p_1 \in [\underline{p}_1, \overline{p}_1], ..., p_m \in [\underline{p}_m, \overline{p}_m]\}$$

where $[\underline{p}_1, \overline{p}_1], ..., [\underline{p}_m, \overline{p}_m]$ are interval parameters (for example E, A, n etc.) and $\diamond B$ is the smallest interval that contains the set B.
In presented example uncertain parametrs may be E, n, L etc.

Steepest Descent Method

In order to find maximum/minimum of the function u it is possible to apply a modified version of 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.

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 $\varphi_i(x_j) = \delta_{ij}$.

Example

Tension-compression problem

$$\begin{cases} -(E(x)A(x)u'(x))' = n(x) \\ u(0) = 0, u(L) = 0 \end{cases}$$

E is a Young modulus and A is an area of cross-section. $u_h(x)$ is finite element approximation given by a weak formulation.

$$\int_0^L E(x)A(x)u_h'(x)v'(x)dx = \int_0^L n(x)v(x)dx, \forall v \in V_h^{(0)}$$

or

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

The Finite Element Method

Weak formulation

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

Approximate solution

$$u_h = \sum_{i=1}^{n} u_i \varphi_i(x), \quad v = \sum_{j=1}^{n} v_j \varphi_j(x)$$

$$\frac{\partial u_h}{\partial x} = \sum_{i=1}^{n} u_i \frac{\partial \varphi_i(x)}{\partial x}$$

$$\frac{\partial v}{\partial x} = \sum_{j=1}^{n} v_j \frac{\partial \varphi_j(x)}{\partial x}$$

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

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

Final system of equations

$$Ku = q$$

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

The Gradient

After discretization

$$Ku = q$$

Calculation of the gradient

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

where $v = \frac{\partial}{\partial p_k} u$.
Presented gradient can be used in the optimization process. Derivative with respect to different parameters p_k can ba calculated simultaneously by using parallel computing.

FEM approximation

The error of the solution can be approximated by the following inequality

$$\|u - u_h\|_E \leq \|u - v\|_E, \forall v(x) \in V_h^{(0)} \subset H_0^1$$

this means that the finite element solution $u_h \in V_h^{(0)}$ is the best approximation of the solution u by the function in $V_h^{(0)}$, where

$$\|u - u_h\|_E^2 = \int_0^1 a(x) \left(u'(x) - u_h'(x)\right)^2 dx$$

FEM approximation

(An apriori error estimate). Let u and u_h be the solutions of the Dirichlet problem (BVP) and the finite element problem (FEM), respectively. Then there exists an interpolation constant C_i, depending only on $a(x)$, such that

$$\|u - u_h\|_E \leq C_i \|hu''\|_a$$

where

$$\|u\|_a^2 = \int_0^1 a(x)\,(u(x))^2\,dx$$

This, however, requires that the exact solution $u(x)$ is known.

FEM approximation

(a posteriori error estimate). There is an interpolation constant C_i depending only on $a(x)$ such that the error in finite element approximation of the Dirichlet boundary value problem (BVP) satisfies

$$\|u - u_h\|_E \leq C_i \sqrt{\int_0^1 \frac{1}{a(x)} h^2(x) R^2(u_h(x)) dx}$$

where $h(x)$ is some weight and

$$R_h(u_h(x)) = f(x) + (a(x)u_h'(x))'$$

is the residual error and u_h is a solution of the Finite Element Method.

Adaptivity

Assume that one seeks an error bound less that a given error tolerance TOL:

$$\|e(x)\|_E \leq TOL$$

Then one may use the following steps as a mesh refinement strategy:

- (i) Make an initial partition of the interval.
- (ii) Compute the corresponding FEM solution $u_h(x)$ and residual $R(u_h(x))$.
- (iii) If $\|e(x)\|_E > TOL$ refine the mesh in the places for which $\frac{1}{a(x)}R^2(u_h(x))$ is large and perform the steps (ii) and (iii) again.

