TECHNISCHE UNIVERSITÄT CHEMNITZ

Sonderforschungsbereich 393

Parallele Numerische Simulation für Physik und Kontinuumsmechanik

Arnd Meyer

A New Efficient Preconditioner for Crack Growth Problems

Preprint SFB393/05-06

Preprintreihe des Chemnitzer SFB 393

ISSN 1619-7178 (Print) ISSN 1619-7186 (Internet)

SFB393/05-06 April 2005

Contents

Author's addresses:

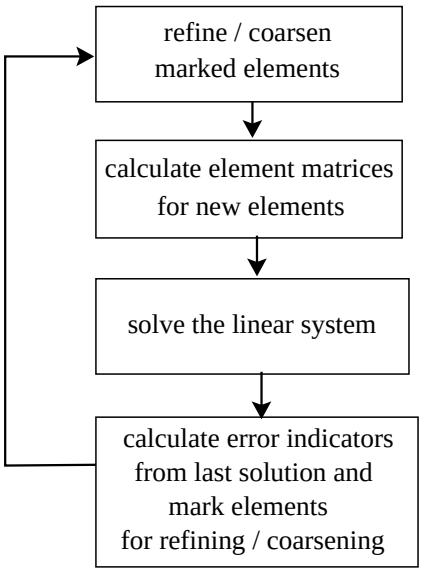
Arnd Meyer
TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

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

1 Introduction

The finite element simulation of crack growth is well understood from the point of modeling this phenomenon. Around an actual crack tip, we are able to calculate stress intensity factors approximately, which give information on the potential growth (or stop) of the existing crack. Additionally we obtain the direction of further crack propagation from approximating the J–integral. For precise results with these approximations based on finite element calculations of the deformation field (at a fixed actual crack situation), a proper mesh with refinement around the crack tip is necessary. From the well established error estimators / error indicators, we are able to control this mesh refinement in using adaptive finite element method.

This means, at the fixed actual crack situation we perform some steps of following adaptive loop:



After three or four sub-calculations in this loop, we end at an approximation of the actual situation that allows us to decide the crack behavior precise enough. After calculating the direction of propagation $\vec{J}$, we are able to return this adaptive loop at a slightly changed mesh with longer crack (for details see [5]).

This new adaptive calculation on a slightly changed mesh was the original challenge for the adaptive solver strategy:
All modern iterative solvers such as preconditioned conjugate gradients (PCGM) use hierarchical techniques for efficient preconditioners. Examples are the Multi-grid method, the hierarchical–basis preconditioner [10] or (especially for 3D) the BPX–preconditioner [2]. The implementation of these multi level techniques requires (among others) a hierarchical order of the unknowns. From the adaptive mesh refinement such a hierarchical node ordering is given by the way, if we store the full edge tree. From this reason we cannot allow

the introduction of some extra edges (“ double” the edges) along the new crack line. A reasonable way out of this problem is discussed in the next Chapter.

2 Data Structure for Maintaining Hierarchies

The idea of the extension of the crack line is given in [5]. We subdivide the existing edges that cut the crack line at the cutting point. Then the usual mesh refinement creates (during “red” or “green” subdivision of some elements) new edges along the crack line together with a proper slightly refined mesh and the correct hierarchies in the new edge tree.

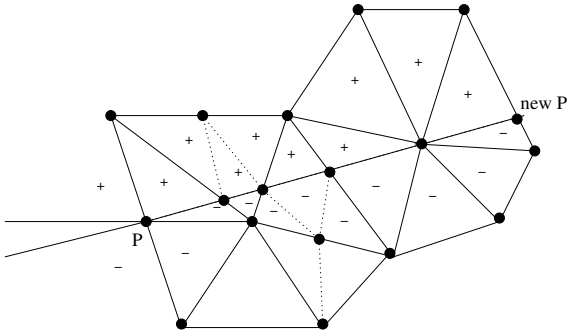


Figure 1: Mesh handling after crack extension from the old crack tip P to “*new* P”

So, there are no “double” edges along the crack line. For defining the double number of unknowns at the so called “crack–nodes”, we introduce a copy of each crack–node and call the old one “–”-node on one side of the crack and the other “+”-node at the other side. Now, the hierarchical preconditioner can access from its edge tree information only the usual and the “–”-values but never the new additional “+”-values. An efficient preconditioner has to combine both informations as it was expected in [5] from a simple averaging technique of the results of two preconditioners (first with usual and “–”-values, then with usual and “+”-values). This simple approach never can lead to a spectrally equivalent preconditioner which is clearly seen in relative high numbers of PCG–iterations. Here, another approach with a basis change and a domain decomposition like method will be given in the next chapter.

3 Two Kind of Basis Functions for Crack Growth Finite Elements

The fact that we have a coherent continuum before the growth of the crack that changes into a slit–domain after growing requires a special finite element treatment. One possibility is the construction of a **new mesh** after crack growing, where the new free crack shores are usual free boundaries (with zero traction boundary condition). This is far away from being efficient because we have a mesh from Chapter 2 and the error indicators will drive the

future refinement and coarsening to the required mesh for good approximating this new situation. So, we start to work with the **existing mesh** with double degrees of freedom at the crack nodes.

From the technique in [5], the crack–line in the undeformed domain is represented by some edges. Each edge refers to its end–nodes. For the calculation of the crack opening these nodes carry twice as much degrees of freedom and are called "crack–nodes".

Let the total number of nodes $N = n + d$ of the actual mesh be split into n usual nodes and d crack–nodes. The degrees of freedom of the crack–nodes are called "–"-values on one shore of the crack and (different) "+"-values on the other shore. A finite element which contains at least one crack–node is called "–"-element, if it refers to "–"-values (lays at "–"-side of the crack) and conversely, a "+"-element refers to "+"-values and lays at the other side of the crack–line (as indicated in Fig.1).

From the usual element by element calculation of the stiffness matrix, using these 2d double unknowns along the d crack–nodes, we understand the resulting stiffness matrix as usual finite element matrix that belongs to the following basis of (vector) ansatz functions:

$$\Phi = \left(\varphi_1 I\,, \ldots,\, \varphi_n I,\ \varphi^-_{n+1} I\,, \ldots,\, \varphi^-_{n+d} I,\ \varphi^+_{n+1} I\,, \ldots,\, \varphi^+_{n+d} I\right).$$

Here, we write $\varphi_k I = (\varphi_k \vec{e}_1 \,\vdots\, \varphi_k \vec{e}_2)$ to specify the two usual vector ansatz functions. Moreover, φ_k $(k = 1 \ldots n)$ denotes a usual hat–function on the usual node k. In contrast to that, φ^-_{n+j} and φ^+_{n+j} are half hat functions with its support around the crack–node $(n + j)$ at the "–"-elements ("+"-elements resp.) only. (If a usual node k lays on the remaining free boundary of the domain, the function φ_k is such a "half" hat function as well). The total number of ansatz functions is $2 \cdot (n + d + d)$, which is the dimension of the resulting stiffness matrix K.

For the efficient preconditioning of this matrix K, we introduce another basis of possible ansatz functions, that span the same $2(n + 2d)$–dimensional finite element space:

$$\tilde{\Phi} = (\varphi_1 I\,, \ldots,\, \varphi_n I,\ \varphi_{n+1} I\,, \ldots,\, \varphi_{n+d} I,\ \tilde{\varphi}_{n+1} I\,, \ldots,\, \tilde{\varphi}_{n+d} I).$$

Here, φ_{n+j} is the usual full (continuous) hat function at the node $(n + j)$ and $\tilde{\varphi}_{n+j}$ is the product of φ_{n+j} with the Heaviside function of the crack–line. This means:

$$\begin{array}{lcll} \varphi_{n+j} & := & \varphi^-_{n+j} + \varphi^+_{n+j} & \text{(a.e.)} \\ \tilde{\varphi}_{n+j} & := & \varphi^-_{n+j} - \varphi^+_{n+j} & \text{(a.e.)}\ . \end{array} \tag{1}$$

Hence, $\tilde{\varphi}_{n+j}$ has a jump from -1 to $+1$ over the crack–line at the crack–node $(n + j)$.

Theoretically, we can use $\tilde{K}$, the stiffness matrix of $\tilde{\Phi}$ instead of K for the same finite element computation of the new crack opening. Note the following differences between these two basis definitions:

Advantages of Φ:

- usual element routines
- usual post–processing (direct calculation of the displacements of both crack shores)
- usual error estimators / error indicators (at the crack the same data as at free boundaries)

Disadvantage of Φ:

- requires special preconditioner for K

For the basis $\tilde{\Phi}$ the reverse properties are true. The use of $\tilde{\Phi}$ would require some special treatment in element routines, post–processing and a new error control.

For $\tilde{K}$ an efficient preconditioner can be found, so by use of the basis transformation (1) we construct an efficient preconditioner for K as well. Obviously

$$\tilde{\Phi} = \Phi D$$

with the block–diagonal matrix

$$D = \text{ blockdiag } (D_1, D_2),$$

where

$$D_1 = I, \quad (2n \times 2n),$$

and

$$D_2 = \begin{pmatrix} I & I \\ I & -I \end{pmatrix}, \quad (2 \cdot (2d) \times 2 \cdot (2d)).$$

This leads to

$$\tilde{K} = DKD. \tag{2}$$

So, if $\tilde{C}$ is a good preconditioner for $\tilde{K}$, then $C = D^{-1}\tilde{C}D^{-1}$ is as good for K.

4 An Efficient DD–Preconditioner for $\tilde{K}$

From the special structure of the matrix $\tilde{K}$ a domain decomposition approach leads to a very efficient preconditioner. Let us recall the structure of K from the definition of the basis Φ:

$$K = \begin{pmatrix} A & B^- & B^+ \\ (B^-)^T & T^- & \mathbb{O} \\ (B^+)^T & \mathbb{O} & T^+ \end{pmatrix}$$

with the blocks:
A $(2n \times 2n)$ of the energy inner products of all usual basis functions $\varphi_i \vec{e}_k$ with itself,

T^- $(2d \times 2d)$ of the energy inner products of all $\varphi^-_{n+i}\vec{e}_k$ with itself,
T^+ $(2d \times 2d)$ of the energy inner products of all $\varphi^+_{n+i}\vec{e}_k$ with itself and
B^-, B^+ contain all energy inner products of $\varphi_i\vec{e}_k (i \leq n)$ with $\varphi^-_{n+j}\vec{e}_l$ (resp. $\varphi^+_{n+j}\vec{e}_l$).

No "+"-node is coupled to a "–"-node, this leads to both zero blocks (the crack-tip is understood as "usual node").

If we perform the transformation (2), the new matrix $\tilde{K}$ possesses a much more simple structure:

$$\tilde{K} = \begin{pmatrix} A & B^- + B^+ & B^- - B^+ \\ \text{sym.} & T^- + T^+ & T^- - T^+ \\ \text{sym.} & T^- - T^+ & T^- + T^+ \end{pmatrix} \tag{3}$$

which is abbreviated as

$$\tilde{K} = \begin{pmatrix} A_0 & B \\ B^T & T \end{pmatrix}$$

with A_0 the leading $2(n+d)$–block and

$$\begin{aligned} B &= \begin{pmatrix} B^- - B^+ \\ T^- - T^+ \end{pmatrix}, \\ T &= T^- + T^+. \end{aligned}$$

Then the matrix A_0 is defined from all energy inner products of all first $2(n+d)$ usual hat functions of the basis $\tilde{\Phi}$. Hence, A_0 is the usual stiffness matrix for the actual mesh without any crack opening.
All the $n+d$ nodes are represented in a hierarchical structure of the edge–tree, so any kind of multi–level preconditioners can easily be applied for preconditioning A_0.
Especially the most simple hierarchical–basis preconditioner as explained in [5] (for details see [10]) would be very cheap but effective.
A resulting good preconditioner for the whole matrix follows from the fine structure of T: If all crack–nodes are ordered in a 1–dimensional chain, then T has block–tridiagonal form for linear elements or block–pentadiagonal form for quadratic elements. Hence, the storage of the sub block T (upper triangle) can be arranged with

8d values (linear elements, fixed bandwidth scheme),
7d values (linear elements, variable bandwidth scheme),
12d values (quadratic elements, fixed bandwidth scheme) or
9d values (quadratic elements, variable bandwidth scheme),

and the Cholesky decomposition of T is of optimal order of complexity due to the fixed bandwidth.
This is exploited best in a domain decomposition–like preconditioner, which results from a simple block factorization of $\tilde{K}$:

$$\tilde{K} = \begin{pmatrix} I & BT^{-1} \\ \mathbb{O} & I \end{pmatrix} \begin{pmatrix} S & \mathbb{O} \\ \mathbb{O} & T \end{pmatrix} \begin{pmatrix} I & \mathbb{O} \\ T^{-1}B^T & I \end{pmatrix}$$

with the Schur–complement matrix

$$S = A_0 - BT^{-1}B^T.$$

The inverse of $\tilde{K}$ can be approximated by the inverse preconditioner $\tilde{C}^{-1}$:

$$\tilde{C}^{-1} = \begin{pmatrix} I & \mathbb{O} \\ -T^{-1}B^T & I \end{pmatrix} \begin{pmatrix} C_0^{-1} & \mathbb{O} \\ \mathbb{O} & T^{-1} \end{pmatrix} \begin{pmatrix} I & -BT^{-1} \\ \mathbb{O} & I \end{pmatrix},$$

when C_0 represents a good preconditioner for S. Note that all other inverts are exact (especially T^{-1}), so the spectrum of $\tilde{C}^{-1}\tilde{K}$ coincides with the spectrum of $C_0^{-1}S$ and additional unities. With proper scaling, we have $\tilde{C}$ for $\tilde{K}$ as good as C_0 for S. The Schur complement is a rank–2d–perturbation of A_0, so we use the same preconditioner C_0 for S as it was explained for A_0. The resulting preconditioner for K follows from (1),(2) as the formula

$$C^{-1} = D \begin{pmatrix} I & 0 \\ -T^{-1}B^T & I \end{pmatrix} \begin{pmatrix} C_0^{-1} & \mathbb{O} \\ \mathbb{O} & I \end{pmatrix} \begin{pmatrix} I & -BT^{-1} \\ 0 & T^{-1} \end{pmatrix} D.$$

For the action $\underline{w} := C^{-1}\underline{r}$ within the PCGM iterations, we need once the action of the (hierarchical) preconditioner C_0 and twice a solver with T, the remaining effort is a multiplication with parts of the stiffness matrix only, which determines a preconditioner of optimal complexity.

5 Numerical Experiments

The power of the above technique can be demonstrated at the example in [5]. Here, we start with a domain of size $(0,4)\times(-1,1)$ with a crack $(0,1)\times\{0\}$ at the beginning. Then, after each 3 adaptive mesh refinements we let the crack grow with constant direction of $(1,0)^T$. Each new crack extension is bounded by 0.25, so it matches with initial finite element boundaries. This results in relative fine meshes near the actual crack tips, but coarsening along the crack path. In the solution method proposed in [5], this example produced relative high number of PCG–iterations in the succeeding steps, which arrived near 200. This demonstrates that the preconditioner in [5] cannot be (near) spectrally equivalent to the stiffness matrix. This situation is drastically improved by introducing the preconditioner of Chapter 4. Now, the total numbers of necessary iterations are bounded near about 30 over all calculations of a fixed crack (compare Fig. 2). After each new crack hop the necessary PCG–iterations are higher due to the fact that no good starting vector for this new changed mesh is available. This leads to the nine peaks in Fig.2. Here, the preconditioner of [5] exceeded more than 100 iterations in this example and over 300 in the other experiment in [5]. Now, the averaged iteration numbers are far less 100 for both examples.

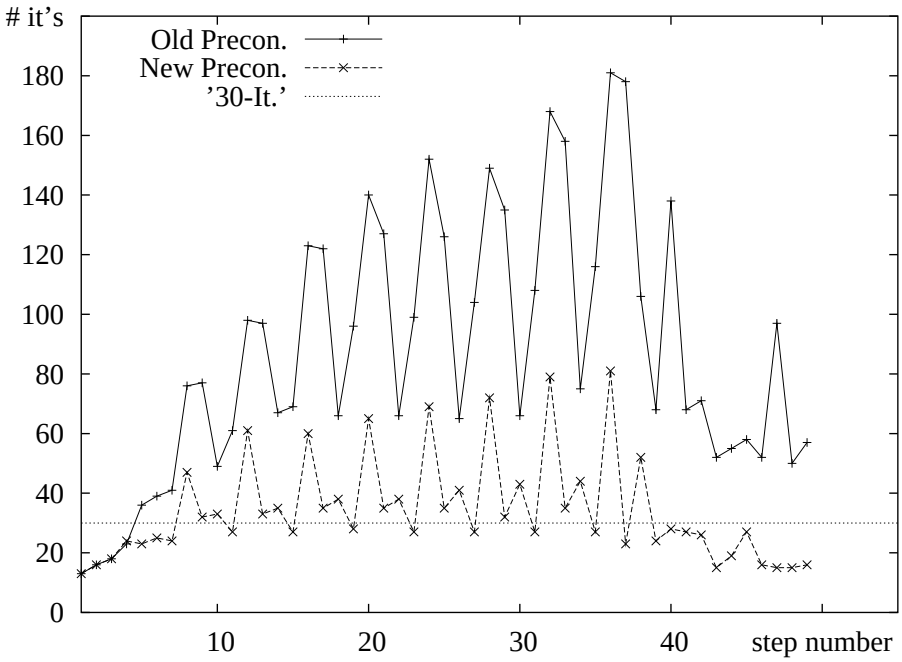


Figure 2: Development of the number of necessary PCG–iterations for the solution method in [5] and the new method

References

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

[2] I.H. Bramble and J. Pasciak, J. E.and Xu: *Parallel Multilevel Preconditioners.* Math. Comp. 55(191), pp. 1–22, 1990.

[3] Jacob Fish and Aditya Nath: *Adaptive and hierarchical modelling of fatigue crack propagation.* International Journal for Numerical Methods in Engineering 36, pp. 2825–2836, 1993.

[4] A. Meyer: *Hierarchical Preconditioners for High Order Elements and Application in Computational Mechanics.* Developments in Computational Mechanics with High Performance Computing, B.Topping (editor), Civil–Comp Press, Edinburgh pp. 107–112, 1999.

[5] A. Meyer, F.Rabold, M.Scherzer: *Efficient Finite Element Simulation of Crack Propagation Using Adaptive Iterative Solvers*
Commun.Numer.Meth Engng 2005,to appear
Preprint-Reihe des Chemnitzer Sonderforschungsbereiches 393, Nr.: 04-01, TU Chemnitz, 2004.

[6] Rolf Rannacher and Franz-Theo Suttmeier: *A feed-back approach to error control in finite element methods: Application to linear elasticity.* Reaktive Strömungen, Diffusion und Transport SFB 359, Interdiziplinäres Zentrum für Wissenschaftliches Rechnen der Universität Heidelberg, 1996.

[7] M. Scherzer and A. Meyer: *Zur Berechnung von Spannungs- und Deformationsfeldern an Interface-Ecken im nichtlinearen Deformationsbereich auf Parallelrechnern.* Preprint-Reihe des Chemnitzer Sonderforschungsbereiches 393 (Numerische Simulation auf massiv parallelen Rechnern), Nr.: 96-03, TU Chemnitz, 1996.

[8] T.J. Stone and I. Babuška: *A numerical method with a posteriori error estimation for determining the path taken by a propagating crack.* Computer Methods in Applied Mechanics and Engineering 160, pp. 245–271, 1998.

[9] P. Wriggers: *Nichtlineare Finite-Element-Methoden.* Springer, Berlin, 2001.

[10] H. Yserentant: *Two Preconditioners Based on the Multilevel Splitting of Finite Element Spaces.* Numerische Mathematik 58, pp. 163–184, 1990.

Other titles in the SFB393 series:

03-01 E. Creusé, G. Kunert, S. Nicaise. A posteriory error estimation for the Stokes problem: Anisotropic and isotropic discretizations. January 2003.

03-02 S. I. Solov'ëv. Existence of the guided modes of an optical fiber. January 2003.

03-03 S. Beuchler. Wavelet preconditioners for the p-version of the FEM. February 2003.

03-04 S. Beuchler. Fast solvers for degenerated problems. February 2003.

03-05 A. Meyer. Stable calculation of the Jacobians for curved triangles. February 2003.

03-06 S. I. Solov'ëv. Eigenvibrations of a plate with elastically attached load. February 2003.

03-07 H. Harbrecht, R. Schneider. Wavelet based fast solution of boundary integral equations. February 2003.

