INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 875–892 (1997)

ADAPTIVE PARALLEL MULTIGRID SOLUTION OF 2D INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

JIAN WU[1]*, HUBERT RITZDORF[1], KEES OOSTERLEE[1], BARBARA STECKEL[1] AND ANTON SCHÜLLER[1]

[1]*Institute of Algorithms and Scientific Computing, GMD—German Research CentrE for Information Technology, Schloss Birlinghoven, D-53754 Sankt Augustin, Germany*

SUMMARY

In this paper an adaptive parallel multigrid method and an application example for the 2D incompressible Navier–Stokes equations are described. The strategy of the adaptivity in the sense of local grid refinement in the multigrid context is the multilevel adaptive technique (MLAT) suggested by Brandt. The parallelization of this method on scalable parallel systems is based on the portable communication library CLIC and the message-passing standards: PARMACS, PVM and MPI. The specific problem considered in this work is a two-dimensional hole pressure problem in which a Poiseuille channel flow is disturbed by a cavity on one side of the channel. Near geometric singularities a very fine grid is needed for obtaining an accurate solution of the pressure value. Two important issues of the efficiency of adaptive parallel multigrid algorithms, namely the data redistribution strategy and the refinement criterion, are discussed here. For approximate dynamic load balancing, new data in the adaptive steps are redistributed into distributed memories in different processors of the parallel system by block remapping. Among several refinement criteria tested in this work, the most suitable one for the specific problem is that based on finite-element residuals from the point of view of self-adaptivity and computational efficiency, since it is a kind of error indicator and can stop refinement algorithms in a natural way for a given tolerance. Comparisons between different global grids without and with local refinement have shown the advantages of the self-adaptive technique, as this can save computer memory and speed up the computing time several times without impairing the numerical accuracy. © 1997 By John Wiley & Sons, Ltd. *Int. J. Numer. Methods Fluids* 24, 875–892, 1997.

(No. of Figs: 13. No. of Tables: 5. No. of Refs: 19.)

KEY WORDS: adaptive parallel multigrid method; local refinement criteria; incompressible Navier–Stokes equations; hole pressure problem

INTRODUCTION

To fit the requirement of more and more accurate results from CFD practice, computational grids for discretization equation systems should be finer and finer, which correspondingly increases the computing time and memory requirements dramatically. On the hardware side, highly parallel computer systems with distributed memory offer the possibility of treating such large-size problems. On the numeric and software side, adaptive methods and algorithms can be used to avoid unnecessary globally fine grids and deal with problems in an economic way. The combination of parallel computing and adaptivity as well as the fastest solution method, multigrid, is therefore very attractive for grand challenges in CFD.

* Correspondence to: Jian Wu, Institute of Algorithms and Scientific Computing, German Research Centre for Information Technology, Schloss Birlinghoven, D-53754 Sankt Augushn, Germany

Contract grant sponsor: German Federal Minister for Education, Science, Research and Technology; Contract grant number: 0I IR 302 A7

CCC 0271–2091/97/090875–18$17.50

© 1997 by John Wiley & Sons, Ltd.

Received September 1995
Revised July 1996

For several years, multigrid solution algorithms have been parallelized successfully, e.g. in the LiSS package developed at SCAI/GMD.[1] This package is based on the portable communication library for block-structured grids, CLIC,[2] and the standard message-passing libraries PARMACS, PVM and MPI. Because the parallel efficiency is dependent upon the load-balancing and communication requirements between different parallel processes, the parallel strategy in the above package is grid partitioning, which means that the flow domain is divided into nearly equal-sized structured blocks and the data are distributed blockwise into different parallel processes. Interprocess communications are needed only for boundary data exchanges. For large-size problems, in which the ratio of the number of grid points on interblock boundaries and the number of interior grid points is very small except at the coarsest level of the multigrid algorithm, the communication in the solution steps can be reduced optimally. Therefore in this package the parallel efficiency is very high.

Upon incorporating adaptivity into such parallel algorithms, a difficulty in redistributing data generated in the adaptive steps to each load balancing occurs. In self-adaptive algorithms a built-in detector first finds the critical subregions in the computational domain at run time and then refines these subregions locally. The refinement process should be stopped at a certain refinement level or when a given tolerance is reached. The data structure of the refined grids is unknown *a priori* and is in general inhomogeneous in each block. It turns out that the load in parallel processes could be quite different if data on refinement stages were to stay in the same processes; then the parallel efficiency would be limited strong by the adaptivity. For this reason a data redistribution strategy is necessary. In fact, exact load balancing for dynamic adaptive algorithms is almost impossible and certainly inefficient, because the communication could require very sophisticated programming and a long computing time. In this case an approximate but efficient load-balancing strategy is more appropriate.

The data structure in the adaptive steps depends on the adaptive multigrid technique and the grid type. In LiSS the multilevel adaptive technique (MLAT) suggested by Brandt[3,4] has been implemented, in which refined grids are treated as extended fine grid levels in the multigrid context. Compared with the strategy of generating globally adaptive grids, local refinement has the advantage that the computational grids need not be globally modified, and that strong distortion and non-uniformity at each level can be avoided. Therefore the following steps are reasonable for approximate load balancing: (a) the type of refined grids in each block should be the same as the primary grid, namely structured grids (logically quadrilateral in 2D and hexahedral in 3D cases); (b) the refined grids are divided into several sub-blocks so that the data redistribution can be realized by sub-block mapping; (c) according to a crude approximation—the computing load is proportional to the number of grid points in a process—the sub-blocks in the adaptive step are redivided and remapped; (d) the sub-blocks should be remapped parallely either to nearest neighbours or along embedded trees. The detailed description of the above steps implemented in LiSS is very complicated. We concentrate only on point (c) in this paper. Clearly, in an adaptive algorithm the parallel efficiency can decrease owing to data redistribution and extra communication. However, the drastic reduction of memory requirements and computing time has shown the advantage of adaptive parallel multigrid algorithms.

Another important factor influencing the efficiency of a dynamic adaptive algorithm is the adaptive criterion employed in it. The choice of adaptive criterion is non-unique and problem-dependent. A suitable criterion for a specific problem should find the refined regions in the flow domain automatically, make them smaller in the next adaptive steps and be capable of stopping the refinement in a natural way. These are the principle requirements for a self-adaptive criterion. In the 'classical' engineering approach, criteria based on variable gradients are often used, but they may not fit the above requirements.[5] In the work of Sonar[5] and Ritzdorf and Stüben,[6] finite element residuals have been used as the refinement criterion for the Euler equations. According to the analysis, this kind of criterion seems to be a suitable one for two-dimensional inviscid compressible transonic flows. In our work this criterion will be used for the incompressible Navier–Stokes equations

 © 1997 by John Wiley & Sons, Ltd.

and compared with other criteria based on gradient-like control functions in the sense of self-adaptivity.

ADAPTIVITY AND PARALLELIZATION

The governing equations are the two-dimensional steady incompressible Navier–Stokes equations. The upwind finite volume discretization scheme used in this work is based on the work of Dick and Linden. This discretization of the convective fluxes has only first-order accuracy. To obtain second-order accuracy, defect correction can be used, in which the right-hand sides of the first-order discrete equations are corrected after every internal iteration by the difference between residuals from the second-order and first-order formulations. In Reference 8 it has been shown that the defect correction converges to the second-order solution. The internal iteration is based on FMG. The defect correction is only carried out on the finest grid with the van Leer κ-scheme[9] for $\kappa = 0$.

The standard full approximate multigrid scheme (FAS-FMG) does not need to be explained in detail. For the purpose of robustness, only the F-cycle is used for first-order accuracy and the V-cycle in the defect correction processing.

