INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:227–244 (DOI: 10.1002/fld.750)

Fourth-order accurate compact-difference discretization method for Helmholtz and incompressible Navier–Stokes equations

R. Steijl and H. W. M. Hoeijmakers*,†

Department of Mechanical Engineering, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

SUMMARY

A fourth-order accurate solution method for the three-dimensional Helmholtz equations is described that is based on a compact finite-difference stencil for the Laplace operator. Similar discretization methods for the Poisson equation have been presented by various researchers for Dirichlet boundary conditions. Here, the complicated issue of imposing Neumann boundary conditions is described in detail. The method is then applied to model Helmholtz problems to verify the accuracy of the discretization method. The implementation of the solution method is also described. The Helmholtz solver is used as the basis for a fourth-order accurate solver for the incompressible Navier–Stokes equations. Numerical results obtained with this Navier–Stokes solver for the temporal evolution of a three-dimensional instability in a counter-rotating vortex pair are discussed. The time-accurate Navier–Stokes simulations show the resolving properties of the developed discretization method and the correct prediction of the initial growth rate of the three-dimensional instability in the vortex pair. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: compact finite-difference schemes; Helmholtz solver; boundary conditions; incompressible Navier–Stokes equations

1. INTRODUCTION

Time-accurate, three-dimensional, high-resolution solutions of the Navier–Stokes equations are of particular interest in the field of direct numerical simulation (DNS) and large-eddy simulation (LES) of turbulent flows as well as in the field of aeroacoustics. These flow problems are characterized by the presence of a large range of both time and length scales. For these applications, spatial discretization methods with a high-order accuracy are of interest. Typically, high-order means an order of spatial accuracy exceeding two. In this work, compact

*Correspondence to: H. W. M. Hoeijmakers, Department of Mechanical Engineering, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
†E-mail: h.w.m.hoeijmakers@ctw.utwente.nl

Contract/grant sponsor: Netherlands Organisation for Scientific Research (NWO)
Contract/grant sponsor: Foundation of Fundamental Research of Matter (FOM); contract/grant number: 97-PR-1598

Published online 8 July 2004

Received 7 September 2002
Revised 5 April 2004

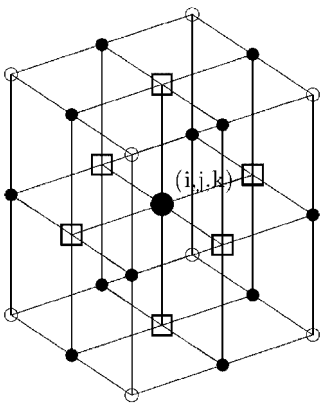

Figure 1. Stencil of compact-difference Laplace operator around node (i, j, k).

finite-difference techniques of fourth-order accuracy are used for the spatial discretization of the Navier–Stokes equations for incompressible flow. The temporal integration of the Navier–Stokes equations employs the fractional time-stepping method of Kim and Moin [1], which is second-order accurate in time. This is a well-known temporal integration method for direct numerical simulation of incompressible flows. The method is semi-implicit, i.e. viscous terms are treated implicitly in combination with explicit integration of the non-linear convection terms. For three-dimensional problems, the method leads to a system of three independent Helmholtz equations for the updated velocity components and a Poisson equation for the pressure, to be solved for every time-step.

The main subject of this article is the spatial discretization method for the Helmholtz and Poisson equations. A fourth-order accurate compact finite-difference discretization method for the Helmholtz equations is presented, with a discretization stencil that has a width of only three mesh points in each spatial direction. This method can similarly be used for the Poisson equation. A conventional fourth-order accurate central-difference method requires a stencil width of five mesh points in each direction. The compact finite-difference stencil for the Laplace operator in three-dimensional space presented here is shown in Figure 1 and involves 27 grid points in total.

Compact finite-difference approximations for the three-dimensional Poisson equation have been known since the mid-1980s. Ananthakrishnaiah *et al.* [2] studied fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients and Dirichlet boundary conditions discretized on a cubic domain with a uniform mesh. For the Poisson equation, a fourth-order accurate scheme was derived that uses 19 grid points within the 27-point stencil shown in Figure 1. A similar derivation, using symbolic computer algebra, was presented by Gupta and Kouatchou [3]. A family of fourth-order accurate schemes was derived on 15, 19 and 21 grid points, respectively, within the 27-point stencil shown in Figure 1. However, their multi-grid solution method failed to converge for the 21-point scheme. The other two schemes resulted in fourth-order accurate solutions for test problems in a cubic domain. The work of Ananthakrishnaiah *et al.* [2] and Gupta and Kouatchou [3] was restricted to Dirichlet boundary conditions.

On the 27-point stencil of Figure 1, a sixth-order accurate approximation for the Poisson equation can be obtained that needs fourth-derivatives of the forcing functions, as is shown by Spotz and Carey [4]. Since the fourth-derivatives cannot be computed within the 27-point stencil, a practical sixth-order accurate method cannot be derived on this stencil.

In the present article, a fourth-order accurate finite-difference approximation for the three-dimensional Helmholtz equation is presented that uses all grid points of the 27-point stencil of Figure 1. The method is more general than the methods of Ananthakrishnaiah *et al.* [2] and Gupta and Kouatchou [3], i.e. a one-parameter family of fourth-order schemes is derived. The parameter can be chosen such that the schemes on 15, 19 and 21 grid points are obtained. An important aspect of the present study is the formulation of Neumann boundary conditions, which often appear in the Poisson problem that typically results from the temporal integration of the incompressible Navier–Stokes equations. Imposing Neumann boundary conditions proves to be a complicated aspect of using three-dimensional compact finite-difference discretization methods.

Section 2 describes the fourth-order accurate discretization method for the Helmholtz equation. The formulation of Neumann boundary conditions is described in Section 3. The object-oriented implementation of the solution method for the Helmholtz equation is described in Section 4. A verification of the discretization method is presented in Section 5. In that section, numerical results for a test problem in a cubic domain are analysed, for Dirichlet boundary conditions as well as for Neumann boundary conditions. The remaining part of this article focuses on numerical simulation of the three-dimensional Navier–Stokes equations for incompressible flow. The solution method for these equations of fourth-order spatial accuracy, that uses the solution method for the Helmholtz equation shown in Section 2, is described in Section 6.

The developed simulation method for the Navier–Stokes equations for incompressible flow has been used in research work on large-scale aircraft wake vortex simulations, simulating the evolution of three-dimensional instabilities in counter-rotating vortex pairs (described in Reference [5]). Numerical results for the long-wavelength instability, or Crow instability (see Reference [6]), in a vortex pair are presented in Section 7. The spatial resolution of the time-dependent flow field is discussed. Finally, conclusions are drawn in Section 8.