03-08 S. I. Solov'ëv. Preconditioned iterative methods for monotone nonlinear eigenvalue problems. March 2003.

03-09 Th. Apel, N. Düvelmeyer. Transformation of hexahedral finite element meshes into tetrahedral meshes according to quality criteria. May 2003.

03-10 H. Harbrecht, R. Schneider. Biorthogonal wavelet bases for the boundary element method. April 2003.

03-11 T. Zhanlav. Some choices of moments of refinable function and applications. June 2003.

03-12 S. Beuchler. A Dirichlet-Dirichlet DD-pre-conditioner for p-FEM. June 2003.

03-13 Th. Apel, C. Pester. Clément-type interpolation on spherical domains - interpolation error estimates and application to a posteriori error estimation. July 2003.

03-14 S. Beuchler. Multi-level solver for degenerated problems with applications to p-version of the fem. *(Dissertation)* July 2003.

03-15 Th. Apel, S. Nicaise. The inf-sup condition for the Bernardi-Fortin-Raugel element on anisotropic meshes. September 2003.

03-16 G. Kunert, Z. Mghazli, S. Nicaise. A posteriori error estimation for a finite volume discretization on anisotropic meshes. September 2003.

03-17 B. Heinrich, K. Pönitz. Nitsche type mortaring for singularly perturbed reaction-diffusion problems. October 2003.

03-18 S. I. Solov'ëv. Vibrations of plates with masses. November 2003.

03-19 S. I. Solov'ëv. Preconditioned iterative methods for a class of nonlinear eigenvalue problems. November 2003.

03-20 M. Randrianarivony, G. Brunnett, R. Schneider. Tessellation and parametrization of trimmed surfaces. December 2003.

04-01 A. Meyer, F. Rabold, M. Scherzer. Efficient Finite Element Simulation of Crack Propagation. February 2004.

04-02 S. Grosman. The robustness of the hierarchical a posteriori error estimator for reaction-diffusion equation on anisotropic meshes. March 2004.

04-03 A. Bucher, A. Meyer, U.-J. Görke, R. Kreißig. Entwicklung von adaptiven Algorithmen für nichtlineare FEM. April 2004.

04-04 A. Meyer, R. Unger. Projection methods for contact problems in elasticity. April 2004.

04-05 T. Eibner, J. M. Melenk. A local error analysis of the boundary concentrated FEM. May 2004.

04-06 H. Harbrecht, U. Kähler, R. Schneider. Wavelet Galerkin BEM on unstructured meshes. May 2004.

04-07 M. Randrianarivony, G. Brunnett. Necessary and sufficient conditions for the regularity of a planar Coons map. May 2004.

04-08 P. Benner, E. S. Quintana-Ortí, G. Quintana-Ortí. Solving Linear Matrix Equations via Rational Iterative Schemes. October 2004.

04-09 C. Pester. Hamiltonian eigenvalue symmetry for quadratic operator eigenvalue problems. October 2004.

04-10 T. Eibner, J. M. Melenk. An adaptive strategy for hp-FEM based on testing for analyticity. November 2004.

04-11 B. Heinrich, B. Jung. The Fourier-finite-element method with Nitsche-mortaring. November 2004.

04-12 A. Meyer, C. Pester. The Laplace and the linear elasticity problems near polyhedral corners and associated eigenvalue problems. December 2004.

04-13 M. Jung, T. D. Todorov. On the Convergence Factor in Multilevel Methods for Solving 3D Elasticity Problems. December 2004.

05-01 C. Pester. A residual a posteriori error estimator for the eigenvalue problem for the Laplace-Beltrami operator. January 2005.

05-02 J. Badía, P. Benner, R. Mayo, E. Quintana-Ortí, G. Quintana-Ortí, J. Saak. Parallel Order Reduction via Balanced Truncation for Optimal Cooling of Steel Profiles. February 2005.

05-03 C. Pester. CoCoS – Computation of Corner Singularities. April 2005.

05-04 A. Meyer, P. Nestler. Mindlin-Reissner-Platte: Einige Elemente, Fehlerschätzer und Ergebnisse. April 2005.

05-05 P. Benner, J. Saak. Linear-Quadratic Regulator Design for Optimal Cooling of Steel Profiles. April 2005.

The complete list of current and former preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.