Bull. Nov. Comp. Center, Num. Anal., 13 (2005), 57–65

On a massively parallel finite element method

Joseph M. Maubach*

Abstract. This paper discusses the key-properties from the Parallel Finite Element Method (PFEM) introduced and examined in [4, 5, 11–15]. It focuses at the PFEM applications in the context of non-conforming finite element basis functions (for maximal parallelism, see [14]) on locally-bisection-refined tensor-product grids (for simple and cheap load balance techniques, see [18, 19, 21]).

The Parallel Finite Element Method is an iterative solution method based on a Red-Black domain decomposition. The method is robust, and can solve elliptic as well as mixed elliptic-hyperbolic and hyperbolic problems (see [14]). The amount of iterations is optimal for a method with only local communication. Nonlinear as well as constraint systems of equations can be solved element-wise in parallel.

1. Introduction

The Parallel Finite Element Method (PFEM) is a finite element discrimination method which — with a proper choice of finite basis functions — can solve linear as well as constrained problems massively in parallel, as is shown in [13, 14].

The latter papers are quite theoretical and therefore somewhat hide the fact that the actual iterative solution algorithm is of a simple nature. In a lucid manner, this paper discusses the PFEM key properties and iterative solution method, as well as some nonlinear constrained applications. For details it refers to published papers. By means of illustration, application of the PFEM to a convection-diffusion problem is discussed.

The remainder of this paper is organized as follows. First, we present the required mathematical context for the PFEM: An example of a standard convection-diffusion problem is introduced in Section 2, the related domain decomposition being presented in Section 3, the induced operator splitting is addressed in 4. This splitting is exploited in the iterative algorithm in Section 6. The rate of convergence results (analyzed in [14]) are summarized in Section 7, and more complex (nonlinear, constraint) applications are mentioned in Section 8.

2. The convection-diffusion problem

First, we formulate the standard unit-square convection-diffusion problem:

$$-\varepsilon\Delta u + \boldsymbol{b}^T\nabla u + cu = f \quad \text{in } \Omega = (0,1)^2, \qquad u|_{\partial\Omega} = 0. \tag{1}$$

*Supported by NWO-Russia Cooperation Grant 047.016.008.

We assume that the domain is first partitioned into a tensor-triangle-grid (such as in Figure 1 on the next page). Afterwards it is refined where needed with the local (n-dimensional) bisection refinement presented in [17–19]. This refinement method is simple and cheap — for the two-dimensional case, see, also [6, 10, 22, 24, 25].

The related standard stream-line upwind Galerkin finite element formulation (SUPG) is: Find a solution u such that $b(u,v) = f(v)$ for all v in a suitable space, where

$$b(u,v) = \int \varepsilon\nabla u\nabla v + \delta\boldsymbol{b}\nabla u\boldsymbol{b}\nabla v + cu(v+\delta\boldsymbol{b}\nabla v) + v\boldsymbol{b}\nabla u + \text{H.O.T.} \quad (2)$$

Here H.O.T. stands for Higher Order Terms which stem from the partial integration of the term $\int(\Delta u)(v+\delta\boldsymbol{b}\nabla v)$. The PFEM induced parallelism (see Section 5) is maximal if one uses non-conforming finite element basis functions: In such a case there are no basis functions which have support points at the vertices of the triangles, which decreases the amount of couplings in the matrix (and, hence, increases the amount of parallelism).

The bilinear operator b in (2) is not coercive for non-conforming bases, whence the PFEM uses the standard skew-symmetrized SUPG formulation: Find a solution u such that $a(u,v) = f(v)$ for all v, where

$$a(u,v) = \int \varepsilon\nabla u\nabla v + \delta\boldsymbol{b}\nabla u\boldsymbol{b}\nabla v + \Big(c - \frac{1}{2}\nabla\cdot(\boldsymbol{b}+\delta c\boldsymbol{b})\Big)uv + (1-\delta c)(v\boldsymbol{b}\nabla u - u\boldsymbol{b}\nabla v) + \text{H.O.T.} \quad (3)$$

In the sequel, for the sake of simplicity, we will use the first order non-conforming linear basis functions on triangles (as in [1]), whence all H.O.T. reduce to zero. For higher order basis functions, see, for instance [9]. The use of formulation (3) introduces an extra (small) error called consistency error, see [1].

