Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 208 (2005) 675–690

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A staggered compact finite difference formulation for the compressible Navier–Stokes equations

Bendiks Jan Boersma [*]

Laboratory for Aero and Hydrodynamics, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

Received 12 July 2004; received in revised form 4 January 2005; accepted 3 March 2005
Available online 14 April 2005

Abstract

In this paper, a compact high order (up to 12th order) numerical method to solve the compressible Navier–Stokes equations will be presented. A staggered arrangement of the variables has been used. It is shown that the method is not only very accurate but numerically also very stable even in the case that not all the energy containing scales in the flow are resolved. This in contrast to standard (collocated) compact finite difference methods. Some results for a turbulent non-reacting and a reacting jet with a Reynolds number of 10,000 and a Mach number of 0.5 are reported.

1. Introduction

High order compact finite difference schemes, see for instance [8,3,10,1], have been widely used for the solution of compressible Navier–Stokes equations, see e.g. [11]. The advantage of compact finite difference schemes over standard finite difference schemes is the good accuracy for a large range of wave numbers in combination with low numerical diffusion and small dispersion errors. That is why sometimes the phrase ‘‘spectral resolution’’ is used in combination with compact finite difference schemes. Furthermore, compact finite difference methods are computationally much more efficient than finite element methods (e.g. spectral elements) with the same order of accuracy.

However, when compact finite difference schemes are used for problems in which non-linear terms play an important role, for instance a turbulent flow, short wave numerical instabilities occur due to aliasing errors. These short waves are due to low molecular damping on the grid scales. Decreasing the grid spacing

[*] Corresponding author. Tel.: +31 15 278 7979; fax: +31 15 278 2947.
E-mail address: b.j.boersma@wbmt.tudelft.nl.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.03.004

will, in general, solve this problem. However, for a large three dimensional calculation this is in general not a desirable option.

Common practice is to remove the numerical instability by spatially filtering the solution with a high order (compact) filter [8,7]. In some studies this filtering is performed every integration step, in other studies this filtering is performed every few hundred steps. Filtering every time step will lead to a large amount of artificial diffusion while filtering say every 100 time steps will lead to spurious fluctuations in time. To avoid these short wave instabilities some researchers have used high order upwind schemes [16]. However, this type of discretization will most likely influence the smaller turbulent scales due to the added numerical viscosity.

Recently, [12] published a compact finite difference method in which the variables were arranged in a staggered formulation, they show that this method is superior to the standard compact finite difference method for a few standard test cases.

In the present paper, we will introduce a staggered finite difference method which differs slightly from the one proposed by [12]. The method we formulate is a straightforward extension of the classical marker and cell method of [5]. It will be shown that this method is able to handle a wide range of physical phenomena such as combustion and aeroacoustics, with rather low numerical resolutions, i.e. it is not necessary to resolve all flow features for a stable numerical calculation. This is an indication that the present method is a good candidate for a large Eddy simulation (LES).

The organization of this paper is as follows. In Section 2, we will introduce the governing equations. In Section 3, we will discuss the numerical method in detail. Next in Section 4, we will give some numerical examples. Finally, we will draw some conclusions in Section 5.

2. Governing equations

In this section, we will give the governing equations for mass, momentum, energy and scalar transport in a compressible fluid [14]. The equation for conservation of mass reads:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}\rho u_i = 0. \tag{1}$$

In which ρ is the fluids density and u_i the velocity vector. The equation for conservation of momentum reads:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}[\rho u_i u_j + p] = \frac{\partial}{\partial x_j}\tau_{ij}. \tag{2}$$

In which p is the pressure and τ_{ij} the viscous stress tensor. Here, we will consider Newtonian flows only and the components of the stress tensor can be written as:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\right),$$

where μ is the dynamic viscosity of the fluid. Which is in the present study assumed to be constant. The governing equation for the total energy E which is the sum of the internal energy $\rho C_v T$ and the kinetic energy $\rho u_i u_i/2$ reads:

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j}(u_j + p)E = \frac{\partial}{\partial x_i}\kappa\frac{\partial T}{\partial x_i} + \frac{\partial}{\partial x_j}u_i\tau_{ij} + h_\mathrm{f}\omega. \tag{3}$$

In which $E = \rho C_v T + \rho u_i u_i/2$ is the total energy, κ the thermal diffusion coefficient, ω a source due to the chemical reaction and h_f the formation enthalpy of a chemical reaction. The thermodynamic quantities P, ρ and T are related to each other by the equation of state for an ideal gas

$$P = \rho R T,$$

where R is the gas constant.

The transport of the chemical species Y^j is governed by the following equation

$$\frac{\partial \rho Y^j}{\partial t} + \frac{\partial}{\partial x_j} \rho u_j Y^j = \frac{\partial}{\partial x_i} \kappa_j \frac{\partial Y^j}{\partial x_i} - \omega, \tag{4}$$

where κ_j is the diffusion coefficient of the jth chemical species. For the source term we use Arrhenius kinetics, i.e.

$$\omega = A[\rho Y^j][\cdots] \exp(-KR/T). \tag{5}$$

In which K is the activation energy of the chemical reaction and A is the pre-exponential factor. The actual form of Eq. (5) depends on the composition of the chemical system, see for instance [14].

All the variables in the equations given above are made non-dimensional using the ambient speed of sound c_∞ as reference velocity scale, the ambient density ρ_∞ as reference density, $\rho_\infty c_\infty^2$ as reference pressure, c_∞^2/C_p as reference temperature, and ambient values for the chemical species. The resulting non-dimensional numbers are the Reynolds, Prandtl, Schmidt and Mach numbers.

3. Computational method

In this section, we will describe the computational method and discuss some implementation issues.

The governing equations given in Section 2 are discretized on a staggered non-uniform grid in physical space. For the calculation of the derivative of a certain function f on this non-uniform grid the function f is mapped on a uniform grid with help of the following transformation

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}x} \rightarrow \frac{\partial f}{\partial x}\left(\frac{\mathrm{d}X}{\mathrm{d}x}\right)^{-1} = \frac{\partial f}{\partial X}, \tag{6}$$

where x is the coordinate on the non-uniform mesh and X the coordinate on the uniform mesh. A known analytical function is used for X so that the $\mathrm{d}X/\mathrm{d}x$ can be calculated exactly. The restriction on the transformation given by Eq. (6) is that $\mathrm{d}X/\mathrm{d}x$ should not become zero inside the computational domain.