MLAT strategy

To explain the MLAT strategy, let us suppose that Ω_l^h ($l = 0, 1, 2, \cdots, l_c$) is a hierarchy of global grids, where Ω_0^h is the finest grid and Ω_{-l}^h (l is the level number, $l = 1, 2, 3, \cdots, l_f$, and h indicates the characteristic mesh size on level 0) are locally refined grids in subdomains of the original computational domain. Such subdomains are detected successively at run time by a certain criterion. As in FMG, in the MLAT at any refinement level $-l$ (on grid Ω_{-l}^h) the corresponding two-grid method uses Ω_{-l}^h and $\bar{\Omega}_{-l+1}^h$, where $\bar{\Omega}_{-l+1}^h$ is defined as

$$\bar{\Omega}_{-l+1}^h := \Omega_l^h \cap \Omega_{-l+1}^h. \tag{1}$$

Along the block boundaries of Ω_{-l}^h, boundary values are interpolated from the current approximation on the coarse grid Ω_{-l+1}^h. Figure 1 shows an example of grids at different refinement levels.

Refinement criteria

From the point of view of computational efficiency we desire that a criterion should be self-adaptive in the sense of error estimation and that the refinement will stop in a natural way for a given tolerance. In the 'classical' adaptive approach, gradients of relevant flow variables are used as detectors for adaption. They can detect e.g. locations of shocks and boundary layers, but have nothing to do with error control, and the values of control qualities may not decrease at a higher refinement level. Thus they cannot yield an automatic stop for a given tolerance in adaptive algorithms. Sonar[5] has used finite element ideas in adaptive control and compared several refinement indicators based on finite element (FE) residuals for finite volume solutions of the Euler equations. They were in the L^2-, H^{-1}- and L^1-norm. In Reference 5 it has been shown that these criteria are well suited for detecting regions which have to be refined for solving two-dimensional inviscid compressible flows, e.g. near shocks. They are proportional to the local mesh size h and decrease when the mesh size decreases during the refinement process. In contrast with them, refinement indicators based on gradients of flow variables cannot be used in an automatically self-adapting code. In the work of Ritzdorf and Stüben[6] the criterion based on FE residuals in the L^1-norm has been accepted and implemented successfully in LiSS for the Euler equations.

Analogously to References 5 and 6, control qualities based on FE residuals in the H^{-1}- and L^1-norm for the incompressible Navier–Stokes equations can be introduced.

© 1997 by John Wiley & Sons, Ltd.

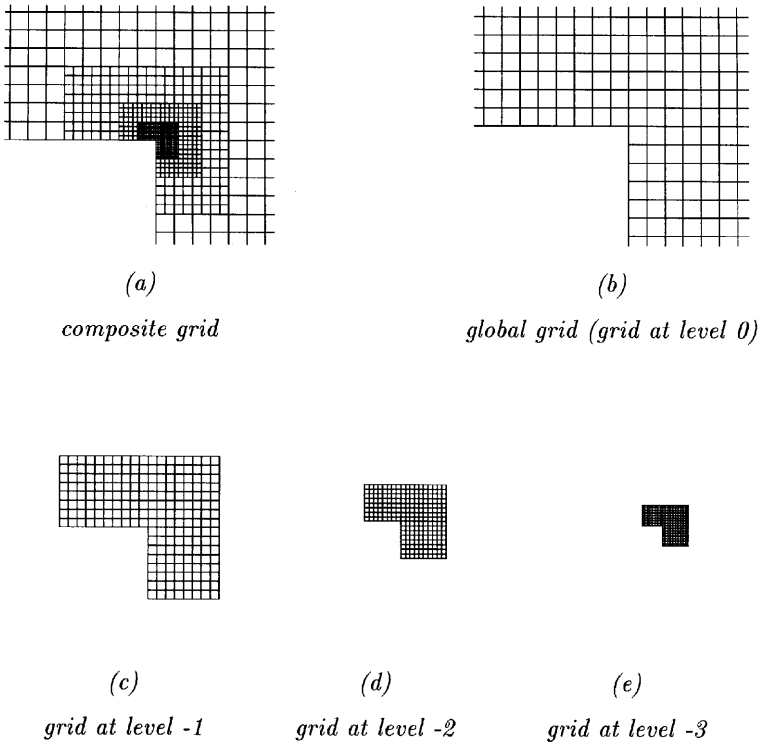


Figure 1. Composite, global and refined grids in MLAT process

Suppose that the discretized equation is $\mathscr{L}q = 0$; then the residual of the equation is

$$r^h := \mathscr{L}q^h. \tag{2}$$

Around any point P at the current level the corresponding rectangular control volume is subdivided into two triangular elements Δ_1 and Δ_2 (see Figure 2). In principle, quadratic elements can be chosen also, but the implementation of triangular elements is easier. The values needed for residual calculations on the four corners of the two triangles can be interpolated linearly from the nine neighbouring grid points. The control quality ψ_{L^1} in the L^1-norm is defined as

$$\psi_{L^1} = \|r^h\|_{L^1} = \sum_{\substack{k=1,2 \\ q=u,v,p}} \int_{\Delta_k} |r^h_{kq}| \mathrm{d}\Omega, \tag{3}$$

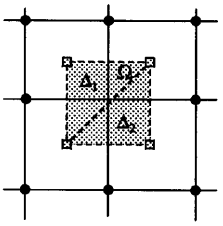


Figure 2. Finite elements Δ_1 and Δ_2 used for residual calculation in control volume Ω

 © 1997 by John Wiley & Sons, Ltd.

were index k denotes a value in the kth triangle Δ_k and r^h_{kq} is the approximate value of the q-residual over Δ_k. That is obtained via the shape functions ϕ_{km} and the values q_m on finite element point m; for example, for $q = u$ is found

$$|r^h_{ku}| = \sum_{m=1,2,3} |r^h_{ku}|_m \phi_{km}, \tag{4}$$

$$|r^h_{ku}|_m = |2u_m \partial_x u_m + v_m \partial_y u_m + u_m \partial_y v_m - \partial_x p_m - 1/Re(\partial_{xx} u_m + \partial_{yy} u_m)|. \tag{5}$$

All primitive variables with index m are average values from neighbouring points and the first and second derivatives are obtained via the weak formulation and shape functions.

The control quality $\psi_{H^{-1}}$ in the H^{-1}-norm is defined as

$$\psi_{H^{-1}} = \|r^h\|_{H^{-1}} = \max_{\substack{k=1,2 \\ m=1,2,3 \\ q=u,v,p}} \frac{\left| \int_{\Delta_k} r^h_{kq} \phi_{km} \mathrm{d}\Omega \right|}{\|\phi_{km}\|_{H^1_0(\Omega)}}. \tag{6}$$

In equation (6), $\|\phi_{km}\|_{H^1_0(\Omega)}$ is defined as

$$\|\phi_{km}\|_{H^1_0(\Omega)} = \sqrt{\left(\int_{\Delta_k} \phi^2_{km} \mathrm{d}\Omega + \int_{\Delta_k} (\partial_x \phi_{km})^2 \mathrm{d}\Omega + \int_{\Delta_k} (\partial_y \phi_{km})^2 \, \mathrm{d}\Omega \right)} \tag{7}$$

and is evaluated exactly.

For comparison, criteria based on control qualities of distances to singularities and on gradient-like qualities can be tested. They may be written as

$$\psi_d = \max\left(\frac{h}{|x - x^*|}, \frac{h}{|y - y^*|} \right), \tag{8}$$

$$\psi_q = \max_m |\Delta q| \sim h \left| \frac{\Delta q}{\Delta x_i} \right|, \tag{9}$$

where x^* and y^* are the co-ordinates of a singularity and in ψ_q a dimension of the mesh size h is added to the gradient for the self-adaptivity. If a region near a solid wall is interesting, the control quality ψ_q for $q = u, v$ can be very small because of the no-slip condition. In this case ψ_q should be scaled by the absolute value of this component as

$$\psi_q = \max_m \left(\frac{|\Delta q|}{\varepsilon + |q|} \right), \quad 0 < \varepsilon \ll 1. \tag{10}$$

In refinement processes after the $-l + 1$ level the chosen control quality is computed. If there is a region with $\psi \geqslant \lambda$ for a given tolerance λ, the refinement at level $-l$ should be carried out. The tolerance λ is problem- and criterion-specific.