Adaptivity

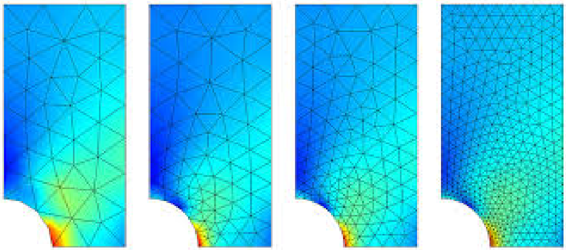

Figure : Adaptive FEM.

Computational method

1. Set some initial grid points $x_0, x_1, \ldots, x_n$ and set $i = 0$.
2. For given sets of grid points
   $x_0^{min,i}, x_1^{min,i}, \ldots, x_n^{min,i}$ for $\underline{u}_h$
   $x_0^{max,i}, x_1^{max,i}, \ldots, x_n^{max,i}$ for $\overline{u}_h$
   find the approximate solutions $\underline{u}_h^i = u_h(p_{min}^i)$, $\overline{u}_h^i = u_h(p_{max}^i)$.
3. If $\|\underline{u}_h^i - \underline{u}_h^{i-1}\| < \varepsilon_1$ and $\|\overline{u}_h^i - \overline{u}_h^{i-1}\| < \varepsilon_2$ then stop. The solution is $\underline{u} \approx \underline{u}_h^i, \overline{u} \approx \overline{u}_h^i$.
4. If $i > i_{max}$ then the method doesn't converge and stop.
5. Find new sets of grid points
   $x_0^{min,i+1}, x_1^{min,i+1}, \ldots, x_n^{min,i+1}$ for $\underline{u}_h$
   $x_0^{max,i+1}, x_1^{max,i+1}, \ldots, x_n^{max,i+1}$ for $\overline{u}_h$
   that minimize error estimator for $\|e\|_E$ and compute new solutions $\underline{u}_h^{i+1} = u_h(p_{min}^{i+1})$, $\overline{u}_h^{i+1} = u_h(p_{max}^{i+1})$ set $i := i + 1$ and go to the point 2.

KKT Conditions

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

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

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.

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

Linearization-Based Algorithm

- *We know:* an algorithm $f(x_1, \ldots, x_n)$ and values $\widetilde{y}_i$ and Δ_i.
- *We need to find:* the range of values $f(x_1, \ldots, x_n)$ when $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- *Algorithm:*
  1) first, we compute $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$;
  2) then, for each i from 1 to n, we compute

$$y_i = f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n);$$

  3) after that, we compute $\overline{y} = \widetilde{y} + \sum\limits_{i=1}^{n} |y_i - \widetilde{y}|$ and

$$\underline{y} = \widetilde{y} - \sum_{i=1}^{n} |y_i - \widetilde{y}|.$$

Taking Model Inaccuracy into Account

- We rarely know the exact dependence $y = f(x_1, \ldots, x_n)$.
- We have an approx. model $F(x_1, \ldots, x_n)$ w/known accuracy ε: $|F(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)| \le \varepsilon$.
- *We know:* an algorithm $F(x_1, \ldots, x_n)$, accuracy ε, values $\widetilde{x}_i$ and Δ_i.
- *Find:* the range $\{f(x_1, \ldots, x_n) : x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]\}$.
- If we use the approximate model in our estimate, we get $\overline{Y} = \widetilde{Y} + \sum\limits_{i=1}^{n} |Y_i - \widetilde{Y}|$.
- Here, $|\widetilde{Y} - \widetilde{y}| \le \varepsilon$ and $|Y_i - y_i| \le \varepsilon$, so $|\overline{y} - \overline{Y}| \le (2n+1) \cdot \varepsilon$.
- Thus, we arrive at the following algorithm.