2. SPATIAL DISCRETIZATION OF HELMHOLTZ/POISSON EQUATIONS

This section discusses the fourth-order accurate compact finite-difference discretization of the Helmholtz equation on a uniformly spaced three-dimensional rectangular domain. The discretization method uses the stencil sketched in Figure 1. The discretization method is similar to methods presented by Ananthakrishnaiah *et al.* [2] and Gupta and Kouatchou [3]. However, the method discussed here is more general than any of the methods discussed in References [2, 3], since it is generalized to Helmholtz equations and has one parameter that can be chosen freely.

2.1. Spatial discretization of Laplace operator

The fourth-order accurate compact finite-difference stencil for the Laplace operator in three dimensions, which relates the Laplace operator in a point (i, j, k) and the one in its 6 immediate

neighbours $(i-1,j,k)$, $(i+1,j,k)$, $(i,j-1,k)$, $(i,j+1,k)$, $(i,j,k-1)$ and $(i,j,k+1)$, the points indicated as $\square$ in Figure 1, is chosen to have the following form:

$$\begin{aligned}
&\Delta f_{i,j,k} + \alpha[\Delta f_{i-1,j,k} + \Delta f_{i+1,j,k} + \Delta f_{i,j-1,k} + \Delta f_{i,j+1,k} + \Delta f_{i,j,k-1} + \Delta f_{i,j,k+1}] \\
&\quad = a_1(f_{i-1,j,k} + f_{i+1,j,k} + f_{i,j-1,k} + f_{i,j+1,k} + f_{i,j,k-1} + f_{i,j,k+1} - 6f_{i,j,k})/h^2 \\
&\qquad + \frac{a_2}{4}(f_{i-1,j-1,k} + f_{i-1,j+1,k} + f_{i+1,j-1,k} + f_{i+1,j+1,k} \\
&\qquad + f_{i-1,j,k-1} + f_{i-1,j,k+1} + f_{i+1,j,k-1} + f_{i+1,j,k+1} \\
&\qquad + f_{i,j-1,k-1} + f_{i,j-1,k+1} + f_{i,j+1,k-1} + f_{i,j+1,k+1} - 12f_{i,j,k})/h^2 \\
&\qquad + \frac{a_3}{4}(f_{i-1,j-1,k-1} + f_{i-1,j+1,k-1} + f_{i+1,j-1,k-1} + f_{i+1,j+1,k-1} \\
&\qquad + f_{i-1,j-1,k+1} + f_{i-1,j+1,k+1} + f_{i+1,j-1,k+1} + f_{i+1,j+1,k+1} - 8f_{i,j,k})/h^2
\end{aligned} \tag{1}$$

the parameters α, a_1, a_2 and a_3 are to be determined by matching the Taylor series expansion of the left-hand side with the one of the right-hand side to the required order. For convenience, the nodes in the stencil are divided in the following groups, indicated with different symbols in Figure 1:

- $\square$: the 6 nodes associated with coefficient a_1;
- $\bullet$: the 12 nodes associated with coefficient a_2;
- $\circ$: the 8 nodes associated with coefficient a_3.

For fourth-order accuracy, the Taylor series coefficients corresponding to terms involving the various even derivatives:

$$\Delta f_{i,j,k}, \quad \left(\frac{\partial^4 f}{\partial x^4} + \frac{\partial^4 f}{\partial y^4} + \frac{\partial^4 f}{\partial z^4}\right), \quad \left(\frac{\partial^4 f}{\partial x^2\,\partial y^2} + \frac{\partial^4 f}{\partial x^2 \partial z^2} + \frac{\partial^4 f}{\partial y^2 \partial z^2}\right) \tag{2}$$

need to be matched. Solving the resulting linear system of equation gives:

$$\alpha = \frac{1}{6}, \quad a_1 = \frac{2}{3} + \beta, \quad a_2 = \frac{4}{3} - 2\beta, \quad a_3 = \beta \tag{3}$$

with β a parameter that can be freely chosen. For specific values of β, the following family of fourth-order accurate stencils results:

- $\beta = 0$: 19-point stencil, removes 8 vertices corresponding to a_3;
- $\beta = -2/3$: 21-point stencil, eliminates the 6 nodes corresponding to coefficient a_1;
- $\beta = +2/3$: 15-point stencil, nodes corresponding to coefficient a_2 eliminated from stencil;
- $\beta \notin \{0, -2/3, +2/3\}$: 27-point stencil involving all nodes.

The first stencil was used in the fourth-order accurate discretization method for the Poisson equation derived by Ananthakrishnaiah *et al.* [2]. The first three stencils were used by Gupta and Kouatchou [3] in their fourth-order accurate discretization methods for the Poisson equation. The last stencil, i.e. with $\beta \notin \{0, -2/3, +2/3\}$, was not shown by Ananthakrishnaiah

et al. [2] or Gupta and Kouatchou [3]. The dependence on β of the resolution properties of the discretization method is studied in Section 5.

2.2. Helmholtz/Poisson solver

A fourth-order accurate solution method for the three-dimensional Helmholtz equation,

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}-\lambda\right) f(x,y,z)=r(x,y,z) \tag{4}$$

with λ a constant parameter in a three-dimensional computational domain with a structured mesh with uniform mesh spacing h in all three co-ordinate directions, can be derived from the compact finite-difference expression given in Equation (1) with the coefficients defined in Equation (3). The derivation starts with expression:

$$\begin{aligned}
&\Delta f_{i,j,k}+\frac{1}{6}[\Delta f_{i-1,j,k}+\Delta f_{i+1,j,k}+\Delta f_{i,j-1,k}+\Delta f_{i,j+1,k}+\Delta f_{i,j,k-1}+\Delta f_{i,j,k+1}]\\
&\quad=\frac{1}{h^2}\left(\frac{2}{3}+\beta\right)(f_{i-1,j,k}+f_{i+1,j,k}+f_{i,j-1,k}+f_{i,j+1,k}+f_{i,j,k-1}+f_{i,j,k+1}-6f_{i,j,k})\\
&\qquad+\frac{1}{4h^2}\left(\frac{4}{3}-2\beta\right)(f_{i-1,j-1,k}+f_{i+1,j-1,k}+f_{i-1,j+1,k}+f_{i+1,j+1,k}\\
&\qquad+f_{i-1,j,k-1}+f_{i+1,j,k-1}+f_{i-1,j,k+1}+f_{i+1,j,k+1}\\
&\qquad+f_{i,j-1,k-1}+f_{i,j+1,k-1}+f_{i,j-1,k+1}+f_{i,j+1,k+1}-12f_{i,j,k})\\
&\qquad+\frac{\beta}{4h^2}(f_{i-1,j-1,k-1}+f_{i+1,j-1,k-1}+f_{i-1,j+1,k-1}+f_{i+1,j+1,k-1}\\
&\qquad+f_{i-1,j-1,k+1}+f_{i+1,j-1,k+1}+f_{i-1,j+1,k+1}+f_{i+1,j+1,k+1}-8f_{i,j,k})+O(h^4)
\end{aligned} \tag{5}$$