Parallelization

An efficient way to achieve load balancing for block-structured CFD problems on distributed memory systems is grid partitioning. In grid partitioning, the computational domain is divided into several blocks. Every block has approximately the same number of grid points. A host process distributes each block to a different node process; each node process receives the grid and its overlap. Then the solution process starts with exchanging and updating necessary data on block boundaries between processes. After global convergence, each node process sends the results to the host process.[10] In this method, data exchange between process (process communication) is only necessary on the overlapping regions along interior boundaries. The grid-partitioning strategy is realized in

© 1997 by John Wiley & Sons, Ltd.

LiSS, a numerical solution package for partial differential equations with multigrid on sequential and parallel computers.[1] LiSS is parallelized with the communication library CLIC for block-structured grids. It is based on the message-passing interface PARMACS[11] and can be ported on other interfaces, PVM and MPI.

Load balancing for MLAT

The refined grids in each block may not necessarily have the same number of grid points and we should remap the refined grids to different processes for load balancing. It is desirable that each process should have the same number of refined grid points, but this is very complicated and sometimes impossible or inefficient, because the number of grid points in each refined area can be extremely different. According to Ritzdorf and Stüben,[6] in LiSS the following remapping strategy has been used.

1. At refinement level $-l$ for each block i (each process has only one block) the number of cells to be refined is n_i. If $\sum_i n_i = 0$, the refinement will be stopped.
2. The optimal cell number to be redistributed to each processor, $n^{(P)} = \sum_i n_i/P$, is computed and broadcasted.
3. All blocks with $n_i \leqslant n^{(P)}$ are distributed to different processes.
4. The new optimal cell number for the remaining $\tilde{P}$ processors is recomputed: $\tilde{n}^{(P)} = \tilde{N}^{-l}/\tilde{P}$, with $\tilde{N}^{-l} = \sum_{n_i > n^{(P)}} n_i$. If there is any block with $n_i \leqslant \tilde{n}^{(P)}$, it should be distributed to a process.
5. Step 4 will be repeated until there is no block with $n_i \leqslant \tilde{n}^{(P)}$.
6. Blocks with $n_i > \tilde{n}^{(P)}$ should be divided into m_i sub-blocks, where m_i is an integer, $m_i = \max(m \leqslant n_i/\tilde{n}^{(P)})$. Each of the sub-blocks has approximately the same cell number $\tilde{n}_{i_k} = n_i/m_i$.
7. Because the total sub-block number $M = \sum_i m_i \leqslant \sum n_i/\tilde{n}^{(P)} = \tilde{N}^{-l}/\tilde{n}^{(P)} \leqslant \tilde{P}$, there may be some free processors. In this case, blocks i containing the largest sub-blocks are resubdivided with $m_i + 1$ until all processors are busy.

The above is only a coarse description. A detailed one is very sophisticated. This strategy does not guarantee exact load balancing and some steps above need to be improved. However, in practice it is a satisfactory load-balancing strategy and relatively easy to programme. In the current version of LiSS, all remapping steps described above can be performed in parallel and all communications can be arranged either to nearest neighbours or along embedded trees. As commented in Reference 6, the communication overhead is merely $O(\log P)$. In practice the time for remapping work is negligible compared with the total work.

Within the multigrid cycle at each refinement level $-l$ the data on grid Ω^h_{-l+1} required for interpolation should first be sent to the related processes of grid Ω^h_{-l} and then interpolations and corrections can be performed. By fine-to-coarse transfer all necessary computations should be done at the finer level (evaluation of residuals, application of full weighting operator, etc.). Only data relevant to the coarse level computation (residuals and current approximations) are redistributed. Examples of the redistribution strategy are shown in the following subsection.

Parallel efficiency and measure of load balancing

In Reference 6 the measurable parallel efficiency $\varepsilon(P)$ has been introduced:

$$\varepsilon(P) = \frac{1}{P} \frac{\sum\limits_{i=1,\cdots,P} a_i}{\max\limits_{i=1,\cdots,P} (a_i + c_i)}, \tag{11}$$

 © 1997 by John Wiley & Sons, Ltd.

where P denotes the number of processors and a_i and c_i denote respectively the (wall-clock) time for the arithmetic and the total communication time (including idle time) on processor i. We can define the parallel efficiency at level $-l$ to be $\varepsilon^{-l}(P)$ as in equation (11). Suppose that the parallel algorithm does not involve substantial additional arithmetic and the floating point performance does not depend sensitively on the mesh size. Then $\varepsilon(P)$ is approximately the real measure of parallel efficiency.

For exact load balancing, every processor (i.e. every block should have the same number of grid points or cells. However, in refinement processes this may not be the case. For justifying a code in real applications, we can introduce a measure of load balancing $\varepsilon_{\mathrm{r}}^{-l}$ for local refinements at level $-l$ and ϵ_{r} for the global refinement process. We define $\varepsilon_{\mathrm{r}}^{-l}$ as

$$\varepsilon_{\mathrm{r}}^{-l} = \frac{\sum_{i=1,2,\ldots,P} n_i^{-l}}{n_{\max}^{-l} P} = 1 - \frac{1}{P n_{\max}^{-l}} \sum_{i=1,2,\ldots,P} (n_{\max}^{-l} - n_i^{-l}), \tag{12}$$

where n_i^{-l} denotes the number of points in processor i at level $-l$ and $n_{\max}^{-l} = \max_{i=1,2,\ldots,P} n_i^{-l}$. The definition of ε_{r} is

$$\varepsilon_{\mathrm{r}} = \frac{\sum_{l=0,1,\ldots} N_{\mathrm{g}}^{-l} \alpha^{-l}}{\sum_{l=0,1,\ldots} P n_{\max}^{-l} \alpha^{-l}}, \tag{13}$$

where $N_{\mathrm{g}}^{-l} = \sum_{i=1,\ldots,P} n_i^{-l}$ is the number of grid points at level $-l$ and α^{-l} is the average work unit for one grid point at level $-l$ relative to the work at level 0 ($\alpha^0 = 1$). We assume in the following text $\alpha^{-l} = 1$.

Generally we have $\varepsilon^{-l}(P) \leqslant \varepsilon_{\mathrm{r}}^{-l}$. If we suppose that each interior grid at the same level has the same work unit, $a_i \sim n_i$, and neglect c_i, then $\varepsilon_{\mathrm{r}}^{-l}$ is identical with the parallel efficiency $\varepsilon^{-l}(P)$. If the number of grid points in several blocks is much larger than in the other blocks, then $\varepsilon_{\mathrm{r}}^{-l}$ tends to be smaller for larger P. According to the remapping strategy described in the previous subsection, when only a few blocks in the original grid need refinements and the processor number of the parallel system is moderate, it is more flexible to remap sub-blocks in the refinement phases. In this case $\sum_{i=1,2,\ldots,P}(n_{\max}^{-l} - n_i^{-l})$ in (12) is small and better load balancing can be obtained.

RESULTS FOR HOLE PRESSURE PROBLEM

The specific flow problem to be tested in the present work is the hole pressure problem. As depicted in Figure 3, the flow in a uniform channel of height $H = 1$ and total length L is disturbed by a cavity of breadth b and depth d on the lower wall of the channel. The problem is to determine the hole pressure, i.e. the difference between the normal stress exerted by the fluid on the bottom of the cavity and that on the channel wall opposite the cavity opening. The hole pressure provides a characterization of non-Newtonian fluids and is also interesting in investigations of Newtonian fluid flows. There are several numerical computations of hole pressure problems for Newtonian fluids in the literature, e.g. by Malkus,[12] Richards and Townsend,[13] Jackson and Finlayson,[14] Crochet *et al.*[15] and Lodge *et al.*[16] In Reference 16 the computational results and their accuracy have been discussed in detail and compared with experimental data. However, Reference 16 is concerned only with the case of small Reynolds numbers.

