Worst case analysis of mechanical structures by interval methods

Arnold Neumaier
(University of Vienna, Austria)

(joint work with Andrzej Pownuk)

Safety

Safety studies in structural engineering are supposed to guard against failure in all reasonable situations encountered during the lifetime of a structure. The uncertainty present in assessing what is 'reasonable' has been under discussion since the dawn of history.

How can we know what will happen to us
when the LORD alone decides? (Proverbs 20:24)

Experience has shown that the LORD decides not arbitrarily, but in accordance with the laws of Nature,

Thus one way of knowing what will happen to us is to compute worst case bounds on critical response variables, given time-proven finite element models together with worst case bounds on the uncertainties of the input variables.

This leads to finite element calculations involving interval parameters.

Finite element structural analysis

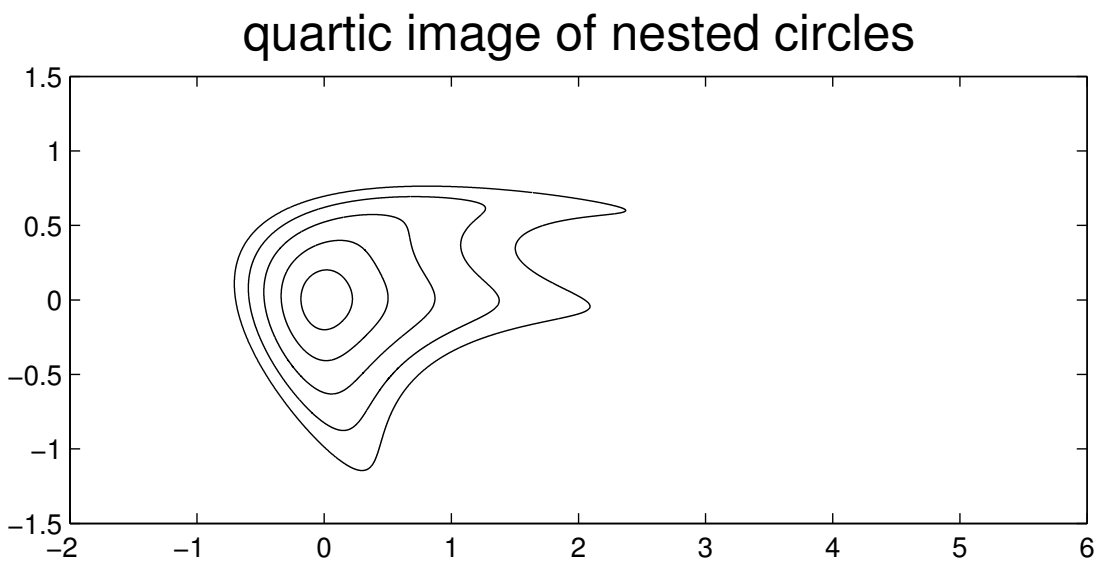
The finite element analysis of mechanical structures amounts in many cases to the solution of a large and sparse linear system with a symmetric, positive definite coefficient matrix.

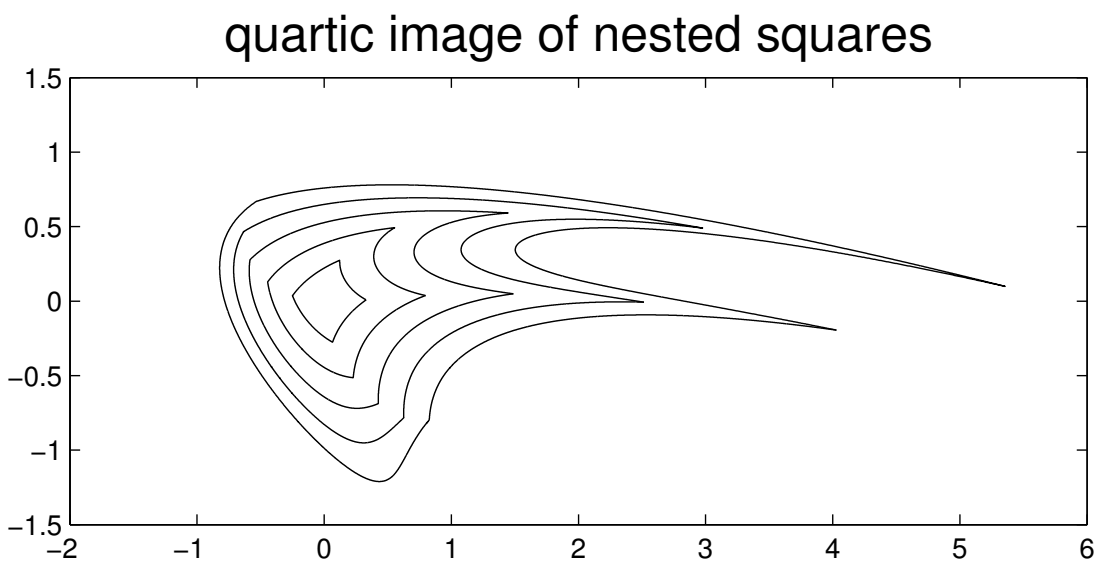
Uncertainties in the material parameters or the execution of a given design result in linear systems with uncertain coefficients.