Resulting Algorithm

- *We know:* an algorithm $F(x_1, \ldots, x_n)$, accuracy ε, values $\widetilde{x}_i$ and Δ_i.
- *Find:* the range $\{f(x_1, \ldots, x_n) : x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]\}$.
- *Algorithm:*
  1) compute $\widetilde{Y} = Y(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ and
  $$Y_i = F(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n).$$
  2) compute $\overline{B} = \widetilde{Y} + \sum\limits_{i=1}^{n} |Y_i - \widetilde{Y}| + (2n+1) \cdot \varepsilon$ and
  $\underline{B} = \widetilde{Y} - \sum\limits_{i=1}^{n} |Y_i - \widetilde{Y}| - (2n+1) \cdot \varepsilon.$
- *Problem:* when n is large, then, even for reasonably small inaccuracy ε, the value $(2n+1) \cdot \varepsilon$ is large.
- *What we do:* we show how we can get better estimates for $\overline{y}$.

How to Get Better Estimates: Idea

- One possible source of model inaccuracy is discretization (e.g., FEM).
- When we select a different combination of parameters, we get an *unrelated* value of inaccuracy.
- So, let's consider approx. errors $\Delta y \stackrel{\text{def}}{=} F(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)$ as *independent* random variables.
- What is a probability distribution for these random variables? We know that $\Delta y \in [-\varepsilon, \varepsilon]$.
- We do not have any reason to assume that some values from this interval are more probable than others.
- So, it is reasonable to assume that all the values are equally probable: a uniform distribution.
- For this uniform distribution, the mean is 0, and the standard deviation is $\sigma = \dfrac{\varepsilon}{\sqrt{3}}$.

How to Get a Better Estimate for $\widetilde{y}$

- In our main algorithm, we apply the computational model F to $n+1$ different tuples.
- Let's also compute $M \stackrel{\text{def}}{=} F(\widetilde{x}_1 - \Delta_1, \ldots, \widetilde{x}_n - \Delta_n)$.
- In linearized case, $\widetilde{y} + \sum\limits_{i=1}^{n} y_i + m = (n+2) \cdot \widetilde{y}$, so

  $\widetilde{y} = \dfrac{1}{n+2} \cdot \left(\widetilde{y} + \sum\limits_{i=1}^{n} y_i + m \right)$, and we can estimate $\widetilde{y}$ as

  $$\widetilde{Y}_{\text{new}} = \frac{1}{n+2} \cdot \left(\widetilde{Y} + \sum_{i=1}^{n} Y_i + m \right).$$

- Here, $\Delta \widetilde{y}_{\text{new}} = \dfrac{1}{n+2} \cdot \left(\Delta \widetilde{y} + \sum\limits_{i=1}^{n} \Delta y_i + \Delta m \right)$, so its variance is $\sigma^2 \left[\widetilde{Y}_{\text{new}} \right] = \dfrac{\varepsilon^2}{3 \cdot (n+2)} \ll \dfrac{\varepsilon^2}{3} = \sigma^2 \left[\widetilde{Y} \right]$.

Estimation of σ^2

- Let us compute $\overline{Y}_{\text{new}} = \widetilde{Y}_{\text{new}} + \sum_{i=1}^{n} |Y_i - \widetilde{Y}_{\text{new}}|$.
- Here, when $s_i \in \{-1, 1\}$ are the signs of $y_i - \widetilde{y}$, we get:

$$\overline{y} = \widetilde{y} + \sum_{i=1}^{n} s_i \cdot (y_i - \widetilde{y}) = \left(1 - \sum_{i=1}^{n} s_i\right) \cdot \widetilde{y} + \sum_{i=1}^{n} s_i \cdot y_i.$$

- Thus, $\Delta\overline{y}_{\text{new}} = \left(1 - \sum_{i=1}^{n} s_i\right) \cdot \Delta\widetilde{y}_{\text{new}} + \sum_{i=1}^{n} s_i \cdot \Delta y_i$, so

$$\sigma^2 = \left(1 - \sum_{i=1}^{n} s_i\right)^2 \cdot \frac{\varepsilon^2}{3 \cdot (n+2)} + \sum_{i=1}^{n} \frac{\varepsilon^2}{3}.$$

