TECHNISCHE UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:
Methoden und Werkzeuge für die Nutzung
paralleler Rechnerarchitekturen

Implicit Extrapolation Methods for Variable Coefficient Problems

M. Jung, U. Rüde

TUM-I9542
SFB-Bericht Nr.342/24/95 A
Dezember 1995

TUM–INFO–12-95-I42-200/1.–FI

Anforderungen an: Prof. Dr. A. Bode
Sprecher SFB 342
Institut für Informatik
Technische Universität München
D-80290 München, Germany

Druck: Fakultät für Informatik der
Technischen Universität München

IMPLICIT EXTRAPOLATION METHODS FOR VARIABLE COEFFICIENT PROBLEMS

M. JUNG* AND U. RÜDE†

Abstract. Implicit extrapolation methods for the solution of partial differential equations are based on applying the extrapolation principle *indirectly.* Multigrid tau-extrapolation is a special case of this idea. In the context of multilevel finite element methods, an algorithm of this type can be used to raise the approximation order, even when the meshes are nonuniform or locally refined. The implicit extrapolation multigrid algorithm converges to the solution of a higher order finite element system. This is obtained without explicitly constructing higher order stiffness matrices but by applying extrapolation in a natural form within the algorithm. The algorithm requires only a small change of a basic low order multigrid method.

Key Words. Finite Elements, Extrapolation, Multigrid, Numerical Quadrature.

AMS(MOS) subject classification. 65F10, 65F50, 65N22, 65N50, 65N55.

1. Introduction.

Implicit extrapolation is an efficient technique to improve the accuracy of a multilevel solver. When combined with extrapolation, the multilevel principle is not only used as the basis for a fast algebraic solver, but also to increase the approximation order. The basic idea of extrapolation is to exploit discretizations on different levels.

In classical Richardson extrapolation, two or more approximations from different meshes are combined linearly to eliminate the dominating terms of the error expansion. For partial differential equations this has been studied in the context of finite difference discretizations, see e.g. Marchuk and Shaidurov [8] and in the framework of finite elements (FE), see e.g. Blum, Lin, and Rannacher [2]. These techniques are *explicit* extrapolation methods, since they use approximate solutions *directly.*

Here we propose a different approach, where extrapolation is applied *indirectly* to intermediate quantities of the solution process. Such methods are called *implicit* extrapolation techniques. Methods of this type may be related to defect correction, and — if combined with multigrid — to τ-extrapolation, see e.g. Brandt [3], Hackbusch [5], Schaffer [11], or Bernert [1]. However, these methods are mathematically still motivated by expansions of the truncation error, which in turn require uniform meshes. A generalization to locally uniform meshes can e.g. be found in McCormick and Rüde [9].

In Jung and Rüde [6] we have presented an implicit finite element extrapolation technique which is based on extrapolating the quadrature rules used to compute the stiffness matrices. In [6] it has been shown that within the nested

* Fakultät für Mathematik, Technische Universität Chemnitz-Zwickau, D-09107 Chemnitz, Germany, e-mail: Dr.Michael.Jung@mathematik.tu-chemnitz.de

† Institut für Informatik, Technische Universität München, Arcisstr. 21, D-80290 München, Germany, e-mail: ruede@informatik.tu-muenchen.de

spaces of a multilevel finite element algorithm, this implicit extrapolation converts an h-hierarchical to a p-hierarchical basis. This improves the approximation order, independent of any uniformity constraints on the mesh and without requiring global asymptotic error expansions. On the other hand, the algorithm presented in [6] is algebraically just a special case of multigrid τ-extrapolation, which differs from the usual multilevel process only by an additional factor appearing in the restriction of the residual. The method is therefore particularly convenient to implement in any given multigrid algorithm.

The analysis of [6] was still restricted to problems with element-wise constant coefficients. In this present paper we will now generalize these results to show that an analogous algorithm can be used for variable coefficients as long as the coefficients are smooth enough to justify higher order approximations at all. The analysis is again based on studying quadrature formulas for the stiffness matrices, and using extrapolation to construct quadrature formulas which are exact for higher order polynomial functions. For variable coefficients, this is now significantly more complicated and our analysis requires nonstandard quadrature rules. These rules and the multilevel algorithm are introduced in detail. The final section presents a numerical example showing the efficiency of the method.

2. The boundary value problem and its finite element discretization . In this paper we consider two-dimensional second order elliptic boundary value problems given in the weak formulation

$$\text{Find } u \in V_0 \text{ such that } \quad a(u,v) = \langle F, v\rangle \quad \text{for all } v \in V_0, \tag{1}$$

with

$$a(u,v) = \int_\Omega (A(x)\nabla_x u, \nabla_x v)\,dx \qquad \text{and} \qquad \langle F, v\rangle = \int_\Omega f v\,dx. \tag{2}$$

Ω is a two-dimensional bounded polygonal domain. The space $V_0 = H_0^1(\Omega)$ is a subspace of the Sobolev space $H^1(\Omega)$, where the functions of V_0 satisfy homogeneous Dirichlet boundary conditions on the boundary $\partial\Omega$. The restriction to this type of boundary conditions is only to keep the exposition as simple as possible. The generalization to somewhat more general boundary conditions is analogous to [6].

Furthermore, we suppose that the 2×2 matrix $A(x) = (a_{ij}(x))_{i,j=1,2}$ is symmetric and positive definite for almost all $x \in \Omega$ with $a_{ij}(x) \in W^2_\infty(\Omega)$. The function f belongs to the space $W^2_q(\Omega)$ with $q \geq 2$. We need these assumptions to obtain a discretization error which is typical for FE discretizations with piecewise quadratic functions and the application of appropriate quadrature rules for the computation of the stiffness matrix and the load vector.

We now discretize (1) by three different finite element spaces. We suppose that two nested triangulations $\mathcal{T}_{l-1}$ and $\mathcal{T}_l$ of the domain Ω are given. The finer triangulation $\mathcal{T}_l$ results from $\mathcal{T}_{l-1}$ by regular refinement, that is by connecting the midpoints of all triangles $\delta_{l-1}^{(r)}$, $r = 1, 2, \ldots, R_{l-1}$, in $\mathcal{T}_{l-1}$. Corresponding to the triangulations $\mathcal{T}_{l-1}$ and $\mathcal{T}_l$ we introduce the finite element spaces

$$V_{l-1}^L = \text{span}\{p_{l-1}^{(i)} : i = 1, 2, \ldots, N_{l-1}\} \subset V_0, \tag{3}$$

$$V_l^L = V_{l-1}^L \cup \text{span}\{p_l^{(i)} : i = N_{l-1}+1, \ldots, N_l\} \subset V_0, \tag{4}$$

$$V_l^Q = V_{l-1}^L \cup \text{span}\{q_{l-1}^{(i)} : i = N_{l-1}+1, \ldots, N_l\} \subset V_0. \tag{5}$$

The trial functions $p_k^{(i)}$, $k = l, l-1$, are continuous and piecewise linear in each triangle of $\mathcal{T}_k$ and they satisfy

$$\begin{aligned} p_{l-1}^{(i)}(x^{(j)}) &= \delta_{ij} \quad \text{for } i, j = 1, 2, \ldots, N_{l-1} \\ p_l^{(i)}(x^{(j)}) &= \delta_{ij} \quad \text{for } i, j = N_{l-1}+1, \ldots, N_l. \end{aligned}$$

Here $x^{(j)} = (x_1^{(j)}, x_2^{(j)})$ denotes the coordinates of the node $P^{(j)}$ and N_k is the number of nodes of $\mathcal{T}_k$ in Ω. δ_{ij} is the Kronecker symbol.

The functions $q_{l-1}^{(i)}$, $i = N_{l-1}+1, \ldots, N_l$, of (5) are continuous and piecewise *quadratic* in each triangle of $\mathcal{T}_{l-1}$. Again, they satisfy

$$q_{l-1}^{(i)}(x^{(j)}) = \delta_{ij} \quad \text{for } i, j = N_{l-1}+1, \ldots, N_l\,.$$

The basis of the space V_l^L we call *h-hierarchical basis* and the basis of the space V_l^Q is called *p-hierarchical basis*.

The finite element subspaces V_{l-1}^L, V_l^L, V_l^Q of (3), (4), and (5), respectively, give rise to the finite element stiffness matrices K_{l-1}^L, K_l^L, and K_l^Q as well as the load vectors $\underline{f}_{l-1}^L$, $\underline{f}_l^L$, and $\underline{f}_l^Q$.

For the computation of the coefficients of the element stiffness matrices and the element load vectors in general we must perform numerical integration. We therefore need an appropriate quadrature rule which guarantees the same FE discretization error as in the case of exact computation of the stiffness matrix and the load vector. To investigate the effect of numerical integration we will use well-known results as e.g. contained in [4]. For the sake of completeness we summarize some of them.

The application of quadrature rules for the computation of the matrix elements and the elements of the load vector results in an approximate bilinear form $\tilde{a}(\tilde{u}, \tilde{v})$ and an approximate right-hand side $\langle \tilde{F}, \tilde{v} \rangle$. Depending on the choice of the quadrature rule and the finite element subspace $\tilde{V}$, i.e. $\tilde{V} = V_{l-1}^L, \tilde{V} = V_l^L$, or $\tilde{V} = V_l^Q$, we will later describe $\tilde{a}(\tilde{u}, \tilde{v})$ in detail.

The approximate bilinear form is called *uniformly* $\tilde{V}$-elliptic, if there exists a constant $\tilde{\alpha} > 0$, $\tilde{\alpha}$ independent of $\tilde{V}$, such that

$$\tilde{a}(\tilde{v}, \tilde{v}) \geq \tilde{\alpha} ||\tilde{v}||_{1,2,\Omega}^2 \quad \text{for all } \tilde{v} \in \tilde{V}.$$

Here $|| \cdot ||_{1,2,\Omega}$ denotes the norm in the Sobolev space $H^1(\Omega)$.