However, as Cramer's rule shows, the uncertainties enter nonlinearly into the equations.

- **FEM equations become nonconvex when data are uncertain**
- **Several percent errors in elasticiy modules, and much larger errors in forces are not uncommon**
- **Current safety regulation laws require worst case analysis**

A nonconvex toy problem





The graphs depict the image of a family of concentric circles and squares (indicating increasing amounts of uncertainty) under the harmless looking function $y = F(x)$ defined by only two nonlinear operations (squaring),

$$u = x_2 - x_1^2, \quad z_1 = cx_1 - su, \quad z_2 = sx_1 + cu,$$

$$v = z_2 - z_1^2, \quad y_1 = cz_1 - sv, \quad y_2 = sz_1 + cv,$$

for $c = 0.4, s = \sqrt{1 - c^2}$.

Typical finite element computations are far more complex.

Large uncertainties imply large nonlinearities and require new methods.

- **Monte Carlo techniques underestimate worst case**
- **Monotonicity-based methods underestimate worst case**
- **Local optimization methods underestimate worst case**
- **$\Rightarrow$ interval analysis, global optimization**

The goal verification problem

A goal verification problem is the quest for verifying that for all parameter combinations in some feasible region, a family of constraints are satisfied, or exhibiting a feasible parameter combination for which some constraint is violated.

In many applications involving the design of a structure, a factory, or a machine, it is important that certain goals are met under a variety of conditions that cannot be known in advance.

Violating the goals may mean physical danger (risk of failure, often life-threatening) or financial danger (loss of money).

Generally, these conditions determining the goal can be specified in terms of a vector x of parameters whose components are unknown.

If x is known to lie in a box $\mathbf{x} = [\underline{x}, \overline{x}]$, and all choices of $x \in C$ are meaningful scenarios, a deterministic worst case analysis is appropriate.

Given the conditions x, the goals are assumed to be expressible in the form of vector constraints

$$F(x) \in \text{int } \mathbf{F} \quad \textbf{for all } x \in \mathbf{x}, \tag{1}$$

where F is a vector-valued function, int $\mathbf{F}$ is an open box of acceptable values of F, and int denotes the interior.

We call (1) the safety constraint(s) since, in the majority of applications, their satisfaction implies that it is safe (for both the designing and the using party) to build and use the structure, factory, or machine, while violation of (1) implies potential danger.

The goal verification problem is to show that

$$F(x) \in \text{int } \mathbf{F} \quad \textbf{for all } x \in \mathbf{x}, \tag{2}$$

or find a counterexample.

The fact that there are infinitely many constraints in (2) makes the problem hard and nonstandard.

Past practice is to check (2) only for a number of randomly or systematically generated sample cases.

This makes it quite possible that the worst case is overlooked. For safety critical applications, a complete search seems imperative, though it is usually regarded as infeasible to do.

Condition (2) is equivalent to checking that the range of F over the box x is contained in the interior of F. In principle, this can be checked by a computation of the range. Using interval analysis, one can often get fairly cheaply enclosures for the range.

However, the wrapping effect produces often overly pessimistic enclosures.

Moreover, if the computation of $F(x)$ involves the solution of linear systems (as in finite element applications), the enclosure algorithms may even fail completely due to overestimation in intermediate results.

Linear systems with uncertain coefficients

We consider the uncertain linear system

$$B(x)u(x) = b(x),$$

where the coefficient matrix $B(x)$ and/or the right hand side $b(x)$ depend on a parameter vector $x \in \mathbf{x}$.

For simplicity, we assume a single safety constraint, expressed in terms of the displacement vector $u(x)$,

$$F(x, u(x)) < 0.$$

Centered form approach

To solve $B(x)u(x) = b(x)$, we choose a center x_0 and write $x = x_0 + s$, $s \in \mathbf{s} = \mathbf{x} - x_0$.