Equation (5) can be used to solve the Helmholtz equation (4) by first developing the left-hand side of Equation (5) into Taylor series expansions up to fourth-order accuracy, this gives:

$$\begin{aligned}
&\Delta f_{i,j,k}+\frac{1}{6}[\Delta f_{i-1,j,k}+\Delta f_{i+1,j,k}+\Delta f_{i,j-1,k}+\Delta f_{i,j+1,k}+\Delta f_{i,j,k-1}+\Delta f_{i,j,k+1}]\\
&\quad=2\Delta f_{i,j,k}+\frac{h^2}{6}\Delta^2 f_{i,j,k}+O(h^4)\\
&\quad=2r_{i,j,k}+\frac{h^2}{6}(\lambda r_{i,j,k}+\Delta r_{i,j,k})+\frac{1}{6}(12+\lambda h^2)\lambda f_{i,j,k}+O(h^4)
\end{aligned} \tag{6}$$

where the last step uses Equation (4). The terms with the function values on the right-hand side of Equation (6) are moved to the left-hand side of Equation (5), resulting in the following

expression on a 27-point stencil:

$$
\begin{aligned}
&-d f_{i,j,k} + \left(\frac{2}{3} + \beta\right)(f_{i-1,j,k} + f_{i+1,j,k} + f_{i,j-1,k} + f_{i,j+1,k} + f_{i,j,k-1} + f_{i,j,k+1}) \\
&\quad + \left(\frac{1}{3} - \frac{\beta}{2}\right)(f_{i-1,j-1,k} + f_{i+1,j-1,k} + f_{i-1,j+1,k} + f_{i+1,j+1,k} \\
&\quad + f_{i-1,j,k-1} + f_{i+1,j,k-1} + f_{i-1,j,k+1} + f_{i+1,j,k+1} \\
&\quad + f_{i,j-1,k-1} + f_{i,j+1,k-1} + f_{i,j-1,k+1} + f_{i,j+1,k+1}) \\
&\quad + \frac{\beta}{4}(f_{i-1,j-1,k-1} + f_{i+1,j-1,k-1} + f_{i-1,j+1,k-1} + f_{i+1,j+1,k-1} \\
&\quad + f_{i-1,j-1,k+1} + f_{i+1,j-1,k+1} + f_{i-1,j+1,k+1} + f_{i+1,j+1,k+1}) \\
&= 2h^2 r_{i,j,k} + \frac{h^4}{6}(\lambda r_{i,j,k} + \Delta r_{i,j,k}) + O(h^6)
\end{aligned}
\tag{7}
$$

where diagonal element $d = (8 + 2\beta) + [(12 + \lambda h^2)/6]\lambda h^2$. The evaluation of Δr in Equation (7) needs to be of second-order accuracy only to guarantee fourth-order accuracy of the method.

Applying Equation (7) to the Helmholtz equation (4) on the domain Ω results in the following system of equations:

$$
A\underline{f} = 2h^2\underline{r} + \frac{h^4}{6}(\lambda\underline{r} + \Delta\underline{r}) + \underline{u}_d + 2h\underline{u}_n + O(h^6) \tag{8}
$$

where A is the sparse discretization matrix, $\underline{f}$ is the vector containing the unknown function values $f_{i,j,k}$, $\underline{r}$ is the vector representing the forcing term of Equation (4) in the mesh points and $\Delta\underline{r}$ is the vector containing the Laplacian of the forcing term in the mesh points. Dirichlet and Neumann boundary conditions are represented by the vectors $\underline{u}_d$ and $\underline{u}_n$, respectively.

3. FORMULATION OF BOUNDARY CONDITIONS

The formulation of Dirichlet boundary conditions for the present fourth-order finite-difference method contains the known function values at the domain boundary in the vector $\underline{u}_d$ on the right-hand side of Equation (8). Since the stencil width is just 3 mesh points, this is sufficient to maintain fourth-order accuracy. Therefore, the formulation of Dirichlet boundary conditions is not more complicated than that for a conventional second-order accurate central-difference method, which also has a stencil width of 3 mesh points.

The formulation of Neumann boundary conditions is significantly more complicated and is discussed in detail here. The implementation of Neumann boundary conditions prescribed on the face $(0, j, k)$ is taken as example. Applying the discretization scheme to the grid points of this face of the domain, introduces mesh points with $i = -1$ into the system. The contribution

of mesh points with $i=-1$ is eliminated using the central-difference expression for the first derivative at the face of the domain considered ($i=0$):

$$f_{1,j,k} - f_{-1,j,k} = 2h\left(\frac{\partial f}{\partial x}\right)_{0,j,k} + \frac{h^3}{3}\left(\frac{\partial^3 f}{\partial x^3}\right)_{0,j,k} + O(h^5) \tag{9}$$

The complicating element is the third-derivative at the boundary $i=0$ that needs to be included in order to maintain fourth-order accuracy of the solution method. The need to compute the third-derivative in the direction normal to the domain boundary can be circumvented by rewriting this derivative as:

$$\begin{aligned}\left(\frac{\partial^3 f}{\partial x^3}\right)_{0,j,k} &= \frac{\partial}{\partial x}\left[\Delta f_{0,j,k} - \left(\frac{\partial^2 f}{\partial y^2}\right)_{0,j,k} - \left(\frac{\partial^2 f}{\partial z^2}\right)_{0,j,k}\right] \\ &= \frac{\partial}{\partial x}\left[r_{0,j,k} + \lambda f_{0,j,k} - \left(\frac{\partial^2 f}{\partial y^2}\right)_{0,j,k} - \left(\frac{\partial^2 f}{\partial z^2}\right)_{0,j,k}\right] \\ &= \left(\frac{\partial r}{\partial x}\right)_{0,j,k} + \lambda\left(\frac{\partial f}{\partial x}\right)_{0,j,k} - \frac{\partial^2}{\partial y^2}\left(\frac{\partial f}{\partial x}\right)_{0,j,k} - \frac{\partial^2}{\partial z^2}\left(\frac{\partial f}{\partial x}\right)_{0,j,k}\end{aligned} \tag{10}$$

where Equation (4) has been used.