We have chosen for a staggered grid in which the scalar variables are located in the center of the cell and vector quantities on the cell faces. This is illustrated in Fig. 1, where we show a grid cell in 2D. The actual discretization is of course performed in three dimensions. The arrangement of the variables is similar to the one used by Harlow and Welch [5] in their classical paper. The staggered arrangement of course leads to some additional work. For instance, if we want to calculate $\partial \rho u/\partial x$ in Eq. (1) we first have to interpolate the density, which is a scalar and therefore located at the cell center, to the face of the cell. With the interpolated value of the density we can form the product ρu. Subsequently, we can calculate the derivate $\partial \rho u/\partial x$ at the cell face and interpolate the result to the center of the cell or we can try to evaluate the derivative directly at the center of the cell using information on the faces. The latter is what we will do in the present study.

3.1. First derivative

In the governing equations (1)–(5) only first order derivatives appear. So we only have to consider a formula for the first derivative. In some studies, see for instance [12], the term on the right hand side of Eq. (2) is recasted in a non-conservative form using the chain rule. This non-conservative form, in which second order derivatives appear is used because of additional the damping of high wave number components.

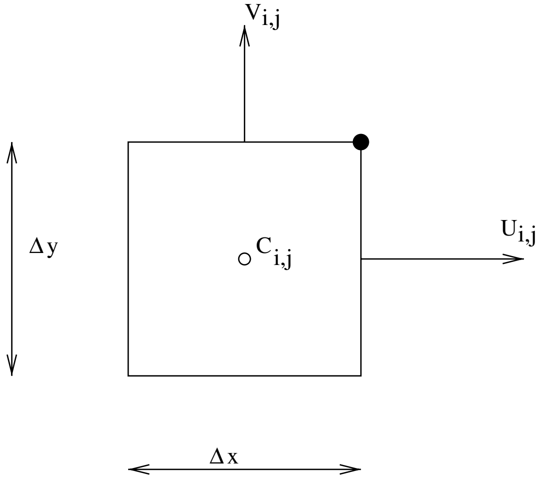

Fig. 1. A two dimensional cell of the computational grid. The actual discretization is performed in three dimensions.

In this study, we have directly discretized the right hand side of Eq. (2) and therefore we only need an expression for the first derivative. It turned out that the additional damping due to the second order derivatives of the velocity are not necessary for stability in the present method.

The first derivative will always be evaluated on a staggered uniform grid as is shown in Fig. 2. If the variables are known at points $i + 1/2, i + 3/2, \ldots$ we can use the following formula to calculate the derivatives at the points $i, i + 1, \ldots$

$$a(f'_{i+1} + f'_{i-1}) + f'_i = \frac{b}{\Delta X}(f_{i+1/2} - f_{i-1/2}) + \frac{c}{\Delta X}(f_{i+3/2} - f_{i-3/2}) + \frac{d}{\Delta X}(f_{i+5/2} - f_{i-5/2}) + \frac{e}{\Delta X}(f_{i+7/2} - f_{i-7/2}). \tag{7}$$

In which f'_i is derivative of f with respect to X in point i and ΔX is the (uniform) grid spacing. The coefficients in the equation above are obtained by Taylor expansions around grid point i. With the five coefficients a, b, c, d and e in Eq. (7), we can obtain an 10th order accurate formulation. The values for a, b, c, d and e for this 10th order scheme are (obtained with the Maple software package):

$$a = 49/190,\ \ b = 12985/14592,\ \ c = 78841/364800,\ \ d = -343/72960,\ \ e = 129/851200.$$

This formula can only be used far away from the boundaries. Closer to the boundaries we have to make the stencil smaller, i.e. $e = 0$. The coefficients for the resulting 8th order scheme read:

$$a = 25/118,\ \ b = 2675/2832,\ \ c = 925/5664,\ \ d = -16/28320,\ \ e = 0.$$

Even closer to the boundary we have to use the 6th and 4th order formulation

$$a = 9/62,\ \ b = 63/62,\ \ c = 17/186,\ \ d = 0,\ \ e = 0,\ \ \mathrm{O}(\Delta X^6),$$

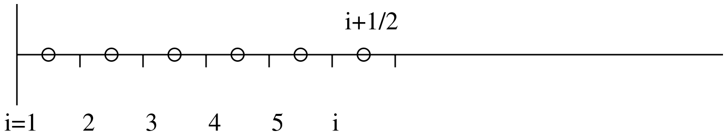

Fig. 2. The distribution of the grid points.

$$a = 1/22,\ \ b = 12/11,\ \ c = 0,\ \ d = 0,\ \ e = 0,\ \ \mathrm{O}(\Delta X^4).$$

At the boundary itself we use a one sided 3rd order accurate formulation

$$f_i' + 23f_{i+1}' = \frac{1}{\Delta X}\left(-25f_{i+1/2} + 26f_{i+3/2} - f_{i+5/2}\right).$$

The approximations above lead to a tridiagonal system with can be inverted easily and efficiently.

In the derivation of Eq. (7) we assumed that the variable was known in between the points i and $i + 1$. In the case that the variables are known on the nodes i and that we want to know the derivative at point $i + 1/2$ we can use the same formula but we have to shift the data over half a grid cell and we have to solve for $n - 1$ instead of n grid points.

3.2. Interpolation and extrapolation

Apart from the relation for the first derivative we also need an interpolation procedure to interpolate variables from the locations i to locations $i + 1/2$ and vice versa. This should preferably be done with a method which has the same formal accuracy as the method which is used to obtain the derivatives. Here, we consider the following compact interpolation rule

$$f_i + a(f_{i+1} + f_{i-1}) = b(f_{i+1/2} + f_{i-1/2}) + c(f_{i+3/2} + f_{i-3/2}) + d(f_{i+5/2} + f_{i-5/2}) + e(f_{i+7/2} + f_{i-7/2}). \tag{8}$$

In the interior we require again 10th order accuracy resulting in the following values for the coefficients a, b, c, d and e (again obtained with the Maple software package):

$$a = 7/18,\ \ b = 1225/1536,\ \ c = 49/512,\ \ d = -7/1536,\ \ e = 1/4608.$$

Closer to the boundaries the stencil has to be made smaller, i.e. $e = 0$:

$$a = 5/14,\ \ b = 25/32,\ \ c = 5/64,\ \ d = -1/448,\ \ e = 0,\ \ \mathrm{O}(\Delta X^8).$$

Even closer to the boundary we have to use 6th an 4th order schemes, i.e.

$$a = 3/10,\ \ b = 3/4,\ \ c = 1/20,\ \ d = 0,\ \ e = 0,\ \ \mathrm{O}(\Delta X^6),$$

$$a = 1/6,\ \ b = 2/3,\ \ \mathrm{O}(\Delta X^4).$$

At the boundary we use again a one sided 3rd order accurate formulation (actually this is an extrapolation instead of an interpolation):

$$f_i = 15/8f_{i+1/2} - 5/4f_{i+3/2} + 3/8f_{i+5/2}.$$

The formulas above result in a tridiagonal system which can be inverted easily. Again we can use the same rule to interpolate from points f_i to points $f_{i+1/2}$ by shifting the data over half a grid cell and evaluating for $n-1$ instead of n points.