For an arbitrary preconditioning matrix J, we compute enclosures

$$JB(x_0 + Ds) \in \mathbf{B}_0 + \mathbf{D}\sum \mathbf{B}_l s_l \quad \textbf{for all } s \in \mathbf{s},$$

$$Jb(x_0 + Ds) \in \mathbf{b}_0 + \mathbf{D}\sum \mathbf{b}_l s_l \quad \textbf{for all } s \in \mathbf{s},$$

$$\{u_0 \mid B_0 u_0 \in \mathbf{b}_0 \textbf{ for some } B_0 \in \mathbf{B}_0\} \subseteq \mathbf{u}_0.$$

If an interval enclosure

$$\left(\mathbf{B}_0 + \mathbf{D}\sum \mathbf{B}_l \mathbf{s}_l\right)^{-1} \subseteq \mathbf{S}$$

exists and

$$\mathbf{X} := \mathbf{S}[\mathbf{b}_1 - \mathbf{B}_1\mathbf{u}_0, \ldots, \mathbf{b}_n - \mathbf{B}_n\mathbf{u}_0],$$

then

$$u(x) \in \mathbf{u}_0 + \mathbf{X}(x - x_0) \quad \textbf{for all } x \in \mathbf{x}.$$

This can be used to check the safety constraint $F(x, u(x)) < 0$ **by another centered form.**

This works well if uncertainties are only in the right hand side (Muhanna & McMullen**), but not for uncertainties in the coefficient matrix.**

A counterexample

$$B(x) = \frac{1}{2}\begin{pmatrix} x_1 + x_2 & x_1 - x_2 \\ x_1 - x_2 & x_1 + x_2 \end{pmatrix}, \quad x \in \mathbf{x} = \begin{pmatrix} [0.5, 1.5] \\ [0.5, 1.5] \end{pmatrix}$$

With $x_0 = \operatorname{mid} \mathbf{x} = \begin{smallmatrix} 1 \\ 1 \end{smallmatrix}$ **, we have**

$$\mathbf{s} = \begin{pmatrix} [-0.5, 0.5] \\ [-0.5, 0.5] \end{pmatrix}, \quad J = B(x_0)^{-1} = I,$$

$$\mathbf{B}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{B}_1 = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \quad \mathbf{B}_2 = \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix},$$

$$\mathbf{B}_0 + \mathbf{B}_1\mathbf{s}_1 + \mathbf{B}_2\mathbf{s}_2 = \begin{pmatrix} [0.5, 1.5] & [-0.5, 0.5] \\ [-0.5, 0.5] & [0.5, 1.5] \end{pmatrix}$$

contains the singular matrix $\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$ **although**

$\det B(x) = x_1 x_2 > 0$ **for all** $x \in \mathbf{x}$**.**

In higher dimensions, the same problem tends to appear already for much smaller uncertainties.

Finite element applications therefore call for a modified approach, which exploits the special form of the finite element equations.

Representing FEM matrices

In many finite element problems, the only uncertainty in the coefficient matrix is in the element stiffness coefficients x_l.
(Additional uncertainty in the forces = right hand sides is allowed, too.)

For example, in a truss structure,

$$x_l = E_l a_l / L_l > 0$$

where l is the element index, E_l the Young modulus describing material properties of the lth bar, a_l its cross section area and L_l its length.

In general, the coefficient matrix depends both on the element stiffness coefficients and on lengths and angles, but if the geometry is assumed fixed then the dependence takes a simple form:

$$B(x) = \sum_{l=1}^{m} x_l A_l^T A_l \tag{3}$$

with extremely sparse matrices A_l with few rows.

An important special case is where each A_l has a single row only. This is the case for truss structures, but not for beams and more complex elements.

In the case of truss structures, we may rewrite (3) **as**

$$B(x) = A^T D(x) A,$$

where

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix}, \quad D(x) = \mathrm{Diag}(x_1, \ldots, x_m).$$

