A sparse mesh for Compact Finite Difference - Fourier solvers with radius-dependent spectral resolution in circular domains.

A. J. Torregrosa, S. Hoyas*, A. Gil*, J. P. G. Galache*

CMT - Motores Térmicos, Universitat Politècnica de València, Spain

Abstract

This paper presents a new method for the resolution of elliptic and parabolic equations in circular domains. It can be trivially extended to cylindrical domains. The algorithm uses a mixed Fourier-Compact Finite Difference method. The main advantage of the method is achieved by a new concept of mesh. The topology of the new grid keeps constant the aspect ratio of the cells, avoiding the typical clustering for radial structured meshes at the center. The reduction of the number of nodes has as a consequence the reduction in memory consumption. In the case of fluid mechanics problems, this technique also increases the time step for a constant Courant number. Several examples are given in the paper which show the potential of the method.

Keywords: Direct Numerical Simulation, parabolic equation, radial mesh, Compact Finite Differences, CFL condition

*Corresponding author. Tel.: +34 96 3877650, fax: +34 96 3877659.
Email address: `serhocal@mot.upv.es` (S. Hoyas)

1. Introduction

The resolution of non-linear parabolic and elliptic equations in circular domains is of great interest for several branches of knowledge. In particular, the understanding of the kinematics and the dynamics of turbulent flows in pipes remains one of the challenges for the next decade. As an example, 50% of the energy losses in large pipes are originated in the first millimeters from the wall [1, 2, 3]. Spectral or spectral-like methods are frequently chosen to solve these sort of problems, due to both their great precision and their high ratio of mesh size to computational cost. These methods, when applied on circular domains, are usually formulated in polar coordinates, with the main advantage that the boundary conditions can be imposed in a straightforward manner. Nevertheless, there are two particularities to be dealt with. On one hand, the origin is a pole and it needs a special treatment, which usually involves the use of artificial boundary conditions. On the other hand, for structured radial grids there is a mesh size reduction at the center. For instance, when dealing with turbulent flows in pipes, the smaller structures, which define the mesh structure, are close to the wall, whereas the larger ones are at the centerline of the pipe, thus giving rise to an unnecessary increase in memory requirements. Additionally, the Courant-Friedrichs-Lewy condition (CFL from now on) would impose a very short time step.

The problem at the origin has been addressed in different ways. Chen et al. [4] used spectral collocation methods in order to increase the order of the equation. Li et al. [5] simulated the Navier-Stokes equations with three Chebycheff-Fourier spectral collocation methods. Pure spectral methods have been also proposed, as the Fourier-Legendre discretization used by

Z. Qiu et al. [6], and the Fourier-generic orthogonal polynomials used by Matsushima and Marcus [7]. A coordinate system transformation is another possible technique. Heinrichs [8] used conformal mapping to transform a Cartesian coordinate system into a polar system, and a similar approach was proposed by Hansen et al. [9]. The transformation was applied to the nodes closest to the center, and the components, already in Cartesian form, were then averaged, giving as a result the value of the pole.

The accumulation of grid points at the center of the domain can also be compensated. Kwan [10] proposed a spectral-Galerkin method with a quadratic transform in the radial direction to improve the clustering at the center. Akselvoll and Moin [11] solved the Navier-Stokes equations by dividing the circular domain into two separate regions, the core and the outer region, respectively. At the core region some of the terms were treated explicitly, whereas at the outer region all the terms were solved implicitly. They also used different time schemes for each region to improve the performance without affecting the accuracy.

An additional possibility is the use of a Fourier-Compact Finite Difference (CFD) discretization developed by Lee [12], as that proposed by Lai [13], who imposed symmetry and antisymmetry conditions at the center (asymptotic behavior) applied on phantom points for odd and even values of each wave number.

In the present paper a new algorithm is presented. It uses a sixth-order CFD method in the radial direction, and spectral Fourier decomposition in the azimuthal one. When applied to an elliptic equation, the algorithm produces a set of one-dimensional radial equations. Each equation set is represented by a compact sixth-order finite difference discretization. As a consequence, the error is bounded by the maximum wave number, and by the radial spacing. The aim of this work is to optimize the CFD-Fourier refinement to solve the system as accurately as possible, minimizing the computational resources required. This objective is achieved by the creation of a structured radial mesh with thresholded aspect-ratio and compatible with the Fourier decomposition in the azimuthal coordinate. The main characteristic of the grid is the reduction of the number of computed wave lengths for the internal radii, so that the memory storage is greatly reduced and the computational time is decreased by some orders of magnitude.

The paper is organized as follows. Section 2 is devoted to explain the numerical method and the meshing techniques. The mesh is described in section 3 and some illustrative examples are given in section 4. Finally, in section 5 the main conclusions of the work are pointed out.

2. Numerical method

2.1. The non-linear elliptic equation

The general two-dimensional non-linear partial differential equation considered in this work is

$$\partial_t \vec{u} + N\left(\vec{u}\right) - \frac{1}{\beta} L\left(\vec{u}\right) = \vec{S}, \tag{1}$$

solved in a circular domain of coordinates r and ϕ. In equation (1) L and N stand for the linear and the non-linear term, respectively, while $\vec{S}$ is a source term that may depend on t, r and θ. β is a positive constant whose finality is to weight the diffusion speed. The smaller it is, the faster is the diffusion.