Using numerical integration, the boundary value problem (1) is approximated by

(6) $$\text{Find } \tilde{u} \in \tilde{V} \text{ such that } \quad \tilde{a}(\tilde{u}, \tilde{v}) = \langle \tilde{F}, \tilde{v} \rangle \quad \text{for all } \tilde{v} \in \tilde{V}.$$

THEOREM 2.1. (First Lemma of Strang) *Let the approximate bilinear form $\tilde{a}$ of (6) be uniformly $\tilde{V}$-elliptic. Then*

$$||u - \tilde{u}||_{1,2,\Omega} \leq c \left(\inf_{\tilde{v} \in \tilde{V}} \left\{ ||u - \tilde{v}||_{1,2,\Omega} + \sup_{\tilde{w} \in \tilde{V}} \frac{|a(\tilde{v}, \tilde{w}) - \tilde{a}(\tilde{v}, \tilde{w})|}{||\tilde{w}||_{1,2,\Omega}} \right\} + \sup_{\tilde{w} \in \tilde{V}} \frac{|\langle F, \tilde{w} \rangle - \langle \tilde{F}, \tilde{w} \rangle|}{||\tilde{w}||_{1,2,\Omega}} \right)$$

with a constant c which does not depend on the space $\tilde{V}$.

Let the solution $u \in H_0^{s+1}(\Omega)$, $a_{ij} \in W_\infty^s(\Omega)$, $i, j = 1, 2$, $f \in W_q^s(\Omega)$ with $q \geq 2$ and $q > 2/s$, and let the FE subspace $\tilde{V}$ contain piecewise polynomials of degree s, i.e. polynomials of degree s on the triangles of the triangulation. Furthermore, let the quadrature rule be exact for polynomial of degree $2s - 2$ on each triangle. Then the following estimate holds (see also [4])

$$\|u - \tilde{u}\|_{1,2,\Omega} \leq ch^s \left(|u|_{s+1,2,\Omega} + \sum_{i,j=1}^{2} \|a_{ij}\|_{s,\infty,\Omega} \|u\|_{s+1,2,\Omega} + \|f\|_{s,q,\Omega} \right).$$

Here $\| \cdot \|_{s+1,2,\Omega}$ and $| \cdot |_{s+1,2,\Omega}$ denote norms in $H_0^{s+1}(\Omega)$ as well as $\| \cdot \|_{s,q,\Omega}$ is a norm in $W_q^s(\Omega)$.

3. A multigrid algorithm with implicit extrapolation step.

In Jung/Rüde [6] we have studied the convergence properties of a multigrid algorithm with implicit extrapolation step. However, the paper [6] iis restricted to problems with piecewise constant functions $a_{ij}(x)$ and $f(x)$ in $\mathcal{T}_{l-1}$. If such a problem is discretized by *linear* elements, and the multigrid algorithm is combined with (implicit) extrapolation, the iterates converge to the solution given by *quadratic* elements. In this paper we will generalize this result to the case of variable coefficients. It will be shown that the extrapolation algorithm converges to the solution obtained with quadratic elements. In the analysis of this more general case, we will use special nonstandard quadrature rules.

In the following, we will give a brief description of the smoothing procedure and the restriction operator used. Then we formulate the multigrid algorithm and study the convergence behavior.

Numbering the nodes in $\mathcal{T}_l$ such that the nodes which are also in the coarse mesh $\mathcal{T}_{l-1}$ appear first, we induce a block partitioning of the stiffness matrices

(7) $$K_l^L = \begin{pmatrix} K_{l,vv}^L & K_{l,vm}^L \\ K_{l,mv}^L & K_{l,mm}^L \end{pmatrix}, \quad K_l^Q = \begin{pmatrix} K_{l,vv}^Q & K_{l,vm}^Q \\ K_{l,mv}^Q & K_{l,mm}^Q \end{pmatrix}.$$

In the multigrid algorithm we use the following smoothing procedures:

- Pre-smoothing $G_l^V(\underline{u}_l^{(j)}, K_l^L, \underline{f}_l^L)$: Let the initial guess $\underline{u}_l^{(j)} = (\underline{u}_{l,v}^{(j)}, \underline{u}_{l,m}^{(j)})^T$ be given. Set $\underline{u}_{l,v}^{(j+1)} = \underline{u}_{l,v}^{(j)}$ and compute an approximate solution $\underline{z}_{l,m}$ of the system

$$K_{l,mm}^L \underline{z}_{l,m} = \underline{f}_{l,m}^L - K_{l,mv}^L \underline{u}_{l,v}^{(j+1)} - K_{l,mm}^L \underline{u}_{l,m}^{(j)} \tag{8}$$

by means of a linear iterative method starting with the zero vector. We suppose that the error transmission operator of the method is of the type

$$M_{l,m} = I_{l,m} - B_{l,mm}^{-1} K_{l,mm}^L .$$

Then set $\underline{u}_l^{(j+1)} = (\underline{u}_{l,v}^{(j+1)}, \underline{u}_{l,m}^{(j)} + \underline{z}_{l,m})^T$.
- Post-smoothing $G_l^N(\underline{u}_l^{(j)}, K_l^L, \underline{f}_l^L)$: We use the same form of algorithm as for pre-smoothing. However, we suppose that the error transmission operator of the iterative method is of the form $M_{l,m} = I_{l,m} - B_{l,mm}^{-T} K_{l,mm}^L$ such that the overall multigrid operator becomes symmetric.
- We need the *injection operator*

$$I_l^{l-1,inj} : \mathrm{R}^{N_l} \longrightarrow \mathrm{R}^{N_{l-1}}$$

in our algorithm.

Algorithm MG–EX

Let an initial guess $\underline{u}_l^{(k,0)}$ be given.

1. Pre-smoothing:

$$\underline{u}_l^{(k,1)} = G_l^V(\underline{u}_l^{(k,0)}, K_l^L, \underline{f}_l^L)\,. \tag{9}$$

2. Coarse grid correction:
   (a) Compute the defect

$$\underline{d}_{l-1}^{(k)} = \frac{4}{3}\left(\underline{f}_{l,v}^L - K_{l,vv}^L \underline{u}_{l,v}^{(k,1)} - K_{l,vm}^L \underline{u}_{l,m}^{(k,1)}\right) - \frac{1}{3}\left(\underline{f}_{l-1}^L - K_{l-1}^L I_l^{l-1,inj} \underline{u}_l^{(k,1)}\right) \tag{10}$$

   (b) Solve

$$K_{l-1}^L \underline{w}_{l-1}^{(k)} = \underline{d}_{l-1}^{(k)}\,, \tag{11}$$

   using μ iteration steps of a usual symmetric multigrid ($(l-1)$–grid) algorithm, starting with the 0 vector and returning an approximate solution $\underline{\tilde{w}}_{l-1}^{(k)}$.

   (c) Correct

$$\underline{u}_l^{(k,2)} = (\underline{u}_{l,v}^{(k,1)} + \underline{\tilde{w}}_{l-1}^{(k)}, \underline{u}_{l,m}^{(k,1)})^T\,. \tag{12}$$

3. Post-smoothing:

$$\underline{u}_l^{(k,3)} = G_l^N(\underline{u}_l^{(k,2)}, K_l^L, \underline{f}_l^L) \tag{13}$$

and set $\underline{u}_l^{(k+1,0)} = \underline{u}_l^{(k,3)}$.

Taking into consideration the definition of the smoothing procedures and the equivalence of step 2(a) to

$$\underline{d}_{l-1}^{(k)} = \left(\frac{4}{3}\underline{f}_{l,v}^L - \frac{1}{3}\underline{f}_{l-1}^L\right) - \left(\frac{4}{3}K_{l,vv}^L - \frac{1}{3}K_{l-1}^L\right)\underline{u}_{l,v}^{(k,1)} - \frac{4}{3}K_{l,vm}^L\underline{u}_{l,m}^{(k,1)}, \tag{14}$$

we can interpret our algorithm as a usual multigrid algorithm in the h-hierarchical basis to solve the system of equations

$$K_l^{L,ex}\underline{u}_l = \underline{f}_l^{L,ex} \tag{15}$$

with

$$K_l^{L,ex} = \begin{pmatrix} \frac{4}{3}K_{l,vv}^L - \frac{1}{3}K_{l-1}^L & \frac{4}{3}K_{l,vm}^L \\ \frac{4}{3}K_{l,mv}^L & \frac{4}{3}K_{l,mm}^L \end{pmatrix} \quad \text{and} \quad \underline{f}_l^{L,ex} = \begin{pmatrix} \frac{4}{3}\underline{f}_{l,v}^L - \frac{1}{3}\underline{f}_{l-1}^L \\ \frac{4}{3}\underline{f}_{l,m}^L \end{pmatrix}. \tag{16}$$

The main result of this section is that the iterates of the algorithm MG–EX converge to a FE solution which has the same order of discretization error as a FE solution obtained by p-hierarchical FE functions ($p = 2$).

Before we prove this fact, we introduce the quadrature rules that are used to compute the stiffness matrices and load vectors.

To obtain the entries of the stiffness matrices K_{l-1}^L, K_l^L, and K_l^Q, respectively, we have to compute

$$a(\tilde{p}_l^{(j)}, \tilde{p}_l^{(i)}) = \int_\Omega \left(A(x)\nabla_x\tilde{p}_l^{(j)}(x), \nabla_x\tilde{p}_l^{(i)}(x)\right) dx\,, \tag{17}$$

where $\tilde{p}_l^{(i)}$, $\tilde{p}_l^{(j)}$ stand for the functions $p_{l-1}^{(i)}$, $p_{l-1}^{(j)}$, $i,j = 1,\ldots,N_{l-1}$, $p_l^{(i)}$, $p_l^{(j)}$, $i,j = N_{l-1}+1,\ldots,N_l$, in the case of the h-hierarchical basis. In the case of the p-hierarchical basis the functions $\tilde{p}_l^{(i)}$, $\tilde{p}_l^{(j)}$ stand for $p_{l-1}^{(i)}$, $p_{l-1}^{(j)}$, $i,j = 1,\ldots,N_{l-1}$, $q_{l-1}^{(i)}$, $q_{l-1}^{(j)}$, $i,j = N_{l-1}+1,\ldots,N_l$.

