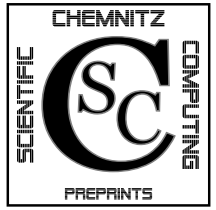
TECHNISCHE UNIVERSITÄT CHEMNITZ

Arnd Meyer

Hierarchical Preconditioners and Adaptivity for Kirchhoff–Plates

CSC/08-03



Chemnitz Scientific Computing Preprints

Chemnitz Scientific Computing Preprints – ISSN 1864-0087

Impressum:

Chemnitz Scientific Computing Preprints — ISSN 1864-0087
(1995–2005: Preprintreihe des Chemnitzer SFB393)

Herausgeber:
Professuren für
Numerische und Angewandte Mathematik
an der Fakultät für Mathematik
der Technischen Universität Chemnitz

Postanschrift:
TU Chemnitz, Fakultät für Mathematik
09107 Chemnitz
Sitz:
Reichenhainer Str. 41, 09126 Chemnitz

http://www.tu-chemnitz.de/mathematik/csc/

Some titles of the former CSC preprint series:

06-01 T. Eibner, J. M. Melenk. p-FEM quadrature error analysis on tetrahedra. October 2006.

06-02 P. Benner, H. Faßbender. On the solution of the rational matrix equation $X = Q + LX^{-1}L^{\top}$. September 2006.

06-03 P. Benner, H. Mena, J. Saak. On the Parameter Selection Problem in the Newton-ADI Iteration for Large Scale Riccati Equations. October 2006.

06-04 J. M. Badía, P. Benner, R. Mayo, E. S. Quintana-Ortí, G. Quintana-Ortí, A. Remón. Balanced Truncation Model Reduction of Large and Sparse Generalized Linear Systems. November 2006.

07-01 U. Baur, P. Benner. Gramian-Based Model Reduction for Data-Sparse Systems. February 2007.

07-02 A. Meyer. Grundgleichungen und adaptive Finite-Elemente-Simulation bei Großen Deformationen. Februar 2007.

07-03 P. Steinhorst. Rotationssymmetrie für piezoelektrische Probleme. Februar 2007.

07-04 S. Beuchler, T. Eibner, U. Langer. Primal and Dual Interface Concentrated Iterative Substructuring Methods. April 2007.

07-05 T. Hein, M. Meyer. Simultane Identifikation voneinander unabhängiger Materialparameter - numerische Studien. Juni 2007.

07-06 A. Bucher, U.-J. Görke, P. Steinhorst, R. Kreißig, A. Meyer. Ein Beitrag zur adaptiven gemischten Finite-Elemente-Formulierung der nahezu inkompressiblen Elastizität bei großen Verzerrungen. September 2007.

07-07 U.-J. Görke, A. Bucher, R. Kreißig Zur Numerik der inversen Aufgabe für gemischte (u/p) Formulierungen am Beispiel der nahezu inkompressiblen Elastizität bei großen Verzerrungen. October 2007.

07-08 A. Meyer, P. Steinhorst. Betrachtungen zur Spektraläquivalenz für das Schurkomplement im Bramble-Pasciak-CG bei piezoelektrischen Problemen. Oktober 2007.

07-09 T. Hein, M. Meyer. Identification of material parameters in linear elasticity - some numerical results. November 2007.

07-10 T. Hein. On solving implicitly defined inverse problems by SQP-approaches. December 2007.

The complete list of SFB393 preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.

TECHNISCHE UNIVERSITÄT CHEMNITZ

Chemnitz Scientific Computing Preprints

Arnd Meyer

Hierarchical Preconditioners and Adaptivity for Kirchhoff–Plates

CSC/08-03

CSC/08-03 ISSN 1864-0087 **Sept. 2008**

Contents

Author's address:

Arnd Meyer
TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

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

# elem.	# d.o.f.'s.	#it. "cccc"	#it. "cccs"	#it. "cscs"	#it. "ccss"
16	100	8	12	12	27
64	324	18	26	24	29
256	1 156	25	27	26	26
1024	4 356	22	23	24	22
4096	16 900	18	19	21	18
16 T	66 564	16	16	16	16
65 T	264 196	14	15	15	14

Table 1: Number of PCGM iterations on 4 examples

combination with the cascadic approach we obtain a small and near fixed number of PCG–iterations required for some given accuracy.