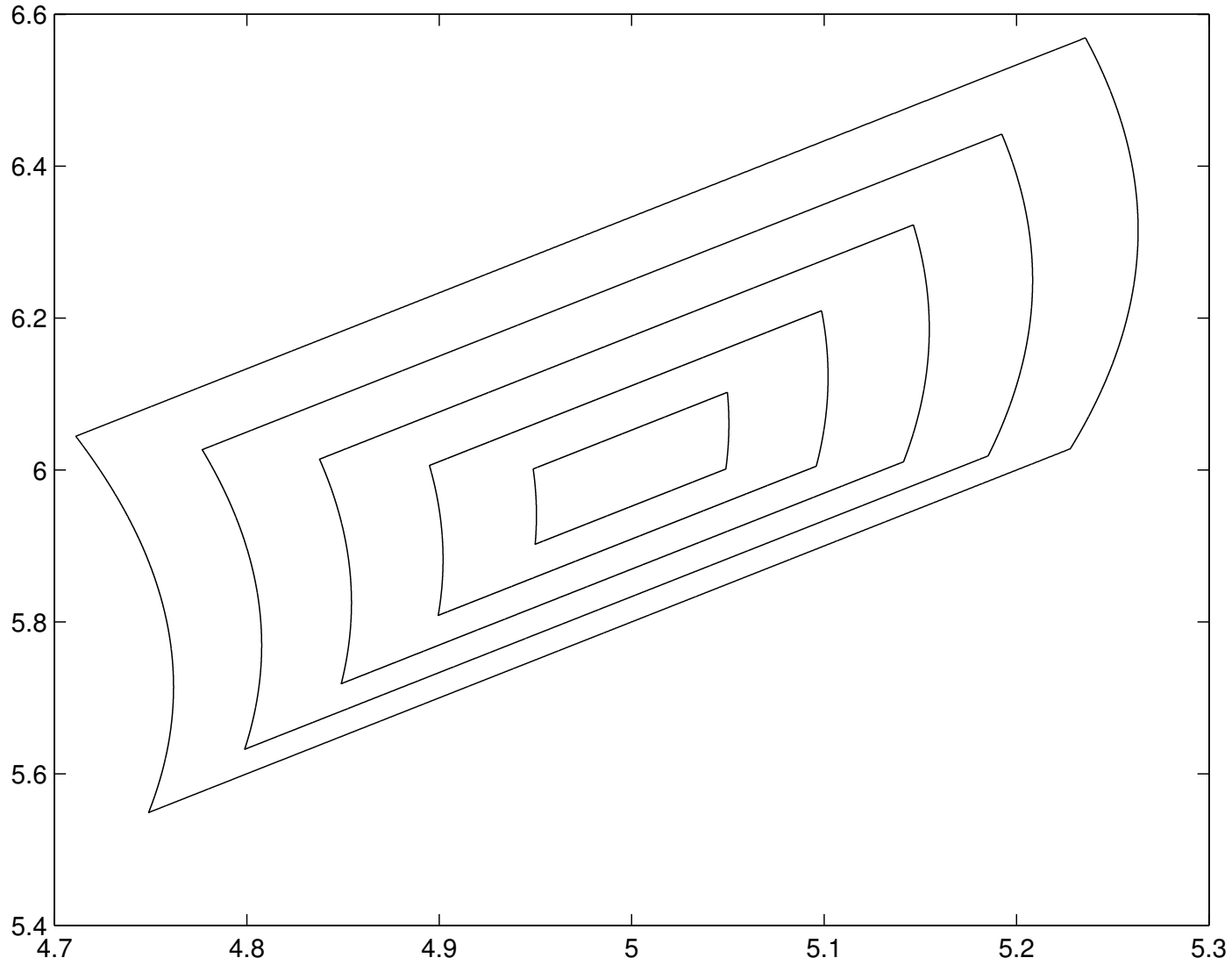
A **is a sparse rectangular matrix, and** $D(x)$ **diagonal with positive diagonal entries.**

Unfortunately, even this special form of $B(x)$ is not enough to make the traditional methods work. The example

$$\begin{pmatrix} x_1 + x_2 & x_2 \\ x_2 & x_1 + x_2 \end{pmatrix} u = \begin{pmatrix} 60 \\ 61 \end{pmatrix}, \quad x \in \begin{pmatrix} [1-\delta, 1+\delta] \\ [5-\delta, 5+\delta] \end{pmatrix},$$

has a coefficient matrix of the form $KA^T D(x)A$ with diagonal $D(x)$ with positive diagonal entries, although not constructed from finite elements.

We display the boundaries of the solution set for $\delta = 0.05, 0.1, 0.15, 0.2, 0.25$


6.6
6.4
6.2
6
5.8
5.6
5.4
4.7
4.8
4.9
5
5.1
5.2
5.3

The monotonicity method and the vertex sensitivity method fail to find the maximum of u_1 since the maximizer is not a corner.

A local optimization may get stuck in the nonglobal minimum. when trying to minimize u_1.

Similar problems can be seen in (higher-dimensional) finite element problems.

However, coefficient matrices of the form $A^T D(x) A$ allow one to iteratively improve the centered form, leading to excellent interval enclosures.

We consider more generally uncertain linear systems of the form

$$(K + BDA)u = a + Fb,$$

with uncertainties in D and b only.