The formulation of the discretization method for mesh points on the face $(0,j,k)$ can be obtained by using Equation (7), Equation (9) and Equation (10). For each grid point $(0,j,k)$ on the domain boundary at which Neumann boundary conditions are imposed, 9 mesh points (for which $i=-1$) of the total of 27 mesh points of the stencil need to be eliminated using Equation (10). The terms $\partial^2/\partial y^2(\partial f/\partial x)$ and $\partial^2/\partial z^2(\partial f/\partial x)$ for these 9 grid points can be combined into a much simpler form using Taylor series expansions. Following this approach, the discretization of Equation (7) at the grid point on the boundary with Neumann conditions ($i=0$) becomes:

$$\begin{aligned}&-d f_{i,j,k} + \left(\frac{2}{3}+\beta\right)(2f_{1,j,k} + f_{0,j-1,k} + f_{0,j+1,k} + f_{0,j,k-1} + f_{0,j,k+1}) \\ &\quad + \left(\frac{1}{3}-\frac{\beta}{2}\right)(2f_{1,j-1,k} + 2f_{1,j+1,k} + 2f_{1,j,k-1} + 2f_{1,j,k+1} \\ &\quad + f_{0,j-1,k-1} + f_{0,j+1,k-1} + f_{0,j-1,k+1} + f_{0,j+1,k+1}) \\ &\quad + \frac{\beta}{2}(f_{1,j-1,k-1} + f_{1,j+1,k-1} + f_{1,j-1,k+1} + f_{1,j+1,k+1}) \\ &= 2h^2 r_{0,j,k} + \frac{h^4}{6}(\lambda r_{0,j,k} + \Delta r_{0,j,k}) + 2h\underline{u}_n(0,j,k) + O(h^5)\end{aligned} \tag{11}$$

with the elements of vector $\underline{u}_n$ given as:

$$\begin{aligned}\underline{u}_n(0,j,k) = 2\left(\frac{\partial f}{\partial x}\right)_{0,j,k} + \frac{h^2}{6}\Bigg[&\left(\frac{2}{3}+\beta\right)\left(\frac{\partial r}{\partial x}\right)_{0,j,k} \\ &+ \left(\frac{1}{3}-\frac{\beta}{2}\right)\left\{\left(\frac{\partial r}{\partial x}\right)_{0,j-1,k} + \left(\frac{\partial r}{\partial x}\right)_{0,j+1,k} + \left(\frac{\partial r}{\partial x}\right)_{0,j,k-1} + \left(\frac{\partial r}{\partial x}\right)_{0,j,k+1}\right\} \\ &+ \frac{\beta}{4}\left\{\left(\frac{\partial r}{\partial x}\right)_{0,j-1,k-1} + \left(\frac{\partial r}{\partial x}\right)_{0,j+1,k-1} + \left(\frac{\partial r}{\partial x}\right)_{0,j-1,k+1} + \left(\frac{\partial r}{\partial x}\right)_{0,j+1,k+1}\right\}\Bigg] \\ &+ \lambda\frac{h^2}{6}\Bigg[\left(\frac{2}{3}+\beta\right)\left(\frac{\partial f}{\partial x}\right)_{0,j,k} \\ &+ \left(\frac{1}{3}-\frac{\beta}{2}\right)\left\{\left(\frac{\partial f}{\partial x}\right)_{0,j-1,k} + \left(\frac{\partial f}{\partial x}\right)_{0,j+1,k} + \left(\frac{\partial f}{\partial x}\right)_{0,j,k-1} + \left(\frac{\partial f}{\partial x}\right)_{0,j,k+1}\right\} \\ &+ \frac{\beta}{4}\left\{\left(\frac{\partial f}{\partial x}\right)_{0,j-1,k-1} + \left(\frac{\partial f}{\partial x}\right)_{0,j+1,k-1} + \left(\frac{\partial f}{\partial x}\right)_{0,j-1,k+1} + \left(\frac{\partial f}{\partial x}\right)_{0,j+1,k+1}\right\}\Bigg]\end{aligned} \tag{12}$$

again with diagonal element $d = (8+2\beta)+((12+\lambda h^2)/6)\lambda h^2$. The discretization of the imposed Neumann boundary condition is based on the fourth-order accurate expression of Equation (9). Therefore, the derivatives of f are assumed to be known from the problem definition either exactly or from approximation using one-sided differences of fourth-order accuracy. As a result of the pre-multiplication with h^2, the evaluation of the derivatives of r needs to be of second-order accuracy only. One-sided differences are used for this purpose.

4. IMPLEMENTATION OF HELMHOLTZ SOLVER

The fourth-order accurate discretization method for the Helmholtz equation given in Equation (7) forms the basis of the solver for three-dimensional Helmholtz problems used in the present work. The implementation of this solver was designed to enable solution of the Helmholtz and Poisson problems in a wide variety of computational domains and combinations of imposed boundary conditions. To achieve this flexibility, the object-oriented programming approach was chosen, using the C++ programming language, where user-defined data types (*classes*, see Reference [7]) are defined that contain data elements and functions/operators. Using this approach, the solution method is implemented as a collection of cooperating *objects*, which are instances of the different classes. A hierarchy of classes is formed using inheritance (construct new classes by re-using already existing class definitions). In the implementation of the present Helmholtz solver, a basis framework is created that includes most of the techniques required to solve the Helmholtz equations in three dimensions, such as the finite-difference expressions for

the spatial derivatives, conjugate gradient methods that solve the linear systems of equations, preconditioners, vector-norm functions, etc. This framework is designed to be independent of the choice of data structure used to represent the unknowns of a particular Helmholtz problem that is to be solved, i.e. this data is hidden in an object (*data encapsulation*). User-defined classes are then added (typically by inheriting from existing classes) to this framework to define the problem-specific domain properties and boundary conditions. In this way, a large range of different Helmholtz problems can be solved with only small additions to the basis framework. In the present article, the developed Helmholtz solver is applied only to model problems with a rectangular computational domain.

The sparse-matrix system of linear equations that results from applying Equation (7) is solved using either the preconditioned conjugate gradient method (for symmetric systems) or the preconditioned bi-conjugate gradient method (for non-symmetric systems). In both cases, a Jacobi preconditioner is used. The implementation of the conjugate gradient methods is based on that of References [8, 9].

5. APPLICATION TO MODEL PROBLEM

The accuracy of the Helmholtz solver, described in Section 2, is analysed in this section by application to a model problem. The model problem involves the solution of the Helmholtz equation, as given in Equation (4), in a unit cube $\Omega = [0,1]^3$ with $N+1$ uniformly spaced grid points, with mesh spacing h, identical in the three co-ordinate directions. A model problem can be formulated by choosing a solution $f(x,y,z)$ in this domain and computing the corresponding forcing function $r(x,y,z)$ by substitution of solution $f(x,y,z)$ in the Helmholtz equation. For brevity, only one such a 'constructed-solution' model problem is considered here, i.e. with the exact solution given as:

$$f(x,y,z) = 1 + x + \cos(2\pi x)\sin(2\pi y)\sin(2\pi z) \tag{13}$$

To confirm the findings of the present analysis, a model problem with an exponential spatial dependence was also studied. Here, the results of this model problem are not discussed.