- Here, $|s_i| \le 1$, so $\left|1 - \sum_{i=1}^{n} s_i\right| \le n + 1$, and

$$\sigma^2 \le \frac{\varepsilon^2}{3} \cdot (2n + 1).$$

Using $\widetilde{Y}_{\rm new}$ (cont-d)

- We have $\Delta \overline{y}_{\rm new} = \left(1 - \sum\limits_{i=1}^{n} s_i\right) \cdot \Delta \widetilde{y}_{\rm new} + \sum\limits_{i=1}^{n} s_i \cdot \Delta y_i$.
- Due to the Central Limit Theorem, $\Delta \overline{y}_{\rm new}$ is $\approx$ normal.
- We know that $\sigma^2 \le \dfrac{\varepsilon^2}{3} \cdot (2n+1)$.
- Thus, with certainty depending on k_0, we have

$$\overline{y} \le \overline{Y}_{\rm new} + k_0 \cdot \sigma \le \overline{Y}_{\rm new} + k_0 \cdot \frac{\varepsilon}{\sqrt{3}} \cdot \sqrt{2n+1}:$$

  - with certainty 95% for $k_0 = 2$,
  - with certainty 99.9% for $k_0 = 3$, etc.
- Here, inaccuracy grows as $\sqrt{2n+1}$.
- This is much better than in the traditional approach, where it grows $\sim 2n+1$.

Resulting Algorithm

- *We know:* $F(x_1, \ldots, x_n)$, ε, $\widetilde{x}_i$ and Δ_i.
- *We want:* to find the range of $f(x_1, \ldots, x_n)$ when $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- *Algorithm:*
  1) compute $\widetilde{Y} = F(\widetilde{x}_1, \ldots, \widetilde{x}_n)$,
  $$M = F(\widetilde{x}_1 - \Delta_1, \ldots, \widetilde{x}_n - \Delta_n), \text{ and}$$
  $$Y_i = F(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n);$$
  2) compute $\widetilde{Y}_{\text{new}} = \frac{1}{n+2} \cdot \left(\widetilde{Y} + \sum_{i=1}^{n} Y_i + M\right)$,
  $$\overline{b} = \widetilde{Y}_{\text{new}} + \sum_{i=1}^{n} \left|Y_i - \widetilde{Y}_{\text{new}}\right| + k_0 \cdot \sqrt{2n+1} \cdot \frac{\varepsilon}{\sqrt{3}};$$
  $$\underline{b} = \widetilde{Y}_{\text{new}} - \sum_{i=1}^{n} \left|Y_i - \widetilde{Y}_{\text{new}}\right| - k_0 \cdot \sqrt{2n+1} \cdot \frac{\varepsilon}{\sqrt{3}}.$$

A Similar Improvement Is Possible for the Cauchy Method

- In the Cauchy method, we compute $\widetilde{Y}$ and the values

$$Y^{(k)} = F(\widetilde{x}_1 + \eta_1^{(k)}, \ldots, \widetilde{x}_n + \eta_n^{(k)}).$$

- We can then compute the improved estimate for $\widetilde{y}$, as:

$$\widetilde{Y}_{\text{new}} = \frac{1}{N+1} \cdot \left(\widetilde{Y} + \sum_{k=1}^{N} Y^{(k)} \right).$$

- We can now use this improved estimate when estimating the differences $\Delta y^{(k)}$: namely, we compute

$$Y^{(k)} - \widetilde{Y}_{\text{new}}.$$

Conclusions

- Presented method allows to find the solution of the two point boundary value problem with uncertain parameters.
- The method takes into account two types of error in numerical solution: approximation errors and uncertainty in the initial data.
- In order to speed up the calculations parallel computing can be applied.
- Similar methodology can be applied for the solution of different types of differential equations.
- The method can be applied for the solution of large scale engineering (solid mechanics, oil engineering, CFM etc.) and scientific problems with uncertain parameters.