References

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

[2] H.Matthes, *Die nichtüberlappende Gebietszerlegungsmethode zur Parallelisierung und Vorkonditionierung iterativer Verfahren zur Lösung von Platten– und Schalenproblemen*, Dissertation, TU Chemnitz (1996).

[3] P.Oswald, *Multilevel Finite Element Approximation: Theory and Applications*, Teubner Skripten zur Numerik, B.G.Teubner Stuttgart 1994.

[4] M.Thess, *Parallel Multilevel Preconditioners for Thin Smooth Shell Finite Element Analysis*, Num.Lin.Alg.with Appl. **5** (1999) 5 ,401-440(1999)

[5] M.Thess, *Parallel Multilevel Preconditioners for Thin Smooth Shell Problems*, Dissertation, TU Chemnitz (1998).

[6] J. Xu, *Iterative Methods by Space Decomposition and Subspace Correction*, SIAM Rev. **34** (1992) 4, 163-184.

[7] H. Yserentant, *Two Preconditioners Based on the Multilevel Splitting of Finite Element Spaces*, Numer. Math. **58** (1990) 163-184.

need a scaling diagonal D in the definition of the preconditioner $C^{-1} = QDQ^T$ in the original paper of Yserentant.

Here, this is no longer true, because we consider a 4th order p.d.e. and the 4 different degrees of freedom per node are of different kind (the value and derivatives, which scale differently w.r.t. h).The best way out is the definition of D in the MDS (multiple diagonal scaling [6]) style, which means that D is the diagonal of the actual hierarchical stiffness matrix K_H.

This information can be derived and stored during the mesh refinement steps: We mark a node as "used" if its diagonal entry of a previous stiffness matrix has been stored in D. Then after mesh refinement, we calculate the diagonal $\tilde{D}$ of the actual stiffness matrix and copy all entries of $\tilde{D}$ belonging to new nodes (not "used") into D. With this simple procedure we have always the correct scaling factor in D.

4 Numerical Example

We demonstrate the power of this preconditioner at some simple rectangular plate examples. We consider $\Omega = (0, 1.5) \times (0, 1)$ with different boundary conditions and constant force everywhere. Each of the 4 boundary edges are either full clamped ("c", means $w = 0$ and $\nabla w = \overrightarrow{0}$) or only softly supported ("s", means $w = 0$ and $\tau \cdot \nabla w = 0$ with the tangential direction τ).

The example "cccs" has soft support on one of the longer edges, "ccss" at two adjacent edges and "cscs" at two opposite (longer) edges. The coarse mesh consists of one rectangle only, we start the calculation after 2 refinements with the given number of elements in the Table 1. The PCGM iteration starts on each new mesh with interpolated values from the next coarser mesh. The relative decrease of the test value $(\underline{r}, \underline{w})$ under 10^{-8} has used as stopping criterion. The following Table 1 contains the number of iterations required until this stopping criterion is fulfilled for these four examples.

5 Conclusion

The most simple hierarchical technique for preconditioning the Kirchhoff–plate equation leads to a very efficient solver. It is necessary to consider the correct refinement formula for the Bogner–Fox–Schmidt element functions as well as the correct scaling diagonal as the main ingreedients for this preconditioner. In

1 Introduction

Our aim is the development of efficient numerical tools for calculating deformations of thin structures as plates or shells considering finite (large) strain equations. For the beginning we start with quick solving the finite element equations of the well-known linear Kirchhoff-plate equation for the small strain case. Here, the main challenge is the fourth order biharmonic equation, that requires both special finite elements (with $C^{(1)}$-functions) and a special preconditioner for the resulting matrix equation.

The paper is organized as follows. We start with a short sketch of deriving the Kirchhoff plate equation and its conformal finite element discretization in Chapter 2. In Chapter 3 we describe the hierarchical preconditioner for the PCGM solver of the discretized problem.