Main Theorem

Let $D_0 \in \mathbb{R}^{n\times n}$ be such that $C := (K + BD_0A)^{-1}$ exists, and put $d = (D_0 - D)v$, where $v = Au$. If there are vectors $w \geq 0$, $w' > 0$ and w'' such that

$$w' \leq w - |D_0 - D||ACB|w, \quad w'' \geq |D_0 - D||ACa + ACFb|,$$

then

$$d \in \mathbf{d} := [-\alpha w, \alpha w], \quad \alpha = \max_i \frac{w_i''}{w_i'}, \tag{4}$$

and the solution u of $(K + BDA)u = a + Fb$ is related to v and d by the equations

$$u = Ca + CFb + CBd, \tag{5}$$

$$v = ACa + ACFb + ACBd. \tag{6}$$

Proof of (5)–(6)**:**

Since $d = (D_0 - D)v$**, and** $(K + BDA)u = a + Fb$**,**
equation $u = Ca + CFb + CBd$ **follows from**

$$\begin{aligned} CBd &= CB(D_0 - D)Au \\ &= C(K + BD_0A)u - C(K + BDA)u \\ &= u - C(a + Fb). \end{aligned}$$

Multiplication with A **gives**

$$v = Au = ACa + ACFb + ACBd.$$

Proof of (4)**:**

Put $\beta = \max_i |d_i|/w_i$**, so that** $|d| \leq \beta w$**,**
with equality in some component i**.**

The definition of $\alpha = \max_i \frac{w_i''}{w_i'}$ **implies** $w'' \leq \alpha w'$**, hence**

$$\begin{aligned} |d| &= |(D_0 - D)(ACa + ACFb + ACBd)| \\ &\leq |D_0 - D||ACa + ACFb| + |D_0 - D||ACB|\beta w \\ &\leq w'' + \beta(w - w') \leq \alpha w' + \beta(w - w'). \end{aligned}$$

Thus $\beta w_i = |d_i| \leq \alpha w_i' + \beta(w_i - w_i')$**, hence** $\beta w_i' \leq \alpha w_i'$**.**

Since $w' > 0$**, we conclude that** $\beta \leq \alpha$**, and the desired conclusion** $d \in \mathbf{d} := [-\alpha w, \alpha w]$ **follows.**

We now assume

$$D \in \mathbf{D}, \quad b \in \mathbf{b}$$

as interval bounds for the data uncertainties.

If we take $D_0 = \operatorname{mid} \mathbf{D}$ and w as the all-one vector then

$$w' := w - |D_0 - \mathbf{D}||ACB|w, \quad w'' := |D_0 - \mathbf{D}||ACa + ACF\mathbf{b}|,$$

are small and satisfy the required conditions if $w' > 0$ (which is a very moderate condition).

By the theorem, $d \in \mathbf{d} := [-\alpha w, \alpha w]$, a narrow box. Using the formulas of the theorem one gets narrow enclosures u for u, v for v and a generally improved enclosure for d.

The enclosures can be further improved by iterating this and intersecting with the previously computed enclosures.

To get realistic bounds on quantities $z = Z(u)$ (such as safety constraints) dependent linearly or nonlinearly on the solution u of the uncertain linear system, one should intersect the simple enclosure $\mathbf{z} = Z(\mathbf{u})$ with the enclosure obtained from the centered form

$$\mathbf{z}' = Z(CF \operatorname{mid} \mathbf{b}) + (\mathbf{S}CF)(\mathbf{b} - \operatorname{mid} \mathbf{b}) + (\mathbf{S}CB)\mathbf{d},$$

where $\mathbf{S} = Z[\operatorname{mid} \mathbf{u}, \mathbf{u}]$ is a slope matrix for Z.

We implemented a Matlab interface to the ANSYS finite element modeling system.