First we explain the quadrature rules used for the computation of the matrices K_{l-1}^L and K_l^L. From (17) we obtain for the entries of K_{l-1}^L

$$\int_\Omega \left(A(x)\nabla_x p_{l-1}^{(j)}(x), \nabla_x p_{l-1}^{(i)}(x)\right) dx = \sum_{r\in\omega_{l-1}^{(ij)}} \int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x p_{l-1}^{(j)}(x), \nabla_x p_{l-1}^{(i)}(x)\right) dx\,, \tag{18}$$

where

$$\omega_{l-1}^{(ij)} = \left\{r \;:\; p_{l-1}^{(i)} \not\equiv 0 \;\text{ and }\; p_{l-1}^{(j)} \not\equiv 0 \;\text{ on }\; \delta_{l-1}^{(r)}\right\}. \tag{19}$$

We transform the integrals over $\delta_{l-1}^{(r)}$ into integrals over the reference element $\Delta = \{(\xi_1, \xi_2) \,:\, 0 \le \xi_1 \le 1,\, 0 \le \xi_2 \le 1,\, \xi_1 + \xi_2 \le 1\}$. This leads to

$$\begin{aligned}
&\int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x p_{l-1}^{(j)}(x), \nabla_x p_{l-1}^{(i)}(x)\right) dx \\
(20) \quad &= \int_\Delta \left(A(x)(J_{l-1}^{(r)})^{-T}\nabla_\xi p_{l-1}^{(j)}(x(\xi)), (J_{l-1}^{(r)})^{-T}\nabla_\xi p_{l-1}^{(i)}(x(\xi))\right) |\det J_{l-1}^{(r)}|\, d\xi \\
&= \int_\Delta \left(\bar{B}^{(r)}(x)\nabla_\xi \varphi_{\beta^{(r)}}(\xi), \nabla_\xi \varphi_{\alpha^{(r)}}(\xi)\right) d\xi
\end{aligned}$$

with $\bar{B}^{(r)}(x) = (J_{l-1}^{(r)})^{-1}A(x)(J_{l-1}^{(r)})^{-T}|\det J_{l-1}^{(r)}|$ and $J_{l-1}^{(r)}$ from the transformation

$$\begin{aligned}
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} x_1^{(r,2)} - x_1^{(r,1)} & x_1^{(r,3)} - x_1^{(r,1)} \\ x_2^{(r,2)} - x_2^{(r,1)} & x_2^{(r,3)} - x_2^{(r,1)} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \begin{pmatrix} x_1^{(r,1)} \\ x_2^{(r,1)} \end{pmatrix} \\
(21) \quad &= J_{l-1}^{(r)} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \begin{pmatrix} x_1^{(r,1)} \\ x_2^{(r,1)} \end{pmatrix}.
\end{aligned}$$

Here $x_i^{(r,\alpha)}$, $i, j = 1, 2$, $\alpha = 1, 2, 3$, denotes the coordinates of the vertices of the triangle $\delta_{l-1}^{(r)}$, and $\alpha^{(r)}$ as well as $\beta^{(r)}$ are the local numbers of the vertices $P^{(i)}$ and $P^{(j)}$. The linear functions $\varphi_{\alpha^{(r)}}$, $\varphi_{\beta^{(r)}}$, $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, 3$, on the reference element are defined by

$$(22) \qquad \varphi_1(\xi) = 1 - \xi_1 - \xi_2\,, \quad \varphi_2(\xi) = \xi_1\,, \quad \text{and} \quad \varphi_3(\xi) = \xi_2.$$

The following equivalent formulation of (20) is the basis of the application of our quadrature rules.

With the directional derivative

$$(23) \qquad \frac{\partial \varphi}{\partial \xi_s} = \frac{\partial \varphi}{\partial \xi_2} - \frac{\partial \varphi}{\partial \xi_1}$$

we obtain

$$\begin{aligned}
&\int_\Delta \left(\bar{b}_{11}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_1}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_1} + \bar{b}_{12}^{(r)}\left(\frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_1}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_2} + \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_2}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_1}\right) + \bar{b}_{22}^{(r)}\frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_2}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_2}\right) d\xi \\
(24) \quad &= \int_\Delta \left(b_{11}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_1}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_1} + b_{22}^{(r)}\frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_2}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_2} + b_{12}^{(r)}\frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_s}\frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_s}\right) d\xi,
\end{aligned}$$

where

$$\begin{aligned}
b_{11}^{(r)}(x(\xi)) &= \bar{b}_{11}^{(r)}(x(\xi)) + \bar{b}_{12}^{(r)}(x(\xi))\,, \qquad b_{22}^{(r)}(x(\xi)) = \bar{b}_{22}^{(r)}(x(\xi)) + \bar{b}_{12}^{(r)}(x(\xi))\,, \\
b_{12}^{(r)}(x(\xi)) &= -\bar{b}_{12}^{(r)}(x(\xi))\,.
\end{aligned}$$

Numerical schemes to evaluate (24) directly have been developed by Rüde [10] and Lyness and Rüde [7]. For the numerical integration of the three terms in (24) we use the following three quadrature rules

$$\int_\Delta v(\xi) d\xi = \text{meas}\Delta\ v(\xi^{(\sigma)}), \qquad \sigma = 1, 2, 3, \tag{25}$$

with

$$\xi^{(1)} = \left(\frac{1}{2}, 0\right), \ \xi^{(2)} = \left(0, \frac{1}{2}\right), \ \text{and} \ \xi^{(3)} = \left(\frac{1}{2}, \frac{1}{2}\right), \tag{26}$$

respectively. Obviously the quadrature rules in (25) are exact for constant functions v.

The elements of the matrix K_l^L are computed in the same way. We can write the expressions for the computation of the matrix elements in the following formulation

$$\begin{aligned} a(\tilde{p}_l^{(j)}, \tilde{p}_l^{(i)}) &= \sum_{r \in \omega_{l-1}^{(ij)}} \int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x \tilde{p}_l^{(j)}(x), \nabla_x \tilde{p}_l^{(i)}(x)\right) dx \\ &= \sum_{r \in \omega_{l-1}^{(ij)}} \int_\Delta \left(A(x)(J_{l-1}^{(r)})^{-T}\nabla_\xi \tilde{p}_l^{(j)}(x(\xi)), (J_{l-1}^{(r)})^{-T}\nabla_\xi \tilde{p}_l^{(i)}(x(\xi))\right) |\det J_{l-1}^{(r)}|\, d\xi \\ &= \sum_{r \in \omega_{l-1}^{(ij)}} \sum_{k=1}^{4} \int_{\Delta^{(k)}} \left(\bar{B}^{(r)}(x)\nabla_\xi \varphi_{\beta^{(r)}}(\xi), \nabla_\xi \varphi_{\alpha^{(r)}}(\xi)\right) d\xi\,, \end{aligned} \tag{27}$$

where again $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, \ldots, 6$, are the local numbers of the nodes $P^{(i)}$ and $P^{(j)}$, $\omega_{l-1}^{(ij)} = \left\{r\ :\ \tilde{p}_l^{(i)} \not\equiv 0 \ \text{and} \ \tilde{p}_l^{(j)} \not\equiv 0 \ \text{on} \ \delta_{l-1}^{(r)}\right\}$, $\Delta = \cup_{k=1}^4 \Delta^{(k)}$ (see also Figure 1), and

$$\begin{aligned} \varphi_1(\xi) &= 1 - \xi_1 - \xi_2\,, \\ \varphi_2(\xi) &= \xi_1\,, \\ \varphi_3(\xi) &= \xi_2\,, \end{aligned} \qquad \varphi_4(\xi) = \begin{cases} 2\xi_1 & \text{in} \ \Delta^{(1)} \\ 2 - 2\xi_1 - 2\xi_2 & \text{in} \ \Delta^{(2)} \\ 0 & \text{in} \ \Delta^{(3)} \\ 1 - 2\xi_2 & \text{in} \ \Delta^{(4)} \end{cases}, \tag{28}$$

$$\varphi_5(\xi) = \begin{cases} 0 & \text{in} \ \Delta^{(1)} \\ 2\xi_2 & \text{in} \ \Delta^{(2)} \\ 2\xi_1 & \text{in} \ \Delta^{(3)} \\ 2\xi_1 + 2\xi_2 - 1 & \text{in} \ \Delta^{(4)} \end{cases}, \quad \varphi_6(\xi) = \begin{cases} 2\xi_2 & \text{in} \ \Delta^{(1)} \\ 0 & \text{in} \ \Delta^{(2)} \\ 2 - 2\xi_1 - 2\xi_2 & \text{in} \ \Delta^{(3)} \\ 1 - 2\xi_1 & \text{in} \ \Delta^{(4)} \end{cases}$$

To compute each integral over $\Delta^{(k)}$ in (27) we use the equivalent formulation of type (24) and a quadrature rule of type (25).

In the case of the p-hierarchical basis, we have to compute the entries of the matrix K_l^Q, i.e. expressions of the form (17), where $\tilde{p}_l^{(i)}, \tilde{p}_l^{(j)}$ stand for the functions $p_{l-1}^{(i)}, p_{l-1}^{(j)}$, $i, j = 1, \ldots, N_{l-1}$, $q_{l-1}^{(i)}, q_{l-1}^{(j)}$, $i, j = N_{l-1} + 1, \ldots, N_l$.