In the case of the Navier-Stokes equations for incompressible flows, it is the Reynolds number. As an example, which will be discussed later on, for the classical heat equation $N(\vec{u}) = 0,\ \ L(\vec{u}) = \nabla^2\vec{u}$ and β is the inverse of the thermal diffusivity, $\alpha = 1/\beta$. This equation is solved in a circular domain with unit radius, so the polar coordinate system (r, θ) is the most appropriate one. The domain is then delimited to $\Omega : [0, 1] \times [0, 2\pi)$.

It is also worth mentioning that in the case of the Navier-Stokes equations, if the domain is periodic along the pipe length the use of a Fourier spectral method reduces the problem to the solution of a (probably large) set of problems like (1).

2.2. General aspects

The Runge-Kutta methods [14] are the time schemes most widely used for fluid dynamics computations. This is due to its high accuracy even when relatively long time steps are used. However, for the sake of simplicity, an Euler implicit method has been chosen here, being straightforward the generalization to more complicated schemes. While in the azimuthal coordinate an obvious choice is the use of a Fourier decomposition method, in the radial direction a CFD scheme [12] has been used, due to its flexibility in choosing the points and to its spectral-like resolution.

2.3. Space discretization: CFD - Fourier

Applying a Fourier spectral expansion along θ, a set of one-dimensional ordinary differential equations is obtained. Let n be the number of points in the azimuthal coordinate. This also is the number of Fourier modes. The equation for each wavenumber κ is

$$\partial_t \vec{\hat{u}}_\kappa + N_\kappa\left(\vec{\hat{u}}_\kappa\right) - \frac{1}{\beta} L_\kappa\left(\vec{\hat{u}}_\kappa\right) = \vec{\hat{S}}_\kappa, \tag{2}$$

where the $\widehat{\ }$ symbol means that the variable is expressed in the wavenumber domain. Both the linear and the non-linear operators depend on the wave number; however, a part of L remains constant. Then, it is convenient to split L into this constant term L_C and the κ-dependent one L_r^κ, as

$$L_\kappa = L_C + L_r^\kappa \tag{3}$$

For example, in the case of the heat equation the operator L is given by

$$L = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}. \tag{4}$$

After the transformation from polar coordinates to the phase space L becomes,

$$L = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) - \frac{\kappa^2}{r^2}, \tag{5}$$

and thus

$$L_C = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) \quad \text{and} \quad L_r^\kappa = -\frac{\kappa^2}{r^2}. \tag{6}$$

Applying the temporal discretization, (explicit for the non-linear part, so that any iterative process may be skipped) equation (2) becomes

$$\frac{\vec{\hat{u}}_\kappa^{n+1} - \vec{\hat{u}}_\kappa^{n}}{\Delta t} + N_\kappa\left(\vec{\hat{u}}_\kappa^{n}\right) - \frac{1}{\beta}\left[L_C + L_r^\kappa\right]\left(\vec{\hat{u}}_\kappa^{n+1}\right) = \vec{\hat{S}}_\kappa. \tag{7}$$

After the discretization in the r direction, equation (4) can be written as

$$\left[-\frac{1}{\beta}\left(L_C + L_r^\kappa\right) + \frac{1}{\Delta t}[I]\right]\vec{\hat{U}}_\kappa^{n+1} = \frac{1}{\Delta t}[I]\vec{\hat{U}}_\kappa^{n} - N_\kappa(\vec{\hat{U}}_\kappa^{n}) + \vec{\hat{S}}_\kappa. \tag{8}$$

Here, $\vec{\widehat{U}}_\kappa$ is an array whose components are $\widehat{U}_{\kappa,i} = \widehat{u}_\kappa(r_i)$ and $[I]$ is the identity matrix. The source term $\vec{\widehat{S}}_\kappa$ is t^n and t^{n+1} dependent. The key point of CFD techniques is that it is possible to find two matrices $[A]$ and $[B]$ such that

$$-\frac{1}{\beta}\,[B]\,L_C(\vec{\widehat{U}}_\kappa) = [A]\,\vec{\widehat{U}}_\kappa, \tag{9}$$

where $[A]$ and $[B]$ are κ-independent. Now, multiplying (8) by $[B]$ and making use of (9) one gets

$$\left[[A] + [B]\left(\frac{1}{\Delta t}[I] + \mathrm{Diag}\left(\frac{\kappa^2}{\beta r^2}\right)\right)\right]\vec{\widehat{U}}_\kappa^{\,n+1} = [B]\left(\frac{1}{\Delta t}[I]\vec{\widehat{U}}_\kappa^{\,n} - N_\kappa(\vec{\widehat{U}}_\kappa^{\,n}) + \vec{\widehat{S}}_\kappa\right) \tag{10}$$

Where $\mathrm{Diag}(v)$ is an operator that puts the vector v on the main diagonal of an identity matrix. Thus a possible large amount of small systems as (10) need to be solved. This can be efficiently implemented in parallel computers.