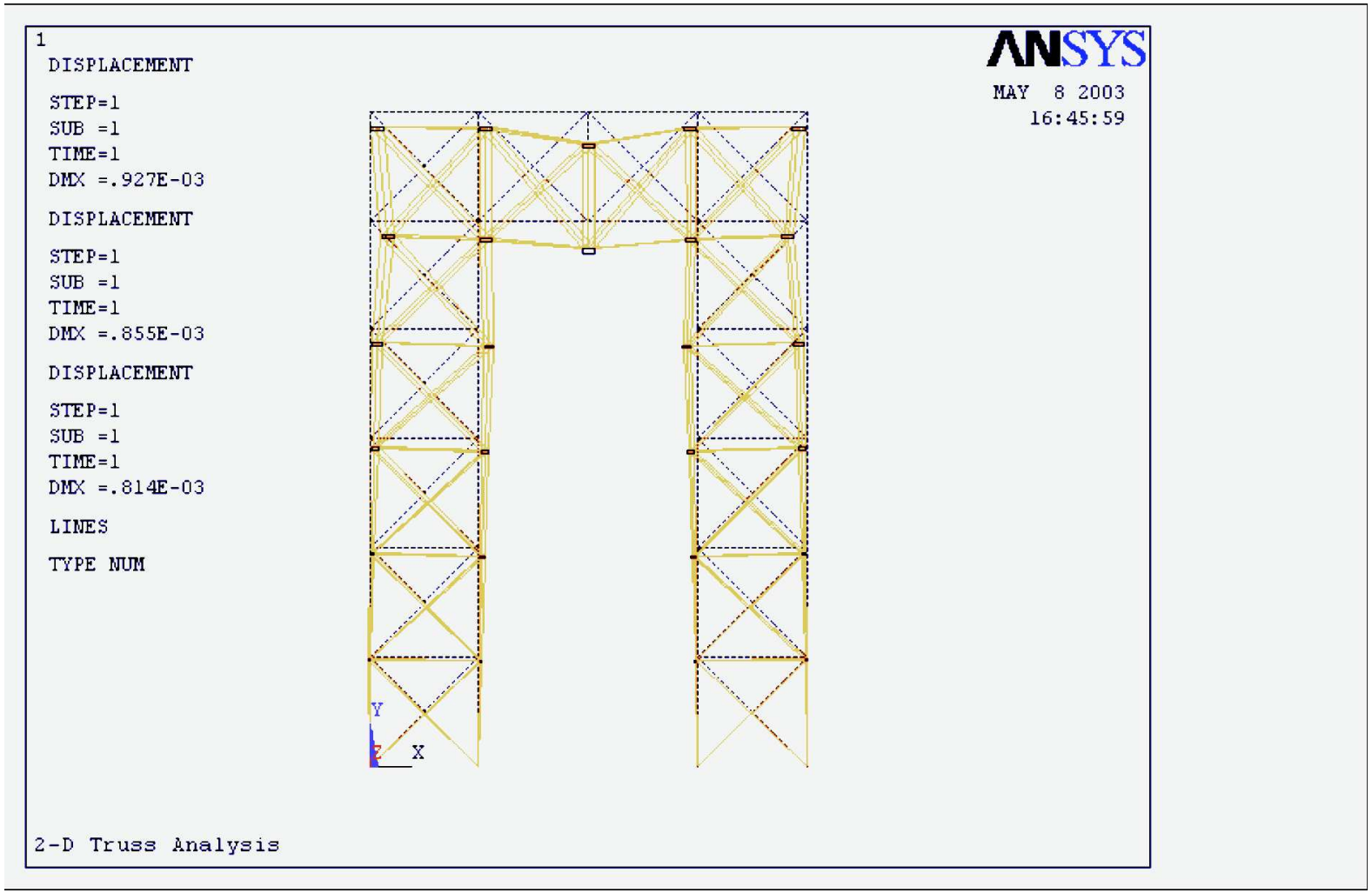
The set of ANSYS commands which describe a particular truss structure can be created by using a standard ANSYS GUI.

Then, using the algorithm presented here it is possible to calculate the interval solution.