In the present work this problem is chosen for testing the present local refinement strategy, since it is a practical problem with application background, the flow domain is simple and there are geometric singularities. The numerical study first examines the comparison with the results from Reference 16 for a small Reynolds number, $Re = 60$, with and without local refinement. Then the case of $Re = 500$

© 1997 by John Wiley & Sons, Ltd.

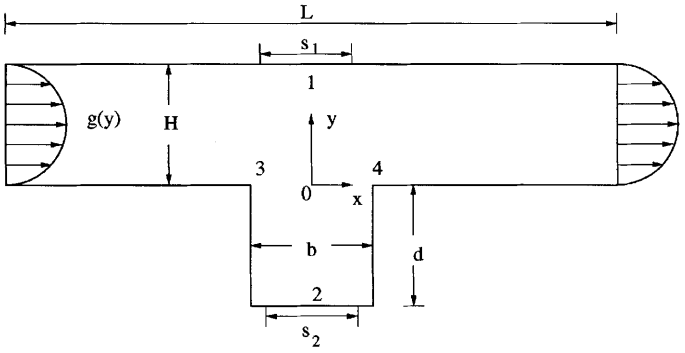


Figure 3. Schematic description of flow domain

is discussed. All the computations in this work were carried out on an IBM Scalable POWERparallel System (SP2) with 10 node processors.

$Re = 60$ without local refinement

The inlet and outlet velocity distribution $g(y)$ in Figure 3 is the same as that of Poiseuille flow in a uniform channel and defined as

$$g(y) = y(1 - y). \tag{14}$$

All the other boundary conditions on the solid walls are no-slip conditions. From the above definition the average inlet velocity U and the flow rate Q are both $1/6$. The breadth b and depth d of the cavity are both 1 and L is 5. The Reynolds number is defined as $Re = Ub/\nu$, where ν is the kinematic viscosity. The dimensionless normal stress σ_{22} acting on both regions of interest, i.e. on the channel wall opposite the cavity opening, s_1, and on the bottom of the cavity, s_2, is given by

$$\sigma_{22} = -p + 2v_y, \tag{15}$$

where the viscosity coefficient does not appear owing to the dimensionless formulation. The hole pressure is defined as the mean difference between the normal stresses on the average intervals s_1 and s_2 (see Figure 3):

$$\Delta = \frac{1}{s_1}\int_{s_1} \sigma_{22}(x, 1)\mathrm{d}x - \frac{1}{s_2}\int_{s_2} \sigma_{22}(x, -1)\mathrm{d}x. \tag{16}$$

Here s_1 and s_2 are equal to s.

It should be noticed that the definition of the Reynolds number in Reference 16 is six times smaller than that in the real computation.

Because of the geometric singularity, a dense grid should be used in the regions near the two upper corners of the cavity. This has also been done in Reference 16 with the finite element method. For the purpose of testing local refinement, it is better to use a globally uniform grid with $h_x = h_y$, where h_x and h_y are the mesh sizes in directions x and y, so that we can determine whether the regions that need to be refined can be detected by our criteria. The flow domain is divided into six rectangular load-balancing blocks as shown in Figure 4. In this case only six processors (exclusive of the host processor) of the parallel computer are needed. Every block has the same number of cells. Because the number of grid points along interior boundaries in each block is not the same and owing to the fact that these grid points belong to only one block, exact load balancing is generally very difficult and unnecessary. The unit length of mesh size without refinement is $h = h_x = h_y = 1/T$. For

 © 1997 by John Wiley & Sons, Ltd.

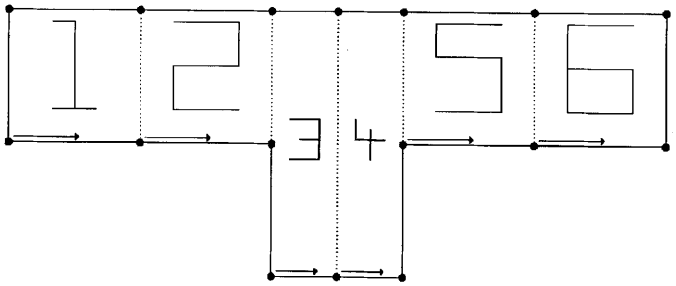


Figure 4. Domain decomposition

convenience in the following text, 'T = 128', 'T = 64', etc. denote cases of global grids with $T = 128, 64, \ldots$.

The first example is T = 128. In this case there are approximately 16,640 grid points in all blocks and $h = 1/128$. The global grid is finer than in Reference 16. The computed streamline pattern and pressure contours are plotted in Figure 5 for $Re = 60$. For comparison the pictures are drawn vertically. All the contour lines have the same values as in Reference 16. For different lengths s the computed hole pressure is presented in Table I. The hole pressures are almost identical with those in Reference 16.

It should be pointed out that the calculated pressure values at the sharp corners of the cavity, points 3 and 4, depend on the local mesh size. The order of the truncation error in a circle with an approximate radius r of order $O(h^*)$ (h^* is the local grid spacing) around these points is lower than that outside the circle. Therefore they are geometric singular points. For the Poisson equations this has been discussed in References 17 and 18. The purpose of employing fine grids is to reduce the radius of the circle and to solve the flow outside the circle near the singularities more accurately rather than to improve the accuracy of the pressure values at the sharp corners. In this work the calculated values of the pressure at the singularities from different grids are presented only as comparative references for the solution accuracy near the singularities.

Table II shows the comparison between the computed pressure at positions $0, \cdots, 4$ in Figure 3, where $p_0 = p(0, 0)$, $p_1 = p(0, 1), p_2 = p(0, -1), p_3 = p(-0{\cdot}5, 0)$ and $p_4 = p(0{\cdot}5, 0)$. Also included

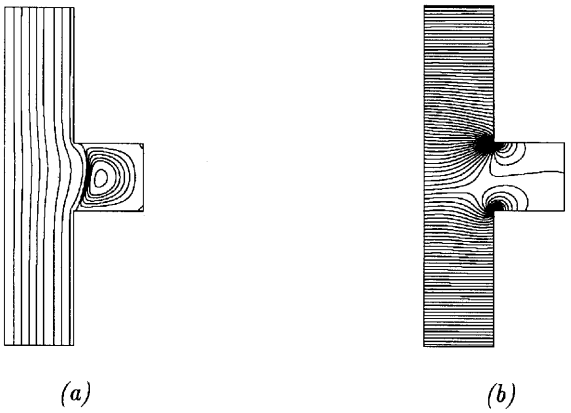


Figure 5. (a) Streamlines and (b) pressure contours for $Re = 60$

© 1997 by John Wiley & Sons, Ltd.

Table I. Hole pressure for different s

	$s = 0{\cdot}0$	$s = 0{\cdot}1$	$s = 0{\cdot}5$	$s = 1{\cdot}0$
Reference 16	0·3795	0·3796	0·3795	0·3829
Present	0·3774	0·3775	0·3786	0·3827

Table II. Computed pressure values on different grids

T	p_0	p_1	p_2	p_3	p_4	Δ	$\|r\|_{\max}$
16	0·3807	0·0097	0·3811	−0·2930	0·2786	0·3735	1×10^{-7}
32	0·3921	0·0125	0·3883	−0·4031	0·3109	0·3802	5×10^{-8}
64	0·3956	0·0134	0·3905	−0·5486	0·3926	0·3822	3×10^{-8}
128	0·3928	0·0137	0·3912	−0·7267	0·5101	0·3827	5×10^{-7}

are the hole pressure Δ and the maximal solution residual $|r|_{\max}$ of the Navier–Stokes equations after 15 defect correction steps. From Table II and Figure 5 we can see that the critical regions of this flow should be near the two singular points. Compared with T = 128, the relative differences of p_0, p_1 and p_2 from T = 64 are smaller than 3%. Since $v_y \ll 1$ and the hole pressure can be approximated by $p_2 - p_1$, the Δ-value is not influenced strongly by the mesh size if h is small enough. On the other hand, the differences of p_3 and p_4 between coarse and fine grid are quite large owing to the discretization errors and they do not converge as the mesh size is decreased, as explained above. This is also shown in Figure 6, in which the computed pressure distribution is plotted at $x = 0{\cdot}5$, near the right singularity.