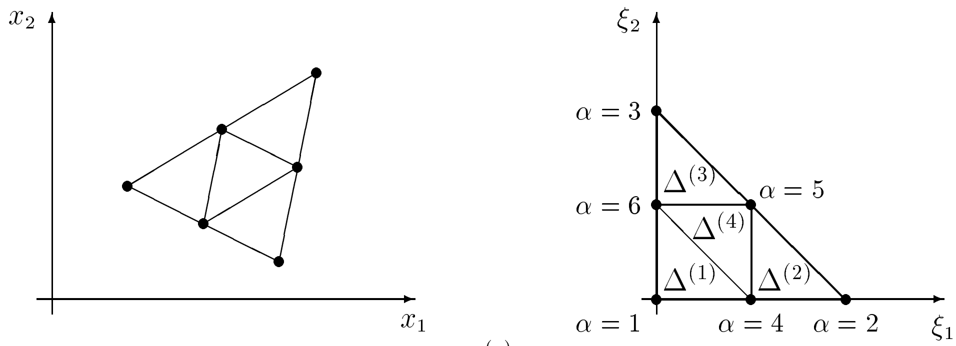

FIG. 1. *An arbitrary triangle* $\delta_{l-1}^{(r)}$ *and the reference element* Δ

Again we get

$$\int_\Omega \Big(A(x)\nabla_x \tilde{p}_l^{(j)}(x), \nabla_x \tilde{p}_l^{(i)}(x)\Big)\, dx = \sum_{r\in\omega_{l-1}^{(ij)}} \int_{\delta_{l-1}^{(r)}} \Big(A(x)\nabla_x \tilde{p}_l^{(j)}(x), \nabla_x \tilde{p}_l^{(i)}(x)\Big)\, dx\,, \tag{29}$$

with $\omega_{l-1}^{(ij)}$ from (19). After the transformation of the integrals over $\delta_{l-1}^{(r)}$ into integrals over the reference element Δ we obtain the integrals

$$\int_\Delta \Big(\bar{B}^{(r)}(x)\nabla_\xi \psi_{\beta^{(r)}}(\xi), \nabla_\xi \psi_{\alpha^{(r)}}(\xi)\Big)\, d\xi\,. \tag{30}$$

The functions $\psi_{\alpha^{(r)}}$ and $\psi_{\beta^{(r)}}$, $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, \ldots, 6$, are defined by

$$\begin{array}{lll} \psi_1(\xi) = 1 - \xi_1 - \xi_2\,, & \psi_2(\xi) = \xi_1\,, & \psi_3(\xi) = \xi_2\,, \\ \psi_4(\xi) = 4\xi_1(1 - \xi_1 - \xi_2)\,, & \psi_5(\xi) = 4\xi_1\xi_2\,, & \psi_6(\xi) = 4\xi_2(1 - \xi_1 - \xi_2)\,. \end{array} \tag{31}$$

The integral (30) we write in the form (24). For the numerical integration of the resulting integrals over Δ we use quadrature rules, which we derive from the quadrature rules (25) by extrapolation. Specifically, we apply for the computation of the first, the second, and the third term the quadrature rules

$$\int_\Delta v(\xi)d\xi \;\approx\; \left\{\frac{4}{3}\left(\frac{1}{4}v(\xi^{(4)}) + \frac{1}{4}v(\xi^{(5)}) + \frac{1}{2}v(\xi^{(6)})\right) - \frac{1}{3}v(\xi^{(1)})\right\} \operatorname{meas}\Delta \tag{32}$$

$$\int_\Delta v(\xi)d\xi \;\approx\; \left\{\frac{4}{3}\left(\frac{1}{4}v(\xi^{(7)}) + \frac{1}{4}v(\xi^{(8)}) + \frac{1}{2}v(\xi^{(9)})\right) - \frac{1}{3}v(\xi^{(2)})\right\} \operatorname{meas}\Delta \tag{33}$$

$$\int_\Delta v(\xi)d\xi \;\approx\; \left\{\frac{4}{3}\left(\frac{1}{4}v(\xi^{(10)}) + \frac{1}{4}v(\xi^{(11)}) + \frac{1}{2}v(\xi^{(12)})\right) - \frac{1}{3}v(\xi^{(3)})\right\} \operatorname{meas}\Delta \tag{34}$$

with $\xi^{(1)}$, $\xi^{(2)}$, $\xi^{(3)}$ from (26) and

$$\begin{array}{llllll} \xi^{(4)} &= (\frac{1}{4}, 0)\,, & \xi^{(5)} &= (\frac{3}{4}, 0)\,, & \xi^{(6)} &= (\frac{1}{4}, \frac{1}{2})\,, \\ \xi^{(7)} &= (0, \frac{1}{4})\,, & \xi^{(8)} &= (0, \frac{3}{4})\,, & \xi^{(9)} &= (\frac{1}{2}, \frac{1}{4})\,, \\ \xi^{(10)} &= (\frac{3}{4}, \frac{1}{4})\,, & \xi^{(11)} &= (\frac{1}{4}, \frac{3}{4})\,, & \xi^{(12)} &= (\frac{1}{4}, \frac{1}{4}) \end{array} \tag{35}$$

(see also Figure 2). A simple calculation shows that the quadrature rules (32) – (34) are exact for quadratic functions.

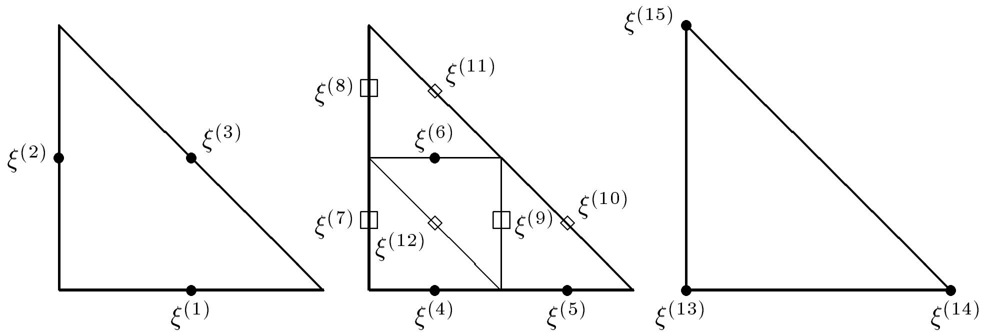

FIG. 2. *Quadrature points of the formulas (25), (32) - (34), (39), and (42)*

Because of the smoothness of the coefficient functions a_{ij} in (2) one can prove that the quadrature rules (25) and (32) – (34) lead for sufficiently small discretization parameters h to a uniformly $\tilde{V}$-elliptic bilinear form $\tilde{a}(.,.)$.

In the following we prove that the extrapolated stiffness matrix in (16) is equal to the stiffness matrix resulting from a discretization with p-hierarchical functions, where we assume that we use the quadrature rules (25) in case of the h-hierarchical basis, and (32) – (34) in case of the p-hierarchical basis.

LEMMA 3.1. *If we compute the element stiffness matrices K_{l-1}^L, K_l^L, and K_l^Q as described above, i.e. by means of the quadrature rules (25) and (32) – (34), the relation*

$$K_l^{L,ex} = K_l^Q \tag{36}$$

holds.

Proof: The proof is based on comparing the matrices $K_l^{L,ex}$ and K_l^Q element by element. The extrapolated stiffness matrix $K_l^{L,ex}$ and the matrix K_l^Q have the block structure

$$K_l^{L,ex} = \begin{pmatrix} \frac{4}{3}K_{l,vv}^L - \frac{1}{3}K_{l-1}^L & \frac{4}{3}K_{l,vm}^L \\ \frac{4}{3}K_{l,mv}^L & \frac{4}{3}K_{l,mm}^L \end{pmatrix}, \quad K_l^Q = \begin{pmatrix} K_{l,vv}^Q & K_{l,vm}^Q \\ K_{l,mv}^Q & K_{l,mm}^Q \end{pmatrix}. \tag{37}$$

The entries of the stiffness matrix K_{l-1}^L are computed using relations (17)–(22) and for the computation of the elements of the matrix K_l^L we use relations (27)–(28).

First, we now prove the identity of the coarse mesh blocks, i.e. $K_{l,vv}^{L,ex} = K_{l,vv}^Q$. Using the quadrature rules of type (25) and the representation (24) with $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, 3$, the elements of the matrix $K_{l,vv}^{L,ex}$ are defined by

$$
\begin{aligned}
K_{l,vv}^{L,ex,(ij)} &= \sum_{r\in\omega_{l-1}^{(ij)}} \Bigg\{ \\
&\frac{4}{3}\sum_{t=4}^{6}\gamma_t^L b_{11}^{(r)}(x(\xi^{(t)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_1}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_1} - \frac{1}{3}\gamma_1^L b_{11}^{(r)}(x(\xi^{(1)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(1)})}{\partial\xi_1}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(1)})}{\partial\xi_1} \\
&+ \frac{4}{3}\sum_{t=7}^{9}\gamma_t^L b_{22}^{(r)}(x(\xi^{(t)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_2}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_2} - \frac{1}{3}\gamma_2^L b_{22}^{(r)}(x(\xi^{(2)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(2)})}{\partial\xi_2}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(2)})}{\partial\xi_2} \\
&+ \frac{4}{3}\sum_{t=10}^{12}\gamma_t^L b_{12}^{(r)}(x(\xi^{(t)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_s}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_s} - \frac{1}{3}\gamma_3^L b_{12}^{(r)}(x(\xi^{(3)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(3)})}{\partial\xi_s}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(3)})}{\partial\xi_s}\Bigg\}
\end{aligned}
$$

with $\gamma_1^L = \gamma_2^L = \gamma_3^L = \operatorname{meas}\Delta$, $\gamma_4^L = \gamma_5^L = \gamma_7^L = \gamma_8^L = \gamma_{10}^L = \gamma_{11}^L = \operatorname{meas}\Delta^{(k)}$, and $\gamma_6^L = \gamma_9^L = \gamma_{12}^L = 2\operatorname{meas}\Delta^{(k)}$.