2 The Kirchhoff–Plate Equation

We consider the deformation of a 3D-domain $\Omega^{3D} = \Omega \times [-\frac{d}{2}, \frac{d}{2}]$ with thickness d in z-direction and $(x, y) \in \Omega \subset \mathbb{R}^2$ a plane. Here, $d << \mathrm{diam}\Omega$, so the well-known Kirchhoff hypothesis about an approximation to the 3D-displacement vector reads as:

$$\boldsymbol{U}(x, y, z) = \begin{pmatrix} -z\ w_x \\ -z\ w_y \\ w \end{pmatrix} \tag{1}$$

with the unknown scalar function $w(x, y) \in \mathbb{H}^2(\Omega)$. The general 3D-solution of a linear elastic deformation of Ω^{3D} reads as

$$a(\boldsymbol{U}, \boldsymbol{V}) = f(\boldsymbol{V}) \quad \forall \boldsymbol{V} \in (\mathbb{H}^1_D(\Omega^{3D}))^3 \tag{2}$$

with

$$a(\boldsymbol{U}, \boldsymbol{V}) = \int\limits_{\Omega^{3D}} \sigma(\boldsymbol{U}) : \epsilon(\boldsymbol{V})\ d\Omega dz \tag{3}$$

$$f(\boldsymbol{V}) = \int\limits_{\Omega^{3D}} \vec{\boldsymbol{f}} \cdot \boldsymbol{V}\ d\Omega dz + \int\limits_{\Gamma_N^{3D}} \vec{\boldsymbol{g}} \cdot \boldsymbol{V}\ d\mathcal{S} \tag{4}$$

for given volume forces $\vec{\boldsymbol{f}}$ and boundary tractions $\vec{\boldsymbol{g}}$ on parts Γ_N^{3D} of $\partial\Omega^{3D}$. In linear elasticity we define

$$\epsilon(\boldsymbol{U}) = \frac{1}{2}\left(Grad\boldsymbol{U} + Grad\boldsymbol{U}^T\right) \tag{5}$$

and

$$\sigma(\boldsymbol{U}) = 2\mu\,\epsilon(\boldsymbol{U}) + \lambda(tr\ \epsilon)I \tag{6}$$

with the Lamé constants μ and λ. Using Cartesian coordinates this means

$$\epsilon_{ij}(\boldsymbol{U}) = \frac{1}{2}(\partial_i \boldsymbol{U}_j + \partial_j \boldsymbol{U}_i) \tag{7}$$

so

$$a_{3D}(\boldsymbol{U},\boldsymbol{V}) = \int\limits_{\Omega_{3D}} 2\mu\ \epsilon_{ij}(\boldsymbol{U})\cdot\epsilon_{ij}(\boldsymbol{V}) + \lambda\ \mathrm{div}\boldsymbol{U}\ \mathrm{div}\boldsymbol{V}\ d\Omega dz \tag{8}$$

The Sobolev space $\mathbb{H}^1_D(\Omega^{3D})$ contains functions $u \in \mathbb{H}^1(\Omega^{3D})$ with zero values at the Dirichlet boundary Γ^{3D}_D. We assume that

$$\Gamma^{3D}_D = \partial\Omega \times \left[-\frac{d}{2}, \frac{d}{2}\right]$$

and

$$\Gamma^{3D}_N = \Omega \times \left\{\pm\frac{d}{2}\right\} \quad \text{(hence: } d\mathcal{S} = d\Omega\text{)}.$$

Under the restriction $\vec{\boldsymbol{f}} = f(x,y)\boldsymbol{e}_3$ and $\vec{\boldsymbol{g}} = g(x,y)\boldsymbol{e}_3$ with scalar functions f and g we can project the problem (2) into the subspace due to (1) and integration over $z \in \left[-\frac{d}{2}, \frac{d}{2}\right]$ leads to the well-known Kirchhoff plate equation

$$a(w,v) = f(v) \quad \forall v \in \mathbb{H}^2_0(\Omega) \tag{9}$$

with

$$a(w,v) = \frac{d^3}{12}\int\limits_{\Omega} 2\mu(\nabla\nabla^T w) : (\nabla\nabla^T v) + \lambda(\Delta w)(\Delta v)\ d\Omega,$$

$$f(v) = d\int\limits_{\Omega} f\,v\ d\Omega + \int\limits_{\Omega} g\,v\ d\Omega.$$

For the finite element equation it is convenient to define

$$\underline{\epsilon}(w) = \begin{pmatrix} w_{xx} \\ w_{yy} \\ 2w_{xy} \end{pmatrix} = D(\nabla)\,w \tag{10}$$

with the differential operator

$$D(\nabla) = (\frac{\partial^2}{\partial x^2},\ \frac{\partial^2}{\partial y^2},\ 2\frac{\partial^2}{\partial x \partial y})^T$$

and the material matrix

$$\underline{C} = \begin{pmatrix} 2\mu+\lambda & \lambda & 0 \\ \lambda & 2\mu+\lambda & 0 \\ 0 & 0 & \mu \end{pmatrix},$$