For the model problem with solution (13), two sets of boundary conditions are considered:

- Dirichlet boundary conditions on the six faces of the computational domain. In this case, the coefficient matrix A in Equation (8) is symmetric and the approximate solution is obtained using the pre-conditioned conjugate gradient method;
- Neumann boundary conditions on the faces $i = 0$ and N and Dirichlet boundary conditions on the 4 other faces. For both model problems, function $f(x,y,z)$ has a non-zero normal derivative at the faces $i = 0$ and N. For $\lambda \neq 0$, the derivative of the forcing term $r(x,y,z)$ is non-zero as well. Therefore, with these model problems, the formulation for the Neumann boundary conditions can be properly tested.

  The coefficient matrix is non-symmetric for this combination of boundary conditions and the pre-conditioned bi-conjugate gradient method is used to obtain the approximate solution.

For the model problem, numerical results have been obtained for various choices of parameter λ, numerical scheme parameter β and imposed boundary conditions, this for various

Table I. Parameters of numerical results for test problem.

λ	b.c.	β	Points in stencil
0	Dirichlet	0	19
0	Dirichlet/Neumann	0	19
10	Dirichlet	0	19
0	Dirichlet	$-2/3$	21
0	Dirichlet	$+2/3$	15
0	Dirichlet	0	19

Table II. Summary of results of test problem.

b.c.	λ	β	h	Max. error	L_2 norm	Iter
Dirichlet	0	0	1/10	4.81351×10^{-3}	6.28649×10^{-5}	15
Dirichlet	0	0	1/20	3.10890×10^{-4}	1.10908×10^{-6}	26
Dirichlet	0	0	1/30	6.00456×10^{-5}	1.11789×10^{-7}	34
Dirichlet	0	0	1/40	1.91310×10^{-5}	2.22963×10^{-8}	38
Dirichlet/Neumann	0	0	1/10	7.03758×10^{-3}	8.83731×10^{-5}	6
Dirichlet/Neumann	0	0	1/20	3.25727×10^{-4}	1.36645×10^{-6}	11
Dirichlet/Neumann	0	0	1/30	5.99446×10^{-5}	1.37586×10^{-7}	17
Dirichlet/Neumann	0	0	1/40	1.88639×10^{-5}	2.75621×10^{-8}	24
Dirichlet	10	0	1/10	4.37620×10^{-3}	5.74069×10^{-5}	14
Dirichlet	10	0	1/20	2.84521×10^{-4}	1.02030×10^{-6}	25
Dirichlet	10	0	1/30	5.50169×10^{-5}	1.02951×10^{-7}	32
Dirichlet	10	0	1/40	1.75359×10^{-5}	2.05409×10^{-8}	37
Dirichlet	0	$-2/3$	1/10	8.09828×10^{-4}	1.08460×10^{-5}	14
Dirichlet	0	$-2/3$	1/20	3.39588×10^{-5}	1.22886×10^{-7}	26
Dirichlet	0	$-2/3$	1/30	5.93874×10^{-6}	1.11038×10^{-8}	45
Dirichlet	0	$-2/3$	1/40	1.82233×10^{-6}	2.12921×10^{-9}	59
Dirichlet	0	$+2/3$	1/10	8.78799×10^{-3}	1.14734×10^{-4}	15
Dirichlet	0	$+2/3$	1/20	5.87690×10^{-4}	2.09498×10^{-6}	29
Dirichlet	0	$+2/3$	1/30	1.14148×10^{-4}	2.12467×10^{-7}	40
Dirichlet	0	$+2/3$	1/40	3.64398×10^{-5}	4.24628×10^{-8}	45

choices of mesh width h. Table I presents the parameters for the various numerical results that are shown in Table II. Table II shows the maximum deviation from the exact solution, the L_2 norm of the error and the number of iterations required for convergence of the preconditioned (bi)-conjugate gradient solver to a tolerance of 10^{-12}. Results are shown for 4 mesh sizes: $h = 1/10$, $1/20$, $1/30$ and $1/40$. In Table II, three different discretization stencils are considered: involving 19 grid points ($\beta = 0$), 21 grid points ($\beta = -2/3$) and 15 grid points ($\beta = 2/3$) within the 27-point stencil of Figure 1. For the 19-point stencil, the effect of imposed boundary conditions can be analysed by comparing the respective results in Table II. The Neumann boundary conditions can be seen to result in a slightly larger numerical error for the coarser meshes. For the finest meshes, the effect is negligible. Also for the 19-point stencil ($\beta = 0$), the dependence of the numerical error on parameter λ can be studied by comparing

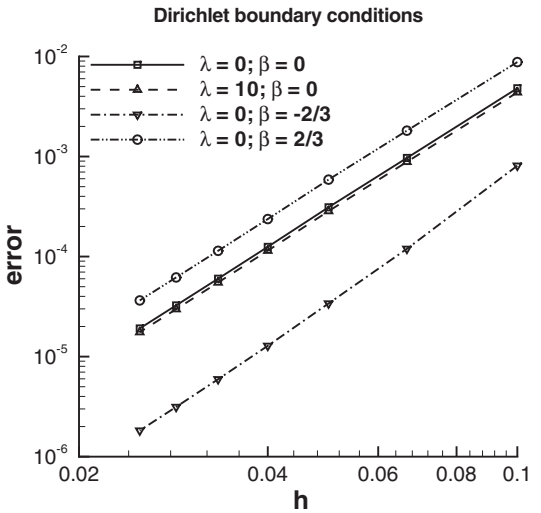

Figure 2. Maximum error versus h, Dirichlet boundary conditions.

results for $\lambda = 0$ and 10 in Table II, both for Dirichlet boundary conditions. An increasing λ can be seen to slightly enhance the convergence rate of the iterative solution method, as a result of the more favourable condition number of the discretization matrix. Furthermore, the numerical error is reduced for an increased λ. From Table II it can be seen that the magnitude of the maximum deviation from the exact solution is larger for the 15-point stencil ($\beta = 2/3$) than that for the 19-point stencil ($\beta = 0$) and the 21-point stencil ($\beta = -2/3$). The convergence behaviour of the pre-conditioned conjugate gradient method is in general most favorable for the 19-point stencil ($\beta = 0$). For Dirichlet boundary conditions the numerical results are compiled in Figure 2, including results for intermediate values of mesh width h. For the 4 different cases, a linear dependence in a double logarithmic axis system can be seen for smaller values of h. The slope of the almost straight lines indicates that the maximum deviation from the exact solution behaves asymptotically as h^4 (the slopes vary from 3.963 to 4.074 for different choices of the parameters β and λ). A similar comparison is shown in Figure 3 for Neumann/Dirichlet boundary conditions. Qualitatively, the results are similar to those for Dirichlet boundary conditions imposed on all 6 faces of the computational domain, i.e. also fourth-order accuracy is achieved. The effect of β on the numerical error is shown in detail in Figure 4 for Dirichlet boundary conditions for values of β within the range $[-1, 1]$. The error as function of β reveals a cusp-like behaviour around $\beta = -3/4$. Around this value of β, the maximum error is significantly lower than for other choices of β. This behaviour is thought to occur as a result of cancellation of parts of the leading term of the truncation error as a result of the particular form of the exact solution, i.e. the trigonometric functions in the three co-ordinate directions. A detailed study of the dependence of the error on β for the model problem with an exponential spatial dependence shows a weaker dependence on β, without any cusp-like behaviour.