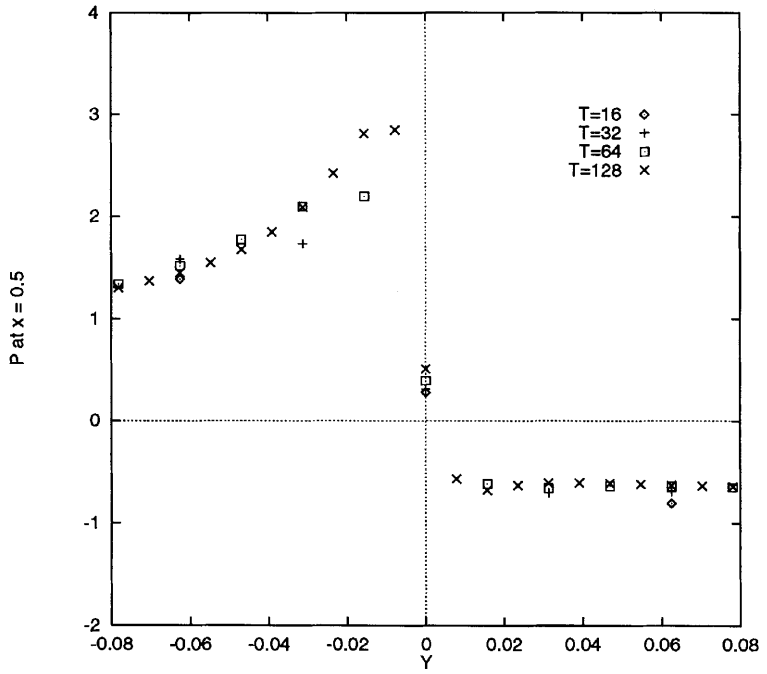


Figure 6. Pressure distribution at $x = 0{\cdot}5$ for $Re = 60$ without local refinement

 © 1997 by John Wiley & Sons, Ltd.

Re=60 with local refinement

It is known that the grid should be finer near the two geometric singular points than it is elsewhere, such as near the inlet or in the lower side of the cavity. Of course, if we are interested in the eddies near the lower corners for large *Re*, a very fine grid in these regions is needed. However, a coarse grid in these regions cannot affect the global accuracy substantially.

The smaller the distance to the singular points is, the larger the discretization errors of pressure are and therefore the finer the grid should be. From this point of view we can use a criterion based on the ratio of the mesh size and the distance to a singularity, namely ψ_d defined by equation (8). Let us refine the grid with $T = 16$ recursively three times with a given tolerance $\lambda = 1/8$ for the reason that at places with $d > 8h$ there is no obvious influence of the singularities. The minimal mesh size after three refinement steps is the same with $T = 128$. After 15 steps of defect correction the maximal global residual $|r|_{\max} \approx 10^{-7}$ has been reached. The relative differences of p_3 and p_4 from $T = 128$ and 2·4% and 3·8%. The difference of the hole pressure Δ is only 0·2% as shown in Table III. The domain decomposition in Figure 4 is chosen such that the two geometrical singularities are not in the same block. With the current strategy in LiSS there are $16 \times 16 \times 4$ elements (levels $0, \ldots, -3$) in each block to be dealt with. This is 16 times less than by T = 128.

These refinement results are satisfactory, but this criterion is not a self-adapting one. We want to find criteria to do similar work self-adaptively. Let us now test the other criteria based on the control qualities introduced above, namely $\psi_{L^1}, \psi_{H^{-1}}$ and ψ_q $(q = u, v, p)$.

In Figure 7 the contours of control qualities ψ for T = 32 are shown. Obviously ψ_u is unsuitable as a self-adapting indicator because it is concentrated near the channel wall, where very fine grids are unnecessary for a small Reynolds number. The contours of other control qualities are concentrated near the singularities. Among them, ψ_{L^1} and $\psi_{H^{-1}}$ are similar to each other, so we need only discuss ψ_{L^1} later. Analogously to the above discussion, in the following and in Figure 9, 'Ref_L1', 'Ref_v' and 'Ref_p' indicate the refinement cases with criteria based on ψ_{L^1}, ψ_v and ψ_p respectively.

In the following computations the global grid is generated with $T = 16$ and three refinement steps are carried out. The refinement tolerances are chosen so that there is no completely refined block. They are $\lambda_{L^1} = 0{\cdot}33 \times 10^{-3}$ and $\lambda_{v,p} = 0{\cdot}4$. The global and refined grids are shown in Figure 8. Although the different criteria have led to quite different geometries of the refined regions, all results in Table III from locally refined grids agree well with those from T = 128. The distribution of computed pressure near the right singularity at $x = 0{\cdot}5$ is plotted in Figure 9. All curves denoting results with different criteria are very close to the results of T = 128. For $Re = 60$ the discretization errors are dependent on mesh sizes only near the local singularities, while the global values from boundary integrals, such as the hole pressure Δ in Table III, are changed only slightly by the local refinement.

The maxima of the control qualities around the two singularities at every refinement level, $\psi^1_{\max}$ and $\psi^2_{\max}$, the number of grid points, N_{g}^{-l}, the maximum of grid points, $N_{\max}^{-l} = n_{\max}^{-l} P$, and the measures of load balancing, $\varepsilon_{\mathrm{r}}^{-l}$ and ε_{r} (approximately for $\alpha^{-l} = 1$ in equation (13)), are given in

Table III. Computed pressure values on different grids

	p_0	p_1	p_2	p_3	p_4	Δ	N_{g}	$\lvert r\rvert_{\max}$
T = 128	0·3928	0·0137	0·3912	−0·7267	0·5101	0·3827	99846	5×10^{-7}
ψ_d	0·3931	0·0103	0·3901	−0·7096	0·4908	0·3819	7088	1×10^{-7}
ψ_{L^1}	0·3878	0·0103	0·3882	−0·7175	0·5149	0·3832	4814	7×10^{-7}
ψ_v	0·3935	0·0124	0·3878	−0·7275	0·5131	0·3776	9026	6×10^{-6}
ψ_p	0·3941	0·0104	0·3906	−0·7282	0·5091	0·3819	4890	2×10^{-7}

© 1997 by John Wiley & Sons, Ltd.

