Technische Universität Chemnitz

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Arnd Meyer

The Adaptive Finite Element Method – Can We Solve Arbitrarily Accurate ?

Preprint SFB393/01-30

Abstract

In the adaptive finite element method, the solution of a p.d.e. is approximated by finer and finer meshes, which are controlled from error estimators. So, starting from a given coarse mesh, some elements are subdivided a couple of times. We investigate the question of avoiding instabilities which limit this process from the fact that nodal coordinates of one element coincide in more and more leading digits. To overcome this problem we demonstrate a simple mechanism for red subdivision of triangles (and hanging nodes) and a more sophisticated technique for general quadrilaterals.

Preprint-Reihe des Chemnitzer SFB 393

SFB393/01-30 **September 2001**

Contents

Author's addresses:

Arnd Meyer
TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

http://www.tu-chemnitz.de/sfb393/

1 Introduction

In the last two decades the use of adaptive mesh refinement in the finite element solution of partial differential equations is widely recognized as having significant potential for improving the efficiency of the solution process. The development of the adaptive finite element method focused mainly on two ingredients, the error estimators for detecting some elements with large contribution to the H^1–error and the mesh subdivision technique for subdividing these elements. Here, we investigate the question whether a very fine mesh in small parts of the domain as a result of continued subdivisions can terminate this process due to numerical instabilities and how to overcome this difficulty.

The main tool that has to be considered for producing or avoiding numerical instabilities is the calculation of the local Jacobian matrix of the mapping from the master–element onto the real element, as usually done in finite element calculations. From this reason our investigations are valid for each kind of 2nd order partial differential equation, so we consider only a model problem for numerical experiments.

2 The adaptive strategy

In the adaptive finite element method, we have some error estimators, which mark some of the elements (with large contribution to the overall error) for subdivision. For a overview of known error estimators see [AO00, Verf96] and the references therein. The marked elements are subdivided into smaller parts due to some strategies.

For 2D–calculations and triangular meshes two basic strategies for one triangle are employed:
The so called "red" subdivision of a triangle into 4 sub–triangles of equal shape and size (see Fig. 1) and the so called "green" subdivision into 2 parts (see Fig. 2).

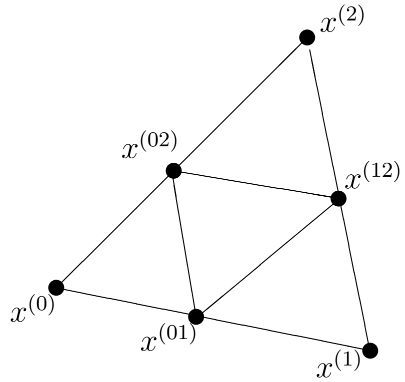

Figure 1: "red"

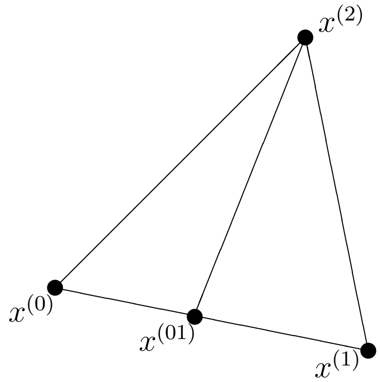

Figure 2: "green"

For the whole mesh we have at least three strategies:
The strategy “green” due to Bänsch [Bän91] uses green subdivisions only. If we start with the red subdivision on all marked triangles we can continue with two possible sub–strategies:

“red–green”	We produce a conforming finite element mesh by adding green subdivisions on triangles with at least one subdivided edge. (These green elements are removed before deeper subdividing in the next step.)
“red–hanging nodes”	No additional subdivision on elements with one subdivided edge – we accept a non–conforming mesh with so called “hanging nodes”. The solver can guarantee that we work in the conforming subspace i.e. we use continuous f.e. functions (see [Mey99]).

In the quadrilateral case similar ideas have been used. The most simple strategy in the mesh–handling is again “red–hanging nodes”, i.e. subdivide each marked element into 4 parts and accept maximal two hanging nodes on two non–opposite sides. This is simply generalized into 3D, whereas the generalization of the “red” triangular case leads to a new property. The resulting sub–tetrahedrons are of the same volume but not of the same shape.

An important fact for the following considerations is the definition of the vertices of the son–elements. In the triangular case the new nodes are calculated first from the formula

$$x^{son} := \frac{1}{2}(x^{father1} + x^{father2})$$

and then used for defining the new sub–edges. If this technique is repeated a couple of times neighboring nodes coincide in more and more leading digits. More precisely, a node x_{son} generated at an L–th level subdivision of some coarse element coincides with its two fathers in about L leading digits (in their binary representation). This means, the coordinates itself are correct until machine accuracy, but differences of neighboring nodes lead to cancellations of about L digits. In classical finite element codes with uniform mesh refinement this played no role, because L was maximally about 8...10 (the amount of storage and work grows with 4^L in 2D!). In an adaptive regime this is not longer true, we can have L larger then 20 (in small parts of the domain, for instance near singularities). So, differences of nodal coordinates should not be allowed (except at the beginning).

3 Unstable calculation of the Jacobian

Let K denote one of the finite elements of the given mesh having n nodes $x_0, \ldots, x_{n-1} \in \mathbb{R}^2$. Here, $n = 3$ or $n = 6$ in the triangular case and usually $n = 4, 8$ or 9 for quadrilaterals.

We denote by $\hat{K}$ the master element which is the unit triangle with vertices

$$\hat{x}^{(0)} = (0,0)^T,\ \hat{x}^{(1)} = e_1 := (1,0)^T \text{ and } \hat{x}^{(2)} = e_2 := (0,1)^T$$

or the unit square $[0,1]^2$ (then $\hat{x}^{(3)} = e := (1,1)^T$), resp. Let $\hat{x} = (\hat{x}_1, \hat{x}_2)^T \in \hat{K}$.

For calculating the element stiffness matrix or for post processing after having the approximate solution and for most of the error estimators, we have to consider gradients at points $x \in K$. Here, the usual technique considers the mapping $x(\hat{x}) : \hat{K} \to K$ and calculates gradients from transformations of master–gradients. Let $N_0(\hat{x}) \ldots, N_{n-1}(\hat{x})$ be the n form functions (defined on $\hat{x} \in \hat{K}$), then usually on some Gaussian points $\hat{g} \in \hat{K}$ we obtain at first the values:

$$\begin{aligned} N(\hat{g}) &= (N_0(\hat{g}), \ldots, N_{n-1}(\hat{g})) \qquad \text{and} \\ \hat{\nabla} N(\hat{g}) &= \left(\hat{\nabla} N_0(\hat{g}), \ldots, \hat{\nabla} N_{n-1}(\hat{g})\right). \end{aligned}$$

Here, $\hat{\nabla} = (\partial/\partial\hat{x}_1, \partial/\partial\hat{x}_2)^T$ means formal differentiation with respect to the master coordinates. The values in $N(\hat{g})$ had $\hat{\nabla} N(\hat{g})$ are given in a stable way, usually these are copies from a fixed list.

From the definition

$$x(\hat{x}) = \sum_{k=0}^{n-1} N_k(\hat{x})\, x^{(k)} \tag{1}$$

of the mapping $\hat{K} \longleftrightarrow K$ with $x^{(0)}, \ldots, x^{(n-1)}$ the nodes of K, we obtain the Jacobian matrix

$$T(\hat{g}) = \hat{\nabla} x^T|_{\hat{g}} = \sum_{k=0}^{n-1} \hat{\nabla} N_k(\hat{g})\, (x^{(k)})^T \tag{2}$$

from a simple matrix–matrix–multiply of $\hat{\nabla} N$ with the nodal coordinates. The values of T seem to be simple short inner products, but are calculated with a serious cancellation of leading digits. Let us consider the most simple triangular case with $n = 3$, then the form functions are

$$N_0(\hat{x}) = 1 - \hat{x}_1 - \hat{x}_2, \quad N_1(\hat{x}) = \hat{x}_1, \quad N_2(\hat{x}) = \hat{x}_2, \tag{3}$$

so

$$\hat{\nabla} N = (-e \;\vdots\; e_1 \;\vdots\; e_2) \tag{4}$$

and

$$T = (x^{(1)} - x^{(0)} \;\vdots\; x^{(2)} - x^{(0)})^T. \tag{5}$$

In the case of 6–node triangles the same is true if the edge mid nodes define straight edges ($x^{(3)} = \frac{1}{2}(x^{(0)} + x^{(1)})$ and so on).

If a quadrilateral is a parallelogram, we have (5) as well, for more general cases see Section 5.

The instability of succeeding calculations results from the use of the inverse of T for instance in:

$$\nabla N_k(g) = T^{-1} \hat{\nabla} N_k(\hat{g}). \tag{6}$$

4 Stable calculation of T for triangles

A stable calculation of T (and especially T^{-1}) is possible, if we avoid these differences of nodal coordinates in (5) for the fine grid elements. This is very easy in the triangular case, if all triangles have straight edges. Then the strategy "red–hanging nodes" leads to sub–triangles which are all similar to one of the given coarse mesh triangles. We suppose that T was calculated in a stable way on these coarse mesh triangles, then all Jacobians on all finer triangles are multiples of the Jacobian of its father triangle (for suitable node numbering), which stabilizes this procedure totally.

For doing this we save the four values of T additionally to the data structure for storing the element. During the "red" subdivision of an element K we calculate the 4 Jacobians of the son elements from T. Let $x^{(0)}, x^{(1)}, x^{(2)}$ the vertices of K, then the 4 sons have to be defined as (see Fig. 1)

$$\begin{array}{llll} K_1: & x^{(0)} & x^{(01)} & x^{(02)} \\ K_2: & x^{(01)} & x^{(1)} & x^{(12)} \\ K_3: & x^{(02)} & x^{(12)} & x^{(2)} \\ K_4: & x^{(12)} & x^{(02)} & x^{(01)} \end{array}$$

So, the Jacobians are $\frac{1}{2}T$ (for K_1, K_2, K_3) and $-\frac{1}{2}T$ (for K_4), which is totally free of rounding errors.

From this fact, the strategy "red–hanging nodes" has a slight advantage.

For "green" triangles such a stabilization by heredity of the Jacobians to the son elements is possible as well. But it requires differences of rows of T:

For the "green" subdivision of the given element K, we have to define the so called subdivision edge of K. Let this be the first edge of each element. The nodes of K are $x^{(0)}, x^{(1)}$ and $x^{(2)}$ (the first two define the subdivision edge). Then, according to Fig. 2 the two sons K_1 and K_2 can be:

$$\begin{array}{llll} K_1: & x^{(0)} & x^{(2)} & x^{(01)} \\ K_2: & x^{(1)} & x^{(2)} & x^{(01)} \end{array}$$

for defining $(x^{(0)}, x^{(2)})$ and $(x^{(1)}, x^{(2)})$ as new subdivision edges (see [Bän91]). From

$$T^T = (x^{(1)} - x^{(0)} \vdots x^{(2)} - x^{(0)})$$

follows with this restriction:

$$T_1^T = (x^{(2)} - x^{(0)} \vdots x^{(01)} - x^{(0)}) = T^T \begin{pmatrix} 0 & 1/2 \\ 1 & 0 \end{pmatrix}$$

$$T_2^T = (x^{(2)} - x^{(1)} \vdots x^{(01)} - x^{(1)}) = T^T \begin{pmatrix} -1 & 1/2 \\ 1 & 0 \end{pmatrix},$$

which are the stable Jacobians of the son elements (here, T_2 produces a rounding error of machine accuracy but not a cancellation of leading digits because the two input vectors are never near to linearly dependent).

5 The case of arbitrary quadrilaterals

If the initial coarse mesh contains quadrilaterals which are not parallelograms then the technique above is impossible due to the fact that T is not constant over $\hat{K}$ in this case. Again a stable calculation has to avoid differences of the nodal coordinates of the vertices..

Let K_0 with the vertices $y^{(0)}, y^{(1)}, y^{(2)}, y^{(3)}$ be one of the quadrilaterals in the initial coarse mesh. Then none of the sub–quadrilaterals which occur later has similarity to K_0 (except for parallelograms). Hence, another technique as in the triangular case is required.

First note that from continued subdividing K_0 all new nodes have the following representation:

$$x = \sum_{i=0}^{3} N_i(\hat{b})\, y^{(i)}, \tag{7}$$

where N_i are the bilinear shape functions:

$$\begin{aligned} N_0(\hat{x}) &= (1-\hat{x}_1)(1-\hat{x}_2) \\ N_1(\hat{x}) &= \hat{x}_1(1-\hat{x}_2) \\ N_2(\hat{x}) &= (1-\hat{x}_1)\hat{x}_2 \\ N_3(\hat{x}) &= \hat{x}_1\hat{x}_2 \end{aligned}$$

and $\hat{b} \in \hat{K}$ is a vector with coordinates in $\mathbb{B}_L$ after L levels of subdivision, where

$$\mathbb{B}_0 = \{0,1\}, \mathbb{B}_1 = \{0, \tfrac{1}{2}, 1\},$$

$$\mathbb{B}_L := \{k \cdot 2^{-L} : k = 0, \cdots, 2^L\}$$

(All values in $\mathbb{B}_L$ have maximally an L–digit binary representation.)

Now let $K \subset K_0$ be one of the sub–quadrilaterals within K_0 with the vertices $x^{(0)}$, $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$, so

$$x^{(i)} = \sum_{i=0}^{3} N_k(\hat{b}^{(i)}) y^{(k)}, \tag{8}$$

where

$$\hat{b}^{(0)} = \hat{b}, \quad \hat{b}^{(1)} = \hat{b} + \nu e_1, \quad \hat{b}^{(2)} = \hat{b} + \nu e_2 \;\text{ and }\; \hat{b}^{(3)} = \hat{b} + \nu e \tag{9}$$

with $\hat{b} \in (\mathbb{B}_L)^2$ the representation of the first node $x^{(0)}$ and $\nu = 2^{-L}$, if this cell occurred in the L–th level of subdivision. (Again the node numeration is restricted to be "similar" to K_0 for all sub–quadrilaterals.)

Now, the Jacobian of K is defined as

$$T(\hat{g}) = \sum_{i=0}^{3} \hat{\nabla} N_i(\hat{g}) (x^{(i)})^T, \tag{10}$$

which leads to an unstable calculation due to:

$$\begin{aligned} \hat{\nabla} N_0 &= \begin{pmatrix} \hat{g}_2 - 1 \\ \hat{g}_1 - 1 \end{pmatrix} =: q(\hat{g}) - e, \\ \hat{\nabla} N_1 &= e_1 - q(\hat{g}),\; \hat{\nabla} N_2 = e_2 - q(\hat{g}),\; \hat{\nabla} N_3 = q(\hat{g}). \end{aligned}$$

Here, for a vector $\hat{g} = (\hat{g}_1, \hat{g}_2)^T \in \hat{K}$ define $q(\hat{g}) = (\hat{g}_2, \hat{g}_1)^T$.
Then (10) reads as

$$T(\hat{g}) = (x^{(1)} - x^{(0)} \vdots x^{(2)} - x^{(0)})^T + q(\hat{g})(x^{(4)})^T \tag{11}$$

with

$$x^{(4)} = x^{(0)} - x^{(1)} - x^{(2)} + x^{(3)}. \tag{12}$$

Note that $x^{(4)} = 0$ for parallelograms.

A stable calculation of $T(\hat{g})$ for the cell K can be obtained from combining (10) with (8) and (9):

$$T(\hat{g}) = \sum_{i=0}^{3} \sum_{k=0}^{3} \hat{\nabla} N_i(\hat{g}) N_k(\hat{b}^{(i)}) \, (y^{(k)})^T =: \sum_{k=0}^{3} v_k (y^{(k)})^T.$$

The vectors v_k are simply calculated as

$$\begin{array}{rclcl} v_0 &=& \nu(q(\hat{b}) - e) &+& \nu^2 q(\hat{g}) \\ v_1 &=& \nu(e_1 - q(\hat{b})) &-& \nu^2 q(\hat{g}) \\ v_2 &=& \nu(e_2 - q(\hat{b})) &-& \nu^2 q(\hat{g}) \\ v_3 &=& \nu q(\hat{b}) &+& \nu^2 q(\hat{g}). \end{array}$$

This results in a much more stable formula for $T(\hat{g})$:

$$T(\hat{g}) = \nu Y^T + \nu \, q(\hat{b} + \nu \hat{g}) \, (y^{(4)})^T, \tag{13}$$

where

$$Y = (y^{(1)} - y^{(0)} \vdots y^{(2)} - y^{(0)})$$

and

$$y^{(4)} = y^{(0)} - y^{(1)} - y^{(2)} + y^{(3)}.$$

Again Y and $y^{(4)}$ are stable calculated from the coordinates of the initial mesh (constant for all K within K_0). The special cell K is represented by $\hat{b}$ the local representation of $x^{(0)}$. The addition $\hat{b} + \nu\hat{g}$ produces a rounding error of machine accuracy only. If K_0 was a parallelogram then $y^{(4)} = 0$ and we have the same result as in the "red" triangular case.

6 The 3D case of arbitrary hexahedrons

Here, let the initial course mesh contain a hexahedron K_0 with the vertices $y^{(0)}, \ldots, y^{(7)}$ which is subdivided into smaller cells analogously to the 2D case. Again we consider one typical sub–cell K after L–th level of subdivision having the nodes $x^{(0)}, \ldots, x^{(7)}$. The nodes in K and K_0 are enumerated in a special way to specify their position. This is most easily fixed by giving their images of the map onto the master element $\hat{K} = [0,1]^3$ (as seen in Fig. 4):

$$
\begin{array}{lllllllll}
\text{Let} & x^{(0)} & \longleftrightarrow & (0,0,0)^T \in \hat{K}, & & & & x^{(7)} & \longleftrightarrow & e, \\
 & x^{(1)} & \longleftrightarrow & e_1, & x^{(2)} & \longleftrightarrow & e_2, & x^{(3)} & \longleftrightarrow & e_3, \\
\text{and} & x^{(4)} & \longleftrightarrow & e - e_1 = (0,1,1), & x^{(5)} & \longleftrightarrow & e - e_2, & x^{(6)} & \longleftrightarrow & e - e_3\ .
\end{array}
$$

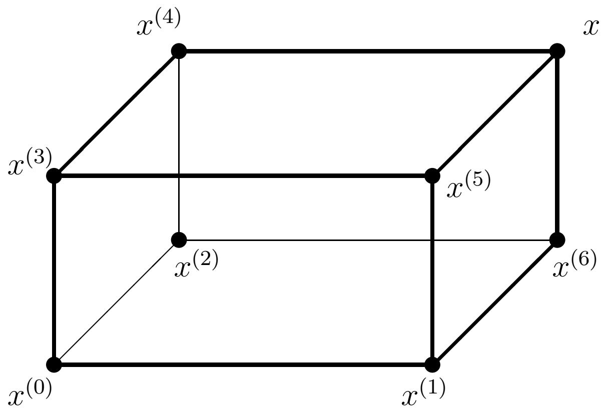

Figure 4: nodes of a hexahedron

This map is defined analogously to Section 5 as

$$x(\hat{x}) = \sum_{i=0}^{7} N_i(\hat{x})\, x^{(i)}, \tag{14}$$

where

$$
\begin{aligned}
N_0(\hat{x}) &= (1-\hat{x}_1)(1-\hat{x}_2)(1-\hat{x}_3) \\
N_1(\hat{x}) &= \hat{x}_1\ (1-\hat{x}_2)(1-\hat{x}_3) \\
N_2(\hat{x}) &= (1-\hat{x}_1)\ \hat{x}_2\ (1-\hat{x}_3) \\
N_3(\hat{x}) &= (1-\hat{x}_1)\ (1-\hat{x}_2)\ \hat{x}_3 \\
N_4(\hat{x}) &= (1-\hat{x}_1)\ \hat{x}_2\ \hat{x}_3 \\
N_5(\hat{x}) &= \hat{x}_1\ (1-\hat{x}_2)\hat{x}_3 \\
N_6(\hat{x}) &= \hat{x}_1\ \hat{x}_2\ (1-\hat{x}_3) \\
N_7(\hat{x}) &= \hat{x}_1\ \hat{x}_2\ \hat{x}_3
\end{aligned}
$$

The same node numbering is used in K_0. Furthermore, we need four abbreviations for vectors which vanish on parallelograms or parallelepipeds:

$$\tilde{x}^{(1)} = x^{(0)} - x^{(2)} - x^{(3)} + x^{(4)} \tag{15}$$

(belongs to the quadrilateral "$\hat{x}_1 = 0$")

$$\tilde{x}^{(2)} = x^{(0)} - x^{(1)} - x^{(3)} + x^{(5)} \tag{16}$$

(to "$\hat{x}_2 = 0$")

$$\tilde{x}^{(3)} = x^{(0)} - x^{(1)} - x^{(2)} + x^{(6)} \tag{17}$$

(to "$\hat{x}_3 = 0$") and let

$$x^{(8)} = -x^{(0)} + x^{(1)} + x^{(2)} + x^{(3)} - x^{(4)} - x^{(5)} - x^{(6)} + x^{(7)}. \tag{18}$$

We use the same definitions for K_0:

$$\begin{aligned}
\tilde{y}^{(1)} &= y^{(0)} - y^{(2)} - y^{(3)} + y^{(4)}, && (19)\\
\tilde{y}^{(2)} &= y^{(0)} - y^{(1)} - y^{(3)} + y^{(5)}, && (20)\\
\tilde{y}^{(3)} &= y^{(0)} - y^{(1)} - y^{(2)} + y^{(6)} && (21)\\
\text{and} \quad y^{(8)} &= -y^{(0)} + y^{(1)} + y^{(2)} + y^{(3)} - y^{(4)} - y^{(5)} - y^{(6)} + y^{(7)}. && (22)
\end{aligned}$$

Furthermore, we will need the following operations on vectors in $\hat{K}$:

$$q(\hat{x}) = \begin{pmatrix} \hat{x}_2\hat{x}_3 \\ \hat{x}_1\hat{x}_3 \\ \hat{x}_1\hat{x}_2 \end{pmatrix} = \prod_{i=1}^{3} \hat{x}_i \left(\frac{1}{\hat{x}_j}\right)_{j=1}^{3} \tag{23}$$

(the 3D analogue to the vector $q(\hat{x})$ in Section 5) and

$$B(\hat{x}) = \begin{pmatrix} 0 & \hat{x}_3 & \hat{x}_2 \\ \hat{x}_3 & 0 & \hat{x}_1 \\ \hat{x}_2 & \hat{x}_1 & 0 \end{pmatrix}. \tag{24}$$

Now, we can express each vertex in K as a trilinear mapping of vectors $\hat{b}^{(i)} \in (\mathbb{B}_L)^3 \subset \hat{K}$:

$$x^{(i)} = \sum N_k(\hat{b}^{(i)}) y^{(k)}, \tag{25}$$

where $\hat{b} = \hat{b}^{(0)} \in (\mathbb{B}_L)^3$ represents the special cell K and the other vectors are:

$$\begin{array}{lll}
\hat{b}^{(1)} = \hat{b} + \nu e_1, & \hat{b}^{(2)} = \hat{b} + \nu e_2, & \hat{b}^{(3)} = \hat{b} + \nu e_3, \\
\hat{b}^{(4)} = \hat{b} + \nu(e - e_1), & \hat{b}^{(5)} = \hat{b} + \nu(e - e_2), & \hat{b}^{(6)} = \hat{b} + \nu(e - e_3), \\
\hat{b}^{(7)} = \hat{b} + \nu e & \text{and again } \nu = 2^{-L}. &
\end{array}$$

For the cell K, we have to use the map onto the master element and to compute its Jacobian (at some Gaussian point $\hat{g} \in \hat{K}$)

$$
\begin{aligned}
x(\hat{g}) &= \sum_{i=0}^{7} N_i(\hat{g}) x^{(i)} && (26)\\
T(\hat{g}) &= \sum_{i=0}^{7} \hat{\nabla} N_i(\hat{g})\, (x^{(i)})^T. && (27)
\end{aligned}
$$

This is again the unstable formula for very fine meshes. For stabilizing this we first calculate a special structure of (27) from considering the gradients of the N_i's:

$$
\begin{aligned}
\hat{\nabla} N_0(\hat{g}) &= -e + (\hat{g}_2 + \hat{g}_3)e_1 + (\hat{g}_1 + \hat{g}_3)e_2 + (\hat{g}_1 + \hat{g}_2)e_3 + q(\hat{g}) && (28)\\
\hat{\nabla} N_1(\hat{g}) &= e_1 - (\hat{g}_2 + \hat{g}_3)e_1 - \hat{g}_1 e_2 - \hat{g}_1 e_3 + q(\hat{g}) && (29)\\
\hat{\nabla} N_4(\hat{g}) &= \hat{g}_2 e_3 + \hat{g}_3 e_2 - q(\hat{g}) && (30)\\
\hat{\nabla} N_7(\hat{g}) &= q(\hat{g}). && (31)
\end{aligned}
$$

The remaining gradients are obtained by cyclically rotating the indices (from $\hat{\nabla} N_1$ get $\hat{\nabla} N_2$ and $\hat{\nabla} N_3$ from $\hat{\nabla} N_4$ the rest). A longer calculation leads to

$$
T(\hat{g}) = X^T + B(\hat{g})\tilde{X}^T + q(\hat{g})(x^{(8)})^T, \tag{32}
$$

when we abbreviate $\tilde{X} = (\tilde{x}^{(1)} \vdots \tilde{x}^{(2)} \vdots \tilde{x}^{(3)})$ and $X = (x^{(1)} - x^{(0)} \vdots x^{(2)} - x^{(0)} \vdots x^{(3)} - x^{(0)})$.

All the three matrices X, $\tilde{X}$ and $x^{(8)}$ contain nodal differences with possible cancelations of leading digits. For a stable calculation we combine this last result with the node definitions in (25), which can be done for the three cell depending parts $X, \tilde{X}$ and $x^{(8)}$ seperately.

For the matrix X we have to consider $x^{(1)} - x^{(0)}$ (resp. $x^{(2)} - x^{(0)}$ and $x^{(3)} - x^{(0)}$):

$$
\begin{aligned}
x^{(1)} - x^{(0)} &= \sum_{k=0}^{7} \left[N_k(\hat{b} + \nu e_1) - N_k(\hat{b}) \right] y^{(k)}\\
&= \nu(y^{(1)} - y^{(0)}) + \nu \hat{b}_2 \tilde{y}^{(3)} + \nu \hat{b}_3 \tilde{y}^{(2)} + \nu \hat{b}_2 \hat{b}_3 y^{(8)},
\end{aligned}
$$

so

$$
X = \nu Y + \nu \tilde{Y} B(\hat{b}) + \nu y^{(8)} q(\hat{b})^T \tag{33}
$$

with $Y = (y^{(1)} - y^{(0)} \vdots y^{(2)} - y^{(0)} \vdots y^{(3)} - y^{(0)})$ and $\tilde{Y} = (\tilde{y}^{(1)} \vdots \tilde{y}^{(2)} \vdots \tilde{y}^{(3)})$.

For the last part $x^{(8)}$ surprisingly a constant vector $\nu^3 y^{(8)}$ (constant for each cell in K_0) is obtained. From the calculation of

$$
\begin{aligned}
&-N_k(\hat{b}) + N_k(\hat{b} + \nu e_1) + N_k(\hat{b} + \nu e_2) + N_k(\hat{b} + \nu e_3)\\
&-N_k(\hat{b} + \nu(e - e_1)) - \ldots\\
&+N_k(\hat{b} + \nu e) \quad \text{for each } k
\end{aligned}
$$

we get $\pm\nu^3$. The signs are such that $y^{(8)}$ arises:

$$x^{(8)} = \nu^3 y^{(8)}. \tag{34}$$

A similar calculation has to be done with the columns of $\tilde{X}$, for instance

$$\begin{aligned} \tilde{x}^{(1)} &= \sum_{k=0}^{7} \left[N_k(\hat{b}) - N_k(\hat{b} + \nu e_2) - N_k(\hat{b} + \nu e_3) + N_k(\hat{b} + \nu e_2 + \nu e_3) \right] y^{(k)} \\ &= \nu^2 \tilde{y}^{(1)} + \hat{b}_1 \nu^2 y^{(8)}, \end{aligned}$$

hence

$$\tilde{X} = \nu^2 \tilde{Y} + \nu^2 y^{(8)} \hat{b}^T. \tag{35}$$

Inserting (33), (34) and (35) into (32),

$$T(\hat{g}) = \nu Y^T + \nu B(\hat{b} + \nu\hat{g})\ \tilde{Y}^T \ + \ \nu \left(q(\hat{b}) + \nu B(\hat{g})\hat{b} + \nu^2 q(\hat{g}) \right)\ (y^{(8)})^T\ , \tag{36}$$

we obtain the main result:

$$T(\hat{g}) = \nu \left(Y^T \ + \ B(\hat{b} + \nu\hat{g})\tilde{Y}^T \ + \ q(\hat{b} + \nu\hat{g})\ (y^{(8)})^T \right)\ , \tag{37}$$

which is again a total stable possibility to obtain the entries of $T(\hat{g})$ up to the last binary digit. This formula can be implemented from having $\hat{b}$ as the representation of the cell $K \subset K_0$ and using the nodes of the coarse mesh cell K_0 in $Y, \tilde{Y}$ and $y^{(8)}$.

7 Numerical Results

For demonstrating the above results, we choose the following widely used benchmark problem having a strong singularity.
We solve the problem:

$$\begin{cases} -\Delta u = 0 & \text{in} \quad \Omega \\ u = 0 & \text{on} \quad \Gamma_D \\ \frac{\partial u}{\partial n} = 1 & \text{on} \quad \Gamma_N, \end{cases}$$

where Ω is a square domain with slit: $\Omega = (0,2)^2 \setminus \{(x,1)^T : x \in [1,2)\}$. We have homogeneous Dirichlet type boundary conditions everywhere on $\partial\Omega$ except at one side of the slit, which is Γ_N.