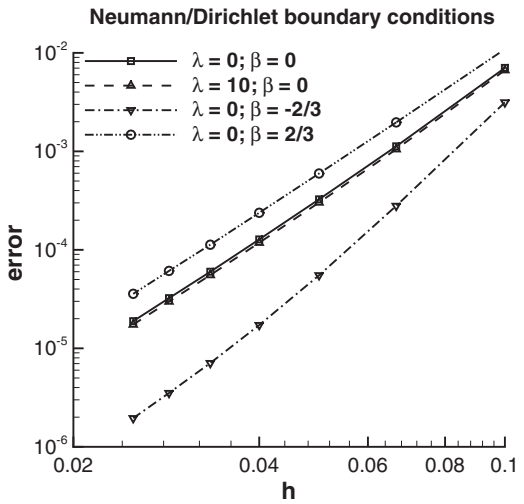

Figure 3. Maximum error versus h, Neumann/Dirichlet boundary conditions.

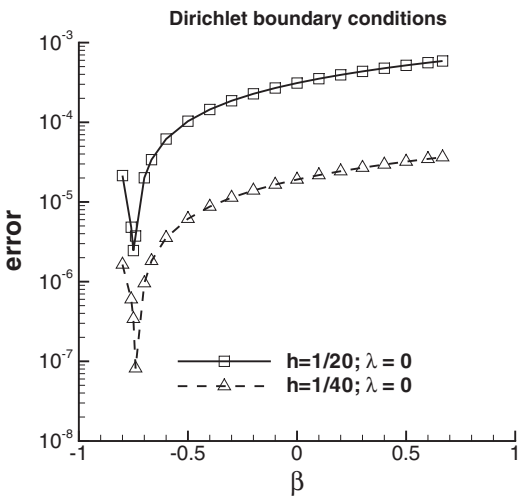

Figure 4. Maximum error versus β, Dirichlet boundary conditions, effect of h.

6. APPLICATION TO INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

The discretization method for the Helmholtz equations of fourth-order accuracy is used as the basis of a discretization method for the three-dimensional Navier–Stokes equations for an incompressible flow of a Newtonian fluid with constant viscosity in Cartesian co-ordinates,

given by

$$\frac{\partial \underline{u}}{\partial t} + \underline{u} \cdot \nabla \underline{u} = -\frac{1}{\rho} \nabla p + \nu \Delta \underline{u} \tag{14}$$

$$\nabla \cdot \underline{u} = 0 \tag{15}$$

where $\underline{u}$ denotes the velocity vector and p is the pressure, ρ the (constant) density and ν the (constant) kinematic viscosity of the medium. Equation (15) is the divergence-free constraint on the velocity field, as a result of the incompressibility of the flow. The Navier–Stokes equations are integrated in time using the fractional time-stepping method of Kim and Moin [1]. This method is second-order accurate in time and involves three steps to advance a solution of the Navier–Stokes equations from time level n to $n+1$. In the first step, the second-order explicit Adams–Bashfort method is applied to the non-linear convection terms and the second-order implicit Crank–Nicholson scheme to the viscous terms. Omitting the pressure term from the momentum equations, for co-ordinate direction $i = 1, 2, 3$, an intermediate velocity field with components v_i can be obtained from the Helmholtz equations:

$$\left(1 - \frac{\nu \delta t}{2} \Delta\right) v_i = -\delta t \left(\frac{3}{2} H_i^n - \frac{1}{2} H_i^{n-1}\right) + \left(1 + \frac{\nu \delta t}{2} \Delta\right) u_i^n \tag{16}$$

where δt is the time-step and H_i^n the non-linear convection terms at time-level n.

In a three-dimensional flow situation, Equation (16) corresponds to a system of three independent Helmholtz equations. Since the pressure term was omitted from the momentum equations, the intermediate velocity field with components v_i will not be divergence-free. The next step in the temporal integration is the computation of a pressure-like variable from the intermediate velocity field v_i. Quantity ϕ, defined as $p = \phi + (\nu \delta t/2)\Delta\phi$ (here, p is the pressure divided by the constant density), can be computed from the Poisson equation

$$\Delta \phi = \frac{1}{\delta t} \frac{\partial v_i}{\partial x_i} \tag{17}$$

where the divergence of the intermediate velocity field forms the forcing term. From the computed pressure-like variable ϕ, a solenoidal velocity field at the new time-level $n+1$ is computed as

$$u_i^{n+1} = v_i - \delta t \frac{\partial \phi}{\partial x_i} \tag{18}$$

The present method for the incompressible Navier–Stokes equations employs a collocated mesh formulation. The spatial discretization method uses the sixth-order accurate compact finite-differences presented by Lele [10] for the first and second derivatives in Equations (16)–(18). The discretization of the non-linear convection terms H_i uses the skew-symmetric form to reduce the aliasing errors, see Reference [11], and conserve kinetic energy. The discretization method developed in this article for the Helmholtz equation is used to discretize Equations (16) and (17). A tenth-order accurate low-pass filter, based on the expression given by Lele [10], is used to suppress the growth of under-resolved modes. This approach forms a suitable alternative for 'artificial viscosity' terms typically used in Navier–Stokes methods based on lower-order discretization methods.

The Navier–Stokes simulation method is implemented using the C programming language. The Helmholtz and Poisson equations are solved using routines from the Helmholtz solver tested in Section 5. The implementation is reported in Reference [12]. Similarly, the implementation of Dirichlet as well as Neumann boundary conditions is identical to the implementation in the Helmholtz solver used in Section 5. For practical simulations of three-dimensional flows, a parallel implementation of the Navier–Stokes method is needed. The parallelization of the Navier–Stokes solver is based on domain decomposition and explicit message-passing between the processes, allowing parallel simulations to be performed on distributed-memory as well as shared-memory computers. The MPI library is used for the required communication and synchronization in simulations on multiple processors. The parallelization is described in more detail in Reference [5].

7. RESULTS FOR INSTABILITY IN VORTEX PAIR

The simulation method for the incompressible Navier–Stokes equations, based on the fourth-order accurate spatial discretization techniques discussed in Section 2, is now applied to the well-studied time-dependent flow of a counter-rotating vortex pair developing a long-wavelength three-dimensional sinusoidal instability.