2.4. Compact sixth-order scheme

In the rest of the paper the operator L will be the Laplacian as it is the most usual one. Then, the implicit linear part to be replaced by the CFD coefficient matrices is

$$L_\kappa(u) = \left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{\Delta t} - \frac{\kappa^2}{r^2}\right)u. \tag{11}$$

Since neither Δt nor κ are constant along the calculation, this operator is split into two parts L_C and L_r^κ as

$$L_C = \left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r}\right)$$
$$L_r^\kappa = \left(\frac{1}{\Delta t} - \frac{\kappa^2}{r^2}\right).$$

The CFD coefficient matrices do not depend neither on Δt nor on κ, and thus they are calculated only once. In order to calculate $[A]$ and $[B]$, both $L_C(u)$ and u are expanded in Taylor series,

$$L_{Cj}(u, r_j) = \sum_{d=0}^{\infty} w_d(r_j)\frac{\partial^d u(r_j)}{\partial r^d}, \tag{12a}$$

$$u_j(r_i) = \sum_{m=0}^{\infty} \frac{\Delta r_{j-i}^m}{m!}\frac{\partial^m u(r_i)}{\partial r^m}. \tag{12b}$$

where w_d represents the coefficient of the d-th derivate and Δr_{j-i} replaces $r_j - r_i$. Here i goes from 1 to $N-1$ (the last point is known) and j indicates the position in the stencil. Upon substitution of (12b) into (12a) one gets the equation for L_C as

$$L_{Cj}(u_{r_i}) = \sum_{m=0}^{\infty} \frac{\Delta r_{j-i}^m}{m!} \sum_{d=0}^{\mathcal{O}} w_d(r_i)\frac{\partial^{d+m} u(r_i)}{\partial r^{d+m}}, \tag{13}$$

In order to obtain the banded matrices $[A]$ and $[B]$, equations (12b) and (13) are substituted into (9). The order of the approximation is fixed by the total number of points in the stencil chosen. Eight points are needed (the order plus two), distributed between $[A]$ and $[B]$. In this case, five points were used for the operator and three points for u, so that $[A]$ is pentadiagonal and $[B]$ tridiagonal. Let α_{ij} be the components of $[A]$ and β_{ij} the components of $[B]$. Normalizing $\beta_{ii} = 1$, from equations (6), (9b) and (10) one gets

$$\sum_{j=i-2}^{i+2} \alpha_{ij} \left(\sum_{m=0}^{d} w_m(r_j) \frac{\Delta r_{j-i}^{d-m}}{(d-m)!} \right) - \sum_{j=i-1}^{i+1} \beta_{ij} \frac{\Delta r_{j-i}^{d}}{d!} = 0. \tag{14}$$

As it has been said before, the i index goes from 0 to $N-1$, leading to one equation for each point (although several points are mixed in each equation). The index j moves in the stencil (and thus in the same row of the matrices). As $[A]$ has a five-point centered stencil, the nodes index j goes from $i-2$ to $i+2$. The case of $[B]$ is similar but with a three-point stencil. These equations are imposed until the desired approximation order is achieved. As a consequence, a truncation error appears in equation (9), which is now written as

$$-\frac{1}{\beta} [B] \, L_C \vec{\widehat{U}}_\kappa = [A] \, \vec{\widehat{U}}_\kappa + \epsilon, \tag{15}$$

where $\epsilon = \mathcal{O}\left(h^6\right)$. In this last expression h is defined as the r increment between two contiguous nodes.

2.5. Boundary conditions

Two locations must be taken into account during the construction of the CFD matrices: the center of the domain and the wall. The center has a pole, which is a critical situation in computation. This problem is solved by calculating a few nodes around the center and using phantom points in order to simulate symmetric and antisymmetric boundary conditions [13]. Due to the periodicity of f, $f(r,\theta) = f(-r, \theta+\pi)$, and thus

$$\sum_{\kappa=-\infty}^{\infty} \widehat{f}_\kappa(r) \exp\left(i\kappa\theta\right) = \sum_{\kappa=-\infty}^{\infty} \widehat{f}_\kappa(-r) \exp\left(i\kappa(\theta+\pi)\right). \tag{16}$$

A small manipulation on the right hand side of this expression leads to

$$\sum_{\kappa=-\infty}^{\infty} \widehat{f}_\kappa(r)\exp\left(i\kappa\theta\right) = \sum_{\kappa=-\infty}^{\infty} (-1)^\kappa \widehat{f}_\kappa(-r)\exp\left(i\kappa\theta\right). \tag{17}$$

Therefore, $\widehat{f}$ is symmetric for odd wave numbers and antisymmetric for even wave numbers:

$$\widehat{f}_\kappa(r) = (-1)^\kappa \widehat{f}_\kappa(-r) \quad \forall\kappa \in \mathbb{N}. \tag{18}$$

In order to keep the same order of approximation at the center, the two first nodes are affected by phantom points; denoting these with an asterisk, one has

$$\begin{aligned} &\sum_{j=1}^{3} \alpha_{1j}\left(\sum_{m=0}^{d} w_m(r_j)\frac{\Delta r_{j-1}^{d-m}}{(d-m)!}\right) - \sum_{j=1}^{2}\beta_{1j}\frac{\Delta r_{j-1}^{d}}{d!} + \cdots \\ &\qquad \cdots \sum_{j=1}^{2} \alpha_{1j}^{*}\,(-1)^\kappa \left(\sum_{m=0}^{d} w_m(r_j)\frac{\Delta r_{j^*-1}^{d-m}}{(d-m)!}\right) - \beta_{11}^{*}\frac{(2r_1)^d}{d!} = 0, \end{aligned} \tag{19a}$$

$$\begin{aligned} &\sum_{j=1}^{4} \alpha_{2j}\left(\sum_{m=0}^{d} w_m(r_j)\frac{\Delta r_{j-2}^{d-m}}{(d-m)!}\right) - \sum_{j=1}^{3}\beta_{2j}\frac{\Delta r_{j-2}^{d}}{d!} + \cdots \\ &\qquad \cdots \alpha_{21}^{*}\,(-1)^\kappa \left(\sum_{m=0}^{d} w_m(r_1)\frac{(2r_1)^{d-m}}{(d-m)!}\right) = 0. \end{aligned} \tag{19b}$$