The mapping, given by Eq. (6) takes care of the effect of the non-uniform grid on the derivatives. For the function f itself we cannot use such a mapping. Formally we could construct an interpolation rule for non-uniform grids. In which the difference in grid-spacings between points $i,i - 1$ and $i,i + 1$ are taken into account. However, in the literature it is shown, see e.g. [17], that with such a non-uniform interpolation the kinetic energy $\rho u_i u_i/2$ is not conserved numerically. [17] shows that only with a uniform (symmetric) interpolation kinetic energy is conserved. Conservation of kinetic energy is a good start for a stable numerical solution.

Therefore, we use the interpolation rule (8) also on non-uniform grids. Formally, this will reduce the local truncation error of the scheme. However, this is the only way to strictly satisfy conservation of kinetic energy on a computational grid, see e.g. [17].

In the following paragraphs we will explain how all the terms in the equations of motion are discretized.

3.3. Discretization of mass

The equation for conservation of mass, Eq. (1) is discretized around the center of the cell, see Fig. 1. First the density is interpolated from the cell center to the cell faces with help of Eq. (8). The fluxes ρu_i on the faces can be calculated. Subsequently, the derivatives of $\partial \rho u_i / \partial x_i$ are calculated at the cell center using Eq. (7) and the equations are advanced in time. The fluxes ρu_i are stored for further usage in the discretization of the other equations.

3.4. Discretization of scalars

For the convective terms in the scalar equations (total energy and species) we use the following procedure. Firstly, the scalars ρY^j and $\rho C_v T$ are interpolated to the cell faces, using the interpolation rule (8). Subsequently, the interpolated scalars are multiplied with the velocity component on the faces. Eq. (7) is then used to calculate the derivative at the cell center. For the diffusive terms, Eq. (7) is used to calculate the gradients, $\partial Y^j / \partial x_i$ and $\partial T / \partial x_i$ at the cell faces. In case of non-constant material properties the material properties μ, κ, κ^j are interpolated to the cell faces, using Eq. (8), and the diffusive fluxes are formed. Next, Eq. (7) is used to calculate the derivative of the diffusive fluxes at the cell-center. Once the convective and diffusive terms are known at the cell-center the equations can be advanced in time.

3.5. Discretization of momentum

The momentum equations are discretized around the velocity points which are located at the cell face. The momentum equation in the x-direction is discretized around the u point in Fig. 1 and the momentum equation in the y-direction is integrated around the v point in Fig. 1.

The convective terms $\partial \rho u_i u_i / \partial x_i$ are calculated by interpolating the velocity components from the cell faces to the cell centers. At the cell centers the product $\rho u_i u_i$ is formed and Eq. (7) is used to calculate the derivate at the cell face. The terms $\partial \rho u_i u_j / \partial x_j$ are calculated by interpolating the flux ρu_i (which is still available from the integration of the equation for conservation of mass) to the corner of the grid cell (the black dot in Fig. 1). The velocity u_j is also interpolated to the same point and the product $\rho u_i u_j$ is formed. Again Eq. (7) is used to calculate the derivative at the cell face. The pressure gradient can be calculated directly with Eq. (7) because of the staggered arrangement. The diffusive terms are calculated in the same manner as the convective terms. After the evaluation of the spatial derivatives the equations can be advanced in time.

3.6. Boundary conditions

The formulation of the boundary conditions is of course problem specific. In this paper, we will consider a simple jet diffusion flame and we will describe the boundary treatment for this problem. A sketch of the geometry we consider is shown in Fig. 3.

The jet flow is to be considered subsonic, i.e. the velocity is always smaller than the speed of sound. At the jet inlet we specify the velocity, density and temperature profiles. For a subsonic flow this type of boundary condition is mathematically ill-posed. Therefore, we have added a small region to the domain in which an artificial convection velocity is added to the equations to make the flow locally supersonic. This is a well known procedure, see for instance [4]. At the side boundaries we set the primitive variable to their reference values and at the outflow we use a simple characteristic boundary condition. Surprisingly, this

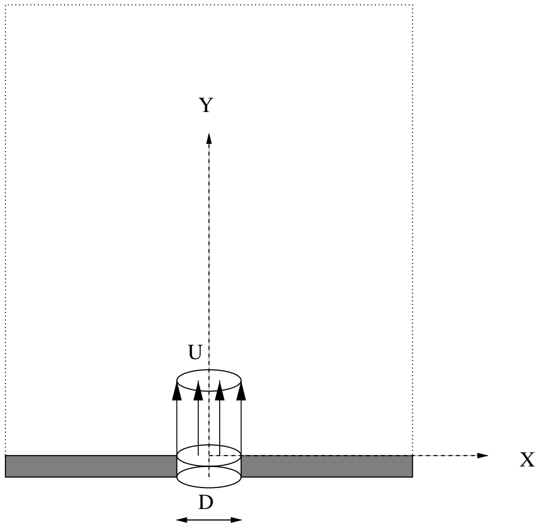

Fig. 3. A sketch of the geometry: A cylindrical jet with velocity U flowing through an orifice with diameter D into a Cartesian box.

gives in the staggered formulation no significant reflections. For a combustion problem, as is discussed here, reflections on the boundaries are not so important. For an acoustic problem, where reflections should be avoided, probably some additional damping layers will be necessary.

The boundary conditions are always applied at the faces of the grid cells. For instance, if we want to apply a specific value for the velocity on the top face of the cell shown in Fig. 1 we can put V_{ij} equal to this value. To apply the boundary conditions to u velocity component we first have to use the extrapolation/interpolation rule (Eq. (8)) to extrapolate the velocity to the point indicated by the black dot in Fig. 1. This extrapolated value is set to zero and the interpolation rule (8) is used to calculate a new value for u_{ij}. Due to the implicit nature of the method this means also that the interior points will be slightly modified.