3. The domain decomposition

Next, the domain Ω is covered with the black and the red colored elements, for instance, as in Figure 1. The research has shown that the use of more than two colors does not being about a faster iterative solution algorithm (for comments on the color strategy see [4, 20]). The notation is as follows: B and R are the sets of all respectively black and red elements, and subdomains are defined: $\Omega_R = \bigcup_{e\in R} e$, etc.

4. The coefficient matrix splits...

The basic idea for the domain decomposition is that integration is additive, the integral over the entire domain Ω is the sum of the integrals over the

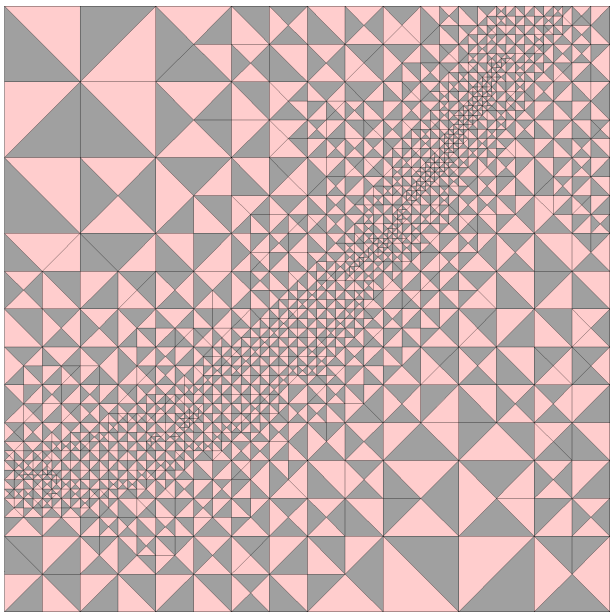

Figure 1. Domain decomposition: $\Omega = \overline{\Omega_R \cup \Omega_B}$, $\Omega_R \cap \Omega_B = \emptyset$

red Ω_R and the black Ω_B domains. Now, the bilinear operator $a(\cdot,\cdot)$ is split to a red and a black part:

$$a_\omega(u,v) = \int_\omega \varepsilon \nabla u \nabla v + \delta \boldsymbol{b} \nabla u \boldsymbol{b} \nabla v + (c - \frac{1}{2}\nabla \cdot (\boldsymbol{b} + \delta c \boldsymbol{b}))uv + (1 - \delta c)(v\boldsymbol{b}\nabla u - u\boldsymbol{b}\nabla v) + \text{H.O.T.}$$

and $a(u,v) := a_\Omega(u,v)$. Based on this, we can split the canonical matrix representation of the bilinear operator $a(\cdot,\cdot)$:

$$A = A_\Omega = A_{\Omega_R} + A_{\Omega_B} =: A_R + A_B.$$

The PFEM method differs from the ADI (alternative direction splitting) where the bilinear operator $e(u,v) = \int \nabla u \nabla v$ is split to direction-dependent parts

$$e_x(u,v) = \int u_x v_x, \quad e_y(u,v) = \int u_y v_y$$

with a related canonical matrix splitting $E = E_x + E_y$.

For more information on the ADI methods see [2, 3, 7, 8].

5. ... To block-diagonal matrices

As shown in [14], a proper (color-respecting) numbering of the degrees of freedom leads to the matrices A_R and A_B *which are block-diagonal under*

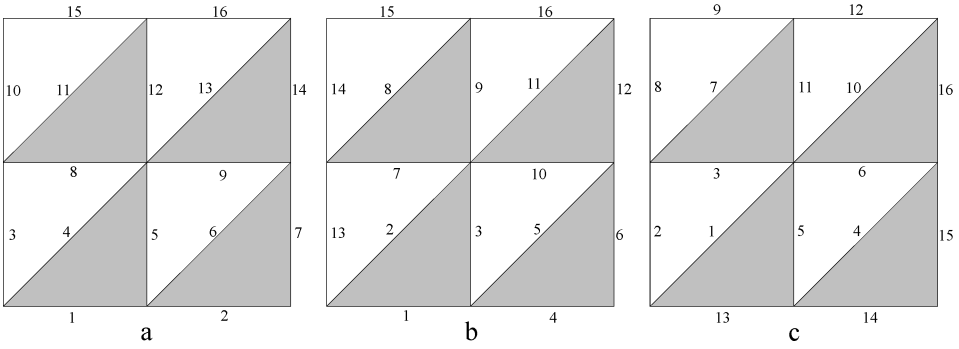

Figure 2. Numbering order: a) left-right/bottom-top, b) black variables first, c) red variables first

permutation. For our Crouziex–Raviart basis this is shown for a simple triangulation in Figure 2.