Regarding the adjacent point to the boundary ($i = N-1$), some modifications needs to be done. In order to retain the same error close to the boundary, this point needs a not-centered stencil, with just one point at its right (the one at the wall) and three at its left.

$$\sum_{j=i-3}^{i+1} \alpha_{ij} \left(\sum_{m=0}^{d} w_m(r_j) \frac{\Delta r_{j-i}^{d-m}}{(d-m)!} \right) - \sum_{j=i-1}^{i+1} \beta_{ij} \frac{\Delta r_{j-i}^{d}}{d!} = 0. \tag{20}$$

This last issue introduces a value outside of the five diagonals of $[A]$. As $[B]$ has a three point stencil, this problem does not appear on it. This anomalous position has to be taken into account when the system of equations is solved. However, it does not increase the complexity of the algorithm.

2.6. CFL condition

The computational cost can be reduced through the use of a sparse mesh, but then the time step is also indirectly increased. The convergence of the algorithm implies a threshold for the CFL number which in turn imposes a maximum value for the time step. The CFL number is defined as

$$CFL = \pi \Delta t \frac{u}{\Delta r} = \pi \Delta t \left[\left(\frac{\bar{u}}{\Delta r} \right)^2 + \left(\frac{\bar{v}}{r \Delta \theta} \right)^2 \right]_{\max}^{1/2}. \tag{21}$$

where $\bar{u}$ and $\bar{v}$ are the main values in each cell (see Fig. 1):

$$\frac{\bar{u}_{n+1/2}}{\Delta r} = \frac{u^n r^n + u^{n+1} r^{n+1}}{\left(r^{n+1}\right)^2 - \left(r^n\right)^2}, \qquad \frac{\bar{v}_{n+1/2}}{r \Delta \theta} = \frac{v^{n+1} + v^n}{\Delta \theta \left(r^{n+1} + r^n\right)}. \tag{22}$$

Assume that $\bar{u} \sim \bar{v}$. For radial structured meshes (“full meshes” from now), $CFL(r_1) \sim CFL(1)/r_1$, whereas for constant-sized meshes (“sparse meshes”) $CFL(r_1) \sim CFL(1)$. Thus, for the same threshold, Δt for the sparse mesh is $1/r_1$ times Δt for the full mesh. Therefore, with the sparse mesh the number of nodes can be reduced and the time step increased.

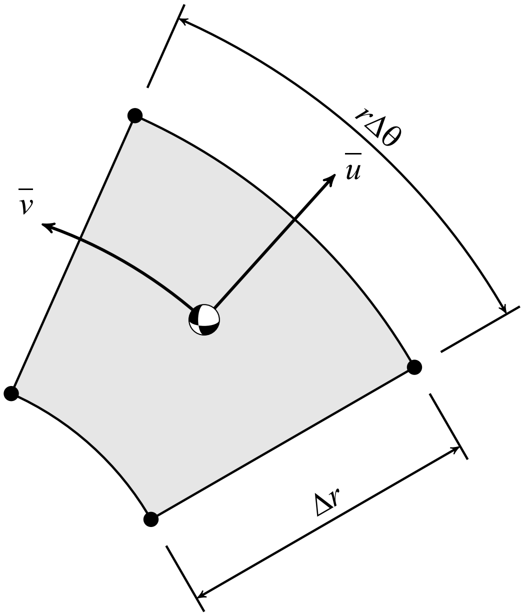


Figure 1: Set of parameters used to obtain the time step for a fixed maximum CFL. Values are averaged along the cell and applied at the center of mass.

3. Mesh

The aim of this paper is to improve the performance of CFD-Fourier by using an optimized mesh. Currently, the use of full meshes for the solution of polar and cylindrical problems is widely extended. In the following, the term “full mesh” will refer to a grid which contains the full tensor product of the discretized independent variables. Full meshes in polar coordinates produce clustering around the center because the density of points is proportional to the inverse of the radius. Here it is proposed that the cell aspect ratio is adapted so that the CFL value is as close as possible to its threshold in the whole domain. A good option is to combine several cells into a single one. Furthermore, in order to have a coherent mesh and good behavior of Fast

Fourier Transform (FFT) operations, the number of nodes in a single radius is taken to be the product of the initial number of nodes at the first radius times a power of two (Fig. 2).