For the entries of the matrix K_l^Q we get by using relations (29)–(31) and the quadrature rules (32)–(34)

$$
\begin{aligned}
K_{l,vv}^{Q,(ij)} &= \sum_{r\in\omega_{l-1}^{(ij)}} \Bigg\{ \\
&\frac{4}{3}\sum_{t=4}^{6}\gamma_t^Q b_{11}^{(r)}(x(\xi^{(t)}))\frac{\partial\psi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_1}\frac{\partial\psi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_1} - \frac{1}{3}\gamma_1^Q b_{11}^{(r)}(x(\xi^{(1)}))\frac{\partial\psi_{\beta^{(r)}}(\xi^{(1)})}{\partial\xi_1}\frac{\partial\psi_{\alpha^{(r)}}(\xi^{(1)})}{\partial\xi_1} \\
&+ \frac{4}{3}\sum_{t=7}^{9}\gamma_t^Q b_{22}^{(r)}(x(\xi^{(t)}))\frac{\partial\psi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_2}\frac{\partial\psi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_2} - \frac{1}{3}\gamma_2^Q b_{22}^{(r)}(x(\xi^{(2)}))\frac{\partial\psi_{\beta^{(r)}}(\xi^{(2)})}{\partial\xi_2}\frac{\partial\psi_{\alpha^{(r)}}(\xi^{(2)})}{\partial\xi_2} \\
&+ \frac{4}{3}\sum_{t=10}^{12}\gamma_t^Q b_{12}^{(r)}(x(\xi^{(t)}))\frac{\partial\psi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_s}\frac{\partial\psi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_s} - \frac{1}{3}\gamma_3^Q b_{12}^{(r)}(x(\xi^{(3)}))\frac{\partial\psi_{\beta^{(r)}}(\xi^{(3)})}{\partial\xi_s}\frac{\partial\psi_{\alpha^{(r)}}(\xi^{(3)})}{\partial\xi_s}\Bigg\}
\end{aligned}
$$

with $\gamma_1^Q = \gamma_2^Q = \gamma_3^Q = \operatorname{meas}\Delta$, $\gamma_4^Q = \gamma_5^Q = \gamma_7^Q = \gamma_8^Q = \gamma_{10}^Q = \gamma_{11}^Q = 0.25\operatorname{meas}\Delta$ and $\gamma_6^Q = \gamma_9^Q = \gamma_{12}^Q = 0.5\operatorname{meas}\Delta$.

If we examine the values of the partial derivatives $\partial\varphi_{\alpha^{(r)}}/\partial\xi_1$, $\partial\varphi_{\alpha^{(r)}}/\partial\xi_2$, $\partial\varphi_{\alpha^{(r)}}/\partial\xi_s$, $\partial\psi_{\alpha^{(r)}}/\partial\xi_1$, $\partial\psi_{\alpha^{(r)}}/\partial\xi_2$, $\partial\psi_{\alpha^{(r)}}/\partial\xi_s$, $\alpha^{(r)} = 1,2,3$, given in Table 1 and the relations $\operatorname{meas}\Delta = 4\operatorname{meas}\Delta^{(k)}$, $k = 1,2,3,4$, $\varphi_{\alpha^{(r)}} = \psi_{\alpha^{(r)}}$, $\alpha^{(r)} = 1,2,3$, we see that

$$K_{l,vv}^{L,ex,(ij)} = K_{l,vv}^{Q,(ij)} \quad \text{for } i,j = 1,2,\ldots,N_{l-1}\,, \quad \text{i.e. } K_{l,vv}^{L,ex} = K_{l,vv}^{Q}\,.$$

In the same way we can prove

$$K_{l,vm}^{L,ex} = K_{l,vm}^{Q}\,, \quad K_{l,mv}^{L,ex} = K_{l,mv}^{Q} \quad \text{and} \quad K_{l,mm}^{L,ex} = K_{l,mm}^{Q}\,.$$

The symbol "–" in Table 1 means that the partial derivative does not exist in this quadrature point. But we do not need these values for the computation of the matrix elements of $K_l^{L,ex}$. ■

TABLE 1

Values of the partial derivatives in the quadrature points

	$\xi^{(1)}$	$\xi^{(4)}$	$\xi^{(5)}$	$\xi^{(6)}$		$\xi^{(2)}$	$\xi^{(7)}$	$\xi^{(8)}$	$\xi^{(9)}$		$\xi^{(3)}$	$\xi^{(10)}$	$\xi^{(11)}$	$\xi^{(12)}$
$\frac{\partial\varphi_1}{\partial\xi_1}$	-1	-1	-1	-1	$\frac{\partial\varphi_1}{\partial\xi_2}$	-1	-1	-1	-1	$\frac{\partial\varphi_1}{\partial\xi_s}$	0	0	0	0
$\frac{\partial\varphi_2}{\partial\xi_1}$	1	1	1	1	$\frac{\partial\varphi_2}{\partial\xi_2}$	0	0	0	0	$\frac{\partial\varphi_2}{\partial\xi_s}$	-1	-1	-1	-1
$\frac{\partial\varphi_3}{\partial\xi_1}$	0	0	0	0	$\frac{\partial\varphi_3}{\partial\xi_2}$	1	1	1	1	$\frac{\partial\varphi_3}{\partial\xi_s}$	1	1	1	1
$\frac{\partial\varphi_4}{\partial\xi_1}$	–	2	-2	0	$\frac{\partial\varphi_4}{\partial\xi_2}$	–	0	0	-2	$\frac{\partial\varphi_4}{\partial\xi_s}$	–	0	0	-2
$\frac{\partial\varphi_5}{\partial\xi_1}$	–	0	0	2	$\frac{\partial\varphi_5}{\partial\xi_2}$	–	0	0	2	$\frac{\partial\varphi_5}{\partial\xi_s}$	–	2	-2	0
$\frac{\partial\varphi_6}{\partial\xi_1}$	–	0	0	-2	$\frac{\partial\varphi_6}{\partial\xi_2}$	–	2	-2	0	$\frac{\partial\varphi_6}{\partial\xi_s}$	–	0	0	2
$\frac{\partial\psi_4}{\partial\xi_1}$	0	2	-2	0	$\frac{\partial\psi_4}{\partial\xi_2}$	0	0	0	-2	$\frac{\partial\psi_4}{\partial\xi_s}$	0	0	0	-2
$\frac{\partial\psi_5}{\partial\xi_1}$	0	0	0	2	$\frac{\partial\psi_5}{\partial\xi_2}$	0	0	0	2	$\frac{\partial\psi_5}{\partial\xi_s}$	0	2	-2	0
$\frac{\partial\psi_6}{\partial\xi_1}$	0	0	0	-2	$\frac{\partial\psi_6}{\partial\xi_2}$	0	2	-2	0	$\frac{\partial\psi_6}{\partial\xi_s}$	0	0	0	2

Next we discuss the computation of the entries of the load vectors. For the entries of the vector $\underline{f}_{l-1}^L$ we get

$$f_{l-1}^{L,(i)} = \langle F, p_{l-1}^{(i)} \rangle = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\delta_{l-1}^{(r)}} f(x) p_{l-1}^{(i)}(x)\, dx = \sum_{r\in\omega_{l-1}^{(i)}} \int_\Delta f(x(\xi))\varphi_{\alpha^{(r)}}(\xi) |\det J_{l-1}^{(r)}|\, d\xi \tag{38}$$

with $\omega_{l-1}^{(i)} = \{r \,:\, p_{l-1}^{(i)} \not\equiv 0 \text{ on } \delta_{l-1}^{(r)}\}$.

The integrals over Δ we compute by using the quadrature rule

$$\int_\Delta v(\xi)\, d\xi = \left(\frac{1}{3}v(0,0) + \frac{1}{3}v(1,0) + \frac{1}{3}v(0,1)\right) \operatorname{meas}\Delta\,. \tag{39}$$

Obviously, this formula is exact for linear functions v.

The entries of the vector $\underline{f}_l^L$ are defined by

$$f_l^{L,(i)} = \langle F, \tilde{p}_l^{(i)} \rangle = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\delta_{l-1}^{(r)}} f(x) \tilde{p}_l^{(i)}(x)\, dx = \sum_{r\in\omega_{l-1}^{(i)}} \int_\Delta f(x(\xi))\varphi_{\alpha^{(r)}}(\xi) |\det J_{l-1}^{(r)}|\, d\xi$$

$$= \sum_{r\in\omega_{l-1}^{(i)}} \sum_{k=1}^{4} \int_{\Delta^{(k)}} f(x(\xi))\varphi_{\alpha^{(r)}}(\xi) |\det J_{l-1}^{(r)}|\, d\xi \tag{40}$$

with $\tilde{p}_l^{(i)} = p_{l-1}^{(i)}$ for $i = 1, \dots, N_{l-1}$, $\tilde{p}_l^{(i)} = p_l^{(i)}$ for $i = N_{l-1}+1, \dots, N_l$ and the functions $\varphi_{\alpha^{(r)}}$ from (28). The integrals over $\Delta^{(k)}$ are computed by a formula of the type (39).