The initial coarse mesh consists of 8 triangles. We compare the three strategies of the mesh refinement given in Section 2:
"B.-green" due to Bänsch, "red–green" and "red–hanging nodes".

For each of this strategies we tested two different experimental runs:

a) with an element routine as in classical finite element calculations, i.e. the Jacobian T is calculated due to (2) or (5) and

b) with the stable calculation due to Section 4 (different for a “red” or “green” subdivision).

The mesh has been refined from an error estimator of residual type. Here, we can simply choose an edge–based estimator

$$\eta_E^2 = |E| \int_E \left[\frac{\partial u_h}{\partial n}\right]^2 ds,$$

where $\left[\frac{\partial u_h}{\partial n}\right]$ denotes the jump of the normal derivative over E for interior edges. For edges along Γ_N this is the difference between $\frac{\partial u_h}{\partial n}$ from element–side and the given Neumann data.

Both in the element routine and in the error estimation we have to form the Jacobian T on each element. This is done with the same technique different in a) and b).

In Figure 3 we have plotted the (square of the) estimated error $\sum_{\forall E} \eta_E^2$ for the three series of runs on the successive meshes over the number of unknowns. On each adaptive step we subdivide at least 10% of all edges of the actual mesh.

From this regime, the runs using a) with unstable T’s brake down after about 20 levels (with calculating “NaN” – not a number, indicated by the vertical line in Fig. 3), whereas the strategy b) with stable T’s continued to more than 30 levels with further decreasing the error continuously. Here, the limit was the maximal available memory.

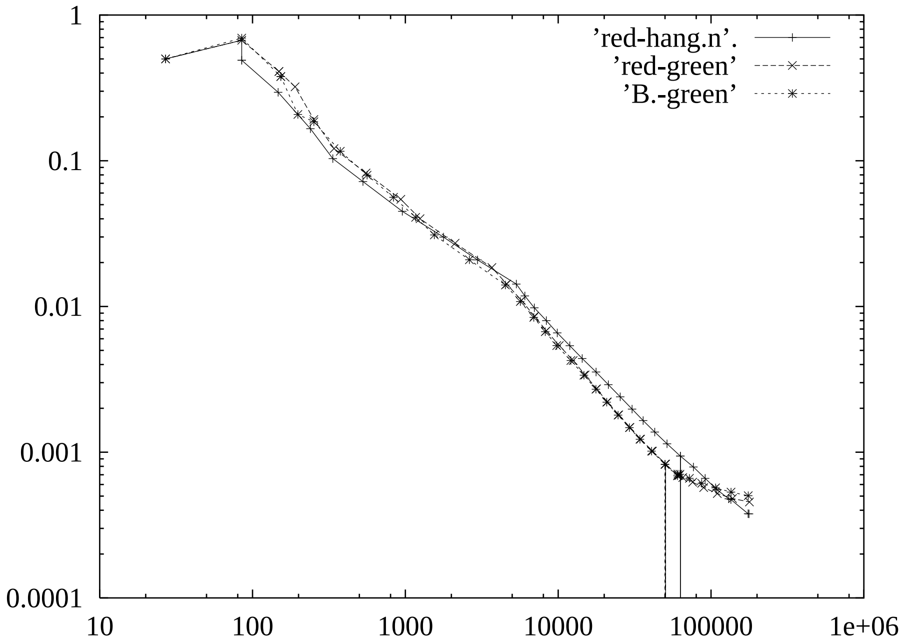

Figure 3: Estimated Error

8 Conclusion

The classical element routines in finite element codes can lead to break downs if they are used in an adaptive regime. The reason is the possibly unstable calculation of the Jacobian–matrices due to serious cancelations of leading digits of nodal coordinates. The way out is different in the cases of triangles and quadrilaterals/hexahedrons. In the triangular case the Jacobian of son elements can be derived from their father element very easily and stable. In case of arbitrary quadrilaterals and hexahedrons, the local Jacobian matrix of a cell $K \subset K_0$ is calculated from the nodes of the coarse element K_0 and a speziell representation vector $\hat{b} \in (\mathbb{B}_L)^d \subset [0,1]^d$ according to the formulas (13) and (37).

References

[AO00] M. Ainsworth, J. T. Oden, A Posteriori Error Estimation in Finite Element Analysis. *Wiley*, New York 2000.

[Bän91] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *IMPACT of Computing in Science and Engineering 3*, 181–191, 1991.

[Mey99] A. Meyer. Projected PCGM for handling hanging nodes in adaptive finite element procedures. Preprint SFB393/99-25, TU Chemnitz, 1999.

[KV00] G. Kunert und R. Verfürth. Edge residuals dominate a posteriori error estimates for linear finite element methods on anisotropic triangular and tetrahedral meshes. *Numer. Math.*, 86(2):283–303, 2000.