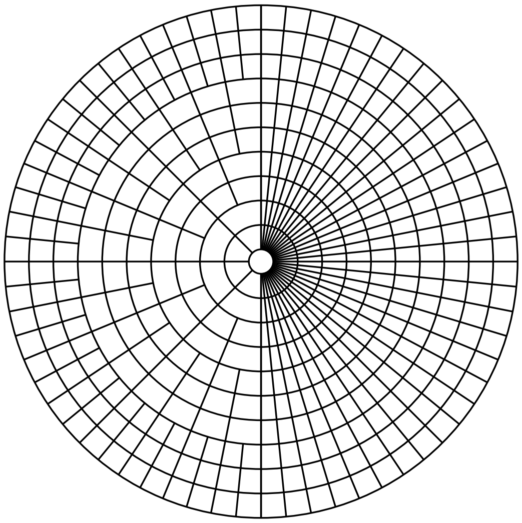


Figure 2: Sparse mesh (left) vs. full mesh (right). Number of nodes reduction can be easily appreciated.

Two different meshes will be studied. On one hand, the r-equidistant mesh. This mesh presents equal properties along the r direction, reason why it will be used in the control set of numerical experiments. On the other hand, a mesh optimized for turbulence studies will be considered. This mesh was created following the methodology proposed by Hoyas and Jiménez [15], adjusting the size of the cell to the most relevant local physical scale. These meshes are compared in Fig. 3.

3.1. Algorithm for the sparse mesh

A new algorithm was developed whose basic idea is to create the mesh for a prescribed aspect ratio. Since in this case the objective was to minimize

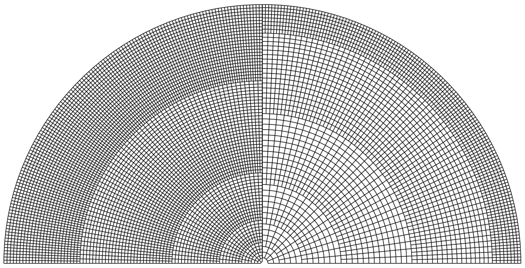


Figure 3: (Left) r-equidistant sparse mesh. (Right) Sparse mesh optimized for turbulence studies at $Re_\tau = 180$

the aspect ratio, the maximum aspect ratio was fixed to $\sqrt{2}$. The four steps of the algorithm are:

1. The number of divisions along the perimeter is calculated from the prescribed maximum mesh size. As an extra condition, the number of divisions is set to a power of two so that the performance of the FFT operations is improved.
2. The divisions along the r direction are generated, taking into account the aspect ratio criterion at the wall and the desired mesh size evolution along the radius. At this point the user must choose the r-distribution to be used, bearing in mind that the optimal mesh for the solution of the Helmholtz equation has a discretization along the r direction different from that of a mesh optimized for turbulence studies.
3. By setting the separation between two neighboring circles, the number of azimuthal divisions for each radius is calculated for a fixed aspect ratio. As in the case of the perimeter, the number of points in each single circle is set to a power of two.
4. The number of divisions at some distances from the wall are corrected

for consistency with the boundary conditions. At the center, if the first discretized r values are far away from the pole, increasing the number of azimuthal divisions contributes to minimize the error at those points, whereas for the nodes closest to the wall the solver needs at least six points (five plus the boundary) to be able to operate, so that the number of divisions along the last six circles has to be the same.

4. Numerical test problems

Three different cases will be analyzed to test the schemes: Laplace equation, Heat Transfer equation and Convective-diffusive equation. The reference to calculate the error will be an analytical solution of these equations, imposing a solution and calculating the source term analytically.

4.1. Laplace equation

First, the order of convergence must be checked. The equation to solve is given by

$$\left(\frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} \right) \phi = 0. \tag{23}$$

The error introduced by our method is dominated by the truncation error studied in section 2.4. A benchmark was designed to study the behavior of the error when solving equation (23) for both the r-equidistant mesh and the turbulence-optimized mesh. Sparse and full meshes, for two wave numbers in the azimuthal direction, κ=10 and κ=30, where employed. Different number of nodes along the perimeter were used: 64, 128, 256, 512, 1024, 2048, 4096, 8192 and 16384. In Fig. 4 the error dependency for $\kappa = 10, 30$, defined as

$$\varepsilon = \left\| u(r_i,\theta_j) - U_{i,j} \right\|_\infty , \tag{24}$$

is shown.

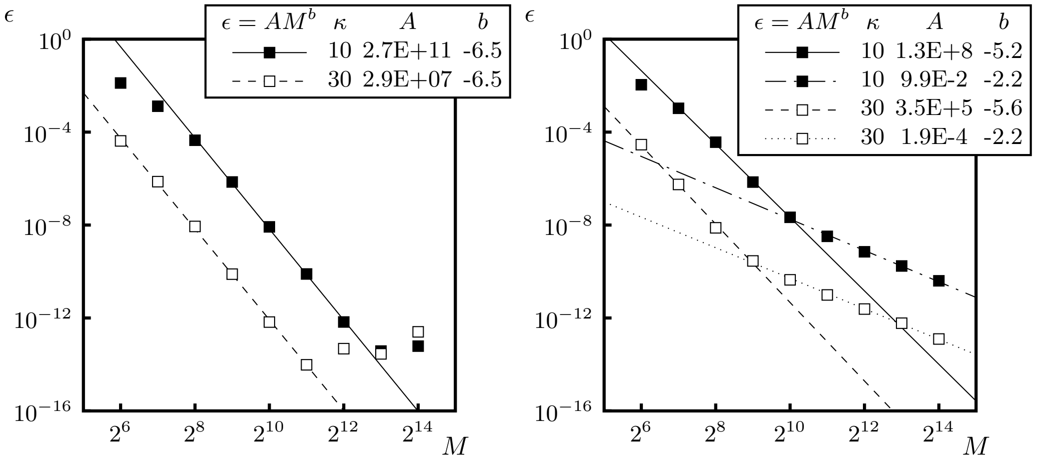
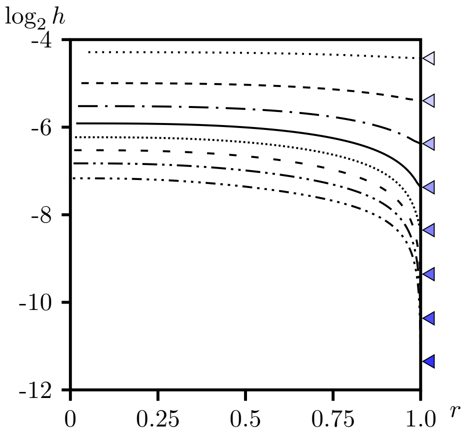