In the case of the p-hierarchical basis, the entries of the load vector $\underline{f}_l^Q$ are given by

$$f_l^{Q,(i)} = \langle F, \tilde{p}_l^{(i)} \rangle = \sum_{r \in \omega_{l-1}^{(i)}} \int_{\delta_{l-1}^{(r)}} f(x) \tilde{p}_l^{(i)}(x)\, dx = \sum_{r \in \omega_{l-1}^{(i)}} \int_{\Delta} f(x(\xi)) \psi_{\alpha^{(r)}}(\xi) |\det J_{l-1}^{(r)}|\, d\xi \tag{41}$$

with $\tilde{p}_l^{(i)} = p_{l-1}^{(i)}$ for $i = 1, \dots, N_{l-1}$, $\tilde{p}_l^{(i)} = q_{l-1}^{(i)}$ for $i = N_{l-1}+1, \dots, N_l$, and the functions $\psi_{\alpha^{(r)}}$ from (31). For the computation of the integrals over Δ we use the quadrature rule

$$\int_{\Delta} v(\xi)\, d\xi = \left(\frac{1}{3} v(\xi^{(1)}) + \frac{1}{3} v(\xi^{(2)}) + \frac{1}{3} v(\xi^{(3)}) \right) \operatorname{meas} \Delta\,. \tag{42}$$

with $\xi^{(\sigma)}$, $\sigma = 1, 2, 3$, from (26). This formula is exact for quadratic functions v.

Lemma 3.2. *If the load vectors $\underline{f}_{l-1}^L$, $\underline{f}_l^L$, and $\underline{f}_l^Q$ are defined as described above, then the relation*

$$\underline{f}_l^{L,ex} = \underline{f}_l^Q \tag{43}$$

holds.

Proof: Using the relations (38), (40) and quadrature rules of type (39) for computing the extrapolated load vector $\underline{f}_l^{L,ex}$ (see (16)) as well as relations (41) and (42) for computing the load vector $\underline{f}_l^Q$, the proof follows immediately. ∎

A consequence of Lemma 3.1 and Lemma 3.2 is the following Theorem.

Theorem 3.3. *If the extrapolated stiffness matrix $K_l^{L,ex}$ and the extrapolated load vector $\underline{f}_l^{L,ex}$ as well as the stiffness matrix K_l^Q and the load vector $\underline{f}_l^Q$ are computed as described above, then the systems of algebraic FE equations*

$$K_l^{L,ex} \underline{u}_l = \underline{f}_l^{L,ex} \qquad \text{and} \qquad K_l^Q \underline{u}_l = \underline{f}_l^Q \tag{44}$$

have the same solution.

Now we can immediately prove the following convergence theorem for the algorithm MG–EX.

Theorem 3.4. *Under the assumption that the extrapolated stiffness matrix $K_l^{L,ex}$, the extrapolated load vector $\underline{f}_l^{L,ex}$, the stiffness matrix K_l^Q, and the load vector $\underline{f}_l^Q$ are computed as discussed in this section, the following statements hold.*

(i) The iterates of algorithm MG–EX converge to a FE solution which has the same discretization error as a FE solution obtained by a FE discretization with p-hierarchical functions.

(ii) *The convergence rate of algorithm MG–EX does not depend on the discretization parameter.*

Proof: The statement (i) follows from the interpretation of algorithm MG–EX as a usual multigrid algorithm for solving the system of algebraic equation $K_l^{L,ex}\underline{u}_l = \underline{f}_l^{L,ex}$ and the equivalence of the systems of algebraic equations $K_l^{L,ex}\underline{u}_l = \underline{f}_l^{L,ex}$ and $K_l^Q\underline{u}_l = \underline{f}_l^Q$.

Statement (ii) we can prove in an analogous way as done for the piecewise constant coefficient case in [6]. ■

Remark: We can also formulate algorithm MG–EX in terms of a piecewise linear nodal basis. All our results are also valid in this case.

4. Numerical results. In this Section we want to confirm our theoretical results by a numerical example. We will illustrate that the iterates of the algorithm MG–EX converge to the FE solution which we would obtain by a discretization of problem (1) with p-hierarchical functions. Furthermore, the numerical example shows that the convergence rate of algorithm MG–EX is independent of the discretization parameter.

All algorithms have been implemented within the multigrid package FEMGP [12]. The computations were performed on a PC 80486 (33 MHz) using the LAHEY-Fortran compiler.

Let us consider the problem (1), where $\Omega = (0,1)\times(0,1)$ and

$$A(x) = \begin{pmatrix} 2 - x_1 + \frac{x_2}{2} & \sin x_1 \sin x_2 \\ \sin x_1 \sin x_2 & e^{x_1+x_2} \end{pmatrix}.$$

The right-hand side $f(x)$ is chosen such that the function

$$u(x) = x_1(1-x_1)\sin x_2 (e^{4x_1^2+x_2^2} - e^{x_1^2+3x_2^2})$$

is the exact solution of problem (1). Figure 3 shows plots of this exact solution.

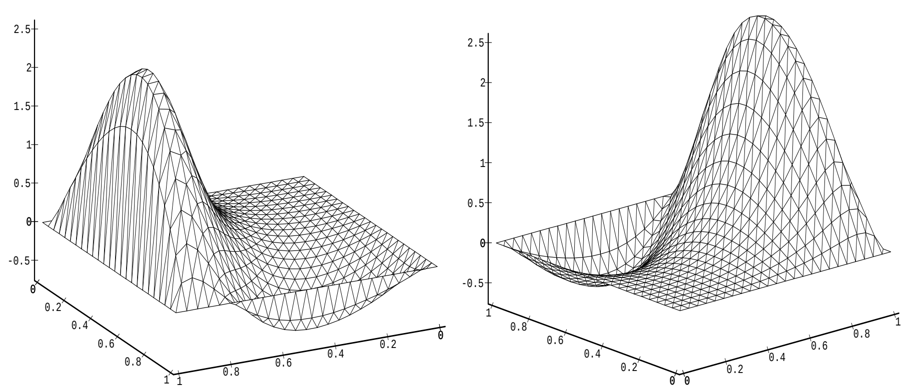

FIG. 3. *Plots of the exact solution* $u(x)$

Starting from the coarsest triangulation $\mathcal{T}_1$ (see Figure 4) the finer triangulations have been generated by dividing all triangles of the triangulation $\mathcal{T}_k$, $k = 1, 2, \dots, l-1$, into four smaller congruent sub-triangles. In Table 2 we give the numbers of nodes and the numbers of triangles in each triangulation.

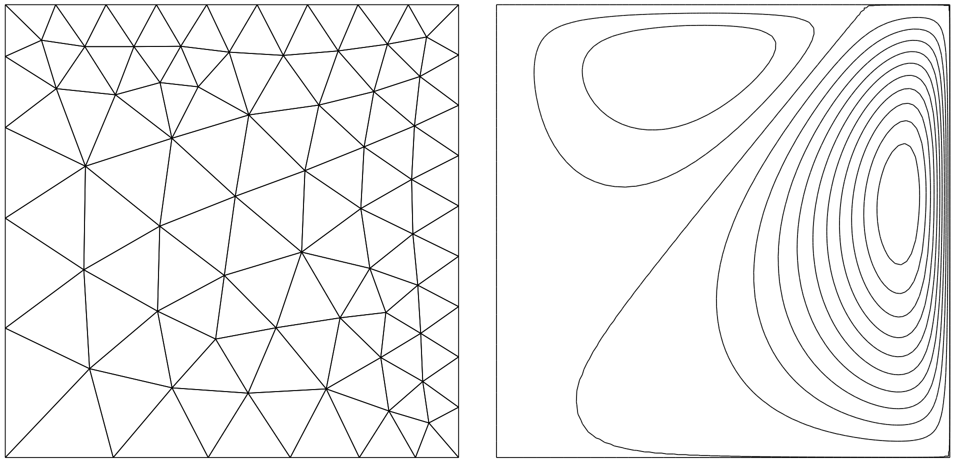

FIG. 4. *Mesh $\mathcal{T}_1$ and iso-lines of the solution u*

TABLE 2
Number of nodes and number of triangles in $\mathcal{T}_k$, $k = 1, 2, \dots, 5$

triangulation	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
number of nodes	75	268	1011	3925	15465
number of triangles	119	476	1904	7616	30464

For Algorithm MG–EX we used as pre-smoother two sweeps of the lexicographically forward Gauss-Seidel method for solving system (8), one iteration step of a $(l-1)$–grid algorithm for solving the coarse-grid system (11), and two sweeps of the lexicographically backward Gauss-Seidel method in the post-smoothing step. The initial guess was obtained by a full multigrid strategy. On the levels $k = 1, 2, \dots, l-1$ a usual multigrid algorithm for solving the corresponding FE equations in the linear nodal basis was performed. Within this k–grid algorithm one V-cycle with two Gauss-Seidel sweeps lexicographically forward in the pre-smoothing step and two Gauss-Seidel sweeps lexicographically backward in the post-smoothing step were used. The convergence criterion for MG–EX was

$$\|\underline{f}_l^{L,ex} - K_l^{L,ex}\underline{u}_l^{(k+1,0)}\| \leq 10^{-4}\,\|\underline{f}_l^{L,ex} - K_l^{L,ex}\underline{u}_l^{(0,0)}\|\,, \tag{45}$$

where $\|\,.\,\|$ denotes the Euclidean norm in the space R^{N_l}, and $\underline{u}_l^{(0,0)}$ is the initial guess.

In Table 3 we present the number of iterations and the CPU-time needed by the application of the algorithm MG–EX. We obtain an improvement of the convergence behavior of our algorithm by introducing additional pre-smoothing and post-smoothing steps, i.e. before step 1 in the algorithm MG–EX we perform one iteration step of the Gauss-Seidel method lexicographically forward and after step 3 one iteration step of the Gauss-Seidel method lexicographically backward applied to the system of algebraic equations $K_l^{L,ex}\underline{u}_l^{L,ex} = \underline{f}_l^{L,ex}$. This is illustrated in column MG–EX(1) of Table 3.