Now, we can derive the 2D-refinement formula from the Kronecker-product definition of Φ^{coars}_k:

$$\begin{aligned} \Phi^{coars}_k &= Q^C_k \otimes\ P^C_k \\ &= (Q^F_k + F^F_1 \cdot A(-h_y) + F^F_2 \cdot A(+h_y)) \\ &\quad \otimes\ (P^F_k + F^F_3 \cdot A(-h_x) + F^F_4 \cdot A(+h_x)) \\ &= Q^F_k \otimes P^F_k\ + (Q^F_k \otimes F^F_3)(I\ \otimes A(-h_x)) + ... \end{aligned} \tag{20}$$

which is exactly the sum (17) with the matrices A_j as

$$I \otimes A(\pm h_x),\ A(\pm h_y) \otimes I,\ \text{or}\ \ A(\pm h_y) \otimes A(\pm h_x)$$

depending on the place of the 8 nodes j relative to the center node k. The implementation is very simple, if the data structure "Edge", as the tree of all edges (from coarse to fine) is defined with 3 informations on each edge: "left" node number (l), "right" node number (r), number of mid-node (m). From the definitions above the following algorithm for the multiply $\underline{w} := Q\underline{y}$ arises. Note that

$$\underline{w} = (\underline{w}^T_1,\ ...\ , \underline{w}^T_n)^T$$

with $\underline{w}_i \in \mathbb{R}^4$ for n nodes of the actual mesh (and $N = 4n$).

Algorithm: Multiply $\underline{w} := Q\underline{y}$

1. $\underline{w} := \underline{y}$
2. for each edge with nodes (l,r,m) from coarse to finest do

$$\left[\begin{array}{l} h_x = (x\text{-coord. of node } r) - (x\text{-coord. of node } l) \\ h_y = (y\text{-coord. of node } r) - (y\text{-coord. of node } l) \\ \\ \text{if}\quad (|h_y| > |h_x|) \quad \text{then} \quad B^+ := A(+h_y) \otimes I \\ \qquad\qquad\qquad\qquad\qquad\quad\ B^- := A(-h_y) \otimes I \\ \\ \qquad\qquad\qquad\qquad\ \ \text{else} \quad B^+ := I \otimes A(+h_x) \\ \qquad\qquad\qquad\qquad\qquad\quad\ B^- := I \otimes A(-h_x) \\ \\ \underline{w}_m := \underline{w}_m + B^+\underline{w}_r + B^-\underline{w}_l \end{array}\right. \tag{21}$$

Note that h_x and h_y have a sign depending on the position of the endpoints of this edge and one of them is zero due to the axis-parallel mesh.

3.2 The Scaling Diagonal

In the case of 2D calculations on 2nd order p.d.e.'s, the size of the elements of the stiffness matrix is fixed independent of the mesh size. From this reason we do not

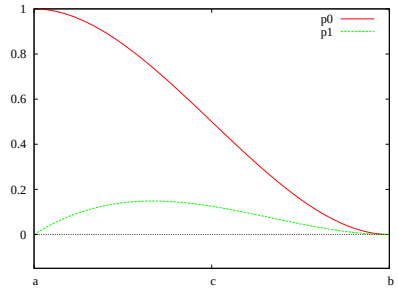


Figure 1: $P_a^C(x)$

with support in $[a,c]$ and a new fine function $F(x) = (f^0(x),\ f^1(x))$ with support in $[a,b]$ and

$$F(c) = (1,0), \qquad F(a) = F(b) = (0,0)$$
$$\partial_x F(c) = (0,1), \quad \partial_x F(a) = \partial_x F(b) = (0,0)$$

acording to Fig.2 and 3.