Figure 4: Error vs. number of nodes along the perimeter M for $\kappa = 10$ (white squares) and $\kappa = 30$ (black squares). Equidistant mesh (left), and turbulence-optimized mesh (right)

The theoretical error is given by $\epsilon = AM^b$, where h should be 6 in our case. In the left part of Fig. 4, the error committed using the equidistant mesh is shown. The slope of this error is even better than the imposed one. Due to round-off error, occurring in very thin meshes, it is impossible to reduce the error below 10^{-13}. The situation is a little bit worse in the case of the turbulence-optimized meshes, where two slopes are presented. The expected tendency is achieved for meshes up to 2^{10} points along the perimeter. After that, due to unequal refinement along the r coordinate a reduced slope is obtained. This can be seen in Fig. 5. The main cause is that when the mesh is refined by a factor of 2 at the wall, at the center it is only reduced by a factor of 1.3 for the thinnest cases. In fact, the slope of 2.2 corresponds with the theoretical expected behavior, as $1.3^6 \simeq 2^{2.2}$. It should be remarked that such a behavior of the error for the last mesh type arises in the case of the Laplacian operator, but is not expected to appear when turbulent flows are calculated.



Legend

		N	M	h_0	h_w
	....	128	20	5.125×10^{-2}	4.651×10^{-2}
	- - - -	256	34	3.139×10^{-2}	2.376×10^{-2}
	— · —	512	52	2.181×10^{-2}	1.204×10^{-2}
	——	1024	73	1.659×10^{-2}	6.033×10^{-3}
	....	2048	96	1.333×10^{-2}	3.065×10^{-3}
	- - -	4096	123	1.086×10^{-2}	1.526×10^{-3}
	— ·· —	8192	156	8.813×10^{-3}	7.581×10^{-4}
	...—...	16384	201	6.957×10^{-3}	3.832×10^{-4}

N :Divisions along the perimeter.
M :Divisions along the radius.
h_0 :Mesh size at the middle.
h_w:Mesh size at the wall.

Figure 5: Evolution of the mesh size Δr along the radius. The mesh size at the middle, h_0, becomes much bigger then the size at the boundary, h_w, when M is increased. The value for h in the different equidistant meshes is marked by the triangles at the left of the figure.

4.2. Heat conduction equation

As an algebraic solution of the convection-diffusion equation is not available for comparison with numerical results, in a first approach it was decided to test the method in the Heat equation,

$$\partial_t \vec{u} - \Delta \vec{u} = \vec{0}. \tag{25}$$

The solution of this equation can be expressed as the product of two infinite series:

$$u_i = \sum_{m=0}^{\infty}\sum_{\kappa=0}^{\infty} A_{\kappa,m} \exp\left(-\lambda_{\kappa,m}^2 t\right) \cos\left(\kappa\theta\right) J_\kappa\left(\lambda_{\kappa,m} r\right) + \cdots$$
$$\sum_{m=0}^{\infty}\sum_{\kappa=0}^{\infty} B_{\kappa,m} \exp\left(-\lambda_{\kappa,m}^2 t\right) \sin\left(\kappa\theta\right) J_\kappa\left(\lambda_{\kappa,m} r\right), \quad (26)$$

where the J_κ are Bessel functions and $\lambda_{\kappa,m}$ is the m-th zero of the κ-th Bessel function. Initial conditions are imposed by giving values to $A_{\kappa,m}$ and $B_{\kappa,m}$. In this way, by choosing m, the behavior of the error can be studied separately for different wave numbers κ and, to some extent, the gradients in the r direction. The method has been checked in several sparse and full meshes. The Fig. 6 shows the relative error in the simulation of many different initial conditions in a Turbulence optimized mesh. It is clear that the error is affected essentially by the time step, as it could be expected since the Euler implicit scheme has order one, and thus the error is directly proportional to the time step. The ratio of the error to the time step has unitary order. Same results where obtained for all meshes.

As not error dependence on sufficiently refined meshes was found, this error has to come from the time step and the Euler method. Therefore, any improvement in the error implies the use of higher-order time schemes.

4.3. A non-linear equation

Once the good behavior of the solver was demonstrated, the final equation to be tested is

$$\partial_t \vec{u} - \frac{1}{\beta}\Delta\vec{u} + \alpha\left(\vec{\nabla}\cdot\left(\vec{u}\otimes\vec{u}\right) + \frac{v}{r}\begin{bmatrix} -v \\ u \end{bmatrix}\right) = \vec{S}. \quad (27)$$

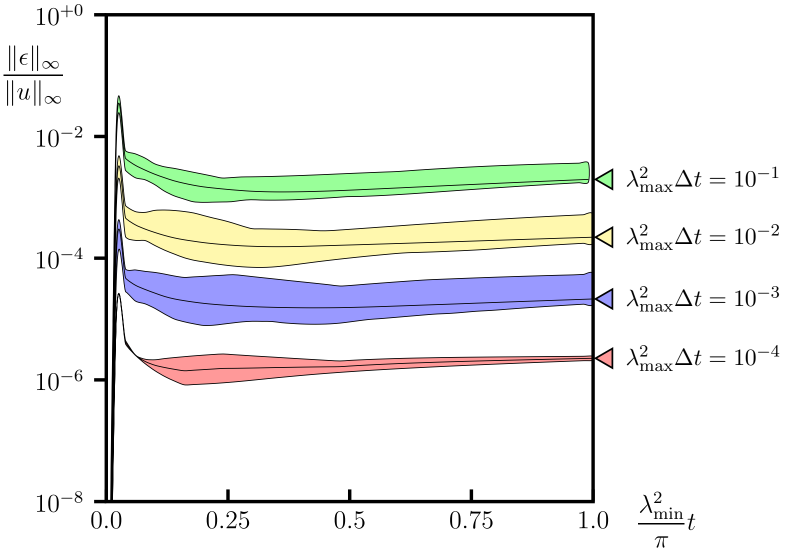


Figure 6: Color online. Relative error in the computation of several random initial conditions. Time interval is π times the characteristic time $1/\lambda^2_{min}$. The area indicates the different errors obtained and the middle line is the mean value. Time step is a factor of the lowest characteristic time $\lambda^{-2}_{\max}$

This equation can be extended by including explicitly the non-linear terms. The solution for general cases cannot be checked, but it is still possible to check the error if a simple source term is imposed:

$$\frac{\partial u}{\partial t} - \frac{1}{\beta}\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right) + \frac{\partial u^2}{\partial r} + \frac{u^2}{r} + \frac{1}{r}\frac{\partial uv}{\partial \theta} - \frac{v^2}{r} = S_u, \tag{28a}$$

$$\frac{\partial v}{\partial t} - \frac{1}{\beta}\left(\frac{\partial^2 v}{\partial r^2} + \frac{1}{r}\frac{\partial v}{\partial r} + \frac{1}{r^2}\frac{\partial^2 v}{\partial \theta^2}\right) + \frac{\partial uv}{\partial r} + \frac{1}{r}\frac{\partial v^2}{\partial \theta} + 2\frac{uv}{r} = S_v. \tag{28b}$$

A benchmark case was designed in order to analyze and understand the increase in the performance of the solver. The value of β was set to 2.5, with a CFL fixed to 0.3 and no source term. As an initial condition, white noise was imposed from $\kappa = 0$ to $\kappa = 16$ for the angular direction, and it was multiplied by r^2 for odd values of κ and by r^3 for even values of κ in order to simulate the initial condition in the r direction consistently with the symmetry and antisymmetry conditions (see equation (18)). The boundary condition was fixed to 0. Sixty-five simulations were performed for meshes with 64, 128, 256, 512 and 1024 divisions along the perimeter, and a single simulation for a mesh with 2048 divisions, due to its long computation time. The algorithm used was the same for both the full and the sparse meshes. The difference between the solutions obtained with both meshes is negligible and thus it is not shown here.

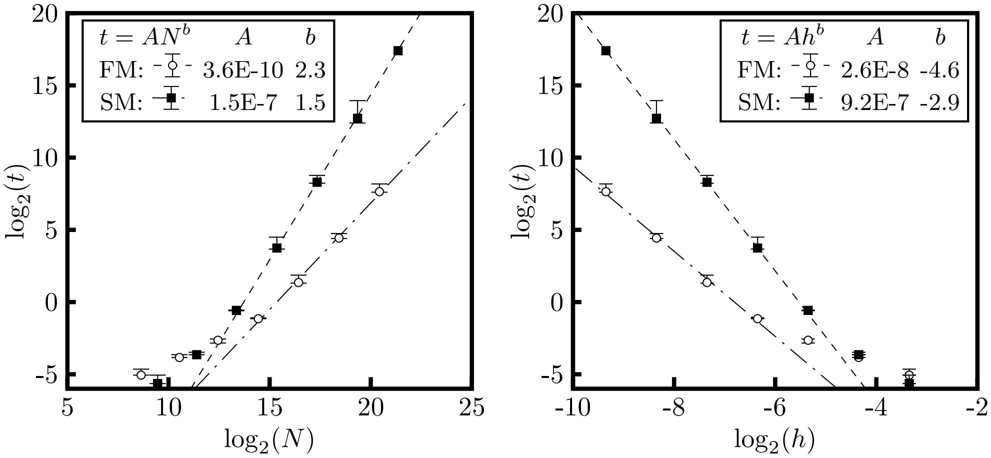


Figure 7: Results of the numerical experiment (time in seconds and mesh size normalized by the radius). Trend lines for full mesh (FM) and sparse mesh (SM) are shown. Left: total number of nodes vs. time; right: mesh size vs. time.