7.1. Crow instability test case

Crow [6] presented a linear stability analysis of small perturbations on a pair of trailing vortex filaments with a relatively small core (core radius r_c less than 10% of the spacing of the vortices) using long-wavelength assumptions. This analysis showed a sinusoidal instability with a typical wavelength of approximately 8 times the initial spacing b_0 of the vortex. The linear perturbations occur in planes that are inclined at approximately 45° with respect to the horizontal plane. The driving mechanism of the evolution of the three-dimensional instability is the mutual velocity induction of the vortices. The analysis showed an exponential growth of the linear perturbation with a growth rate directly proportional to the straining field induced by mutual velocity induction, i.e. $\Gamma/2\pi b_0^2$, with Γ the circulation of the vortices. Using this fact, we introduce a non-dimensional time $T = t/t_{\mathrm{ref}}$, with $t_{\mathrm{ref}} = 2\pi b_0^2/\Gamma$. Characteristic time scale t_{ref} is the time in which the unperturbed vortex pair would move one vortex spacing downward as a result of the mutual velocity induction.

In the temporal evolution of the Crow instability, two stages can be discerned:

- The *linear* stage with an exponential growth at the growth rate predicted by linear stability analysis. This growth rate is maintained up to amplitudes for which the assumption of linearity becomes invalid. At the same time, the planes in which the instability occurs, rotate from the initial inclination angle of 45 to 60° or more. These effects have been observed by various researchers, see e.g. References [13–16]. Figure 5 shows the vortex pair in the computational domain at $T = 0.637$ for a Reynolds number based on circulation $Re_\Gamma = \Gamma/\nu = 6.7 \times 10^5$. Shown are iso-surfaces of pressure. At this stage of the evolution of the Crow instability, the perturbation still has a sinusoidal shape.
- The *non-linear* stage characterized by the reconnection of the vortices and the subsequent formation of vortex ring-like structures. Using a further development of the Navier–

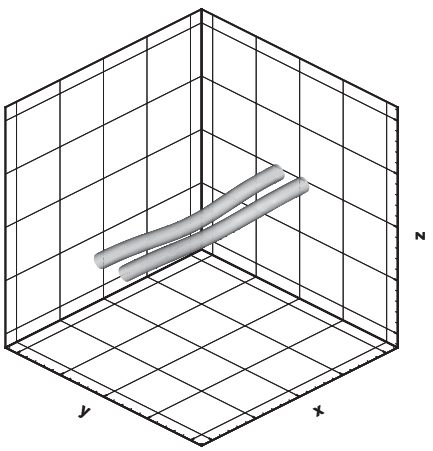

Figure 5. Linear stage of Crow instability: $T = 0.637$, $Re_\Gamma = 6.7 \times 10^5$, $r_c/b = 0.2$, 81^3 mesh.

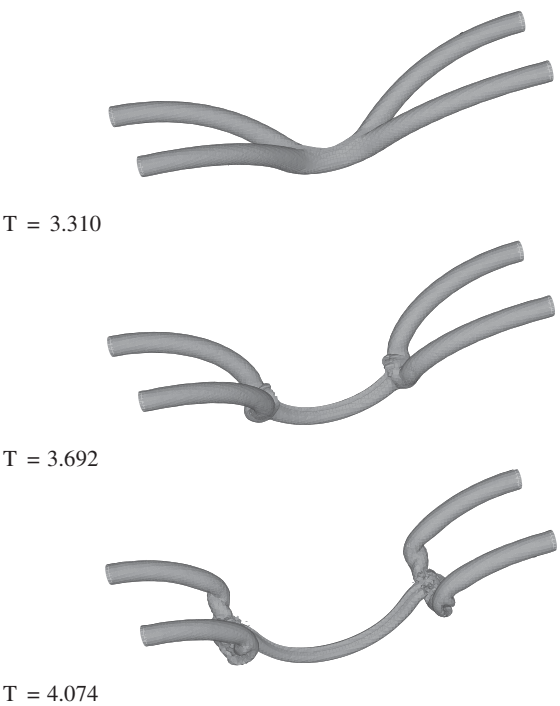

Figure 6. Vortex reconnection in Crow instability evolution: $Re_\Gamma = 1.67 \times 10^4$, $r_c/b = 0.2$, 128^3 mesh.

Stokes method described here, this stage is shown is Figure 6. The figure shows iso-surfaces of the magnitude of vorticity. Results have been obtained on a 128^3 mesh for a Reynolds number based on circulation $Re_\Gamma = 1.67 \times 10^4$.

For the purpose of the present article, we use the linear stage of the Crow instability as a validation test case for the numerical method, since it allows a comparison with the growth rate predicted by linear stability theory. The vortices have a Lamb–Oseen azimuthal velocity profile, which in a cylindrical co-ordinate system is given as:

$$u_\theta(r) = \frac{\Gamma}{2\pi} \frac{[1 - \exp(-\gamma r^2/r_c^2)]}{r} \tag{19}$$

where r_c represents the vortex core radius and r is the radial distance from the vortex axis. The constant γ in the exponential function is chosen to position the maximum of the azimuthal velocity at r_c (this leads to the constraint $(2\gamma + 1)e^{-\gamma} = 1 \rightarrow \gamma \cong 1.256$). Here, the spacing b_0 between the vortices in the unperturbed situation is chosen to be 5 times the vortex core radius, i.e. $r_c/b_0 = 0.2$. For this choice of r_c/b_0, linear stability theory predicts a wavelength of maximum amplification λ of 8 b_0. A cubic computational domain is used with uniform mesh. The dimensions of the domain are set at 8 b_0, i.e. the wavelength of the perturbation in the axial direction of the vortex pair. To account for the downward motion of the vortex pair, a constant vertical velocity, which is equal to the descent velocity, is superposed on the velocity field. This has an effect similar to using a frame of reference that moves at constant speed.

7.2. Numerical results for Crow instability

For the Crow instability test case, simulations have been performed for different meshes and Reynolds numbers. The following boundary conditions are imposed on the computational domain:

- Dirichlet boundary conditions at infinity, i.e. at the 4 domain boundaries in the y and z co-ordinate directions shown in Figure 5, for the Helmholtz equations for the velocity updates as well as the pressure Poisson equation.
- Neumann boundary conditions in the periodic axial direction, i.e. at $x = 0$ and L_x, for the Helmholtz equations for the updates of the velocity in the y and z co-ordinate directions and for the pressure Poisson equation.
- Dirichlet boundary condition for the velocity component in the x co-ordinate direction. At $x = 0$ and L_x, velocity component in the axial direction is set to 0.