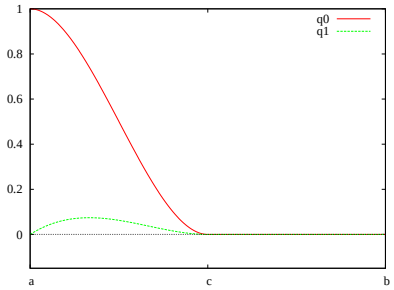


Figure 2: $P_a^F(x)$

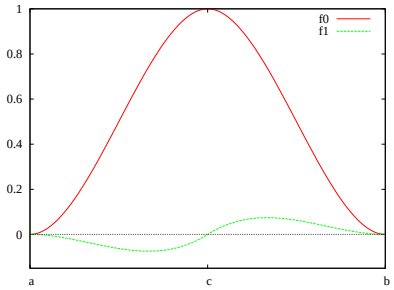


Figure 3: $F(x)$

Then we easily calculate

$$P_a^C(x) = P_a^F(x) + F(x) \cdot A(+h)$$

for

$$A(h) = \begin{pmatrix} 1/2 & h/8 \\ -3/(2h) & -1/4 \end{pmatrix} \quad \text{and} \quad h = b - a. \tag{18}$$

If the coarse mesh function P_b^C belongs to the right point b (instead of a) the same formula with $-h$ is correct:

$$P_b^C(x) = P_b^F(x) + F(x) \cdot A(-h). \tag{19}$$

such that

$$a(w,v) = \frac{d^3}{12} \int_\Omega (Dv)^T \underline{C} (Dw)\ d\Omega \tag{11}$$

From this abbreviation the structure of a finite element stiffness matrix is simply deduced. Let $(\varphi_1, ..., \varphi_n)$ be the basis of the n finite element ansatz functions, then

$$\widehat{D} = (D\varphi_1 \vdots \ .... \ \vdots D\varphi_n)$$

defines the stiffness matrix

$$K = \frac{d^3}{12} \int_\Omega \widehat{D}^T \underline{C} \widehat{D}\ d\Omega.$$

Analogously we can use $(\varphi_1, ..., \varphi_n)$ as the n form functions of one element (with a much smaller n), then

$$K_{el} = \frac{d^3}{12} \int_{el} \widehat{D}^T \underline{C} \widehat{D}\ d\Omega \tag{12}$$

is the element stiffness matrix.

The appropriate finite element discretization of the equation (11) requires $\mathbb{H}^2$-functions for the global ansatz functions φ_i. This can be achieved with the so–called Bogner-Fox-Schmidt elements. Here, we assume Ω consisting of a union of axis-parallel rectangles, then the 16 cubic form functions $\varphi_{kl}^{ab}(x,y)$ $(a,b,k,l \in \{0,1\})$ can fulfill these requirements, if they are defined with the help of the 1D–functions:

$$\begin{aligned} \hat{p}_0^0(s) &= \tfrac{1}{4}(1-s)^2(2+s) & \hat{p}_1^0(s) &= \hat{p}_0^0(-s) \\ \hat{p}_0^1(s) &= \tfrac{1}{4}(1-s)^2(1+s) & \hat{p}_1^1(s) &= -\hat{p}_0^1(-s) \\ & & &\text{for } s \in [-1,1] \end{aligned} \tag{13}$$

After transformation of $[-1,1]$ onto the x-interval of the element we have $P = (p_0^0(x),\ p_1^0(x),\ p_0^1(x),\ p_1^1(x))$ and after transformation of $[-1,1]$ onto the y-interval we have $Q = (q_0^0(y),\ q_1^0(y),\ q_0^1(y),\ q_1^1(y))$. Then

$$\varphi_{kl}^{ab}(x,y) = p_k^a(x)\ q_l^b(y) \tag{14}$$

or the basis $\Phi = (\varphi_{00}^{00}\ , \ldots,\ \varphi_{11}^{11})$ can be written as the Kronecker–product $Q(y) \otimes P(x)$ of the one dimensional functions. This will be used later for deriving the refinement formula of the preconditioner. The calculation of the element matrix is easily done from (12) using

$$\widehat{D} = \begin{pmatrix} Q & \otimes & \partial_{xx}\ P \\ \partial_{yy}\ Q & \otimes & P \\ \partial_y\ Q & \otimes & \partial_x\ P \end{pmatrix} \tag{15}$$

At each node of the finite element mesh, we have 4 ansatz functions with support of the 4 adjacent elements. The dimension of the linear system is four times the number of nodes. The entries of the solution vector can be partitioned into 4-dimensional sub-vectors belonging to each node

$$w_{\text{node}} = (w^{00}_{\text{node}},\ w^{10}_{\text{node}},\ w^{01}_{\text{node}},\ w^{11}_{\text{node}}),$$

which have the following meaning (due to the properties of the ansatz functions)

$$w^{ab}_{\text{node}} = \frac{\partial^a}{\partial x^a}\,\frac{\partial^b}{\partial y^b}\ w|_{\text{node}} \tag{16}$$

3 Quick Solvers for the Stiffness Matrix

For efficient solving the discretized finite element linear equations, we use the preconditioned conjugate gradient method (PCGM). Then the number of arithmetic operations is proportional to the number of unknowns for each step of the iteration. It remains to construct a preconditioner that guarantees a small number of iterations (until a given accuracy) that does not grow after mesh refinement. In [2, 4, 5] these preconditioners are derived as Additive Schwarz Method similar to the BPX-preconditioner[1]. Here, we consider an easier approach based on the hierarchical basis. Even with a very slow growth of the number of iterations (as $\log N$), this would be an efficient solver from two reasons.

First, the implementation is very cheap and second we will embed it into the adaptive mesh refinement, where the so-called cascadic effect dampens the required iteration numbers additionally.

The idea behind is the definition of the hierarchical basis of the finite element space $\mathbb{V}_N$ of the ansatz functions. If $\Phi = (\varphi_1,\ \ldots\ ,\varphi_N)$ is the original (nodal) basis in $\mathbb{V}_N$ and $\Psi = (\psi_1,\ \ldots\ ,\psi_N)$ the hierarchical one, then it exists an $(N \times N)$-matrix Q that transforms one into another,

$$\Psi = \Phi\ Q.$$

From the fact that the stiffness matrix K_H belonging to Ψ is much better conditioned than our usual stiffness matrix K belonging to Φ, we arrive at a preconditioner.

If $\lambda_i(K_H)$ denote the eigenvalues of K_H, then

$$\lambda_i(K_H) = \lambda_i(Q^T K Q) = \lambda_i(Q\,Q^T K),$$

so C with

$$C^{-1} = Q\,Q^T$$

is a "good" preconditioner for K. This was first discovered by Yserentant for the Laplace equation on linear triangular finite elements [7]. The multiply with the (theoretically dense) matrices Q and Q^T is done in $\mathcal{O}(N)$ operations only from the hierarchical definition of the mesh. Analogously to this 2nd order equation, we can derive a hierarchical refinement formula for our cubic functions as well, which is used for the preconditioning action.

3.1 The Multiply with Q

Let

$$\Phi^{coars}_k = (\varphi^{00}_k,\ \varphi^{10}_k,\ \varphi^{01}_k,\ \varphi^{11}_k)$$

be the 4 ansatz functions belonging to the mesh point k of a fixed mesh with mesh spacings $h^+_x,\ h^-_x$ in x-direction and $h^+_y,\ h^-_y$ in y-direction around the k-th node. After mesh refinement as subdivision of all elements around this node, we have

$$\Phi^{Fine}_k = (f^{00}_k,\ f^{10}_k,\ f^{01}_k,\ f^{11}_k)$$

the new (fine) functions around k-th node with half support in both directions and 8 new quadruples of functions belonging to the 8 new nodes around this node. These are $\Phi^{Fine}_j = (f^{00}_j,\ \ldots f^{11}_j)$ with 8 new indices $j \in \mathcal{N}_k$ (the set of all new neighbors of k). The refinement formula

$$\Phi^{coars}_k = \Phi^{Fine}_k + \sum_{j \in N_k} \Phi^{Fine}_j A_j \tag{17}$$

requires the (4x4)-matrices A_j, which can be deduced from the 1-dimensional case, following [4].

We need to consider the refinement formula of the 2 functions

$$P^C_a(x) = (p^0(x),\ p^1(x)),$$

defined on an interval $[a, b]$ belonging to the left end point a. (So:

$$\begin{array}{ll} P^C_a(a) = (1,0), & P^C_a(b) = (0,0), \\ \partial_x P^C_a(a) = (0,1), & \partial_x P^C_a(b) = (0,0), \end{array}$$

see Fig. 1)

The subdivision of this interval is done by introducing $c = \frac{a+b}{2}$ as new grid point. Now we have the fine grid function

$$P^F_a(x) = (q^0(x),\ q^1(x))$$