In Fig. 7 the dependence of the computation time with the total number of nodes N and the mesh size h is shown.

With the exception of the highest values of h, the results from this numerical experiment are clearly indicative of the improvement achieved in the performance of the algorithm. In the case of the very coarse grid, the number of nodes is not enough to see the advantages of the new methodology (first points of Fig. 7). This is a consequence of the extra computational work needed for initialization, for instance, to create the FFT planning. In other words, the initialization takes more time than the solver. Because of that, only the thinnest meshes has been taken into account for the conclusions. It is also worthy to note that in the log-log plots of Fig. 7, it can be seen that the improvement in computation time increases exponentially. In order to quantify the effect observed, one may simply divide the "sparse mesh" result by the "full mesh" results, assuming the same h; this gives

$$\frac{t_{SM}}{t_{FM}} = 34.901 h^{1.61}. \tag{29}$$

This means that, for the same global accuracy (same h), if the mesh size is reduced by a half, then the time ratio is increased by a factor of around 3.

5. Conclusions

A new algorithm has been applied to the solution of equations with different elliptic operators, mainly obtained from the discretization of parabolic equations. The results of the benchmark cases considered show that the global error is controlled by the mesh size and the approximation order of the Compact Finite Difference scheme, and that the distribution of the nodes along the r direction affects the local error. In order to optimize the mesh, the order of the time scheme has to be sufficiently high so as to produce errors with the same order of magnitude as the error arising from the space discretization, and with the time step as close as possible to the threshold given by the convergence criterion.

As a result, the use of a sparse mesh with time-dependent non-linear solvers in polar or cylindrical coordinates has beneficial effects on computation time without penalties in the global error. The grid refinement along the radius produces 2^n nodes, an optimal distribution for Fourier methods. Furthermore, the CFL condition becomes far less restrictive than for the full mesh, as a result from the larger spacing between neighboring points, especially at the central nodes. This effect is magnified when the transport term is much larger than the diffusion term.

The CFL condition, which restricts the time step due to convergence criterion, relieves the time step restriction for sparse meshes. As a second consequence, the memory consumption is reduced by a half as a result of leaving out unnecessary nodes. Therefore, problems with a prohibitive consumption of resources can be addressed thanks to the great reduction achieved in the number of iterations (longer time steps) and to the halving of memory requirements.

Acknowledgments

This work was supported by the Spanish Government in the frame of the Project "Métodos LES para la simulación de chorros multifásicos", grant ENE2010-18542. We also are very grateful to the one of the referees for his/her work.

Bibliography

[1] O. Shishkina, C. Wagner, A fourth order finite volume scheme for turbulent flow simulations in cylindrical domains, Comput. Fluids 36 (2007) 484–497.

[2] X. Wu, P. Moin, A direct numerical simulation study on the mean velocity characteristics in turbulent pipe flow, J. Fluid Mech. 608 (2008) 81–112.

[3] C. Chin, A. S. H. Ooi, I. Marusic, H. M. Blackburn, The influence of pipe length on turbulence statistics computed from direct numerical simulation data, Phys. Fluids 22 (2010) 115107.

[4] H. Chen, Y. Su, B. D. Shizgal, A direct spectral collocation Poisson solver in polar and cylindrical coordinates, J. Comput. Phys. 160 (2000) 453–469.

[5] B. W. Li, Y. R. Zhao, Y. Yu, Z.-D. Qian, Three-dimensional transient Navier-Stokes solvers in cylindrical coordinate system based on a spectral collocation method using explicit treatment of the pressure, Int. J. Numer. Methods Fluids 66 (2011) 284–298.

[6] Z. Qiu, Z. Zeng, H. Mei, L. Li, L. Yao, L. Zhang, A Fourier-Legendre spectral element method in polar coordinates, J. Comput. Phys. 231 (2012) 666–675.

[7] T. Matsushima, P. S. Marcus, A spectral method for polar coordinates, J. Comput. Phys. 120 (1995) 365–374.

[8] W. Heinrichs, Spectral collocation schemes on the unit disc, J. Comput. Phys. 199 (2004) 66–86.

[9] M. O. L. Hansen, J. N. Sørensen, W. Z. Shen, Vorticity-velocity formulation of the 3D Navier-Stokes equations in cylindrical coordinates, Int. J. Numer. Methods Fluids 41 (2003) 29–45.

[10] Y. Y. Kwan, Efficient spectral-galerkin methods for polar and cylindrical geometries, Appl. Numer. Math. 59 (2009) 170–186.

[11] K. Akselvoll, P. Moin, An efficient method for temporal integration of the Navier-Stokes equations in confined axisymmetric geometries, J. Comput. Phys. 125 (1996) 454–463.

[12] S. Lele, Compact finite-difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[13] M. C. Lai, A simple compact fourth-order Poisson solver on polar geometry, J. Comput. Phys. 182 (2002) 337–345.

[14] P. R. Spalart, R. Moser, M. Rogers, Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions, J. Comput. Phys. 96 (1991) 297–324.

[15] S. Hoyas, J. Jiménez, Scaling of the velocity fluctuations in turbulent channels up to Re_τ=2003, Phys. Fluids 18 (2006) 011702.