First, the matrix A related to the numbering of the degrees of freedom in Figure 2a is not a diagonal matrix after permutation, as is indicated by:

$$A = \begin{pmatrix} 1 & & & -1 & & & & & & & \dots \\ & 1 & & & & -1 & & & & & \dots \\ & & 1 & -1 & & & & & & & \dots \\ -1 & & -1 & 4 & -1 & & & -1 & & & \dots \\ 0 & & & -1 & 2 & -1 & & & 0 & & \dots \\ & -1 & & & -1 & 4 & -1 & & -1 & & \dots \\ & 0 & & & & -1 & 1 & & & & \dots \\ & & 0 & -1 & & & & 2 & & & \dots \\ & & & & 0 & -1 & & & 2 & & \dots \\ & & & & & & & & & 1 & \dots \\ & & & & & & & -1 & & -1 & \dots \\ & & & & & & & 0 & & & \dots \\ & & & & & & & & -1 & & \dots \\ & & & & & & & & 0 & & \dots \\ & & & & & & & & & 0 & \dots \\ & & & & & & & & & & \dots \end{pmatrix} \tag{4}$$

However, different numberings of the degrees of freedom such as in Figure 2-(Middle) or 2-(Right) with related permutations

$$\begin{aligned} \pi_1 &= (1\ 4\ 13\ 2\ 3\ 5\ 6\ 7\ 10\ 14\ 8\ 9\ 11\ 12\ 15\ 16) \\ \pi_2 &= (13\ 14\ 2\ 1\ 5\ 4\ 15\ 3\ 6\ 8\ 7\ 11\ 10\ 16\ 9\ 12) \end{aligned}$$

result in block-diagonal matrices, as shown in Figure 3.

Sometimes it is possible that the neighboring elements (elements which share a facet) have the same color (see for instance Figure 1 and [5]). In such

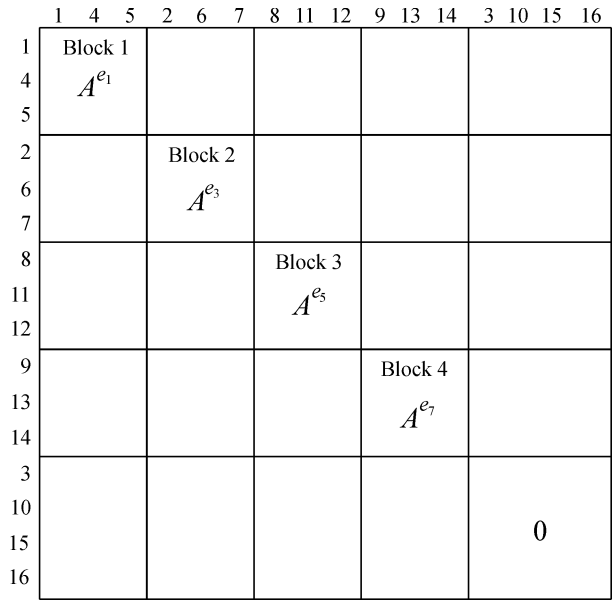

Figure 3. The block diagonal structure of $\pi_2^* A_B \pi_2$

a case, one obtains a diagonal block related to the cluster of identically colored elements. For instance, for the case of a cluster of two identically colored neighbors, the total of four non-conforming linear basis functions forms a cluster.

6. The iterative solution methods

The iterative solution methods for the convection-diffusion *linear case* are based on Peaceman–Rachford [23] and D'yakunov [3], for the *nonlinear case* (a minimum surface equation, see [14]) are based on Kellogg [7], and for the *constraint nonlinear case* are based on Lions and Mercier [16] LM-1 and LM-2. This paper presents the linear convection-diffusion PFEM and, hence, the algorithm [23]:

Let $\boldsymbol{u}^{(0)} \in \mathbb{R}^N$ and $\rho \in (0, \infty)$. Iterate

$$(\rho I_N + A_R)\boldsymbol{v} = (\rho I_N - A_B)\boldsymbol{u}^{(k)} + \boldsymbol{b};$$
$$(\rho I_N + A_B)\boldsymbol{u}^{(k+1)} = (\rho I_N - A_R)\boldsymbol{v} + \boldsymbol{b}$$

until $\|A\boldsymbol{u}^{(k)} - \boldsymbol{b}\|_2 < \epsilon$. Here $\boldsymbol{b}$ is the entire right-hand side vector (integrated over the entire domain Ω). (Algorithm [3] employs the colored right-hand sides $\boldsymbol{b}_B$ and $\boldsymbol{b}_R$.)