Table 3
Comparison of the algorithm MG–EX and the algorithm MG–EX(1)

l	Algorithm MG–EX		Algorithm MG–EX(1)	
	number of iterations	CPU-time	number of iterations	CPU-time
3	13	2.42 sec	6	1.60 sec
4	13	11.31 sec	6	7.58 sec
5	14	52.46 sec	6	33.49 sec

Finally, we compare the discretization errors $\|u - u_l^{L,ex}\|$ and $\|u - u_l^Q\|$ in the H^1-norm and in the L_2-norm. Here $u_l^{L,ex}$ denotes the FE solution obtained by algorithm MG–EX (interpreted as a piecewise quadratic function) and u_l^Q the FE solution obtained directly by a discretization with piecewise p-hierarchical functions.

Table 4
Comparison of the discretization errors

Level l	$\|u - u_l^{L,ex}\|_{H^1}$	$\|u - u_l^{L,ex}\|_{L_2}$	$\|u - u_l^Q\|_{H^1}$	$\|u - u_l^Q\|_{L_2}$
3	0.23541-00	0.17906-02	0.23541-00	0.17906-02
4	0.59806-01	0.22421-03	0.59806-01	0.22421-03
5	0.15007-01	0.26197-04	0.15007-01	0.26198-04

Table 4 shows that the algorithm MG–EX yields discretization errors which are typical for discretizations with piecewise quadratic functions, i.e. we can observe an error of order $O(h_l^2)$ in the H^1-norm and $O(h_l^3)$ in the L_2-norm.

5. Conclusion. In this paper we have presented the analysis of an algorithm which can be understood algebraically as multigrid with τ-extrapolation. In practice, this algorithm is simple to implement, once a multigrid algorithm is available. However, we have shown that the algorithm converges to the same solution as a higher order finite element discretization. The algorithm can thus be used on unstructured meshes in an adaptive refinement setting. Furthermore, it is independent of global error expansions, and can thus be applied locally.

REFERENCES

[1] K. Bernert, *Tau-extrapolation — theoretical foundation, numerical experiment and application to the Navier-Stokes equations*, 1995, accepted for publication in SISC, (German version available as *Tau-extrapolation – theoretische Grundlagen, numerische Experimente und Anwendung auf die Navier-Stokes-Gleichungen*, Preprint SPC 94_7, Fakultät für Mathematik, Technische Universität Chemnitz-Zwickau, 1994).

[2] H. Blum, Q. Lin, and R. Rannacher, *Asymptotic Error Expansions and Richardson Extrapolation for Linear Finite Elements*, Numer. Math., 49 (1986), pp. 11-37.

[3] A. Brandt, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, GMD Studien, 85 (1984).

[4] P. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam, 1978.

[5] W. Hackbusch, *Multigrid Methods and Applications*, Springer Series Comput. Math., Springer Verlag, Berlin, 1985.

[6] M. Jung and U. Rüde, *Implicit extrapolation methods for multilevel finite element computations*, to be published in SISC 17-1, 1996, (extended version available as Preprint SPC 94_11, Fakultät für Mathematik, Technische Universität Chemnitz-Zwickau, 1994).

[7] J.N. Lyness and U. Rüde, *Two-dimensional numerical integration of functions containing derivatives*, (in preparation).

[8] G. Marchuk and V. Shaidurov, *Difference Methods and their Extrapolations*, Springer, New York, 1983.

[9] S. McCormick and U. Rüde, *On Local Refinement Higher Order Methods for Elliptic Partial Differential Equations*, International Journal of High Speed Computing, 2 (1990), pp. 311-334. Also available as TU-Bericht I-9034.

[10] U. Rüde, *Extrapolation techniques for constructing higher order finite element methods*, Bericht I-9304, Institut für Informatik, TU München, 1993.

[11] S. Schaffer, *Higher Order Multigrid Methods*, Math. Comp., 43 (1984), pp. 89-115.

[12] T. Steidten and M. Jung, *Das Multigrid-Programmsystem FEMGPM zur Lösung elliptischer und parabolischer Differentialgleichungen einschließlich mechanisch-thermisch gekoppelter Probleme (Version 06.90)*, Programmdokumentation, Sektion Mathematik, Technische Universität Karl-Marx-Stadt (Chemnitz-Zwickau), 1990.

SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

342/18/90 A	Christoph Zenger: SPARSE GRIDS
342/19/90 A	Michael Griebel, Michael Schneider, Christoph Zenger: A combination technique for the solution of sparse grid problems
342/20/90 A	Michael Griebel: A Parallelizable and Vectorizable Multi- Level-Algorithm on Sparse Grids
342/21/90 A	V. Diekert, E. Ochmanski, K. Reinhardt: On confluent semi-commutations-decidability and complexity results
342/22/90 A	Manfred Broy, Claus Dendorfer: Functional Modelling of Operating System Structures by Timed Higher Order Stream Processing Functions
342/23/90 A	Rob van Glabbeek, Ursula Goltz: A Deadlock-sensitive Congruence for Action Refinement
342/24/90 A	Manfred Broy: On the Design and Verification of a Simple Distributed Spanning Tree Algorithm
342/25/90 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Peter Luksch, Roland Wismüller: TOPSYS - Tools for Parallel Systems (User's Overview and User's Manuals)
342/26/90 A	Thomas Bemmerl, Arndt Bode, Thomas Ludwig, Stefan Tritscher: MMK - Multiprocessor Multitasking Kernel (User's Guide and User's Reference Manual)
342/27/90 A	Wolfgang Ertel: Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems
342/28/90 A	Rob van Glabbeek, Frits Vaandrager: Modular Specification of Process Algebras
342/29/90 A	Rob van Glabbeek, Peter Weijland: Branching Time and Abstraction in Bisimulation Semantics
342/30/90 A	Michael Griebel: Parallel Multigrid Methods on Sparse Grids
342/31/90 A	Rolf Niedermeier, Peter Rossmanith: Unambiguous Simulations of Auxiliary Pushdown Automata and Circuits
342/32/90 A	Inga Niepel, Peter Rossmanith: Uniform Circuits and Exclusive Read PRAMs
342/33/90 A	Dr. Hermann Hellwagner: A Survey of Virtually Shared Memory Schemes
342/1/91 A	Walter Vogler: Is Partial Order Semantics Necessary for Action Refinement?
342/2/91 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Rainer Weber: Characterizing the Behaviour of Reactive Systems by Trace Sets
342/3/91 A	Ulrich Furbach, Christian Suttner, Bertram Fronhöfer: Massively Parallel Inference Systems

342/4/91 A	Rudolf Bayer: Non-deterministic Computing, Transactions and Recursive Atomicity
342/5/91 A	Robert Gold: Dataflow semantics for Petri nets
342/6/91 A	A. Heise; C. Dimitrovici: Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften
342/7/91 A	Walter Vogler: Asynchronous Communication of Petri Nets and the Refinement of Transitions
342/8/91 A	Walter Vogler: Generalized OM-Bisimulation
342/9/91 A	Christoph Zenger, Klaus Hallatschek: Fouriertransformation auf dünnen Gittern mit hierarchischen Basen
342/10/91 A	Erwin Loibl, Hans Obermaier, Markus Pawlowski: Towards Parallelism in a Relational Database System
342/11/91 A	Michael Werner: Implementierung von Algorithmen zur Kompaktifizierung von Programmen für VLIW-Architekturen
342/12/91 A	Reiner Müller: Implementierung von Algorithmen zur Optimierung von Schleifen mit Hilfe von Software-Pipelining Techniken
342/13/91 A	Sally Baker, Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Udo Graf, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Rainer Knödlseder, Jaroslav Kremenek, Siegfried Langenbuch, Robert Lindhof, Thomas Ludwig, Peter Luksch, Roy Milner, Bernhard Ries, Thomas Treml: TOPSYS - Tools for Parallel Systems (Artikelsammlung); 2., erweiterte Auflage
342/14/91 A	Michael Griebel: The combination technique for the sparse grid solution of PDE's on multiprocessor machines
342/15/91 A	Thomas F. Gritzner, Manfred Broy: A Link Between Process Algebras and Abstract Relation Algebras?
342/16/91 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Thomas Treml, Roland Wismüller: The Design and Implementation of TOPSYS
342/17/91 A	Ulrich Furbach: Answers for disjunctive logic programs
342/18/91 A	Ulrich Furbach: Splitting as a source of parallelism in disjunctive logic programs
342/19/91 A	Gerhard W. Zumbusch: Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme
342/20/91 A	M. Jobmann, J. Schumann: Modelling and Performance Analysis of a Parallel Theorem Prover
342/21/91 A	Hans-Joachim Bungartz: An Adaptive Poisson Solver Using Hierarchical Bases and Sparse Grids
342/22/91 A	Wolfgang Ertel, Theodor Gemenis, Johann M. Ph. Schumann, Christian B. Suttner, Rainer Weber, Zongyan Qiu: Formalisms and Languages for Specifying Parallel Inference Systems

342/23/91 A	Astrid Kiehn: Local and Global Causes
342/24/91 A	Johann M.Ph. Schumann: Parallelization of Inference Systems by using an Abstract Machine
342/25/91 A	Eike Jessen: Speedup Analysis by Hierarchical Load Decomposition
342/26/91 A	Thomas F. Gritzner: A Simple Toy Example of a Distributed System: On the Design of a Connecting Switch
342/27/91 A	Thomas Schnekenburger, Andreas Weininger, Michael Friedrich: Introduction to the Parallel and Distributed Programming Language ParMod-C
342/28/91 A	Claus Dendorfer: Funktionale Modellierung eines Postsystems
342/29/91 A	Michael Griebel: Multilevel algorithms considered as iterative methods on indefinite systems
342/30/91 A	W. Reisig: Parallel Composition of Liveness
342/31/91 A	Thomas Bemmerl, Christian Kasperbauer, Martin Mairandres, Bernhard Ries: Programming Tools for Distributed Multiprocessor Computing Environments
342/32/91 A	Frank Leßke: On constructive specifications of abstract data types using temporal logic
342/1/92 A	L. Kanal, C.B. Suttner (Editors): Informal Proceedings of the Workshop on Parallel Processing for AI
342/2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS
342/2-2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS - Revised Version (erschienen im Januar 1993)
342/3/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/4/92 A	Claus Dendorfer, Rainer Weber: Development and Implementation of a Communication Protocol - An Exercise in FOCUS
342/5/92 A	Michael Friedrich: Sprachmittel und Werkzeuge zur Unterstüt- zung paralleler und verteilter Programmierung
342/6/92 A	Thomas F. Gritzner: The Action Graph Model as a Link between Abstract Relation Algebras and Process-Algebraic Specifications
342/7/92 A	Sergei Gorlatch: Parallel Program Development for a Recursive Numerical Algorithm: a Case Study
342/8/92 A	Henning Spruth, Georg Sigl, Frank Johannes: Parallel Algorithms for Slicing Based Final Placement