3.7. Time advancement

For the time advancement we use standard integration methods for Navier–Stokes equations, like a 4th order Runge–Kutta method or a 3rd order Adams–Bashforth method. The Runge–Kutta method is computationally more efficient and we will in general use this method.

3.8. Parallel implementation

The algorithm is implemented in Fortran 77/90 using the message passing interface (MPI) for the parallelization. Two data distributions are used one in which the full data set denoted by $N_x \times N_y \times N_z$ is distributed on N_{cpu} processors as $N_x \times N_y \times N_z/N_{\text{cpu}}$ and one in which the data is distributed as $N_x \times N_y/N_{\text{cpu}} \times N_z$. The redistribution of the date between the two distributions is performed with help of the MPI-routine MPI_ALLTOALL. The calculation of derivatives, and interpolations in the x and y directions are performed on the first data distribution. Derivatives and interpolation in z-direction are performed on the second data distribution. The code scales reasonable well as can be seen from Table 1 in which we show CPU times for a grid of 128^3 on a SGI-ORIGIN 3800 parallel processor.

Table 1
CPU times on a SGI-ORIGIN 3800

N_{cpu}	time/Δt	time/$(\Delta t) \cdot N_{\mathrm{cpu}}$
1	409 s	409 s
2	208 s	416 s
4	99 s	397 s
8	42 s	333 s
16	19 s	304 s
32	12 s	371 s

N_{cpu} denotes the number of processors used, time/Δt is the wall clock time for a single Runge–Kutta 4 timestep. The last column is the CPU time which is actually charged by the operating system. The calculation using 16 CPU's is thus the most efficient.

3.9. Numerical tests

In this section, we will present the results of some basic test which we performed with the staggered formulation.

It is common practice to calculate the modified wave number of a numerical scheme. Modified wave numbers are derived by substituting exp(iwx) in the difference equation. For instance, for the second order central difference

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{f_{i+1} - f_{i-1}}{2\Delta x}.$$

We find the following expression after the substitution of exp(iwx)

$$\frac{\exp[\mathrm{i}w(x + \Delta x)] - \exp[\mathrm{i}w(x - \Delta x)]}{2\Delta x} = \exp(\mathrm{i}wx)\frac{\mathrm{i}\sin(w\Delta x)}{\Delta x} = \mathrm{i}w' \exp(\mathrm{i}wx),$$

where w is the wave number and w' the so-called modified wave number. Depending on the finite difference scheme w' can have different forms. $w' = w$ denotes exact differentiation. It is straightforward to derive the modified wave number for the staggered scheme, Eq. (7) and the results looks promising. However, if in an actual staggered discretization df/dx is calculated, this requires for the convective terms (which are in a turbulent flow very important) always an interpolation in combination with a staggered compact finite difference. The same is true for conventional (second order) staggered finite difference methods. The modified wave number of this combined operation is not so easily to calculate by hand. Therefore, we calculate the modified wave number numerically by assuming a single wave and differentiating this wave numerically for different values of Δx and compare this with the exact result. The advantage of this numerical approach to calculate the modified wave number is that also the effect of the lower accuracy near the boundaries is taken into account. In Fig. 4, we show the modified wave numbers of the staggered compact finite differences schemes combined with interpolation procedure with the same order. For comparison we also include the modified wave number for a collocated 6th order compact finite difference method [8]. It should be noted that in the numerical calculation of the modified wave number we used lower order accuracy near the boundaries of the domain as outlined in Section 3.1.

The figure shows that the 10th order staggered scheme has the best properties. Although differences are very small and can hardly be considered significant. The most important conclusion we can draw from Fig. 4 is that the staggered schemes have similar resolving capabilities as the collocated schemes and that no spectacular improvement in the results should be expected.

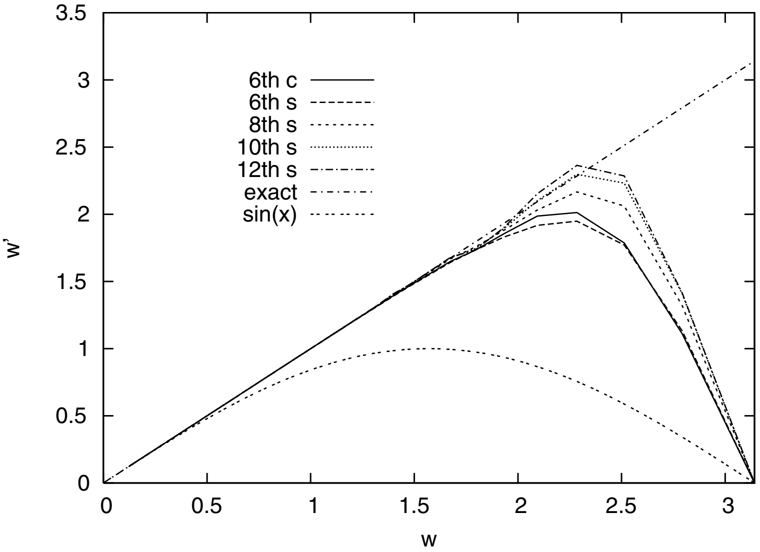

Fig. 4. The modified wave number for the staggered compact finite difference methods (denoted by s). For comparison we have also included the 6th order collocated compact scheme (denoted by c).

Secondly, let us consider the differential equation

$$\frac{\partial C}{\partial t}+\frac{\partial uC}{\partial x}=0, \quad C(x,0)=0,\ \ C(0,t)=1,\ \ C(L,t)=0,\ \ u(x,t)=1. \tag{9}$$

The differential equation is discretized on a staggered grid, see Fig. 2. The interpolation procedure is used to interpolate the scalar C to the position of the velocity point u. The product uC is formed and Eq. (7) is used to calculate the derivative at the position of the original C point. The equations are subsequently integrated in time with a 3rd order Adams–Bashforth scheme. Boundary conditions are imposed by extrapolating the scalar field to the boundary point. The scalar on the boundary is set to the desired value and the result is interpolated back to the original C point. This procedure is fairly complicated for such a simple equation. However, this is exactly the same strategy as we use for the Navier–Stokes equations.

In Fig. 5, we show the numerical solution of Eq. (9) for four different orders of the spatial discretization at time $t = 12$. For comparison we also included the result for a non-staggered grid with a standard 6th order compact finite difference. All the numerical solutions show a similar behavior, a slight overshoot/undershoot. The non-staggered 6th order compact finite difference is better than the staggered 6th order. Increasing the order of the staggered schemes slightly improves the results. The performance of the staggered scheme is a somewhat disappointing when compared to the collocated scheme. This is due to the additional interpolations which are necessary in the staggered formulation. A stability analysis for the staggered compact finite difference scheme is presented in Appendix A.