The advantages of the algorithm are:

- The algorithm converges for all $\rho \in (0, \infty)$, the optimum value estimated to be $\rho = O(h)$;
- The iterations *involve no inner-product*, i.e., *no global communication*;
- The matrices $\rho I_N + A_R$, etc. are *block-diagonal under permutation*, whence inversion stands for inversion of three by three matrices.

There is no need to calculate the inner product of each iteration to check whether convergence is reached, in practice, we just check each 10 iterations. This means that massive amounts of data entirely can be processed in parallel, without need for any global intervention.

The main difference in the nonlinear and the constraint nonlinear algorithms [16] is that there, after operations $(\rho I_N + A_R)^{-1}$ and $(\rho I_N + A_B)^{-1}$, one needs an extra projection onto a constrained space.

7. The rate of convergence

For our convection–diffusion problem, the rate of convergence of the Peaceman–Rachford and LM-1 and LM-2 does not depend on the amount of diffusion, *hyperbolic problems are solved just as efficient as elliptic problems.* More precisely, let $h > 0$ be a minimum over all grid element diameters. Assume the operator is maximally monotone (the bilinear operator $a(\cdot, \cdot)$ is maximally monotone [13]), and let the finite element space be split appropriately [14]. Then, as is proven in [14]:

- The amount of iterations of both Peaceman–Rachford and LM-1 and LM-2 is optimal for local communication: $O(h^{-1})$ for sufficiently h;
- This amount of iterations does not depend on

$$0 \leq \varepsilon_{\min} \leq \varepsilon \leq \varepsilon_{\max} \leq 1,$$

  nor on $\boldsymbol{b}$ and c;
- The optimal value for ρ is $O(h)$.

8. Nonlinear applications

The PFEM can also solve (nonlinear) variational inequalities, as well as constrained (linear) systems of equations:

- Nonlinear operator problems (see [13]):

$$-\Delta u + e^{au} + cu = \boldsymbol{f}; \tag{5}$$

- The obstacle problem and other constrained problems (see, also, [13]):

$$\begin{aligned} &-\Delta u - f \geq 0, \quad u - g \geq 0 \quad \text{in } \Omega; \\ &(-\Delta u - f)(u - g) = 0 \quad \text{in } \Omega; \\ &u = 0 \quad \text{at } \partial\Omega; \end{aligned} \tag{6}$$

- Fluid Flow (Stokes) and Solid Mechanics (Signorini's).

In all cases, the operator must be of the maximally monotone type and the constraints must be pointwise.

The constraints can be dealt with in parallel as well: First the constraints over Ω reduce to constraints over the colored domains Ω_R and Ω_B, and next, the constraints over, for instance, Ω_R decouple into constraints over (clusters of, or individual) red elements. Hence, projections can be elements wise (or a cluster of identically colored elements) in parallel.

9. Conclusions

For problem (5), the PFEM is *faster even in serial mode* than the Walker and Eisenstat global-convergent nonlinear damped Newton solvers. Next, constrained problems are solved much *more accurately than with a penalty method* (see [13]) because there are *no penalty terms*. Furthermore, *local constraints can be computed explicitly* (per element or cluster) — hence very efficiently. The PFEM method *scales better in parallel than the famous Jacobi method* (D.O. Neil, Pittsburgh Super Computer Center). Last but not least, the amount of iterations for the convection–diffusion problem is *optimal* for a method with local communications only ($O(h^{-1})$), both for elliptic and mixed elliptic-hyperbolic problems.

References

[1] Crouzeix M., Raviart P.A. Conforming and nonconforming finite element methods for solving the stationary Stokes equations I // RAIRO. — 1973. — Vol. 3. — P. 33–76.

[2] Douglas J., Rachford H.H. On the numerical solution of the heat conduction problem in 2 and 3 space variables // Transactions of the AMS. — 1956. — Vol. 82. — P. 421–439.