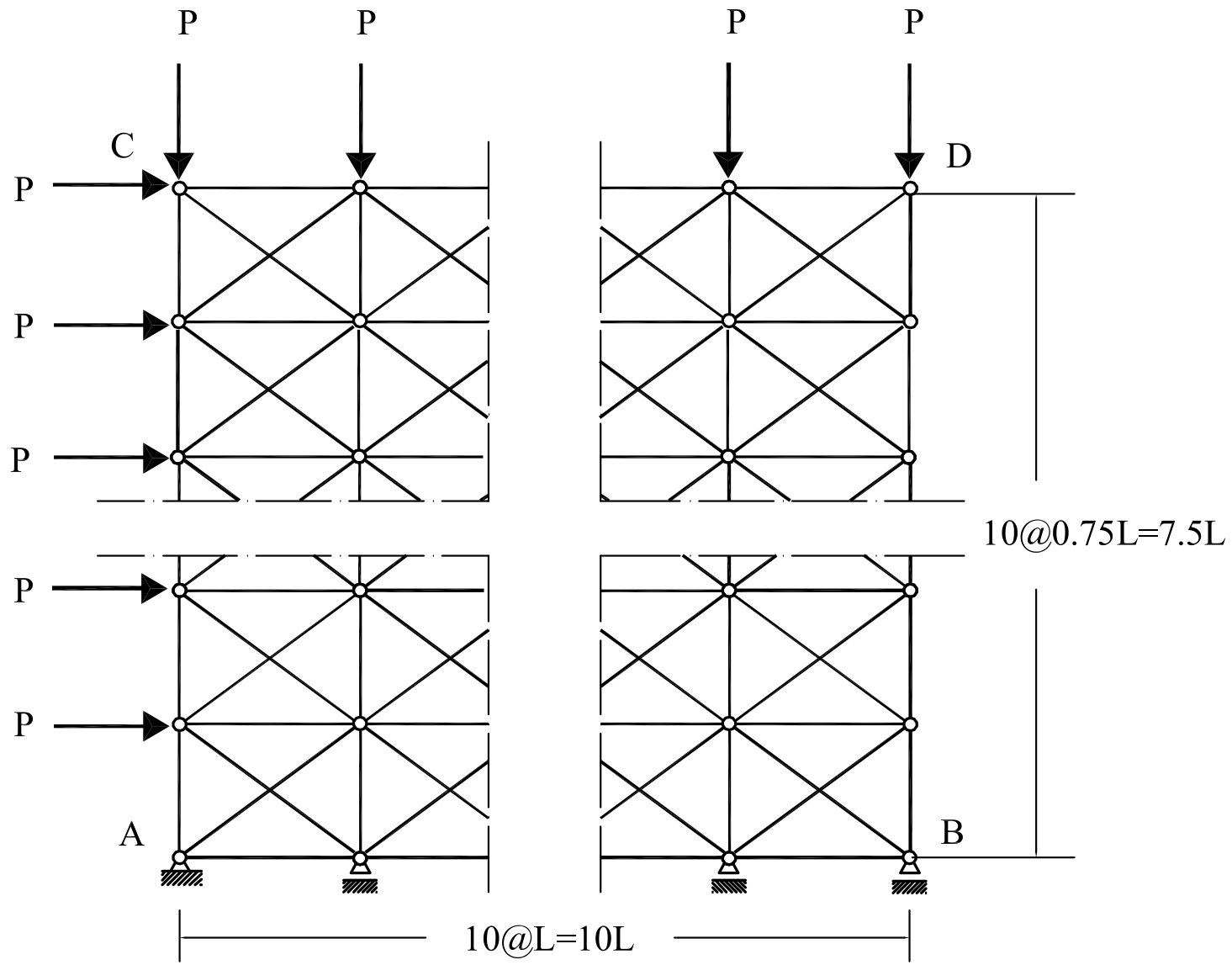
Finally it is possible to plot the interval solution in the ANSYS GUI.

The following calculations were done in Matlab, using the Intlab package of RUMP for the interval calculations.

The computer used was an AMD Athlon MP 2000+ with a 1680 MHz CPU and 3GB memory, running under LINUX.


1
DISPLACEMENT
STEP=1
SUB =1
TIME=1
DMX =.927E-03
DISPLACEMENT
STEP=1
SUB =1
TIME=1
DMX =.855E-03
DISPLACEMENT
STEP=1
SUB =1
TIME=1
DMX =.814E-03
LINES
TYPE NUM
ANSYS
MAY 8 2003
16:45:59
Y
Z
X
2-D Truss Analysis

10-bay 10-floor truss structure used in work by Muhanna & Mullen



Test case: 10-bay 10-floor truss

- **230 equations**
  (FEM equations for displacements of 121 nodes)
- **420 two-sided bound constraints**
  (stiffness uncertainties, one per element)
- **total of 650 variables**
  (230 linear and 420 nonlinear ones)
- **stiffness uncertainty** $\alpha = 5\%$
  α **defines search region**

Sizes increase proportional to the number of bays and to the number of floors.

The monotonicity method is too slow (exponential amount of work).

The vertex sensitivity, although faster and sharper than Monte Carlo methods, underestimates the worst case.
The monotonicity assumption is no longer valid at the specified range of uncertainty (already for the 5×5 grid).

Traditional interval methods already fails for tiny search regions (uncertainty of $\alpha = 0.1\%$).

The relative uncertainty in the stiffness (diagonal entries of D) is varied to be able to assess the degradation of the bounds as the uncertainty increases.

The following table lists the widths of the enclosures of the displacements in horizontal ($\mathbf{u}_x$) and vertical ($\mathbf{u}_y$) direction of the upper right corner, and compares it with the widths $\mathbf{u}'_x$ and $\mathbf{u}'_y$ of the enclosure obtained by the element-by-element method of Muhanna & Mullen**.**

uncertainty	1%	2%	3%	4%	5%
$\mathrm{wid}\,\mathbf{u}'_x$	**0.301**	**0.744**	**1.423**	**2.521**	**4.457**
$\mathrm{wid}\,\mathbf{u}_x$	**0.254**	**0.516**	**0.788**	**1.070**	**1.362**
$\mathrm{wid}\,\mathbf{u}'_y$	**0.168**	**0.438**	**0.879**	**1.638**	**3.045**
$\mathrm{wid}\,\mathbf{u}_y$	**0.136**	**0.278**	**0.425**	**0.578**	**0.737**

With increasing uncertainty, our bounds are increasingly better than those from the element-by-element method.