From the test presented in this section, we can conclude that the staggered formulation has similar resolving capabilities as the collocated formulation. The only added value of the staggered formulation is the increased numerical stability. The cost we have to pay for this is of course additional CPU time due to the many interpolations which have to be carried out on the staggered grid. The increased numerical stability is illustrated by the next example which is impossible to calculate with a collocated compact finite difference method.

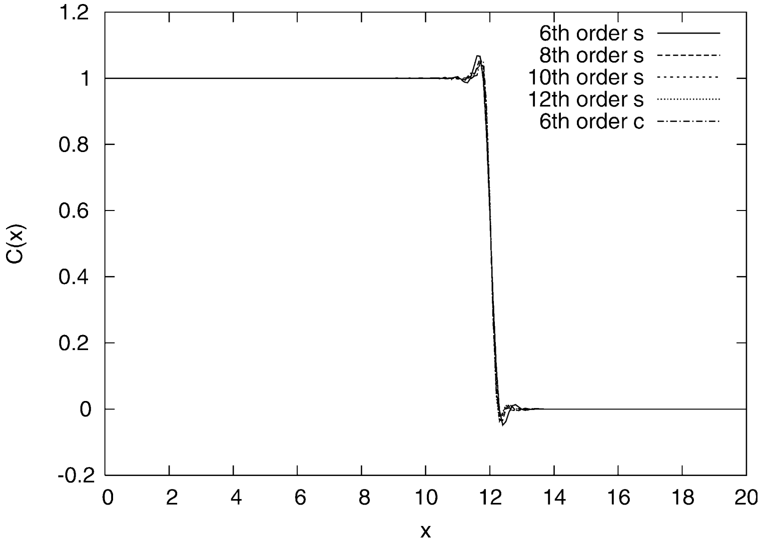

Fig. 5. Advection of a passive scalar with different orders of the spatial discretization, (s) denotes a staggered formulation and (c) a collocated formulation.

4. Jet flow

We have performed numerical simulations of a round turbulent jet with a Reynolds number based on the jet nozzle diameter and velocity of 10,000. The numerical resolution consisted of 192 × 128 × 128 grid points in the x (downstream) and y, z (cross stream) directions.[1] This grid is certainly not fine enough to resolve all the turbulent scales in the flow. However, it is well known that this type of flow, i.e. free shear flows, is governed by large scales and resolving these large scales will produce accurate flow statistics. At the plane $x = 0$ the following relation is used for the velocity in the x-direction:

$$u(0,y,z) = Ma\left[\frac{1}{2} - \frac{1}{2}\tanh\left(30\sqrt{y^2+z^2} - \frac{1}{4}\right)\right], \tag{10}$$

where Ma is the Mach number. The other two velocity components at the plane $x = 0$ are set to zero. The density and temperature are equal to the reference values. This is a well posed boundary conditions, because the flow is locally supersonic due to the additional supersonic convection velocity at in and outflow (see also paragraph 3.6). To trigger the transition to a turbulent state small white noise perturbations with a maximum amplitude of $1 \times 10^{-4}Ma$ are superimposed on the velocity components in the y and z directions

[1] The following relations have been used for the grids in x and y directions (the grid in the z-direction is identical to the grid in the y-direction):

$$x_i = \left(\frac{(i-0.5)3.75}{192} + 1\right)^2, \quad i = 1,\ldots,192,$$

$$y_j = 2.2\cdot\frac{(j-63.5)\cdot 2.75^3}{128} + \frac{(j-63.5)\cdot 2.75}{128}, \quad j = 1,\ldots,128.$$

at the inflow plane, $x = 0$ The equations are integrated in time with the 4th order Runge–Kutta method. The CFL number was $\approx$0.5 and the number of time steps $\approx$20,000.

4.1. Non-reacting case

First, we will consider a non-reacting jet, i.e. the chemical species can be considered as passive scalars. In Fig. 6, we show a snapshot of the passive scalar and a zoom-in close to the jet nozzle. The lines in this figure denote the (rather course) computational grid in physical space. Near the jet nozzle the flow is laminar and a few diameters downstream a Kelvin–Helmholtz instability develops, eventually resulting in a fully three dimensional turbulent flow. It should be noted that the scalar is discretized with a fully central method, i.e. no artificial dissipation. Low order central methods would give a considerable over and undershoot, resulting in significant negative concentrations. Common practice is to remove these negative concentrations using a TVD scheme, see for instance [15,9]. The present high order central method, with the associated high resolution on the small scales, does not produce significant over and undershoots.

In Fig. 7, we compare the decay of the jet centerline velocity, labeled "Fine", with other simulations [2], and experiments [13,6]. The decay rate of the present calculation is in good agreement with earlier experimental and numerical studies. To investigate the grid (in)dependence of the solution we have also performed a calculation on a resolution of $128 \times 80 \times 80$. The result of this simulation is labeled "Coarse" in Fig. 7. Close to the jet nozzle there is a considerable difference between the coarse and fine grid results. The reason for this is that on the coarse grid the transition from a laminar to a turbulent state is not properly resolved. Far downstream of the jet nozzle, the turbulent length scales are large and both grids are able to resolve these scales in some detail. The difference between the $192 \times 128 \times 128$ result and results reported in the literature can be attributed to compressibility effects ($Ma = 0.5$ vs. $Ma = 0$), the size of the computational box (no special treatment has been used to calculate the entrainment of the jet) and to some extend grid dependence. The grid dependence could be investigated by performing a simulation on a much finer grid, but this is beyond the scope of the present paper.

4.2. Reacting jet

Now we will consider the case in which there is a chemical reaction with heat release. We will consider the simple chemical reaction:

$$\mathrm{F} + \mathrm{O} \rightarrow \mathrm{P}.$$

The jet fluid is assumed to be pure fuel, F and the surrounding is to be assumed pure oxidizer, O. The reaction rate of this chemical reaction is given by the Arrhenius equation

$$\omega = A(\rho^2)[\mathrm{F}][\mathrm{O}]\exp(-K/RT).$$

The fuel and oxidizer concentration [F] and [O] are normalized between 0 and 1. In the jet nozzle the fuel concentration is 1 and the oxidizer concentration is 0. In the free stream the fuel concentration is 0 and the oxidizer concentration is 1. In the region where fuel and oxidizer are mixed the product [F][O] is non-zero. If the temperature in this region is sufficiently high, ω will be non-zero and there will be a heat production $h_f\omega$. The value of ω depends of course on the pre exponential factor A but also on the value of K, i.e. the activation energy. For, most fuels the value of the activation is pretty high. Thats why we aim at high values of KR, i.e. the chemical reaction is only significant when T is rather high. The Reynolds and Mach numbers are the same as in the previous case, i.e. 10,000 and 0.5.