[3] D'yakunov E.G. On a method of solving the Poisson equation // Soviet Math. Doklady. — 1962. — Vol. 3. — P. 320–323.

[4] Ervin V., Layton W., Maubach J. Some graph coloring problems // Algoritmen in de Algebra. Seminar on Algebraic Algorithms, University of Nijmegen, the Netherlands / Ed. A.H.M. Levelt. — 1993. — P. 39–48.

[5] Jeurissen R., Layton W. Load balancing via graph coloring: an algorithm // Computers and Mathematics with Applications. — 1993.

[6] Joe B., Liu A. On the shape of tetrahedra from bisection // Math. Comp. — 1944. — Vol. 63. — P. 141–154.

[7] Kellogg R.B. An alternating direction method for operator equations // J. Soc. Ind. Appl. Math. — 1964. — Vol. 12. — P. 848–854.

[8] Kellogg R.B. Nonlinear alternating direction algorithm // Math. Comp. — 1969. — Vol. 23. — P. 23–28.

[9] Knobloch P. New nonconforming finite elements for solving the incompressible Navier-Stokes equations // Numerical mathematics and advanced applications. Proc. ENUMATH 2001, 4-th European Conference, Ischia, July 2001 / Ed. F. Brezzi et al. — Springer, 2003. — P. 123–132.

[10] Kossaczky I. A recursive approach to local mesh refinement in two and three dimensions // J. Comput. and Appl. Math. — 1994. — Vol. 55. — P. 275–288.

[11] Layton W., Maubach J., Rabier P., Sunmonu A. Parallel finite element methods // Proc. ISMM Int. Conf. on Parallel and Distributed Computing and Systems. Pittsburgh, Pennsylvania, USA / Ed. R. Melhem. — 1992. — P. 299–304.

[12] Layton W., Maubach J., Rabier P. Robust methods for highly non-symmetric problems // Contemporary Mathematics. — 1994. — Vol. 180. — P. 265–270.

[13] Layton W., Maubach J., Rabier P. Parallel algorithms for maximal monotone operators of local type // Numerische Mathematik. — 1995. — Vol. 71. — P. 29–58.

[14] Layton W., Maubach J., Rabier P. Robustness of an elementwise parallel finite element method for convection diffusion problems // SIAM J. Sci. Comput. — 1998. — Vol. 19, No. 6. — P. 1870–1891.

[15] Layton W.J., Rabier P. Peaceman-Rachford procedure and domain decomposition for finite element problems // J. Numer. Linear Algebra with Appl. — 1995. — Vol. 2. — P. 363–394.

[16] Lions P.-L., Mercier B. Splitting algorithms for the sum of two non-linear operators // SIAM J. Numer. Anal. — 1979. — Vol. 16. — P. 964–979.

[17] Maubach J. Iterative Methods for Non-Linear Partial Differential Equations. — Amsterdam: C.W.I. Press, 1994. — (CWI Tract; 92).

[18] Maubach J. Local bisection refinement for n-simplicial grids generated by reflections // SIAM J. Sci. Comput. — 1995. — Vol. 16. — P. 210–227.

[19] Maubach J. The efficient location of simplicial neighbors for locally refined n-simplicial grids // Proc. Int. Meshing Roundtable. Pittsburgh, October. — 1996. — P. 137–153.

[20] Maubach J. On optimal order algebraic multilevel preconditioners for locally refined grids // Proc. Summer School on Super Computing. University of Nijmegen, the Netherlands, May. — 1997. — P. 1–15.

[21] Maubach J. Space-filling curves for 2-simplicial meshes created with bisections and reflections // Appl. Math. — 2004 (submitted).

[22] Mukherjee A. An Adaptive Finite Element Code for Elliptic Boundary Value Problems in Three Dimensions with Applications in Numerical Relativity: Phd. Thesis. — Penn State University, 1996.

[23] Peaceman D.W., Rachford H.H. The numerical solution of parablokic and elliptic differential equations // SIAM J. — 1955. — Vol. 3 — P. 28–41.

[24] Plaza A. et al. Mesh quality improvement and other properties in the four-triangles longest-edge partition // Comput. Aided Geom. Design. — 2004. — Vol. 21. — P. 353–369.

[25] Rivara M.C., Levin C. A 3-d refinement algorithm suitable for adaptive and multi-grid techniques // J. Comput. and Appl. Math. — 1992. — Vol. 8. — P. 281–290.