342/9/92 A	Herbert Bauer, Christian Sporrer, Thomas Krodel: On Distributed Logic Simulation Using Time Warp
342/10/92 A	H. Bungartz, M. Griebel, U. Rüde: Extrapolation, Combination and Sparse Grid Techniques for Elliptic Boundary Value Problems
342/11/92 A	M. Griebel, W. Huber, U. Rüde, T. Störtkuhl: The Combination Technique for Parallel Sparse-Grid-Preconditioning and -Solution of PDEs on Multiprocessor Machines and Workstation Networks
342/12/92 A	Rolf Niedermeier, Peter Rossmanith: Optimal Parallel Algorithms for Computing Recursively Defined Functions
342/13/92 A	Rainer Weber: Eine Methodik für die formale Anforderungsspezifkation verteilter Systeme
342/14/92 A	Michael Griebel: Grid– and point–oriented multilevel algorithms
342/15/92 A	M. Griebel, C. Zenger, S. Zimmer: Improved multilevel algorithms for full and sparse grid problems
342/16/92 A	J. Desel, D. Gomm, E. Kindler, B. Paech, R. Walter: Bausteine eines kompositionalen Beweiskalküls für netzmodellierte Systeme
342/17/92 A	Frank Dederichs: Transformation verteilter Systeme: Von applikativen zu prozeduralen Darstellungen
342/18/92 A	Andreas Listl, Markus Pawlowski: Parallel Cache Management of a RDBMS
342/19/92 A	Erwin Loibl, Markus Pawlowski, Christian Roth: PART: A Parallel Relational Toolbox as Basis for the Optimization and Interpretation of Parallel Queries
342/20/92 A	Jörg Desel, Wolfgang Reisig: The Synthesis Problem of Petri Nets
342/21/92 A	Robert Balder, Christoph Zenger: The d-dimensional Helmholtz equation on sparse Grids
342/22/92 A	Ilko Michler: Neuronale Netzwerk-Paradigmen zum Erlernen von Heuristiken
342/23/92 A	Wolfgang Reisig: Elements of a Temporal Logic. Coping with Concurrency
342/24/92 A	T. Störtkuhl, Chr. Zenger, S. Zimmer: An asymptotic solution for the singularity at the angular point of the lid driven cavity
342/25/92 A	Ekkart Kindler: Invariants, Compositionality and Substitution
342/26/92 A	Thomas Bonk, Ulrich Rüde: Performance Analysis and Optimization of Numerically Intensive Programs
342/1/93 A	M. Griebel, V. Thurner: The Efficient Solution of Fluid Dynamics Problems by the Combination Technique
342/2/93 A	Ketil Stølen, Frank Dederichs, Rainer Weber: Assumption / Commitment Rules for Networks of Asynchronously Communicating Agents

342/3/93 A	Thomas Schnekenburger: A Definition of Efficiency of Parallel Programs in Multi-Tasking Environments
342/4/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: A Proof of Convergence for the Combination Technique for the Laplace Equation Using Tools of Symbolic Computation
342/5/93 A	Manfred Kunde, Rolf Niedermeier, Peter Rossmanith: Faster Sorting and Routing on Grids with Diagonals
342/6/93 A	Michael Griebel, Peter Oswald: Remarks on the Abstract Theory of Additive and Multiplicative Schwarz Algorithms
342/7/93 A	Christian Sporrer, Herbert Bauer: Corolla Partitioning for Distributed Logic Simulation of VLSI Circuits
342/8/93 A	Herbert Bauer, Christian Sporrer: Reducing Rollback Overhead in Time-Warp Based Distributed Simulation with Optimized Incremental State Saving
342/9/93 A	Peter Slavkovsky: The Visibility Problem for Single-Valued Surface (z = f(x,y)): The Analysis and the Parallelization of Algorithms
342/10/93 A	Ulrich Rüde: Multilevel, Extrapolation, and Sparse Grid Methods
342/11/93 A	Hans Regler, Ulrich Rüde: Layout Optimization with Algebraic Multigrid Methods
342/12/93 A	Dieter Barnard, Angelika Mader: Model Checking for the Modal Mu-Calculus using Gauß Elimination
342/13/93 A	Christoph Pflaum, Ulrich Rüde: Gauß' Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids
342/14/93 A	Christoph Pflaum: Convergence of the Combination Technique for the Finite Element Solution of Poisson's Equation
342/15/93 A	Michael Luby, Wolfgang Ertel: Optimal Parallelization of Las Vegas Algorithms
342/16/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: Pointwise Convergence of the Combination Technique for Laplace's Equation
342/17/93 A	Georg Stellner, Matthias Schumann, Stefan Lamberts, Thomas Ludwig, Arndt Bode, Martin Kiehl und Rainer Mehlhorn: Developing Multicomputer Applications on Networks of Workstations Using NXLib
342/18/93 A	Max Fuchs, Ketil Stølen: Development of a Distributed Min/Max Component
342/19/93 A	Johann K. Obermaier: Recovery and Transaction Management in Write-optimized Database Systems

342/20/93 A Sergej Gorlatch: Deriving Efficient Parallel Programs by Systemating Coarsing Specification Parallelism

342/01/94 A Reiner Hüttl, Michael Schneider: Parallel Adaptive Numerical Simulation

342/02/94 A Henning Spruth, Frank Johannes: Parallel Routing of VLSI Circuits Based on Net Independency

342/03/94 A Henning Spruth, Frank Johannes, Kurt Antreich: PHIroute: A Parallel Hierarchical Sea-of-Gates Router

342/04/94 A Martin Kiehl, Rainer Mehlhorn, Matthias Schumann: Parallel Multiple Shooting for Optimal Control Problems Under NX/2

342/05/94 A Christian Suttner, Christoph Goller, Peter Krauss, Klaus-Jörn Lange, Ludwig Thomas, Thomas Schnekenburger: Heuristic Optimization of Parallel Computations

342/06/94 A Andreas Listl: Using Subpages for Cache Coherency Control in Parallel Database Systems

342/07/94 A Manfred Broy, Ketil Stølen: Specification and Refinement of Finite Dataflow Networks - a Relational Approach

342/08/94 A Katharina Spies: Funktionale Spezifikation eines Kommunikationsprotokolls

342/09/94 A Peter A. Krauss: Applying a New Search Space Partitioning Method to Parallel Test Generation for Sequential Circuits

342/10/94 A Manfred Broy: A Functional Rephrasing of the Assumption/Commitment Specification Style

342/11/94 A Eckhardt Holz, Ketil Stølen: An Attempt to Embed a Restricted Version of SDL as a Target Language in Focus

342/12/94 A Christoph Pflaum: A Multi-Level-Algorithm for the Finite-Element-Solution of General Second Order Elliptic Differential Equations on Adaptive Sparse Grids

342/13/94 A Manfred Broy, Max Fuchs, Thomas F. Gritzner, Bernhard Schätz, Katharina Spies, Ketil Stølen: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems

342/14/94 A Maximilian Fuchs: Technologieabhängigkeit von Spezifikationen digitaler Hardware

342/15/94 A M. Griebel, P. Oswald: Tensor Product Type Subspace Splittings And Multilevel Iterative Methods For Anisotropic Problems

342/16/94 A Gheorghe Ştefănescu: Algebra of Flownomials

342/17/94 A Ketil Stølen: A Refinement Relation Supporting the Transition from Unbounded to Bounded Communication Buffers

342/18/94 A Michael Griebel, Tilman Neuhoeffer: A Domain-Oriented Multilevel Algorithm-Implementation and Parallelization

342/19/94 A	Michael Griebel, Walter Huber: Turbulence Simulation on Sparse Grids Using the Combination Method
342/20/94 A	Johann Schumann: Using the Theorem Prover SETHEO for verifying the development of a Communication Protocol in FOCUS - A Case Study -
342/01/95 A	Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A	Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law
342/03/95 A	Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space
342/04/95 A	Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes
342/05/95 A	Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems
342/06/95 A	Maximilian Fuchs: Formal Design of a Model-N Counter
342/07/95 A	Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology
342/08/95 A	Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies
342/09/95 A	Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness
342/10/95 A	Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design
342/11/95 A	Thomas Schnekenburger: The ALDY Load Distribution System
342/12/95 A	Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm
342/13/95 A	Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming
342/14/95 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks
342/15/95 A	Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs
342/16/95 A	Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik
342/17/95 A	Georg Stellner: Using CoCheck on a Network of Workstations
342/18/95 A	Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications

SFB 342 : Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B	Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B	Jörg Desel: On Abstraction of Nets
342/3/90 B	Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B	Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B	Barbara Paech1: Concurrency as a Modality
342/2/91 B	Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier- Toolbox -Anwenderbeschreibung
342/3/91 B	Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B	Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B	Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B	Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B	W. Reisig: Concurrent Temporal Logic
342/1/92 B	Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B	Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B	Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung
342/2/93 B	Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B	Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS