INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 26, 303–322 (1998)

# ON THE USE OF ADAPTIVE HIERARCHICAL MESHES FOR NUMERICAL SIMULATION OF SEPARATED FLOWS

DEBORAH M. GREAVES[1]* AND ALISTAIR G. L. BORTHWICK[2]

[1] *Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, U.K.*
[2] *Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.*

## SUMMARY

This paper describes the use of adaptive hierarchical grids to predict incompressible separated flow at low Reynolds number. The grids consist of a quadtree system of hierarchical Cartesian meshes which are generated by recursive subdivision about seeding points. The governing equations are discretized in collocated primitive variable form using finite volumes and solved using a pressure correction scheme. The mesh is locally adapted at each time step, with panel division or removal dependent on the vorticity magnitude. The resulting grids have fine local resolution and are economical in array size. Results are presented for unidirectional, impulsively started flow past a circular and a square cylinder at various Reynolds numbers up to 5000 and 250 respectively. It is clear that hierarchical meshes may offer gains in efficiency when applied to complex flow domains or strongly sheared flows. However, as expected, the stepped approximation to curved boundaries resulting from the Cartesian quadtree representation adversely affects the accuracy of the results for flow past a circular cylinder. © 1998 John Wiley & Sons, Ltd.

*Int. J. Numer. Meth. Fluids*, **26**: 303–322 (1998).

KEY WORDS: hierarchical grids; quadtrees; adaptive remeshing; separated flows

## 1. INTRODUCTION

Viscous flow simulations are often calculated on non-adaptive structured meshes which allow fast numerical methods to be employed. However, adaptive local mesh refinement should be advantageous for modelling unsteady flows and flows with high local gradients, in that efficiency gains can be achieved in the relative distribution of calculation points. This paper considers the application of adaptive hierarchical Cartesian meshes to the simulation of impulsively started flow past bluff cylinders at low Reynolds numbers where the flow is essentially laminar.

Quadtree grids have been chosen as the basis of this work because they have a hierarchical tree structure which aids storage and neighbour finding, they are quick to generate and are suitable for adaptive remeshing applications. Moreover, quadtree grids can be generated automatically about any boundary, no matter how complex or irregular, and can provide regions of very high resolution where required (within the limitations of the computer).

The use of quadtree grids in image processing is described by Samet.[1,2] Quadtree algorithms have also been used to generate meshes for a variety of numerical schemes. For example, Yerry and

*Correspondence to: D. M. Greaves, Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, U.K.

Contract grant sponsor: U.K. EPSRC Marine Technology Directorate Ltd.

CCC 0271–2091/98/030303–20 $17.50
Received June 1996
Revised March 1997

Shephard[3] describe quadtree-based finite element mesh generation for structural analysis, in which triangular meshes are obtained from quadtree grids. Van Dommelen and Rundensteiner[4] simulated flow past a cylinder using a discrete vortex scheme whereby the stream function equation was solved on an adaptive quadtree mesh, created using the locations of the discrete vortices as grid seeding points. Quadtree grids have been applied with multigrid iterations to the solution of species transport and linearized shallow flow problems in complex domains by Gáspár and Józsa,[5] Gáspár and Simbierowicz,[6] Józsa and Gáspár[7] and Gáspár *et al.*[8] Quadtree finite element methods using quadrilateral and cubic elements were described by Young *et al.*[9] for compressible flows. De Zeeuw *et al.*[10] and Evans *et al.*[11] have obtained finite volume solutions to the Euler equations on quadtree grids.

This paper presents the quadtree algorithm, adaptive remeshing, finite volume discretization and solution of the Navier–Stokes equations using Patankar's[12] SIMPLE technique, and validation of the scheme for low-Reynolds-number separated flows.

### *2.1. Glossary of Terms*

| | |
|---|---|
| *seeding points* | points about which the quadtree grid is generated |
| *second-generation neighbour* | adjacent panel that has a division level of two less or greater than the panel in question |
| *regularized mesh* | mesh in which no adjacent edge ratio exceeds two |
| *neighbour finding* | determination of panels having face and/or corner adjacency to the panel in question |
| *tree searching* | traversal of the tree data structure to locate a given panel reference |
| *parent* | panel which has been divided to produce children |
| *children* | panels resulting from division of parent |
| *ancestor* | larger panel which encompasses the panel in question |
| *binary transform* | function transforming 1 to 2 and 2 to 1 |
| *leaf* | undivided panel |
| *neighbourhood* | set of reference numbers of a panel's face and corner neighbours |
| *nearest common ancestor* | the smallest panel which is an ancestor of the given panels |
| *hanging node* | an element's vertex located at the centre of an adjacent element's face. |

## 2. QUADTREE GRID GENERATION

The steps required to generate a quadtree grid for a maximum division level equal to $M_{\max}$ may be summarized as follows.

1. Define the set of seeding points $P_n$ about which the mesh will be generated.
2. Define a unit square or rectangle (root panel) which surrounds the normalized domain of interest.
3. Divide the root panel into four quadrant panels.
4. Consider each panel in turn; if the panel contains a point, continue with step 5; otherwise check the next panel.
5. Check whether the maximum division level $M_{\max}$ has been reached. If so, the division of the panel in question is complete, so go to step 4 and check the next panel. When all panels considered have reached the maximum division $M_{\max}$, the mesh generation is complete. Otherwise continue.
6. Divide the panel into four congruent panels, return to step 4 and check the next panel.

It is desirable to generate additional panels such that the maximum panel edge length ratio between two adjacent panels does not exceed two. This limits the variation of discretization arrangements to a manageable number. The regularization procedure determines hypothetical second-generation face neighbours for each panel, searches the tree to see if they exist within the mesh, further divides the panel if they do and so yields a *regularized* mesh.

The quadtree algorithm is illustrated by the following example of its application to a simple seeding problem. Consider the unit square domain shown in Figure 1, containing three points about which a fine mesh is required. The domain is initially divided into four subregions or panels as shown in Figure 2. Each panel is then considered in turn: if the panel contains a point, then it is subdivided to create four new panels; if not, then the panel is not divided further. This results in the mesh shown in Figure 3. Each of the new panels is then considered in turn and divided if it contains a point, and so the algorithm is repeated until the required resolution is achieved. In this case, if the maximum number of divisions required equals four, then the mesh produced is that shown in Figure 4. The simple quadtree algorithm produces a selectively refined Cartesian mesh with a recognizable hierarchical tree structure. Figure 5 illustrates the tree structure corresponding to the mesh in Figure 4. After regularization the final mesh is as shown in Figure 6.

The numbering system used here is based on that described by Van Dommelen and Rundensteiner.[4] The reference number $N$ of a given panel is made up of a sequence of successive integer translations at each generation level, which contain information regarding the position of the panel and by manipulation lead to neighbour finding within the grid. A quadtree panel in a regularized grid has eight possible neighbours of the same size and 12 possible neighbours one level smaller in size. Face neighbours, with reference number $NF$, of a panel lie adjacent to it and have a common side. Corner neighbours, with reference number $NC$, share a common corner with the panel. Ancestral second-generation face neighbour panels, with reference number $NSG$, have an edge length four times the size of a given panel and the two panels will be two generation levels apart.

For the test cases considered in this paper, the points $P_n$ lie on the boundary of a square or circular cylinder, and so a boundary-generated base mesh is created which may be adapted with further divisions during the solution.

### 2.1. Numbering system

The panel reference number $N$ is essentially a list of binary translations describing the position of the panel. Herein, four arrays of eight digits have been used to represent the reference number, giving

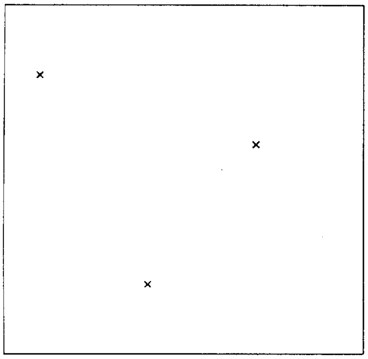

Figure 1. Seed points

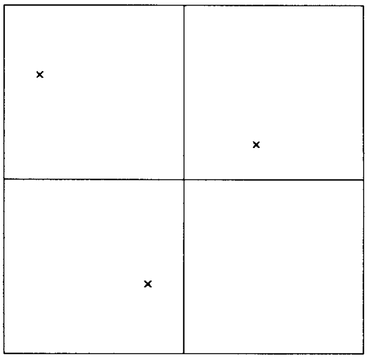

Figure 2. Grid after first-level subdivision

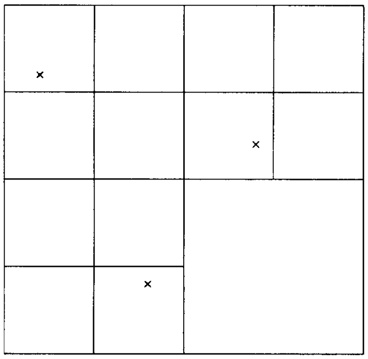

Figure 3. Grid after second-level subdivision

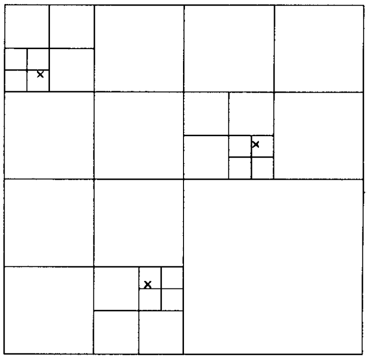

Figure 4. Grid after fourth-level subdivision

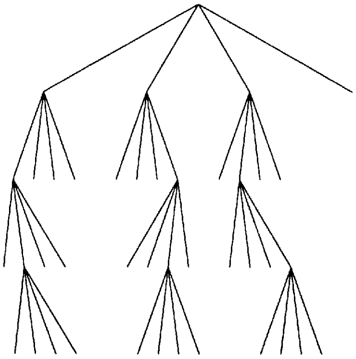

Figure 5. Tree structure

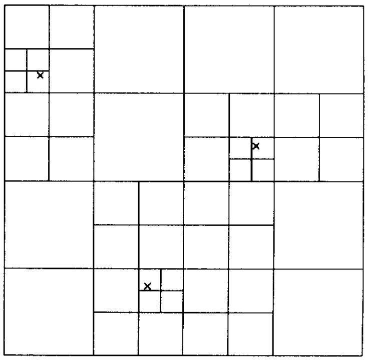

Figure 6. Regularized grid

a total of 32 digits corresponding to a maximum of 16 subdivisions of the root panel. To each binary digit $+1$ is added so that no zeros are contained in the significant part of the reference number, and trailing zeros are appended to give the required 32 digits. As division occurs, the positions of each of the four new panels produced are described by an $x$ and $y$ binary translation 0 or 1; these digits are converted to base two and appended to the number of the *parent* panel being divided to produce the number of each *child* or new panel. Hence the reference number of a given panel contains in reverse order the reference numbers of its parent and all its ancestor panels.

Translations at each level correspond to a separate co-ordinate system for each divided panel. The local numbering of new panels at an arbitrary level is shown in Figure 7. Using this approach, the binary path of panel B in Figure 8 is 101101, and so its reference number is $N_B = (21221200, 00000000, 00000000, 00000000)$. From the truncated reference number (which has trailing zeros removed) the following information can be deduced.

*Generation level.* The number of divisions of the original root panel required to produce a panel of given size is called the generation level of the panel, $M$, which is equal to the number of non-zero digits in $N$ divided by two. For example, the generation level of panel B in Figure 8 is equal to 3.

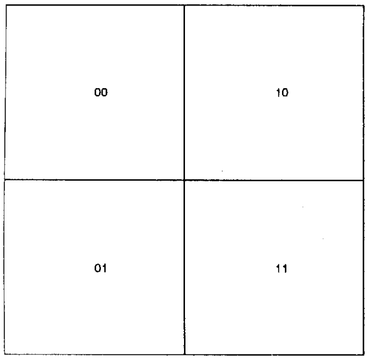

Figure 7. Quadrant order and binary representation

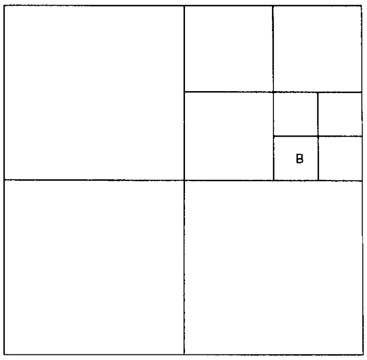

Figure 8. Example grid

*Parent panel.* The reference number of the parent panel of a given panel is found by converting the last two non-zero digits of $N$ to zero. For example, the parent panel of panel B in Figure 8 has reference number (21220000, 00000000, 00000000, 00000000).

*Centre co-ordinates.* The $x$- and $y$-co-ordinates at the centre of the panel are calculated by successive translations, starting from the centre co-ordinates of the root panel. At each level $M$ the panel edge length is equal to $2^{-M}$. This calculation is expressed as

$$x = \sum_{m=1}^{M} \frac{1}{2^{m+1}}(2N_x(m) - 3), \tag{1}$$

$$y = \sum_{m=1}^{M} \frac{1}{2^{m+1}}(3 - 2N_y(m)), \tag{2}$$

where $N_x(m)$ and $N_y(m)$ are the $m$th and $(m+1)$th integer values in $N$ if $N$ is viewed as a concatenation of 32 integers. Thus, for panel B in Figure 8, $N_x(1) = 2$, $N_x(2) = 2$, $N_x(3) = 1$, $N_y(1) = 1$, $N_y(2) = 2$ and $N_y(3) = 2$, and so, using equations (1) and (2), the co-ordinates at the centre of panel B are (0·3125, 0·0625).

*Neighbour finding.* All possible neighbours of any panel can be determined from its reference number $N$. This is a useful attribute when regularizing the mesh and for solving the discretized equations using mesh sweeps. Once the complete set of reference numbers of a panel's potential neighbours has been determined, the tree is searched to identify actual neighbours within the grid.

The tree search starts from the nearest common ancestor of a given panel and its neighbour and ends when a leaf is reached. The reference number *NCA* of the nearest common ancestor shared by two panels is given by consecutive digits common to both panels' reference numbers. Thus, if digits at level $m$ were modified in order to construct a given neighbour reference number, then *NCA* is obtained by discarding digits at levels $\geqslant m$.

Reference numbers increase down through the generations of a family, so that none of the descendants of a given panel can have a reference number greater than that of the panel's next *sister* panel (where sister panels have the same generation level). If a panel P, of generation level $M$, is required and its reference number is greater than that of panel J but less than that of the next panel K, each of a generation level less than $M$, it follows that panel J is an ancestor of P and the children of J should be searched next.

General rules for neighbour finding can be summarized as follows.

1. Break down the reference number into concatenated integer components ($N_x$ and $N_y$).
2. Calculate $\lim x$ and $\lim y$; working backwards through $N_x$ and $N_y$, these are the levels at which $N_x(i)$ first becomes not equal to $N_x(i-1)$ and $N_y(i)$ first becomes not equal to $N_y(i-1)$ respectively.
3. Perform *binary transforms* on $N_x$ and $N_y$ as illustrated in the detailed example below. (In this context, binary transforms are transposes of the digits 1 and 2.)
4. Construct the neighbour reference number by recombining the $N_x$ and $N_y$ components and adding trailing zeros.
5. Determine *NCA* by discarding digits that are not common to both the panel and its neighbour reference numbers.
6. Search the tree from *NCA* until a leaf is found.

In the following example, reference numbers are determined for the four face neighbours and four corner neighbours of the panel marked D in Figure 9, which has a truncated reference number equal to 211221. The reference number is decomposed into $N_x = (2, 1, 2)$ and $N_y = (1, 2, 1)$, and so $\lim x = 3$ and $\lim y = 3$. Each panel may have two face neighbours and one corner neighbour which share the same parent panel. In Figure 9, panel D has the same parent as its west and south face neighbours and its southwest corner neighbour. The reference number of the west face neighbour is determined by performing a binary transform on $N_x(M)$ and recombining the $N_x$ and unaltered $N_y$ components, thus $NF_{\mathrm{W}} = (21121100, 00000000, 00000000, 00000000)$. The reference number for the east face neighbour is created by performing binary transforms on $N_x(\lim x - 1)$ to $N_x(M)$, giving $NF_{\mathrm{E}} = 21221100, 00000000, 00000000, 00000000)$. Reference numbers for the north and south face neighbours are created in a similar fashion, but by performing binary transforms on the corresponding $N_y$ components, i.e. on $N_y(M)$ for the south neighbour and then on $N_y(\lim y - 1)$ to $N_y(M)$ for the north neighbour, giving $NF_{\mathrm{S}} = (21122200, 00000000, 00000000, 00000000)$ and $NF_{\mathrm{N}} = (21112200, 00000000, 00000000, 00000000)$. The southwest corner neighbour is created by binary transforms on $N_x(M)$ and $N_y(M)$, giving the number $NC_{\mathrm{SW}} = (21121200, 00000000, 00000000, 00000000)$. The reference number for the southeast corner neighbour is created by carrying out binary transforms on $N_x(\lim x - 1)$ to $N_x(M)$ and on $N_y(M)$, giving $NC_{\mathrm{SE}} = (21221200, 00000000, 00000000, 00000000)$. The northwest corner neighbour is created by binary transforms on $N_x(M)$ and on $N_y(\lim y - 1)$ to $N_y(M)$, giving $NC_{\mathrm{NW}} = (21111200, 00000000, 00000000, 00000000)$. Finally, the reference number for the northeast corner neighbour is created by binary transforms on $N_x(\lim x - 1)$ to $N_x(M)$ and on

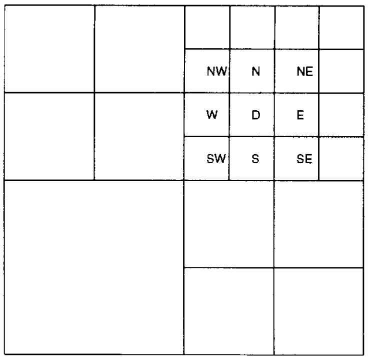

Figure 9. Neighbours of cell D

$N_y(\lim y - 1)$ to $N_y(M)$, giving $NC_{\mathrm{NE}} = (21211200, 00000000, 00000000, 00000000)$. $NCA$ for panel D and its face neighbours W, S and SW is determined by comparing $N$ with $NF_{\mathrm{W}}$, $NF_{\mathrm{S}}$ and $NC_{\mathrm{SW}}$ in turn. In each case the common consecutive digits result in $NCA = (21120000, 00000000, 00000000, 00000000)$. Similarly, by comparing $N$ with $NF_{\mathrm{E}}$ (or $NF_{\mathrm{W}}$), the reference number of the nearest common ancestor shared between panels D and E (or N) is $NCA = (21000000, 00000000, 00000000, 00000000)$.

Consider panel B with $N = (21122100, 00000000, 00000000, 00000000)$. Let us conduct a tree search for a panel with reference number $NF_{\mathrm{N}} = (21112200, 00000000, 00000000, 00000000)$ in order to check whether it exists as a leaf. Starting from the nearest common ancester with $NCA = (21000000, 00000000, 00000000, 00000000)$, the reference numbers of its children are checked and $NF_{\mathrm{N}}$ is found to lie between (21110000, 00000000, 00000000, 00000000) and (21120000, 00000000, 00000000, 00000000). The children of (21110000, 00000000, 00000000, 00000000) are then checked. In this case the reference number of one of the children does match $NF_{\mathrm{N}}$, which is not further divided and therefore corresponds to a leaf, and so the tree search is terminated.

## 3. FINITE VOLUME DISCRETIZATION OF GOVERNING EQUATIONS

The governing equations in primitive variable form for a two-dimensional incompressible flow without body forces are the mass conservation equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3}$$

and the Navier–Stokes momentum conservation equations

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial vu}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{4}$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \tag{5}$$

where $x$ and $y$ define an orthogonal Cartesian co-ordinate system, $u$ and $v$ are the corresponding velocity components, $t$ is time, $p$ is pressure, $\rho$ is the fluid density and $\nu$ is the fluid kinematic viscosity.

The governing equations are discretized using finite volumes with collocated primitive variables and solved by a pressure correction method based on that described by Perić *et al.*[13] and Hortmann *et al.*[14] for regular grids. In essence, the scheme is similar to SIMPLE,[12] but applied to hierarchical grids with interpolations to deal with the non-staggered arrangement of variables.

Figure 10 shows a typical cell with pressure and velocity components located together at its centre. The cell is a control volume of length d$x$ and width d$y$. In the discretized equations, subscript P refers to values belonging to the cell centre, subscripts E, W, N and S denote neighbouring cell centres, subscripts e, w, n and s correspond to the cell faces and subscript Po represents central cell values from the previous time step. Both momentum conservation equations are integrated over the control volume and rearranged to give

$$a_{\mathrm{P}}u_{\mathrm{P}} = \sum a_{\mathrm{nb}}u_{\mathrm{nb}} + a_{\mathrm{Po}}u_{\mathrm{Po}} + (p_{\mathrm{w}} - p_{\mathrm{e}})\,\mathrm{d}y, \tag{6}$$

$$a_{\mathrm{P}}v_{\mathrm{P}} = \sum a_{\mathrm{nb}}v_{\mathrm{nb}} + a_{\mathrm{Po}}v_{\mathrm{Po}} + (p_{\mathrm{s}} - p_{\mathrm{n}})\,\mathrm{d}x, \tag{7}$$

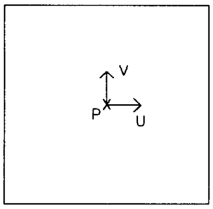

Figure 10. Collocated variables

where

$$\sum a_{\mathrm{nb}} u_{\mathrm{nb}} = a_{\mathrm{e}} u_{\mathrm{E}} + a_{\mathrm{w}} u_{\mathrm{W}} + a_{\mathrm{n}} u_{\mathrm{N}} + a_{\mathrm{s}} u_{\mathrm{S}}, \tag{8}$$

$$\sum a_{\mathrm{nb}} v_{\mathrm{nb}} = a_{\mathrm{e}} v_{\mathrm{E}} + a_{\mathrm{w}} v_{\mathrm{W}} + a_{\mathrm{n}} v_{\mathrm{N}} + a_{\mathrm{s}} v_{\mathrm{S}}. \tag{9}$$

The coefficients $a$ combine convective and diffusive fluxes across control volume faces in accordance with Patankar's[12] power-law scheme. For collocated variables the coefficients $a$ are identical for both momentum equations because they share the same control volume. The expressions for the coefficients used herein are given by

$$a_{\mathrm{e}} = D_{\mathrm{e}} [\![ 0, (1 - 0{\cdot}1|P_{\mathrm{e}}|)^5 ]\!] + [\![ -F_{\mathrm{e}}, 0 ]\!], \tag{10}$$

$$a_{\mathrm{w}} = D_{\mathrm{w}} [\![ 0, (1 - 0{\cdot}1|P_{\mathrm{w}}|)^5 ]\!] + [\![ F_{\mathrm{w}}, 0 ]\!], \tag{11}$$

$$a_{\mathrm{n}} = D_{\mathrm{n}} [\![ 0, (1 - 0{\cdot}1|P_{\mathrm{n}}|)^5 ]\!] + [\![ -F_{\mathrm{n}}, 0 ]\!], \tag{12}$$

$$a_{\mathrm{s}} = D_{\mathrm{s}} [\![ 0, (1 - 0{\cdot}1|P_{\mathrm{s}}|)^5 ]\!] + [\![ F_{\mathrm{s}}, 0 ]\!], \tag{13}$$

$$a_{\mathrm{P}} = a_{\mathrm{e}} + a_{\mathrm{w}} + a_{\mathrm{n}} + a_{\mathrm{s}} + \rho \Delta x \Delta y / \Delta t, \tag{14}$$

where the double brackets indicate taking the maximum of the values contained therein. The diffusive fluxes $D_i$, convective fluxes $F_i$, and Peclet numbers $P_i$ are defined as

$$D_{\mathrm{e}} = \mu A_{\mathrm{e}} / \delta x_{\mathrm{e}}, \qquad F_{\mathrm{e}} = \rho u_{\mathrm{e}} A_{\mathrm{e}}, \qquad P_{\mathrm{e}} = F_{\mathrm{e}} / D_{\mathrm{e}}, \tag{15}$$

$$D_{\mathrm{w}} = \mu A_{\mathrm{w}} / \delta x_{\mathrm{w}}, \qquad F_{\mathrm{w}} = \rho u_{\mathrm{w}} A_{\mathrm{w}}, \qquad P_{\mathrm{w}} = F_{\mathrm{w}} / D_{\mathrm{w}}, \tag{16}$$

$$D_{\mathrm{n}} = \mu A_{\mathrm{n}} / \delta y_{\mathrm{n}}, \qquad F_{\mathrm{n}} = \rho v_{\mathrm{n}} A_{\mathrm{n}}, \qquad P_{\mathrm{n}} = F_{\mathrm{n}} / D_{\mathrm{n}}, \tag{17}$$

$$D_{\mathrm{s}} = \mu A_{\mathrm{s}} / \delta y_{\mathrm{s}}, \qquad F_{\mathrm{s}} = \rho v_{\mathrm{s}} A_{\mathrm{s}}, \qquad P_{\mathrm{s}} = F_{\mathrm{s}} / D_{\mathrm{s}}, \tag{18}$$

where $\mu$ is the dynamic viscosity and $\rho$ is the density. $A_i$ is the $i$th face area for the control volume and $\delta_i$ is the distance between the centre of the control volume and its neighbour in the relevant $i$-direction. It should be noted that evaluating the convective fluxes independently would not necessarily conserve these quantities.

Using linear interpolation to obtain the cell face pressures, the discretized momentum equations become

$$a_P u_P = \sum a_{nb} u_{nb} + a_{Po} u_{Po} + \frac{(p_W - p_E)\,dy}{2}, \tag{19}$$

$$a_P v_P = \sum a_{nb} v_{nb} + a_{Po} v_{Po} + \frac{(p_S - p_N)\,dx}{2}. \tag{20}$$

Here the pressure gradient in the momentum equations has effectively been discretized using first-order central differences, which, if used for all equations, could give a solution independent of the central pressure $p_P$. To prevent such a checkerboard effect occurring, the cell face velocities were determined using a special interpolation procedure. Consider the east face velocity $u_e$ for example. With reference to Figure 11, the discretized $u$-momentum equations for nodes P and E are given by

$$u_P = \left(\frac{\sum a_{nb} u_{nb} + a_{Po} u_{Po} + (p_w - p_e)\,dy}{a_P}\right)_P, \tag{21}$$

$$u_E = \left(\frac{\sum a_{nb} u_{nb} + a_{Po} u_{Po} + (p_w - p_e)\,dy}{a_P}\right)_E. \tag{22}$$

The cell face velocity $u_e$ is obtained by linear interpolation of the terms involving velocities in the equations above, and combined with a pressure gradient centred on the face velocity. Hence

$$u_e = \left(\overline{\frac{\sum a_{nb} u_{nb} + a_{Po} u_{Po}}{a_P}}\right)_e + \left(\overline{\frac{1}{a_P}}\right)_e (p_P - p_E)\,dy, \tag{23}$$

where the overbar indicates linear interpolation. Thus the cell face velocity depends on the pressure gradient across the cell face.

The discretized continuity equation is

$$\rho(u_e - u_w)\,dy + \rho(v_n - v_s)\,dx = 0. \tag{24}$$

Let the corrected values of pressure and velocity be the sum of a continuously updated guessed value (denoted by superscript 'asterisk') and its correction (denoted by superscript 'prime'):

$$p = p^* + p', \qquad u = u^* + u', \qquad v = v^* + v'. \tag{25}$$

Neglecting the neighbour velocity correction terms, the $u$- and $v$-velocity may be expressed as

$$u_P = u_P^* + \left(\frac{p'_W - p'_E}{2}\right)\frac{dy}{a_P}, \tag{26}$$

$$v_P = v_P^* + \left(\frac{p'_S - p'_N}{2}\right)\frac{dx}{a_P}. \tag{27}$$

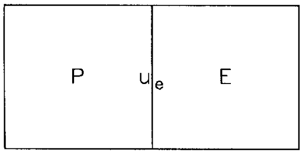

Figure 11. Cell face velocity

After substitution of the equivalent velocity correction formulae for the four face velocities into the discretized continuity equation, the following Poisson-type equation for the pressure corrections is obtained:

$$a_{\mathrm{pP}}p_{\mathrm{P}} = a_{\mathrm{pE}}p'_{\mathrm{E}} + a_{\mathrm{pW}}p'_{\mathrm{W}} + a_{\mathrm{pN}}p'_{\mathrm{N}} + a_{\mathrm{pS}}p'_{\mathrm{S}} = b, \tag{28}$$

where the neighbour coefficients are again given by (10)–(14). The term $b$ represents a mass source which approaches zero as the velocity and pressure corrections successively improve the corrected values, and is given by

$$b = \rho(u^*_{\mathrm{w}} - u^*_{\mathrm{e}})\,\mathrm{d}y + \rho(v^*_{\mathrm{s}} - v^*_{\mathrm{n}})\,\mathrm{d}x. \tag{29}$$

In cases where the adjoining cells have different sizes in the quadtree grid, *hanging nodes*, which lie at the midside of an adjacent cell, will occur and the neighbour contributions to (4) and (5) have to be calculated by interpolation. A weighted average of the smaller cell values is used if the neighbour cell is smaller than the central cell; alternatively, linear extrapolation is used if the neighbour cell is larger than the central cell. Further details of the interpolation scheme are given in Section 5 and by Greaves.[15]

At each time step a single iteration on the momentum equations (8) and (9) is first performed using current estimates of velocity and pressure; next the pressure correction equation (17) is solved using a maximum of six inner iterations, which were found to be sufficient for convergence; the calculated pressure corrections (14) are then used to update the pressure and velocity fields (15) and (16). This completes one outer iteration, which is repeated until the sum of normalized residuals for all variables has converged to less than 0·01. The calculation then proceeds to the next time step.

## 4. ADAPTIVE MESH REFINEMENT

The base quadtree grid (originally determined by seeding points lying on the cylinder boundary) is adapted throughout the simulation in order to provide maximum resolution in areas of rapidly changing flow. This is achieved through the addition of new panels in regions of high absolute vorticity and the removal of panels not included in the base grid in regions of low absolute vorticity.

Maximum and minimum vorticity limits are specified in advance, and after each time step calculation the vorticity in each panel is calculated. If the absolute vorticity in a given panel is greater than the prescribed maximum, then that panel is divided, whereas if the absolute vorticity is less than the prescribed minimum and the panel is not intrinsic to the base grid, then that panel may be removed. In fact, panels are only removed in groups of four, when all four share a single parent and all have values of absolute vorticity less than the prescribed minimum. Hence the mesh adapts at each time step, providing selective refinement in regions of high vorticity. In this way the grid structure mimics vortical features of the flow. It should be noted that the hierarchical nature of quadtree grids is ideally suited to adaptive remeshing, because the addition/removal of leaf panels does not disrupt the tree data structure.

Once remeshing is complete, the grid is again regularized by performing further divisions to ensure that the maximum panel edge ratio is 2:1. The velocity and pressure values at newly generated panels are then interpolated from neighbouring panels. Finally, the mesh is reordered to eliminate removed panels from the tree and to ensure that no gaps occur in the list of panels, and the flow computation then proceeds to the next time step.

## 5. RESULTS

Convergence tests are initially presented for simulation of potential flow past an approximately circular cylinder. In order to verify the grid-adaptive numerical model, three unidirectional, impulsively started flows are then considered, namely flow past a circular cylinder at $Re = 200$ and 5000 and flow past a square cylinder at $Re = 250$. Finally, simulations of flow past a backward-facing step are presented and comparisons drawn between adaptive grid results and those obtained on a globally refined grid.

For flow past cylinders the Reynolds number $Re$ is defined by

$$Re = U_o d/v, \tag{30}$$

where $U_o$ is the inflow velocity, $d$ is the cylinder diameter (or width for the square cylinder) and $v$ is the kinematic viscosity of the fluid. Herein the characteristic dimension of the cylinder is $d = 0{\cdot}02344\,\mathrm{m}$, the inflow incident velocity is $U_o = 0{\cdot}02344\,\mathrm{m\,s^{-1}}$, the fluid density is $\rho = 1000\,\mathrm{kg\,m^{-3}}$ and the fluid kinematic viscosity $v$ is assigned according to the required Reynolds number. The normalized evolution time $t^*$ is defined as

$$t^* = tU_o/d, \tag{31}$$

where $t$ is the time from the impulsive start.

The lay-out of the computational domain is the same in all cases. The characteristic dimension of the cylinder, $d$, is approximately 42 times smaller than the overall length of the domain (equal to 1 m), and the cylinder is positioned such that the inflow boundary is $11d$ upstream of the centre of the cylinder. At each time step the calculation was considered to have converged when the sum of normalized residuals fell below $5{\cdot}0 \times 10^{-4}$. All simulations were performed on a SUN SPARC 10 workstation.

### 5.1. *Potential flow past a circular cylinder*

Unidirectional potential flow past a circular cylinder has an analytical solution and is used to test the convergence of the quadtree grid-based scheme. Solutions are obtained on grids having various levels of maximum resolution for comparison purposes. The governing equations are solved in irrotational form by setting the dynamic viscosity coefficient $\mu$ to zero in the momentum equations, and 'slip' boundary conditions are used at the cylinder wall. The analytical solution for potential flow past a circular cylinder predicts that the pressure distribution around the cylinder is symmetric so that the resultant drag and lift force components are both zero, in accordance with D'Alembert's paradox.

Quadtree grids of maximum division levels 9, 10, 11 and minimum level 5, and of maximum division level 12 and minimum level 6 were tested. Figure 12 shows a portion of the level 10 grid close to the cylinder. The grids have a stepped profile which approximates the curved boundary of the circular cylinder in a fractal-like manner; clearly the surface roughness reduces for grids generated to higher division levels.

The number of cells and CPU time required for convergence using each grid are summarized in Table I, as well as drag and lift coefficients for the four calculations. Comparison of the drag coefficients obtained suggests that the solutions converge as the grid resolution is increased, although comparison of the lift coefficients obtained is not conclusive.

### 5.2. *Unidirectional flow past a circular cylinder at Re = 200*

The early stages of impulsively started laminar flow past a circular cylinder consist of flow separation followed by the development of a symmetric pair of Föppl vortices which elongate with

Table I. Convergence test for potential flow simulation showing variation in drag and lift coefficients with increasing grid resolution at cylinder

| *Maximum grid resolution* | *Number of cells* | *CPU time (s)* | *Drag coefficient* $C_D$ | *Lift coefficient* $C_L$ |
|---|---|---|---|---|
| 9 levels | 1282 | 1969 | 0·33530 | $5{\cdot}99052 \times 10^{-2}$ |
| 10 levels | 1464 | 4143 | 0·13267 | $9{\cdot}34512 \times 10^{-3}$ |
| 11 levels | 1818 | 11031 | $6{\cdot}11832 \times 10^{-2}$ | $-1{\cdot}20148 \times 10^{-2}$ |
| 12 levels | 5578 | 40997 | $5{\cdot}76151 \times 10^{-4}$ | $-1{\cdot}39376 \times 10^{-2}$ |

time. At Reynolds numbers greater than approximately 80 the downstream stagnation streamline between the vortices begins to waver and the instability grows until a vortex detaches from the cylinder and regular vortex shedding commences. Eventually, limit cycle conditions are reached where vortices shed repeatedly from alternate sides of the body and an unsteady periodic wake forms. Here results are compared with experimental visualizations obtained by Bouard and Coutanceau[16] and numerical data provided by Franke *et al.*[17] for impulsively started flow past a circular cylinder at $Re = 200$.

Figure 12 shows a portion of the initial quadtree computational grid containing a total of 1464 panels generated about a circle of seeding points. The grid has a maximum division level $M_{\max} = 10$ at the boundary of the cylinder, is regularized and has extra grid uniformity imposed to a resolution level of 5. Initial values of all variables are zero everywhere to provide an impulsive start, and boundary conditions of non-slip velocity and zero normal pressure gradient are applied at the cylinder surface. A non-dimensional time step $\Delta t^* = 0{\cdot}025$ satisfied the Courant condition. During the flow simulation the quadtree grid was adapted every 20 time steps, with individual panels subdivided if their non-dimensional vorticity magnitude was greater than 1·0 and their division level was less than 9, or groups of four replaced by their parent if the vorticity magnitude fell below 3·0 and the division level was greater than 8.

Figure 13 illustrates a portion of the adapted grid at normalized time $t^* = 2{\cdot}5$ when a symmetric pair of vortices is beginning to develop behind the cylinder. Extra mesh resolution is evident in the near wake behind the cylinder; the total number of grid panels is 1644. The relationship between the

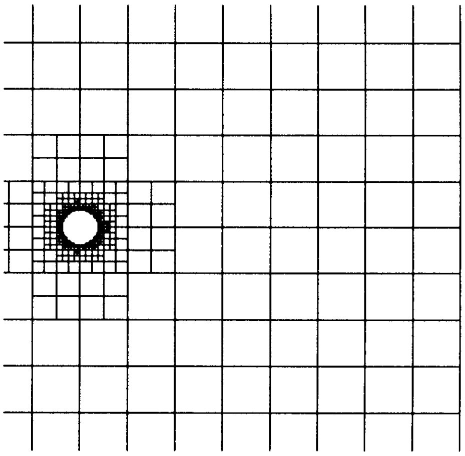

Figure 12. Initial grid, $Re = 200$, and potential flow (level 10)

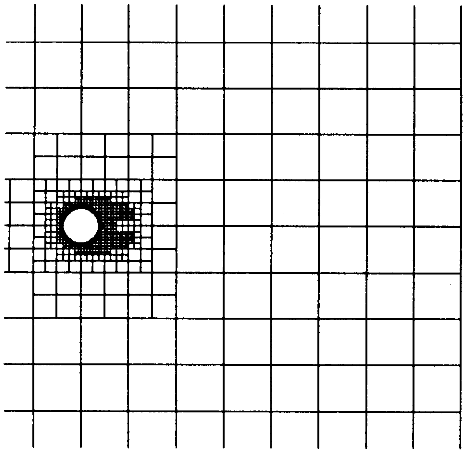

Figure 13. Adapted grid, $t^* = 2{\cdot}5$

locally resolved region of the adapted grid and the vorticity is evident by comparing Figure 13 with the vorticity contours plotted in Figure 14. From the stream function contours in Figure 15 it is clear that the reattachment length is approximately equal to the diameter of the cylinder. The flow pattern indicated by the streamlines and the position of separation points are in close agreement with Bouard and Coutanceau's[16] experimental results. The slightly discontinuous contours in the figures are due to the plotting routine.

Figure 16 shows a portion of the adapted grid at a later time $t^* = 9{\cdot}675$; the locally resolved part of the mesh has grown in conjunction with the elongated vortex pair shown in Figure 17 and contains 2804 panels. The streamlines plotted in Figure 18 show that the vortices are asymmetric and are beginning to waver. At this stage the calculation required 1·017 s of CPU time per iteration and 8552 iterations to converge.

For the present simulation, vortex shedding established itself without external forcing, being initiated by the accumulation of rounding errors in the solution. A typical adapted grid (of 2622 cells) obtained during a vortex-shedding cycle is given in Figure 19 at $t^* = 152{\cdot}063$; 11,462 iterations at 0·512 s of CPU time per iteration were required for convergence. Again the locally refined mesh bears a striking resemblance to the regions of high vorticity portrayed in Figure 20. The corresponding streamlines are given in Figure 21; an attached primary anticlockwise vortex is

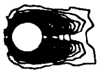

Figure 14. Vorticity contours

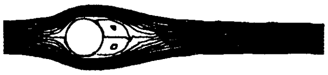

Figure 15. Stream function contours

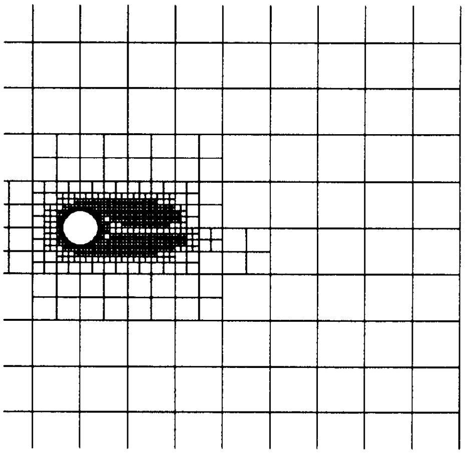

Figure 16. Adapted grid, $t^* = 9{\cdot}7$

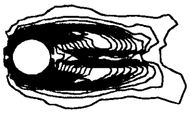

Figure 17. Vorticity contours

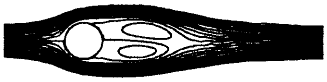

Figure 18. Stream function contours

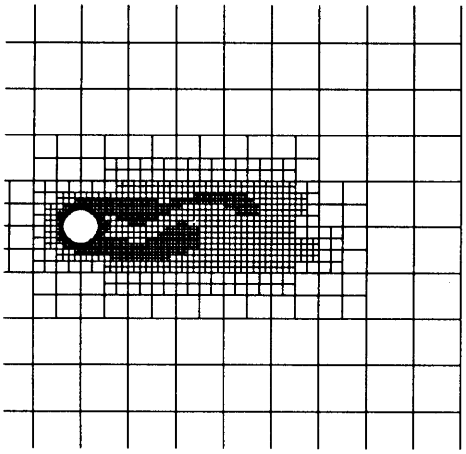

Figure 19. Adapted grid, $t^* = 152\cdot1$

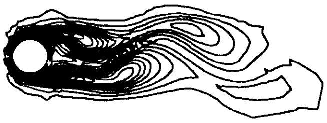

Figure 20. Vorticity contours

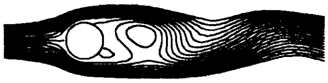

Figure 21. Stream function contours

developing at the lower rear of the cylinder and a recently detached clockwise vortex is visible in the near wake approximately one cylinder diameter downstream of the cylinder. In general the flow patterns calculated using the present method were very similar to those published by Franke *et al.*[17] throughout the vortex-shedding cycle. However, the far wake is narrower than that predicted by Franke *et al.*[17] This discrepancy may be due to numerical diffusion or to roughness from the stepped boundary approximation.

### *5.3. Unidirectional flow past a circular cylinder at $Re = 5000$*

The early stages of flow past a circular cylinder at $Re = 5000$ were simulated for comparison with flow visualizations by Bouard and Coutanceau.[16] In this case the initial grid, shown in Figure 22, has a maximum division level equal to 11 and a base uniform grid level equal to 6. Close examination of the surface of the cylinder indicates that it is significantly rough, with a maximum step height approximately equal to $\frac{1}{48}$th of its radius. The initial and boundary conditions were identical to the above case. Throughout the flow simulation the grid was adapted every 20 time steps.

Figures 23–25 depict the adapted grid, vorticity contours and streamlines obtained using the present model at $t^* = 2\cdot5$. The streamlines agree reasonably with the flow pattern in Figure 26, visualized experimentally by Bouard and Coutanceau.[16] A kink is clearly seen in the exterior boundary of the recirculating zone, though the secondary eddies obtained experimentally by Bouard and Coutanceau[16] are not predicted using the present scheme, indicating insufficient mesh resolution in the immediate vicinity of the cylinder. Also, the width of the wake is narrower than that produced experimentally. As before, discrepancies in the flow features close to the cylinder may have arisen from its stepped surface causing additional separation, whereas those in the wake may be due to

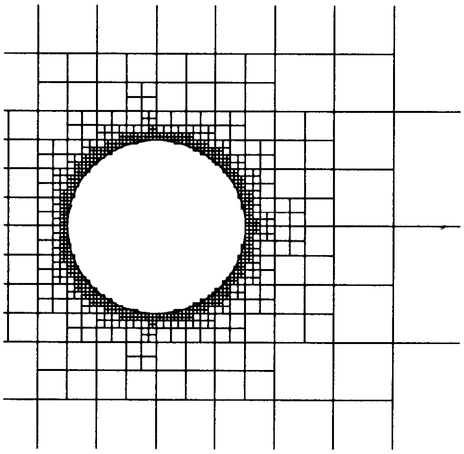

Figure 22. Initial grid, $Re = 5000$

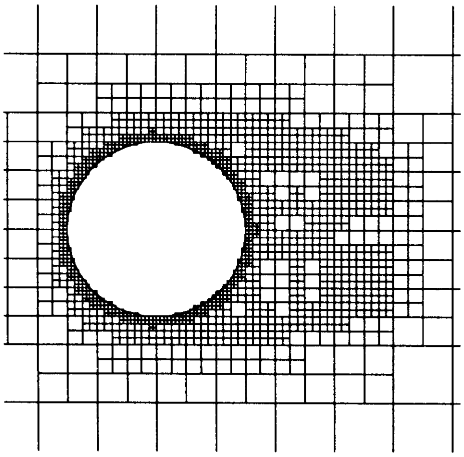

Figure 23. Adapated grid, $t^* = 2{\cdot}5$

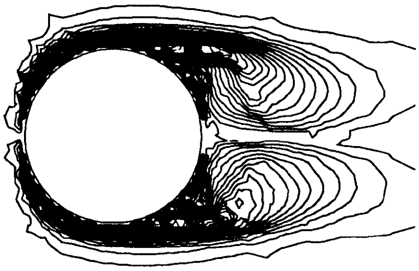

Figure 24. Vorticity contours

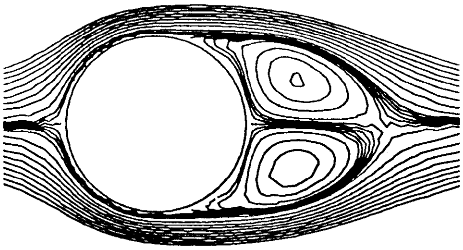

Figure 25. Stream function contours

numerical diffusion. It should be noted that although the present method predicts the commencement of impulsively started, separated, fully laminar flow at $Re = 5000$, a numerical model which includes turbulence would be necessary to simulate properly the longer-term flow features.

### *5.4. Unidirectional flow past a square cylinder at $Re = 250$*

Results are presented here for vortex shedding from a square cylinder in unidirectional flow at $Re = 250$. The initial grid is shown in Figure 27 and consists of a total of 1696 panels generated from seeding points representing the boundary of the square cylinder; the lowest level of resolution is 5 and $M_{\max} = 10$. Maximum and minimum levels of grid resolution for adaptation are 10 and 9 respectively. As before, the simulation started from rest, with velocities and pressure equal to zero everywhere. The non-dimensional time step is $\Delta t^* = 0{\cdot}025$ and the mesh was adapted every 40 time steps. Once vortex shedding was established, a typical adapted grid contained 7234 panels and required 6368 iterations for convergence, with 3·76 s of CPU time taken per iteration.

Figure 28 illustrates a portion of the adapted grid at $t^* = 124{\cdot}0$. Figures 29 and 30 show the corresponding vorticity and stream function contours. The presence of detached vortices may be clearly discerned in both the adapted grid and the vorticity contour plot. The streamline pattern in Figure 30 demonstrates that separation is occurring from the front corners of the square cylinder. A

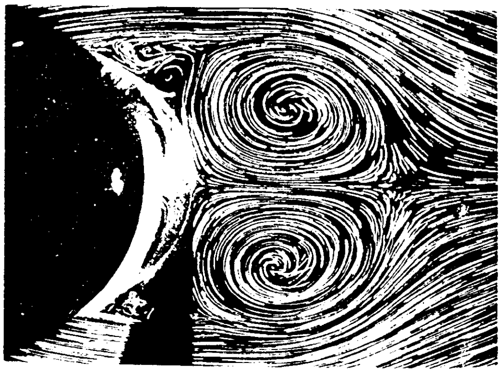

Figure 26. Experimental results of Bouard and Coutanceau[16]

nascent vortex is developing along the cylinder's lower side and a recently detached clockwise vortex can be seen in the downstream wake. It should be noted that the sharp corners of the square are singular points in the flow. Although the meshing is refined close to the boundary, the corner flow cannot be predicted properly. Even so, the results agree favourably with visualized experimental flow patterns and theoretical predictions produced by Okajima[18] and Davis and Moore.[19] However, the wake width predicted by the present method appears slightly narrower than that predicted by Okajima.[18]

### *5.5. Backward-facing step*

The numerical model was used to simulate laminar flow past a backward-facing step at $Re = Uh/v = 50$. Here $U$ is the mean velocity across the inlet, $h$ is the step height and $v$ is the fluid kinematic viscosity. The upstream boundary is located one step dimension $h$ in front of the expansion, the channel width is $2h$ and the overall channel length is $14h$. A parabolic velocity profile is specified at the upstream boundary. Flow simulations were undertaken for three grid configurations, namely a base unadapted quadtree grid, an adapted grid and an equivalent globally refined grid. The base and

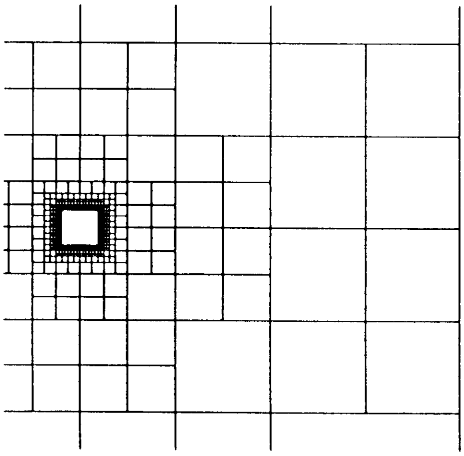

Figure 27. Initial grid, $Re = 250$

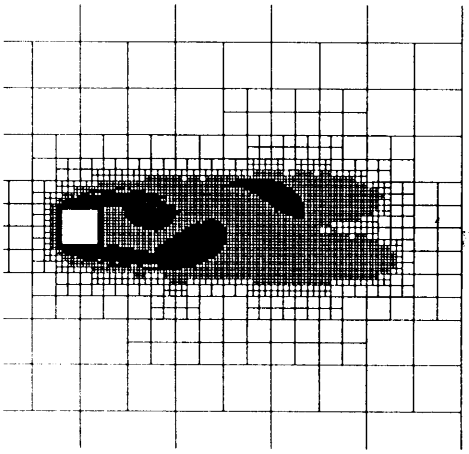

Figure 28. Adapted grid, $t^* = 124{\cdot}0$

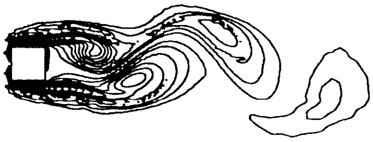

Figure 29. Vorticity contours

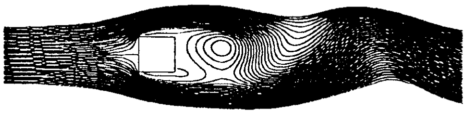

Figure 30. Stream function contours

adapted quadtree grids had a maximum of seven and a minimum of five division levels—the adapted grid is illustrated in Figure 31—and the globally refined grid had seven levels. Figure 32 depicts the steady state velocity vectors obtained using the adapted grid in the vicinity of the step. The flow structure in the entire channel is shown in Figure 33. Table II lists grid details, the number of cells, CPU time and the predicted flow reattachment length. The results suggest that with a careful choice of maximum and minimum division levels within the adapted grid, worthwhile savings can be achieved through employing an adaptive mesh.

### 5.6. *Hanging nodes*

The quadtree grid poses a discretization problem for the Navier–Stokes equations because it is difficult to obtain a first differential across adjacent cells of different sizes. This problem is due to *hanging nodes* which lie on adjacent cell faces in irregular non-conforming grids. In this work an averaging method is used to overcome the hanging node problem. Momentum and pressure discretization equations are constructed on a regular stencil of neighbours having the same cell size as the central cell; for example, if the east neighbour pressure value is required and the east neighbour

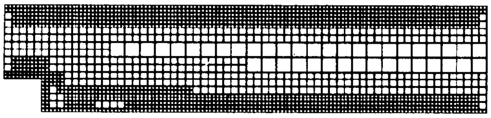

Figure 31. Adapted grid

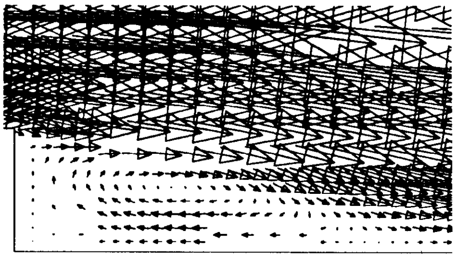

Figure 32. Velocity vectors, detail at step

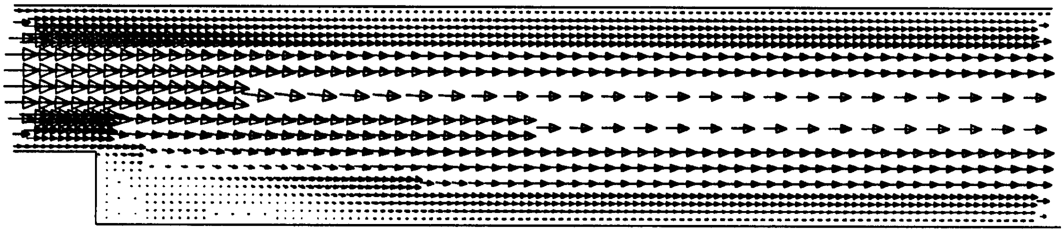

Figure 33. Velocity vectors, adapted grid

Table II. Summary of results for simulation of flow past a backward-facing step at $Re = 50$

| *Grid* | *Number of cells* | *CPU time (s)* | *Reattachment length* |
|---|---|---|---|
| Globally refined uniform grid | 3622 | 5128 | 3·55 |
| Base quadtree grid | 853 | 1277 | 2·63 |
| Adapted quadtree grid | 1879 | 2944 | 2·95 |
| Adapted quadtree grid with interface correction | 1879 | 3140 | 3·00 |

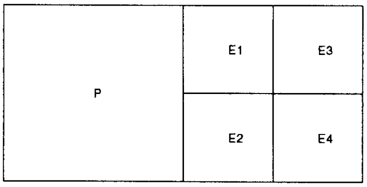

Figure 34. East neighbour

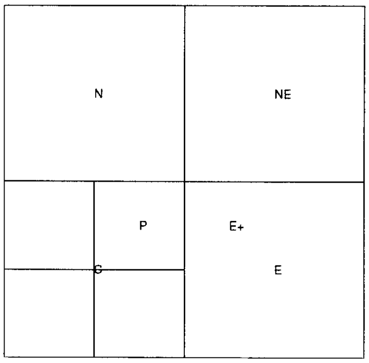

Figure 35. East neighbour

cells are one level smaller than the central cell P, as in Figure 34. The east pressure value used in calculations is the weighted average of the four adjacent cell pressures:

$$p_{\mathrm{E}} = \frac{15}{16}\left(\frac{p_{\mathrm{E1}} + p_{\mathrm{E2}} + p_{\mathrm{E3}} + p_{\mathrm{E4}}}{4}\right) + \left(\frac{1}{16}\frac{p_{\mathrm{E1}} + p_{\mathrm{E2}}}{2}\right) \tag{32}$$

When the east neighbour cell is larger than the central cell, the pressure value of the larger cell is obtained by bilinear interpolation between the large cell and its surrounding neighbours. In Figure 35 the pressure belonging to the fictitious east neighbour position $\mathrm{E}^{+}$ is calculated for cell P as follows:

$$p_{\mathrm{E}^{+}} = \frac{9p_{\mathrm{E}} + 3p_{\mathrm{C}} + 3p_{\mathrm{NE}} + p_{\mathrm{N}}}{16} \tag{33}$$

A disadvantage of this method is that it leads to a conservation error: the mass flux leaving a large cell will not necessarily equal the sum of mass fluxes entering the two smaller cells. The significance of this error was investigated by applying an alternative scheme in which cell face fluxes are corrected to enforce conservation at interfaces. The reattachment predicted by this method is also given in Table II and is found to agree slightly better with results from the globally refined grid.

## 6. CONCLUSIONS

Quadtree-based hierarchical meshes are efficient in controlling the overall number of grid panels while providing fine resolution at boundaries and at flow features. The grids are readily adapted according to vorticity criteria and are therefore closely linked to the evolving flow field. For each flow simulation considered here, the streamline patterns agree reasonably with published data, though the wake tends to be narrower, and in some cases the length of the separation bubble shorter, than obtained by other numerical methods or experiment. These underpredictions may be due to numerical diffusion caused by interpolation errors associated with adaptive regridding of the hierarchical quadtree grids.

The discrepancies noted for the circular cylinder simulations arise from the stepped approximation to the curved boundary. It is recommended that alternative approximations to the curved cylinder boundary should be investigated, such as a triangular mesh. In addition, the solution was found to converge very slowly using point-by-point iterations on the quadtree grids. The use of fast iterative solvers such as multigrids should accelerate convergence of the Poisson equation (28). Multigrid iterations may be ideally suited to quadtree grids, as the tree structure inherently contains nested grids. Gáspár *et al.*,[8] Gáspár and Jósza[5] and Gáspár and Simbierowicz[6] have investigated the use of multigrid techniques with quadtree grids for solving convection–diffusion problems.

In this work the method has been applied solely to subcritical flows for which the maximum Reynolds number is equal to 5000. However, the additional resolution achieved in the wake indicates that it should be possible to simulate flows at higher Reynolds numbers. Used in combination with a turbulence model, the adaptive method with sufficiently high resolution may provide a convenient representation of small-scale eddies as they develop in post-critical turbulent flows.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of U.K. EPSRC Marine Technology Directorate Ltd. in funding this work, and collaborative input from Dr. C. Gáspár and Dr. J. Józsa of the Technical University of Budapest, Hungary.

## REFERENCES

1. H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.
2. H. Samet, *Applications of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.
3. M. A. Yerry and M. S. Shephard, 'A modified quadtree approach to finite element mesh generation', *IEEE Trans. Comput. Graphics Appl.*, **CGA-3**, 39–46 (1983).
4. L. Van Dommelen and E. A. Rundensteiner, 'Fast adaptive summation of point forces in the two-dimensional Poisson equation', *J. Comput. Phys.*, **83**, 126–147 (1989).
5. C. Gáspár and J. Józsa, 'A coupled Lagrangian particle tracking and quadtree-based adaptive multigrid method with application to shear layer evolution', *Proc. XXIV IAHR Congr.*, Madrid, September 1991.
6. C. Gáspár and P. Simbierowicz, 'Difference schemes in tree-structured multigrid context', *Proc. IX Int. Conf. on Computational Methods in Water Resources*, Denver, CO, June 1992.
7. J. Józsa and C. Gáspár, 'Fast, adaptive approximation of wind-induced horizontal flow patterns in shallow lakes using quadtree-based multigrid method', *Proc. IX Int. Conf. on Computational Methods in Water Resources*, Denver, CO, June 1992.
8. C. Gáspár, J. Józsa and P. Simbierowicz, 'Lagrangian modelling of the convective diffusion problem using unstructured grids and multigrid technique', *Proc. 1st Int. Conf. on Water Pollution (Modelling, Measuring and Prediction*), Southampton, September 1991.
9. D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant and J. E. Bussoletti, 'A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics', *J. Comput. Phys.* **92**, 1–66 (1991).
10. D. de Zeeuw and K. G. Powell, 'An adaptively refined Cartesian mesh solver for the Euler equations', *J. Comput. Phys.* **104**, 56–68 (1993).
11. A. Evans, M. J. Marchant, J. Szmelter and N. P. Weatherill, 'Mesh adaptivity with the quadtree method', *Proc. 3rd Int. Conf. on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Barcelona, Spain, 3–7 June, 1991.*
12. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
13. M. Perić, R. Kessler and G. Scheuerer, 'Comparison of finite-volume numerical methods with staggered and collocated grids', *Comput. Fluids*, **16**, 389–403 (1988).

14. M. Hortmann, M. Perić and G. Scheuerer, 'Finite volume multigrid prediction of laminar natural convection: bench-mark solutions', *Int. j. numer. meth. fluids*, **11**, 189–207 (1990).
15. D. M. Greaves, 'Numerical modelling of laminar separated flows and inviscid steep waves using adaptive hierarchical meshes', *D. Phil. Thesis*, Oxford University, 1995.
16. R. Bouard and M. Coutanceau, 'The early stage of development of the wake behind an impulsively started cylinder for $40 < Re < 10^4$', *J. Fluid Mech.*, **101**, 583–607 (1980).
17. R. Franke, W. Rodi and B. Schönung, 'Numerical calculation of laminar vortex shedding flow past cylinders', *J. Wind Engng. Ind. Aerodyn.*, **35**, 235–257 (1990).
18. A. Okajina, 'Strouhal numbers of rectangular cylinders', *J. Fluid Mech.* **121**, 379–398 (1982).
19. R. W. Davis and E. F. Moore, 'A numerical study of vortex shedding from rectangles', *J. Fluid Mech.*, **116**, 475–506 (1982).