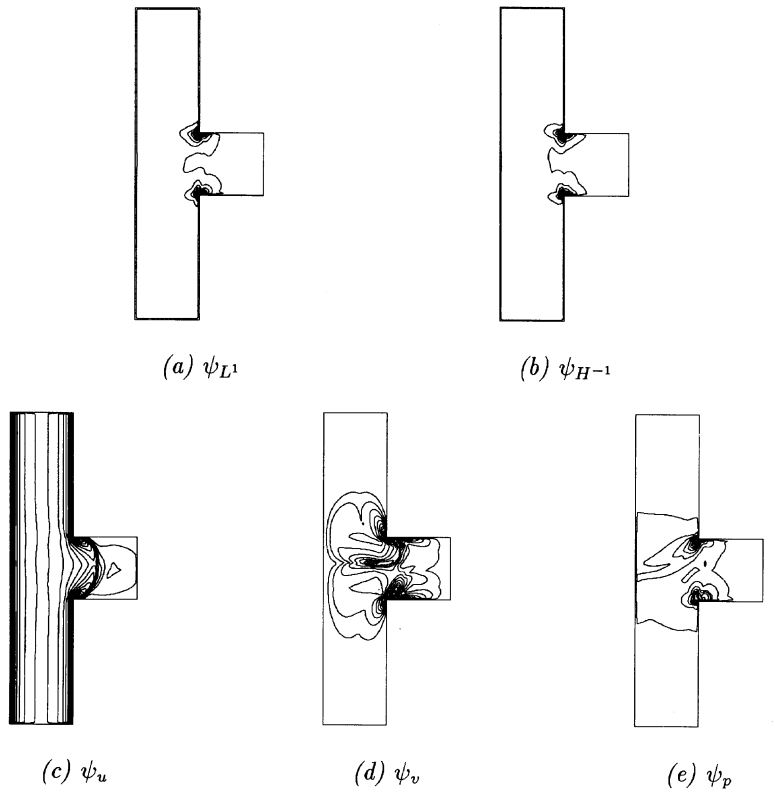


Figure 7. Contours of control qualities

Table IV. The maxima of ψ_{L^1} have been reduced proportionally to the mesh size. This means that $\psi_{L^1} \sim h^n$ and $n \approx 1$. At the third level, ψ_{L^1} is smaller than the tolerance $\lambda = 0{\cdot}33 \times 10^{-3}$, so further refinement can be stopped automatically. The maxima for Ref_v decrease only slowly and those for Ref_p even increase around the singularities. This means that $\psi_{v,p} \sim h^n$ and $n < 1$. In this case the refinement process does not tend to a natural stop. For Ref_L1 the measure of load balancing at each

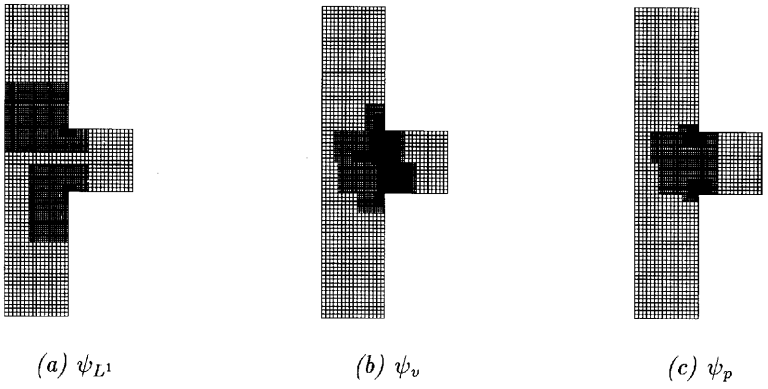


Figure 8. Locally refined grids for $Re = 60$

 © 1997 by John Wiley & Sons, Ltd.

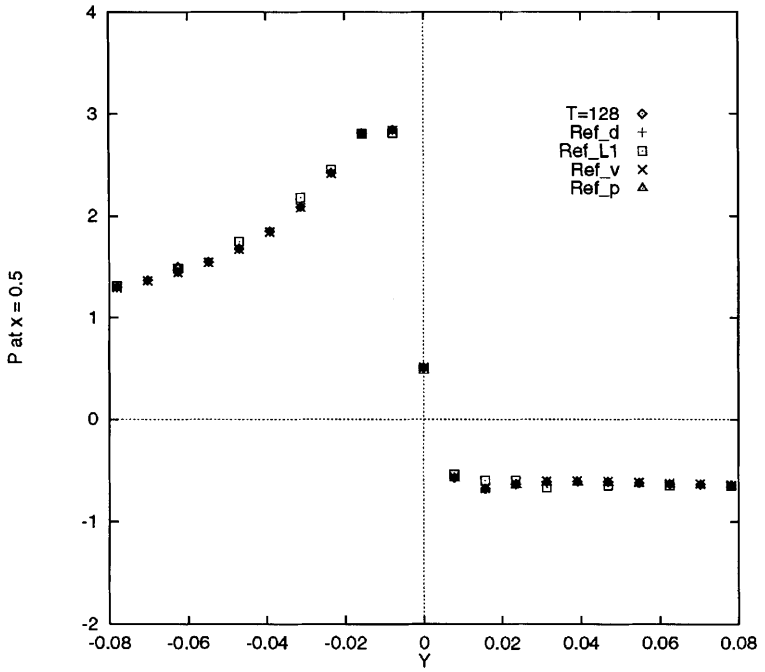


Figure 9. Pressure distribution at $x = 0{\cdot}5$ for $Re = 60$ with local refinement

Table IV. Maxima of control qualities and load-balancing efficiencies

	l	$\psi^1_{\max}$	$\psi^2_{\max}$	N_g^{-l}	$N_{\max}^{-l}$	ε_r^{-l}
ψ_{L^1}	0	$2{\cdot}068 \times 10^{-3}$	$1{\cdot}482 \times 10^{-3}$	1750	1782	0·982
$\lambda = 0{\cdot}33 \times 10^{-3}$	1	$1{\cdot}030 \times 10^{-3}$	$0{\cdot}709 \times 10^{-3}$	2408	3354	0·736
	2	$0{\cdot}616 \times 10^{-3}$	$0{\cdot}378 \times 10^{-3}$	452	570	0·793
	3	$0{\cdot}314 \times 10^{-3}$	$0{\cdot}193 \times 10^{-3}$	204	270	0·756
		Global values $N_g, N_{\max}, \varepsilon_r$		4814	5976	0·805
ψ_v	0	1·1167	1·3233	1750	1782	0·982
$\lambda = 0{\cdot}4$	1	0·8507	1·0214	1690	2142	0·789
	2	0·782	0·7838	2968	7326	0·405
	3	0·7059	0·6896	2618	3990	0·656
		Global values $N_q, N_{\max}, \varepsilon_r$		9026	15240	0·592
ψ_p	0	0·9815	1·3521	1750	1782	0·982
$\lambda = 0{\cdot}4$	1	0·9073	1·3658	1288	1890	0·681
	2	0·9287	1·2145	994	1326	0·75
	3	1·0602	1·165	858	1734	0·495
		Global value $N_q, N_{\max}, \varepsilon_r$		4890	6732	0·726

© 1997 by John Wiley & Sons, Ltd.

level, ε_r^{-l}, is over 70%. The global load-balancing measure ε_r is over 80%. The refined regions for Ref_v and Ref_p only shrink slowly and more grid points are needed in the refinement steps. The number of grid points at refinement levels with lower load balancing is large; for example, for Ref_v at level -2, $\varepsilon_r^{-l} = 0{\cdot}405$ and $N_g^{-l} = 2968$, while for Ref_p at level -3, $\varepsilon_r^{-l} = 0{\cdot}495$ and $N_g^{-l} = 858$. Thus the global load balancing is lower owing to the non-optimal grid point distribution at refinement levels (for Ref_v it is about 60% and for Ref_p it is about 72·6%).

It appears that the computational accuracy does not depend on the refinement criteria directly when the regions near singularities are refined, but the control quality ψ_{L^1} of FE residuals is more suitable to serve as a refinement criterion near a geometric singularity since it provides residual estimations and allows automatic self-adaptation. After the computation it is certain that the residuals with the computed results in an FE norm are within a given tolerance. The gradient-like qualities ψ_v and ψ_p have nothing to do with the residual or error estimations and do not reach a natural stop of the refinement processes. The better load balancing by Ref_L1 does not depend directly on the refinement criterion also. However, the drastically reduced refinement regions using this criterion reduce the influence of the lower load balancing at the refinement levels in comparison with the global load balancing.

Re = 500 *with local refinement*

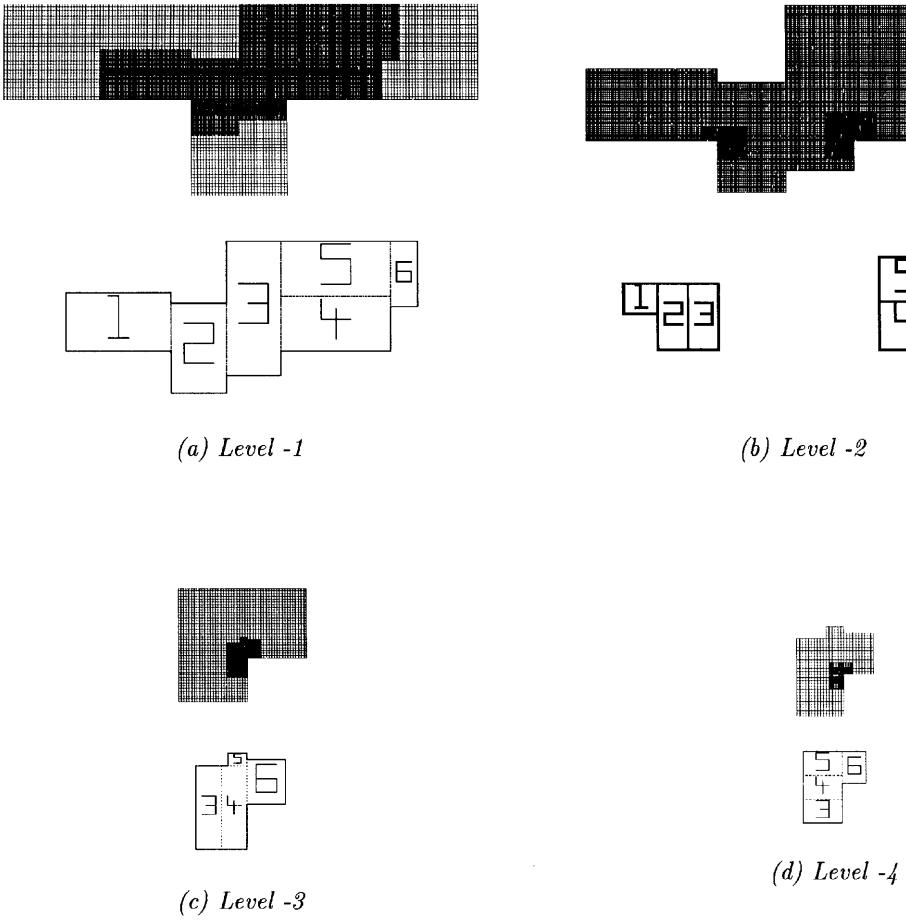
In this subsection we shall discuss the case of a relatively high Reynolds number, $Re = 500$, mainly from a numerical point of view without concern for its physical meaning.

The computation with refinement is started from a global grid with $T = 32$. The refinement criterion is based on ψ_{L^1} and the tolerance is $\lambda = 0{\cdot}104 \times 10^{-4}$ (with this tolerance there will be no globally refined blocks and the refinement process will be stopped at refinement level $l_f = 5$ automatically). The computed results of Ref_2, Ref_3 and Ref_4 (denoting $l_f = 2, 3, 4$) will be discussed below. The maximal residual after 25 steps of defect correction is $O(10^{-5})$.

At each level the whole refined region is divided into six blocks. Some of the refined grids and subdomains are shown in Figure 10. In this figure the numbers in blocks are sub-block numbers and the numbers outside the blocks are codes indicating boundary conditions along interior boundaries between grids of different refinement levels. The refined regions are also concentrated around the singularities and are reduced drastically at higher refinement levels. Near the singularities the pressure contours computed from T = 128 are reproduced by the computations with local refinement; see Figure 11.

The computed pressure at positions $0, \cdots, 4$, the hole pressure, the number of grid points at all refinement levels, N_g, the minimal mesh size $h_{\min}$ and the measure of global load balancing, ε_r, for T = 128, T = 32, Ref_2, Ref_3 and Ref_4 are given in Table V. The computed hole pressure Δ for T = 128 is 0·7413, about twice that for $Re = 60$. The results of Ref_2, Ref_3 and Ref_4 differs by less than 1% from those of T = 128. The results for p_0, p_1, p_2 and Δ from T = 32 are obviously improved by local refinement. With the same minimal mesh size $h_{\min} = 1/128$ the computed pressures p_3 and p_4 at two singularities from T = 128 and Ref_2 agree well. For Ref_3 and Ref_4 with further refinements, p_3 and p_4 are different from T = 128. Figure 12 shows the improvement of T = 32 by the local refinement. The pressure distributions for T = 128 and Ref_2 near $x = 0{\cdot}5, y = 0$ are identical and the results for Ref_3 and Ref_4 with more refinements are quite close to T = 128 up to $|\Delta y| \geqslant 0{\cdot}03$. In regions of $|\Delta y| < 0{\cdot}03$ the curves with finer grids are smoother than those of T = 128. We believe that the results for Ref_3 and Ref_4 should be more accurate because of their finer grids.

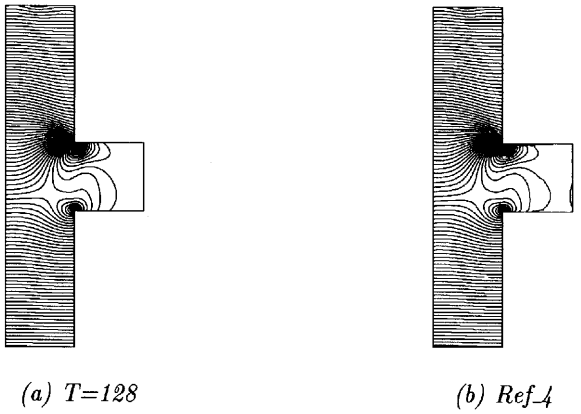
 © 1997 by John Wiley & Sons, Ltd.



(a) Level -1 *(b) Level -2*

(c) Level -3 *(d) Level -4*

Figure 10. Grids and subdomains at refinement levels



(a) T=128 *(b) Ref_4*

Figure 11. Pressure contours for $Re = 500$

© 1997 by John Wiley & Sons, Ltd.

Table V. Computed pressure values on different grids for $Re = 500$

	p_0	p_1	p_2	p_3	p_4	Δ	N_g	h_{min}	ε_r
T = 128	0·6456	0·0612	0·7838	−1·0896	1·7535	0·7413	99846	1/128	0·99
T = 32	0·6072	0·0516	0·7516	−1·1432	1·9885	0·7170	6566	1/32	0·99
Ref_2	0·6507	0·0611	0·7856	−1·0898	1·7477	0·7464	21074	1/128	0·788
Ref_3	0·6545	0·0620	0·7890	−1·2023	1·8245	0·7491	22186	1/256	0·811
Ref_4	0·6558	0·0622	0·7898	−1·3843	1·9729	0·7498	22410	1/512	0·800

The total number of grid points for Ref_4 is 22,410, which is about 3·4 times the number of grid points for $T = 32$ without refinement. Compared with the case of T = 128, the computer memory requirement is reduced 4·46 times. The global parallel efficiency $\varepsilon(P)$ (defined in equation (11)) with the local refinement based on the wall-block time is nearly 68·5%, which is satisfactory. In Figure 10 it is also shown how the sub-blocks at the four refinement levels are remapped into different processes. All the refinement process have good global load balancing, $\varepsilon_r \approx 80\%$ in Table V. It can be seen from Figure 10 that if we could remap the sub-blocks in a more flexible way, instead of only sending one sub-block into one processor, the parallel efficiency might be improved. Unsatisfactory examples of the present remapping strategy are block 6 at level −1 and block 5 at level −3 in Figure 10. On the other hand, if the number of grid points in refined regions is reduced very quickly as in the case discussed here, i.e. $N_g^{-3} \approx 0{\cdot}05N_g^{-2}$ and $N_g^{-4} \approx 0{\cdot}01N_g^{-3}$ in Table V, this unfavourable remapping cannot affect the global load balancing strongly. Another important factor for the real computational efficiency may be the relation between the geometry of refined regions and the MLAT