In Fig. 8, we show the density distribution in the jet. Cold fluid (fuel) is entering the domain through the jet nozzle. Due to the chemical reacting heat is released and the density drops. Realistic density ratios (unburned/burned) are 4–6. The present model is able to handle density ratios in this range (see legend Fig. 8). In

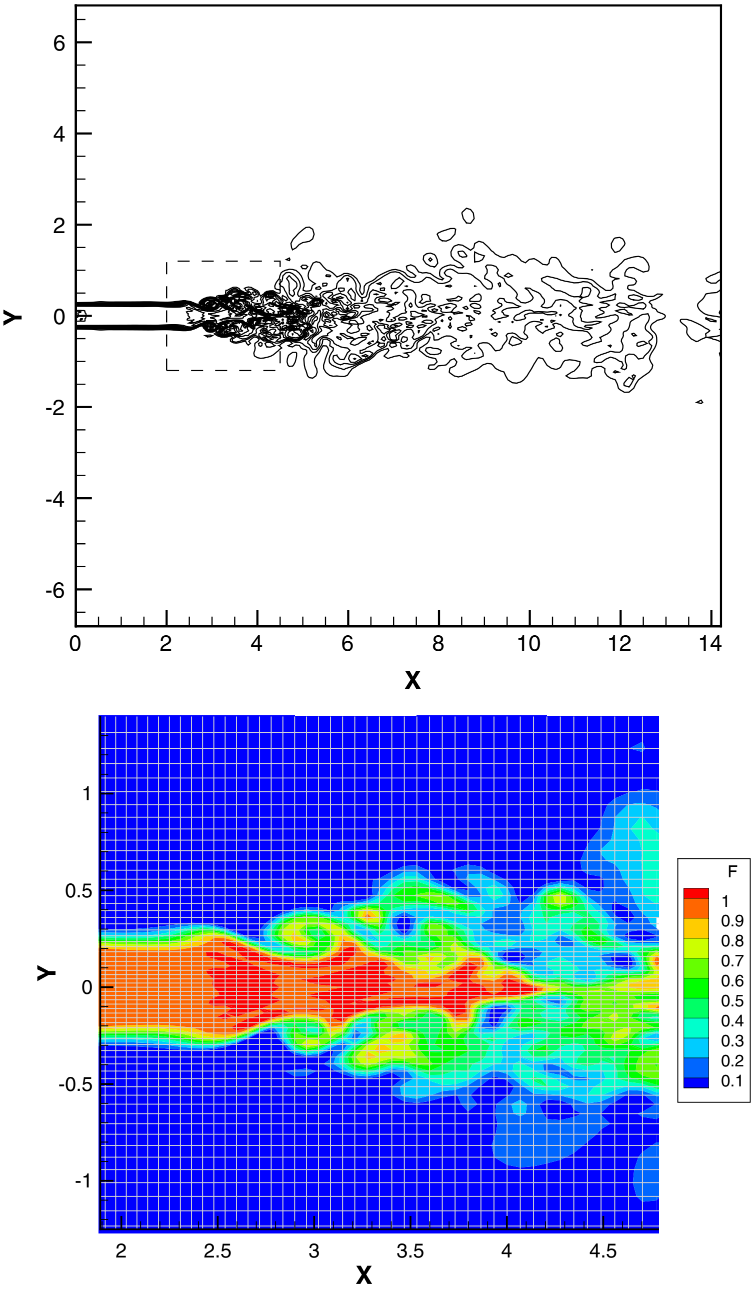

Fig. 6. A snapshot of the passive scalar close to the jet nozzle, for the non-reacting case. Top: The full computational domain. Bottom, zoom in of an area close to the jet nozzle (box in the top figure). The lines denote the computational grid.

Fig. 9, we show the fuel F distribution in the jet. If we compare the distribution with the distribution in Fig. 6 for the non-reacting case we see a considerable decrease in F in the downstream region. For a realistic flame one would expect a zero fuel concentration far downstream. That is not the case in the present flame. This is

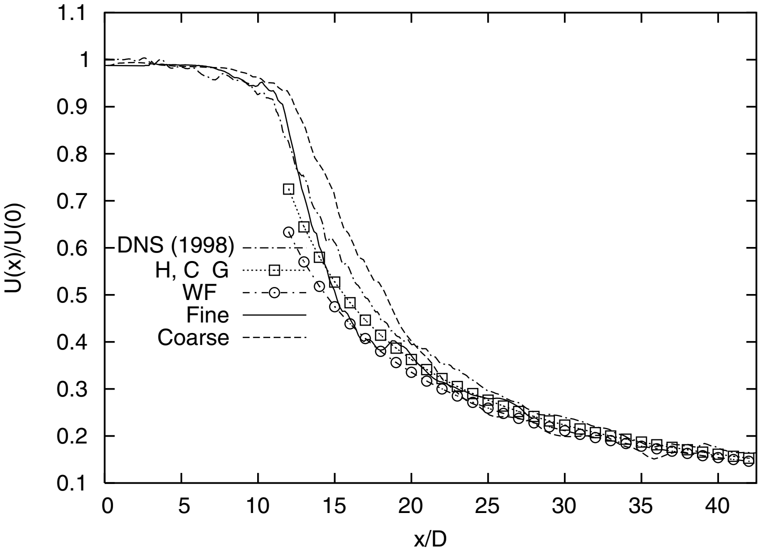

Fig. 7. The decay of the jet centerline velocity as a function of the distance to the jet nozzle, for the non-reacting jet. DNS (1998) denotes the result form [2] (incompressible flow). “Fine” denotes the result with the present model ($Ma = 0.5$) on a grid of $192 \times 128 \times 128$. “Coarse” denotes the results with the present model on a resolution of $128 \times 80 \times 80$. The other curves are the results given in the experimental papers of [13,6].

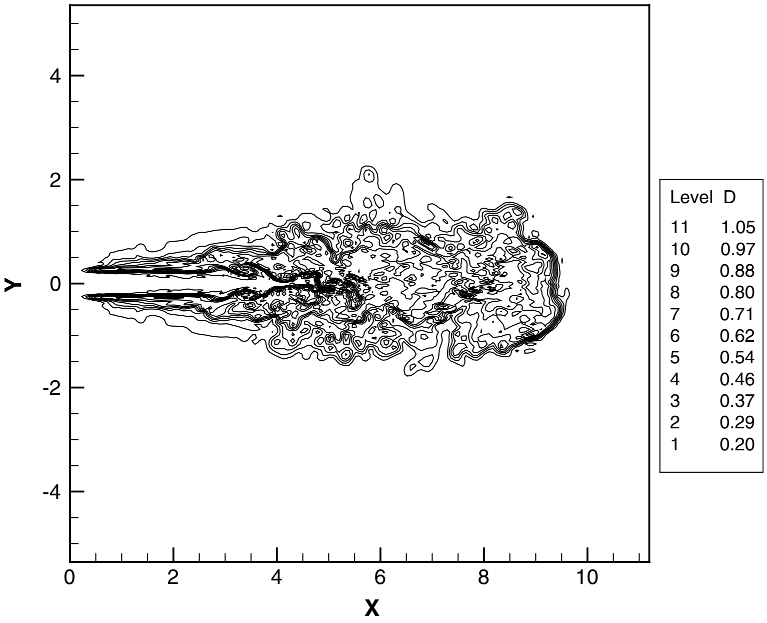

Fig. 8. The density distribution in the reacting jet. Cold fluid (ρ = 1) is entering the domain through the jet nozzle.

probably due to inadequate turbulent mixing of fuel and oxidizer. Finally, we show in Fig. 10 the reaction rate ω. The reaction rate is only non-zero in regions where the fuel and oxidizer are mixed and the temperature is high. The distribution of ω is not smooth, but this is not a problem for the present numerical method.

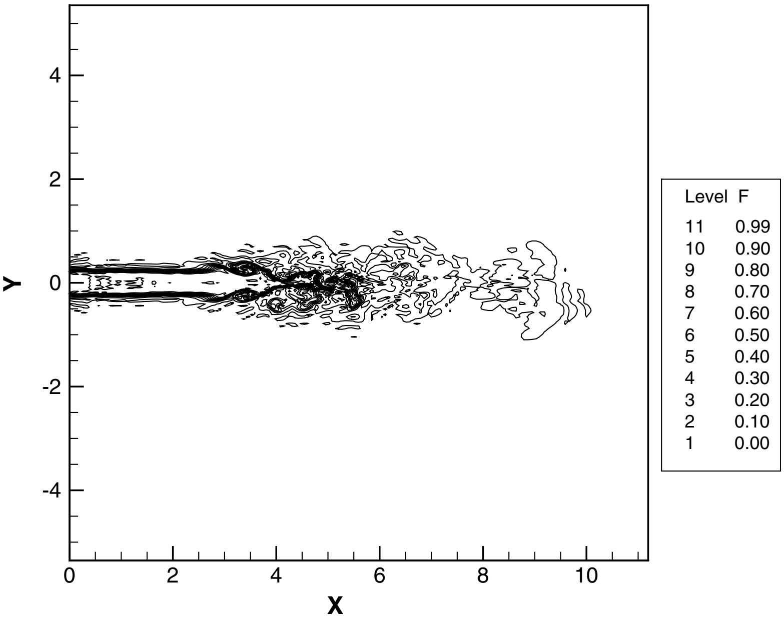

Fig. 9. The distribution of the fuel F in the reacting jet.

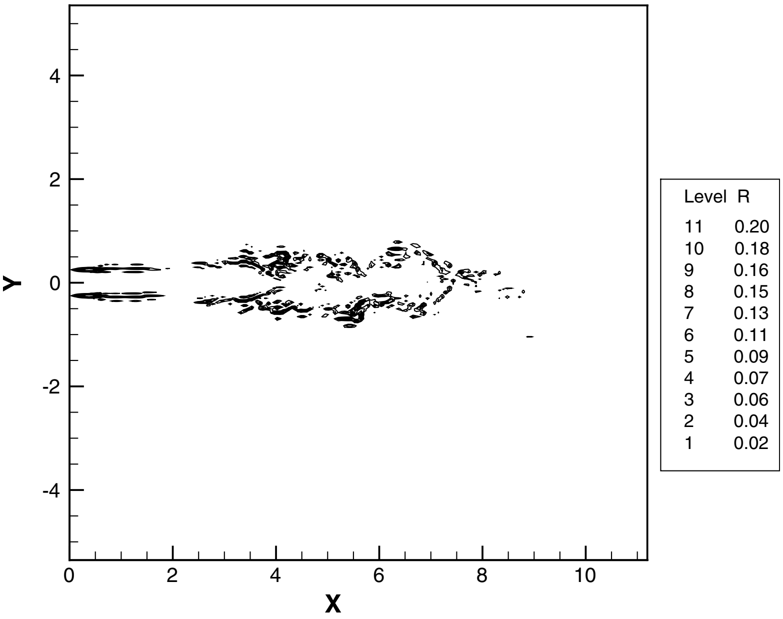

Fig. 10. The reaction rate ω in the turbulent jet.

5. Conclusion

In this paper, we have presented a staggered formulation for the compressible Navier–Stokes equations using high order compact finite differences. The arrangement of the variables on the computational grid is similar as was used by Harlow and Welch [5]. We have shown that the resolution characteristics of the staggered method are similar to a collocated compact finite difference method. The advantage of the present staggered method is that it is very stable, which is not the case with a collocated finite difference method and that no ad-hoc filtering of the velocity field is necessary to avoid numerical instabilities. To demonstrate the numerical stability we have shown results from two test cases in which the numerical resolution is to coarse to resolve all scales of motion.

The ability to under resolve fluid motions makes the present numerical method a good candidate for a LES. In a LES the unresolved scales are modeled with a model, the so-called subgrid model. Modern subgrid scales models do not always have a positive eddy viscosity and one cannot rely on the stabilizing effect of the subgrid model.

Appendix A. Stability analysis of the linear advection equation

In Section 3.9, we considered the numerical solution of the following equation

$$\frac{\partial C}{\partial t}+\frac{\partial C}{\partial x}=0. \tag{11}$$

Here, we will perform a stability analysis of the solution method for this equation. The first step, discussed in Section 3.9, was to interpolated $C_{1/2,\ldots,n-1/2}$ from the midpoints to the gridpoints $\hat{C}_{0,\ldots,n}$. In matrix notation this can be written as

$$\mathbf{T_1}\hat{\mathbf{C}}=\mathbf{A}\mathbf{C},$$

where $\mathbf{T_1}$ is a $(n+1)\times(n+1)$ matrix containing the left hand side of Eq. (8) with boundary conditions, $\hat{\mathbf{C}}$ a vector with $n+1$ elements, $\mathbf{A}$ a matrix with dimension $(n+1)\times n$ containing the right hand side of Eq. (8) and $\mathbf{C}$ is a vector with n elements. The second step in the discretization of Eq. (11) is the compact differentiation. This can be expressed in matrix form as

$$\mathbf{T_2}\mathbf{C}'=\mathbf{B}\hat{\mathbf{C}},$$

where $\mathbf{T_2}$ is a matrix (with $n\times n$ elements) containing the left hand side of Eq. (7), $\mathbf{B}$ a matrix containing the right hand side of Eq. (7) and $\mathbf{C}'$ is a vector containing the spatial derivatives at the grid points. With some matrix algebra we can write

$$\mathbf{C}'=\mathbf{T}_2^{-1}\mathbf{B}\mathbf{T}_1^{-1}\mathbf{A}\mathbf{C}.$$

The solution of Eq. (11) is stable if the norm of the semi-discrete solution

$$\|\mathbf{C}\|_t^2=\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{C}^{\mathrm{T}}\cdot\mathbf{C})=-\left[\mathbf{C}^{\mathrm{T}}(\mathbf{T}_2^{-1}\mathbf{B}\mathbf{T}_1^{-1}\mathbf{A})^{\mathrm{T}}\mathbf{C}+\mathbf{C}^{\mathrm{T}}(\mathbf{T}_2^{-1}\mathbf{B}\mathbf{T}_1^{-1}\mathbf{A})\mathbf{C}\right]$$

is constant or decreases in time, i.e. the eigenvalues of

$$(\mathbf{T}_2^{-1}\mathbf{B}\mathbf{T}_1^{-1}\mathbf{A})^{\mathrm{T}}+(\mathbf{T}_2^{-1}\mathbf{B}\mathbf{T}_1^{-1}\mathbf{A}) \tag{12}$$

should be positive (or zero).

In Fig. 11, we have plotted the eigenvalues of Eq. (12) for the 10th order spatial discretization given in Sections 3.1 and 3.2. All the eigenvalues are positive and the numerical simulation of the differential equa-

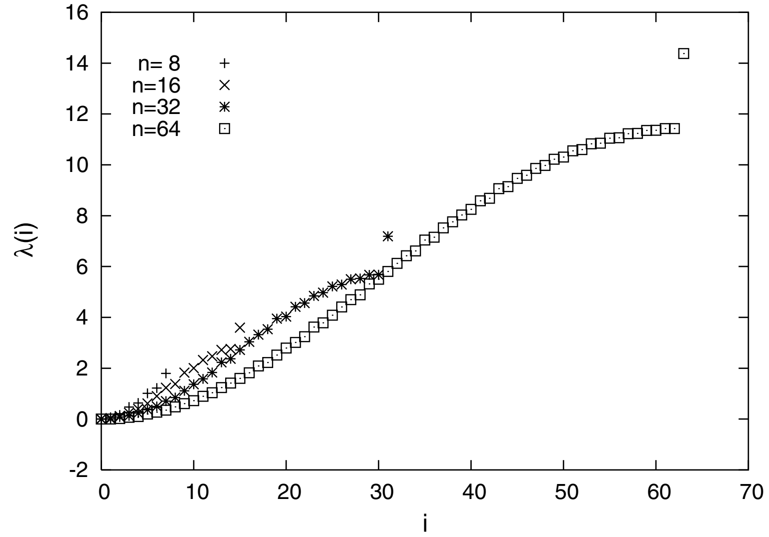

Fig. 11. The eigenvalues $\lambda(i)$ of the matrix given by Eq. (12) for a grid with 8, 16, 32 and 64 points.

tion will be stable. The eigenvalue with the highest index does not follow the curve. This is caused by the different (one sided) formulation on the boundaries of the domain.

References

[1] G. Ashcroft, X. Zhang, Optimized prefactored compact schemes, J. Comp. Phys. 190 (2003) 459.

[2] B.J. Boersma, G. Brethouwer, F.T.M. Nieuwstadt, A numerical investigation on the effect of the inflow conditions on the the self-similar region of a round jet, Phys. Fluids 10 (1998) 899–909.

[3] P. Chu, C. Fan, A three point combined compact difference scheme, J. Comp. Phys. 140 (1998) 370.

[4] J. Freund, Noise sources in a low-Reynolds-number turbulent jet at Mach 0.9, J. Fluid Mech. 438 (2001) 277–305.

[5] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface, Phys. Fluids 8 (1965) 2182.

[6] H.J. Hussein, S.P. Capp, W.K. George, Velocity measurements in a high Reynolds number momentum-conserving axisymmetric turbulent jet, J. Fluid Mech. 258 (1994) 31–76.

[7] C.A. Kennedy, M.H. Carpenter, Several new numerical methods for compressible shear-layer simulations, Appl. Numer. Math. 14 (1994) 397.

[8] S.K. Lele, Compact finite differences with spectral like resolution, J. Comp. Phys. 103 (1992) 16.

[9] C.L. Lubbers, G. Brethouwer, B.J. Boersma, Simulation of the mixing of a passive scalar in a free round turbulent jet, Fluid Dyn. Res. 28 (1999) 189–208.

[10] K. Mahesh, A family of high order finite difference schemes with good spectral resolution, J. Comp. Phys. 145 (1998) 332.

[11] P. Moin, K. Mahesh, Direct numerical simulation: a tool in turbulence research, Annu. Rev. Fluid Mech. 30 (1998) 539.

[12] S. Nagarajan, S.K. Lele, J.H. Ferziger, A robust high-order compact method for large eddy simulation, J. Comp. Phys. 191 (2003) 392.

[13] N.R. Panchapakesan, J.L. Lumley, Turbulence measurements in axisymmetric jets of air and helium. Part 1. Air jet, J. Fluid Mech. 246 (1993) 197–224.

[14] N. Peters, Turbulent Combustion, Cambridge University Press, Cambridge, MA, 2000.

[15] H. Pitsch, H. Steiner, Large-eddy simulation of a turbulent piloted methane/air diffusion flame (Sandia flame D), Phys. Fluids 12 (2000) 2541–2554.

[16] T.K. Sengupta, G. Ganeriwal, S. De, Analysis of central and upwind compact schemes, J. Comp. Phys. 192 (2003) 677.

[17] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, J. Comp. Phys. 187 (2003) 343.