The times for the computation of the enclosures of all components of the solution were 20–25 times the time needed for the computation of the solution with the midpoint stiffness coefficients.

This compares very favorably with Monte Carlo simulations, the monotonicity method and the vertex sensitivity method, although all these only give inner enclosures of the solution set.

For smaller examples where it was feasible to run the vertex sensitivity method, the comparison of the resulting inner enclosure with our outer enclosure shows that for the range of uncertainties considered here, our bounds are optimal within a few percent of the width.

To see how work and results scale with dimension and uncertainty, we consider a n-bay n-floor truss with variable n, with the same other specifications as in the previous example.

truss size $n \times n, \quad n =$	**10**	**20**	**30**	**40**	**50**	**60**
number of rows of A	**420**	**1640**	**3660**	**6480**	**10050**	**14520**
number of columns of A	**230**	**860**	**1890**	**3320**	**5150**	**7380**

For $n = 60$, the memory capacity (about 1GB of free memory) of Matlab, with which we did our tests, was insufficient to store the dense matrix ACA^T (with $14520^2 \approx 2.1 \cdot 10^8$ entries) needed in the iteration for our bounds.

$n = 10$, uncertainty	1%	5%	10%	15%	20%	25%
$\underline{\mathbf{u}}_x$	22.02	21.46	20.64	19.60	18.21	16.13
$\overline{\mathbf{u}}_x$	22.28	22.83	23.66	24.70	26.08	28.17
CPU time	0.19	0.22	0.25	0.28	0.28	0.28
time ratio	21	25	29	32	32	32
$n = 20$, uncertainty	1%	5%	10%	15%	20%	25%
$\underline{\mathbf{u}}_x$	51.40	50.09	48.05	45.27	40.61	27.55
$\overline{\mathbf{u}}_x$	52.00	53.31	55.35	58.13	62.80	75.85
CPU time	3.94	4.56	4.98	5.15	5.13	5.16
time ratio	86	102	110	114	114	114
$n = 30$, uncertainty	1%	5%	10%	15%	20%	25%
$\underline{\mathbf{u}}_x$	83.81	81.66	78.22	73.13	61.08	−0.69
$\overline{\mathbf{u}}_x$	84.78	86.93	90.37	95.46	107.51	169.28
CPU time	42.90	44.98	47.67	47.78	47.61	47.81
time ratio	358	379	397	403	402	402
$n = 40$, uncertainty	1%	5%	10%	15%	20%	25%
$\underline{\mathbf{u}}_x$	118.20	115.16	110.18	102.07	73.22	−136.94
$\overline{\mathbf{u}}_x$	119.56	122.60	127.59	135.69	164.54	374.71
CPU time	2m:55	3m:04	3m:09	3m:09	3m:09	3m:09
time ratio	663	704	728	723	730	738
$n = 50$, uncertainty	1%	5%	10%	15%	20%	25%
$\underline{\mathbf{u}}_x$	231.98	226.39	217.63	202.90	127.98	−522.85
$\overline{\mathbf{u}}_x$	234.52	240.11	248.87	263.59	338.51	989.35
CPU time	9m:18	8m:56	9m:13	9m:09	9m:10	9m:10
time ratio	1280	1115	1154	1151	1143	1120

In the table, lower and upper bounds are outward rounded. CPU times are given in seconds or minutes : seconds.

The time ratio (significant only in the leading digits, due to random variations under repetition) indicates the number of trial point evaluations (using the direct sparse linear solver of Matlab) that can be made in the same time.

As can be seen, it is much smaller than the dimensions of the problem, leading to a time advantage over all current methods for approximate worst case analysis.

For uncertainties up to about 15%, the enclosures are of high quality.

On the other hand, one can see that the enclosures get wider as either the problem size or the uncertainty get larger.

In extremal cases (25% uncertainty for $n \geq 30$), not even the sign of the displacement is guaranteed, probably an artifact of the method caused by overestimation of the worst case.

Conclusion

The new method gives fast, reliable and accurate worst case bounds for uncertain linear systems.

The method scales without difficulties to high-dimensional problems, hence appear to be suitable for many previously untractable instances of worst case analysis in structural engineering.

The method works not only for truss structures but applies apply to all kinds of finite element equations. Applications to other mechanical structures are in preparation.

Further details can be found in the manuscript

A. Neumaier and A. Pownuk, Linear systems with large uncertainties, with applications to truss structures, 2004

A preprint is available from

http://www.mat.univie.ac.at/~neum/papers.html#linunc