The parameters of the different simulations are shown in Table III. The mesh size shown in Table III is the mesh size of the unpartitioned mesh. All simulations have been performed on 4 processors of an SGI/Origin 2000 (with R10K processors). For this purpose, 4 partially-overlapping sub-domains have been used. The sixth column in Table III gives the total number of grid points. The total amount of internal memory required for each simulation is shown in the last column of Table III. Figure 7 shows the normalized amplitude of the sinusoidal perturbation (normalization with the amplitude of the initial amplitude used in the simulation) versus non-dimensional time T. The amplitude of the perturbation in the Navier–Stokes simulations is determined from the variation in axial direction of the lateral location of the centres of the vortex filament. The computed growth is compared with the exponential growth rate predicted by the inviscid asymptotic theory of Widnall *et al.* [17]. At later times, i.e. $T > 2$, the growth rate of the instability mode in the Navier–Stokes simulations is reduced. This can be attributed to the effect of viscosity, which reduces the circulation of the vortices and

Table III. Parameters for Crow instability test case.

Re_Γ	$\Gamma(m^2/s)$	$\nu(m^2/s)$	Mesh	# blocks	# grid points	Memory (MB)
6.7×10^3	10.0	1.5×10^{-3}	81^3	4	6.04×10^5	148
6.7×10^5	10.0	1.5×10^{-5}	81^3	4	6.04×10^5	148
6.7×10^5	10.0	1.5×10^{-5}	121^3	4	1.93×10^6	384

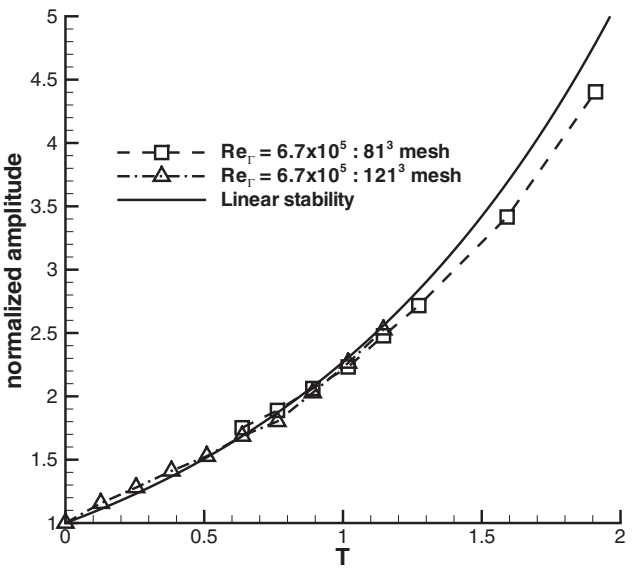

Figure 7. Growth of Crow instability mode: effect of mesh resolution.

increases the radius of the vortex cores. At later stages shown in Figure 7, the amplitude of the perturbation is such that predictions of linear stability theory are no longer valid. Comparing the results for the 81^3 and 121^3 meshes, it can be seen that for both meshes the computed growth curves are nearly identical and show a very good agreement with the result from linear stability theory. Therefore in the period of the time considered, the discretization method for the Navier–Stokes equations leads to a sufficient resolution of the evolving flow field for Reynolds numbers as high as 6.7×10^5. The mesh requirements found here are consistent with those reported for the high-order accurate Navier–Stokes method of Reference [14].

8. CONCLUDING REMARKS

Presented are a fourth-order discretization method for Helmholtz equations and the application of this discretization method in a numerical method for time-accurate simulations of the three-dimensional incompressible Navier–Stokes equations. The fourth-order accuracy of the Helmholtz solver is demonstrated in a mesh-refinement study for a simple model problem with

an exact solution. Furthermore, the dependence of the magnitude of the numerical error and the convergence behaviour of the iterative solution method on the numerical scheme parameter β is studied. The resolving properties of the Navier–Stokes solver are shown in a numerical study of a time-dependent three-dimensional flow, i.e. the evolution of a three-dimensional instability in a counter-rotating vortex pair. The growth rate of the instability obtained from the numerical simulations shows a good agreement with available analytical predictions from linear stability theory.

ACKNOWLEDGEMENTS

The authors thank the Netherlands Organisation for Scientific Research (NWO) for support to the first author through the foundation of Fundamental Research of Matter (FOM) under Grant 97-PR-1598 'Dynamics of aircraft wake vortices in the atmosphere'.

REFERENCES

1. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Fluid Mechanics* 1985; **162**:339–363.
2. Ananthakrishnaiah U, Manohar R, Stephenson JW. Fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients. *Numerical Methods for Partial Differential Equations* 1987; **3**:229–240.
3. Gupta MM, Kouatchou J. Symbolic derivation of finite difference approximations for the three-dimensional Poisson equations. *Numerical Methods for Partial Differential Equations* 1998; **14**:595–606.
4. Spotz WF, Carey GF. A high-order compact formulation for the 3D Poisson. *Numerical Methods for Partial Differential Equations* 1996; **12**:235–243.
5. Steijl R. Computational study of vortex pair dynamics. *Ph.D. Thesis*, University of Twente, 2001.
6. Crow SC. Stability theory for a pair of trailing vortices. *AIAA Journal* 1970; **8**(12):2172–2179.
7. Stroustrup B. *The C++ Programming Language.* Addison-Wesley: Reading, MA, 1986.
8. Golub GH, Van Loan CF. *Matrix Computations* (3rd edn). The Johns Hopkins University Press: Baltimore, MD, 1996.
9. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in C*: *The Art of Scientific Computing.* Cambridge University Press: Cambridge, 1994.
10. Lele SK. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics* 1992; **103**:16–42.
11. Kravchenko AG, Moin P. On the effect of numerical errors in Large Eddy Simulations of turbulent flows. *Journal of Computational Physics* 1997; **131**:310.
12. Steijl R, Hoeijmakers HWM. Efficient implementation of a parallel high-order accurate incompressible Navier–Stokes solver. *AIAA Paper 2000–2676*, 2000.
13. Moore DW. Finite amplitude waves on aircraft trailing vortices. *Aero Quarterly* 1972; **19**:307.
14. Robins RE, Delisi DP. Numerical simulations of three-dimensional trailing vortex evolution. *AIAA Journal* 1997; **35**(9):1552–1555.
15. Garten JF, Werne J, Fritts DC, Arendt S. Direct numerical simulations of the Crow instability and subsequent vortex reconnection in a stratified fluid. *Journal of Fluid Mechanics* 2001; **426**:1–45.
16. Zheng ZC. Thin-tube simulations for sinusoidal instability in a counter-rotating vortex pair. *International Journal for Numerical Methods in Fluids* 2002; **39**:301–324.
17. Widnall SE, Bliss D, Zalay A. Theoretical and experimental study of the stability of a vortex pair. In *Aircraft Wake Turbulence and its Detection*, Olsen JH, Goldburg A, Rogers M (eds), Plenum Press: New York, 1970; 305–329.