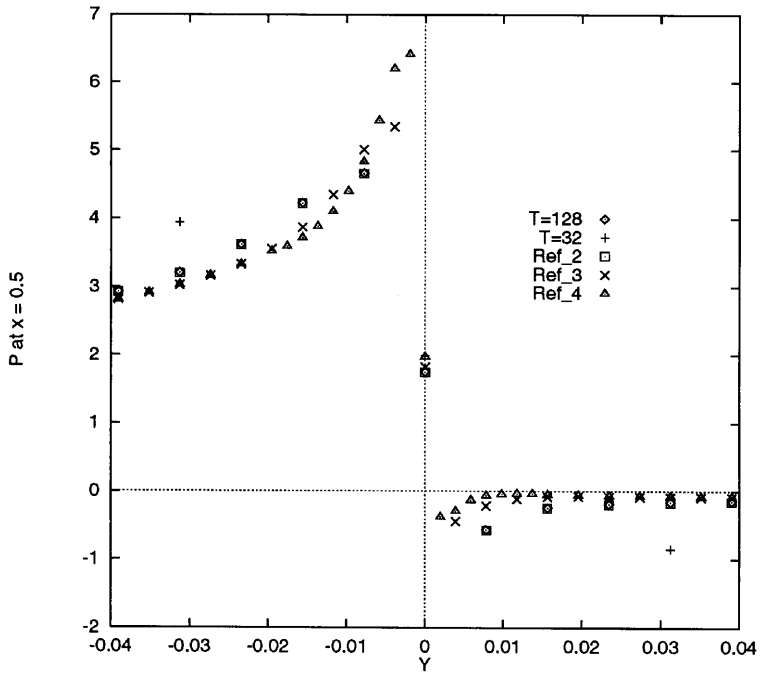


Figure 12. Pressure distribution at $x = 0{\cdot}5$ for $Re = 500$ with local refinement

 © 1997 by John Wiley & Sons, Ltd.

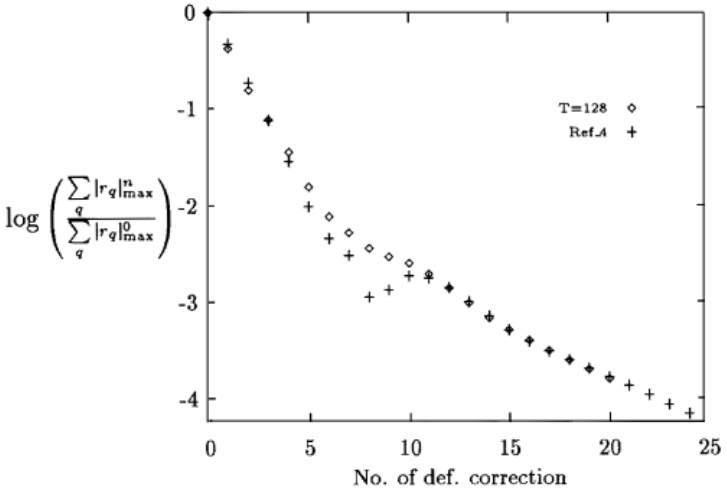


Figure 13. Convergence history for $Re = 500$

efficiency and the ratio of communication and arithmetic operations in the refinement steps. These are not discussed in this paper.

Figure 13 shows the convergence history in the sense of $|r|^n_{\max}/|r|^0_{\max}$ for two cases, T = 128 and Ref_4. After 10 defect correction steps the relative residuals are identical. From the run time documents of two examples we have observed that the maximal solution time (including communication time) in one processor with Ref_4 (wall-clock time 73·86 s) is 4·2 times less than with $T = 128$ (wall-clock time 312·4 s). This means about 4·2 times speed-up by the adaptive technique.

CONCLUDING REMARKS

From the observations and discussion above we can conclude the following:

1. The adaptivity in the sense of local grid refinement is advantageous since it can solve these problems in a faster and cheaper way.
2. The adaptive parallel multigrid algorithm introduced in the present work is very efficient on message-passing parallel systems.
3. To obtain higher computational accuracy, different local refinement criteria can be used, but the FE residual criterion tested in this work is favourable compared with those based on gradient-like control qualities near a geometrical singularity in the self-adapting sense.
4. The relation between the parallel efficiencies and the MLAT has not been addressed in this paper. According to the remapping strategy described in this work, the smaller a local refinement region is, the more flexible the remapping process is and the easier we can achieve easier load balancing. In the case of quickly shrunk refinement regions (e.g. with the FE residual criterion) the influence of the lower load balancing at the refinement levels on the global load balancing can be reduced.
5. In the future a more efficient remapping strategy and refinement criteria for 3D problems should be developed and tested.

© 1997 by John Wiley & Sons, Ltd.

ACKNOWLEDGEMENT

This work was supported by the German Federal Minister for Education, Science, Research and Technology under contract 01 IR 302 A7 (POPINDA Project).

REFERENCES

1. H. Ritzdorf, A., Schüller, B. Steckel and K. Stüben, 'LiSS—an environment for the parallel differential equations on general 2D domains', *Parallel Comput.*, **20**, 1559–1570 (1994).
2. B. Eisfeld, H. Ritsdorf, H.-M. Bleeke and N. Kroll, 'POPINDA Ein BMFT-Verbundprojekt mit DASA, Deutsche Airbus, DLR, Dornier, GMD und IBM zur protablen Parallelisierung industrieller Anwendungen', *Proc. 3rd Workshop on Scientific Computing*, 30–43, Braunschweig, 4–6.10. 1994.
3. A. Brandt, 'Multi-level adaptive solutions to boundary-value problems', *Math. Comput.*, **31**, 333–390 (1977).
4. A. Brandt, 'Multigrid techniques: 1984 guide, with applications to fluid dynamics', *GMD-Studie 85*, GMD, St. Augustin, 1984.
5. Th. Sonar, 'Strong and weak norm refinement indicators based on the finite element residual for compressible flow', in *Impact of Computing in Science and Engineering*, **5**, 111–127 (1993).
6. H. Ritzdorf and K. Stüben, 'Adaptive multigrid on distributed memory computers', in P. W. Hemker and P. Wesseling (eds), *Multigrid Methods IV*, Birkhäuser, 77–95, 1993.
7. E. Dick and J. Linden, 'A multigrid method for steady incompressible Navier–Stokes equations based on flux difference splitting', *j. numer. methods fluids*, **14**, 1311–1323 (1992).
8. W. Hackbusch, *Multi-grid Methods and Applications*, Springer, Berlin, 1985, p. 288.
9. B. van Leer, 'Flux-vector splitting for Euler equations', *Lecture Notes Phys.*, **170**, 507–512 (1982).
10. L. Linden, B. Steckel and K. Stüben, 'Parallel multigrid solution of the Navier–Stokes equations', *Parallel Comput.*, **7**, 461–475 (1988).
11. R. Calkin, R. Hempel, H. C. Hoppe and P. Wypior, 'Portable programming with the PARMACS message-passing library', *Parallel Comput.*, **20**, 615–632 (1994).
12. D. S. Malkus, 'Calculation of hole error by finite element methods', in C. Klason and J. Kubat (eds), *Prov. VIIth Int. Congr. on Rheology*, 1976, p. 618.
13. G. D. Richards and P. Townsend, 'A finite element computer model of the hole pressure problem', *Rheol. Acta*, **20**, 261 (1981).
14. N. R. Jackson and B. R. Finlayson, 'Calculation of hole pressure: I. Newtonian fluids', *J. Non-Newtonian Fluid Mech.*, **10**, 55–69 (1982).
15. M. J. Crochet, S. Dupont and J. M. Marchal, 'Numerical calculation of the Newtonian liquid circular hole pressure', *J. Rheol.*, **30**, 91 (1986).
16. A. S. Lodge, W. G. Pritchard and L. R. Scott, 'The hole-pressure problem', *IMA J. Appl. Math.*, **46**, 39–66 (1991).
17. P. Laasonen, 'On the truncation error of discrete approximations to the solutions of Dirichlet problems in a domain with corners', *J. Assoc. Comput. Mech.*, **5**, 32–38 (1958).
18. W. Kaspar and R. Remke, 'Die numerische Behandlung der Poisson-Gleichung auf einem Gebiet mit einspringenden Ecken', *Computing*, **22**, 141–151 (1979).

 © 1997 by John Wiley & Sons, Ltd.