[Verf96] R. Verfürth. A review of a posteriori error estimation and adaptive mesh–refinement techniques. *Wiley and Teubner*, Chichester and Stuttgart 1996.

Other titles in the SFB393 series:

01-01 G. Kunert. Robust local problem error estimation for a singularly perturbed problem on anisotropic finite element meshes. January 2001.

01-02 G. Kunert. A note on the energy norm for a singularly perturbed model problem. January 2001.

01-03 U.-J. Görke, A. Bucher, R. Kreißig. Ein Beitrag zur Materialparameteridentifikation bei finiten elastisch-plastischen Verzerrungen durch Analyse inhomogener Verschiebungsfelder mit Hilfe der FEM. Februar 2001.

01-04 R. A. Römer. Percolation, Renormalization and the Quantum-Hall Transition. February 2001.

01-05 A. Eilmes, R. A. Römer, C. Schuster, M. Schreiber. Two and more interacting particles at a metal-insulator transition. February 2001.

01-06 D. Michael. Kontinuumstheoretische Grundlagen und algorithmische Behandlung von ausgewählten Problemen der assoziierten Fließtheorie. März 2001.

01-07 S. Beuchler. A preconditioner for solving the inner problem of the p-version of the FEM, Part II - algebraic multi-grid proof. March 2001.

01-08 S. Beuchler, A. Meyer. SPC-PM3AdH v 1.0 - Programmer's Manual. March 2001.

01-09 D. Michael, M. Springmann. Zur numerischen Simulation des Versagens duktiler metallischer Werkstoffe (Algorithmische Behandlung und Vergleichsrechnungen). März 2001.

01-10 B. Heinrich, S. Nicaise. Nitsche mortar finite element method for transmission problems with singularities. March 2001.

01-11 T. Apel, S. Grosman, P. K. Jimack, A. Meyer. A New Methodology for Anisotropic Mesh Refinement Based Upon Error Gradients. March 2001.

01-12 F. Seifert, W. Rehm. (Eds.) Selected Aspects of Cluster Computing. March 2001.

01-13 A. Meyer, T. Steidten. Improvements and Experiments on the Bramble–Pasciak Type CG for mixed Problems in Elasticity. April 2001.

01-14 K. Ragab, W. Rehm. CHEMPI: Efficient MPI for VIA/SCI. April 2001.

01-15 D. Balkanski, F. Seifert, W. Rehm. Proposing a System Software for an SCI-based VIA Hardware. April 2001.

01-16 S. Beuchler. The MTS-BPX-preconditioner for the p-version of the FEM. May 2001.

01-17 S. Beuchler. Preconditioning for the p-version of the FEM by bilinear elements. May 2001.

01-18 A. Meyer. Programmer's Manual for Adaptive Finite Element Code SPC-PM 2Ad. May 2001.

01-19 P. Cain, M.L. Ndawana, R.A. Römer, M. Schreiber. The critical exponent of the localization length at the Anderson transition in 3D disordered systems is larger than 1. June 2001

01-20 G. Kunert, S. Nicaise. Zienkiewicz-Zhu error estimators on anisotropic tetrahedral and triangular finite element meshes. July 2001.

01-21 G. Kunert. A posteriori H^1 error estimation for a singularly perturbed reaction diffusion problem on anisotropic meshes. August 2001.

01-22 A. Eilmes, Rudolf A. Römer, M. Schreiber. Localization properties of two interacting particles in a quasi-periodic potential with a metal-insulator transition. September 2001.

01-23 M. Randrianarivony. Strengthened Cauchy inequality in anisotropic meshes and application to an a-posteriori error estimator for the Stokes problem. September 2001.

01-24 Th. Apel, H. M. Randrianarivony. Stability of discretizations of the Stokes problem on anisotropic meshes. September 2001.

01-25 Th. Apel, V. Mehrmann, D. Watkins. Structured eigenvalue methods for the computation of corner singularities in 3D anisotropic elastic structures. October 2001.

01-26 P. Cain, F. Milde, R. A. Römer, M. Schreiber. Use of cluster computing for the Anderson model of localization. October 2001. Conf. on Comp. Physics, Aachen (2001).

01-27 P. Cain, F. Milde, R. A. Römer, M. Schreiber. Applications of cluster computing for the Anderson model of localization. October 2001. Transworld Research Network for a review compilation entitled "Recent Research Developments in Physics", (2001).

01-28 X. W. Guan, A. Foerster, U. Grimm, R. A. Römer, M. Schreiber. A supersymmetric Uq[(osp)(2—2)]-extended Hubbard model with boundary fields. October 2001.

01-29 K. Eppler, H. Harbrecht. Numerical studies of shape optimization problems in elasticity using wavelet-based BEM. November 2001.

01-30 A. Meyer. The adaptive finite element method - Can we solve arbitrarily accurate? November 2001.

01-31 H. Harbrecht, S. Pereverzev, R. Schneider. An adaptive regularization by projection for noisy pseudodifferential equations of negative order. November 2001.

01-32 G. N. Gatica, H. Harbrecht, R. Schneider. Least squares methods for the coupling of FEM and BEM. November 2001.