

# Spectral Methods for Partial Differential Equations

Bruno Costa <sup>1</sup>

## ABSTRACT

In this article we present the essential aspects of spectral methods and their applications to the numerical solution of Partial Differential Equations. Starting from the fundamental ideas, we go further on building auxiliary techniques, as the treating of boundary conditions, and presenting accessory tools like mapping and filtering, finishing with a complete algorithm to solve a classical problem of fluid dynamics: The flow through a circular obstacle. We also present a short comparison with Finite Differences, showing the superior efficiency of spectral methods in problems with smooth solutions.

Several equations like the wave equation in one spatial dimension, Burgers in a 2D domain and a simple multidomain setting for the Navier-Stokes 2D are solved numerically and the results are presented at the applications sections. We end with a brief presentation on the software PseudoPack2000 and a quick discussion on the relevant literature.

## 1 Introduction

Spectral methods have been used extensively during the last decades for the numerical solution of partial differential equations (PDE) due to their bigger accuracy when compared to Finite Differences (FD) and Finite Elements (FE) methods. The rate of convergence of spectral approximations depends only on the smoothness of the solution, yielding the ability to achieve high precision with a small number of data. This fact is known in literature as "spectral accuracy". In FD and FE, the order of

---

<sup>1</sup>This work was supported by CNPq grant # 300315/98-8.

accuracy depends on some fixed negative power of  $N$ , the number of gridpoints being used, even when only analytic functions are involved.

The expression spectral methods has different meanings for several subareas of Mathematics, like Functional Analysis and Signal Processing. In this article, spectral methods has the meaning of a high accuracy numerical method to solve Partial Differential Equations. The numerical solution is expressed as a finite expansion of some set of basis functions. When the PDE is written in terms of the coefficients of this expansion, the method is known as a Galerkin spectral method. Spectral collocation methods, also known as pseudospectral methods, is another subclass of spectral methods and are similar to Finite Differences methods due to direct use of a set of gridpoints, which are called "collocation points". A third class are the Tau spectral methods. These methods are similar to the Galerkin spectral methods, however the expanding basis is not obliged to satisfy boundary conditions, requiring extra equations. In this article, we will mainly concentrate on collocation methods. The reader interested on Galerkin and Tau methods must check the works of Canuto et al [8] and Gottlieb and Orszag [20].

In the past several years, the activity on both theory and application of spectral methods have been concentrated on collocation spectral methods. One of the reasons is that collocation methods deals with nonlinear terms more easily than Galerkin methods. The nonlinear terms are treated on a FD manner, via gridpoints values multiplication. The underlying idea of a collocation spectral method is to approximate the unknown solution in the entire computational domain by an interpolating high order polynomial at the collocation points. The spatial derivatives of the solution are approximated by the derivatives of the polynomial and the time derivative, if it exists, is solved through classical finite differences schemes. Periodic and non-periodic problems are respectively treated with trigonometric and algebraic polynomials. Some of the methods commonly used in the literature are the Fourier collocation methods for periodic domains and the Jacobi polynomials for non-periodic domains, with the Chebyshev and Legendre polynomials as special cases.

Pseudospectral methods have a wide range of applications going from 3-D seismic wave propagation to turbulence, combustion, non-linear optics and passing through aero-acoustics and electromagnetics. The material exposed in this article intends to enable the interested reader to solve the ubiquitous Navier-Stokes equations, whose solution is exemplified in the case of the two-dimensional flow through an obstacle. Thus, the necessary main ingredients as mapping and boundary conditions are treated in some extent along with several basic examples.

This article is divided as follows: In Section 2, we state the fundamentals of spectral methods by describing the Galerkin spectral methods for linear and nonlinear problems. The collocation version of spectral methods comes in Section 3 where we introduce polynomial interpolating algorithms. In Section 4, we discuss mapping techniques for spectral methods in one and two dimensions in space. The subjects of filtering and boundary conditions are dealt with in Section 5, together with applications to basic equations of fluid dynamics. We finish the article with a quick description of the Library PseudoPack and a brief discussion on the relevant literature.

## 2 Preliminaries

We start this section showing a direct comparison between spectral methods and Finite Differences. The main idea to be passed is the following:

*Spectral Methods achieve a greater precision with a smaller number of points than Finite Difference methods.*

The table below, extracted from [8] shows the error decay of approximating the first derivative of  $e^{\sin x}$  through a second and a fourth order FD method and a Fourier spectral collocation method:

N	FD 2nd order	FD 4th order	Fourier Spectral
16	3.4020e-001	1.3844e-001	4.3179e-003
32	9.3589e-002	1.5750e-002	1.7619e-007
64	2.5833e-002	1.1318e-003	2.3870e-014
128	6.5118e-003	7.5900e-005	7.2054e-014

The graph below in Figure (1) shows the decay of the error with varying number of points.

Figure 1: Error comparison between Spectral and Finite Differences schemes.

In [35] one can find quick Matlab codes to generate graphics like the one above.

## 2.1 Linear periodic problems

In spectral methods the solution of a PDE is approximated by a finite linear combination of a chosen set of orthogonal functions  $\{\varphi_j(x)\}_{j=0}^{\infty}$  as

$$f_N(x) = \sum_{k=0}^N \hat{f}_k \varphi_k(x), \quad (1a)$$

where the  $\hat{f}_k$ 's are called the spectral coefficients of  $f$  with respect to the basis  $\{\varphi_k\}$ ;  $\hat{f}_k$  is a measure of how much of  $f$  is in the "direction" of  $\varphi_k$  and can be evaluated through an integral like

$$\hat{f}_k = \int f(x) \varphi_k(x) w(x) dx, \quad (2)$$

where  $w(x)$  is a weight function associated to the basis  $\{\varphi_k\}$ .

For instance, if we consider periodic functions in the interval  $[0, 2\pi]$ , one such a basis is the Fourier basis formed by sines and cosines as  $\{\cos kx, \sin kx\}_{k=0}^{\infty}$ , which in its concise complex form becomes  $\{e^{ikx}\}_{k=-\infty}^{\infty}$ . If  $f$  satisfies some conditions, like continuity and bounded variation, one can prove that  $f_N$  given by  $f_N(x) = \sum_{k=0}^N \hat{f}_k e^{ikx}$  with  $\hat{f}_k = \int f(x) e^{ikx} dx$ , converges to  $f$  uniformly as  $N$  increases.

The derivative of  $f$  is approximated by  $f'_N$  which is given by

$$f'_N(x) = \sum_{k=0}^N \hat{f}_k \varphi'_k(x). \quad (3)$$

$f'_N$  is named the Galerkin spectral derivative of  $f$ . Once we know the specifics about the derivatives  $\varphi'_k$  (a recurrence formulae, in general), we can work out formula (3) and find out the coefficients  $\hat{f}'_k$  to write  $f'_N(x) = \sum \hat{f}'_k \varphi_k(x)$ .

For instance, let  $f$  be an even periodic function in  $[0, 2\pi]$ . An appropriate basis to approximate  $f$  is the Fourier Basis  $\{\cos kx\}_{k=0}^{\infty}$ . If we write

$$f_N(x) = \sum_{k=0}^N \hat{f}_k \cos kx,$$

we easily see that an approximation to the second derivative of  $f$  is given by:

$$f''_N(x) = \sum_{k=0}^N \left( -k^2 \hat{f}_k \right) \cos kx,$$

meaning that once we know the spectral coefficients  $\hat{f}_k$  of  $f$  we obtain an approximation to  $f''$  by multiplying each  $f_k$  by  $(-k^2)$ . In this way, double differentiation in spectral space is equal to multiplication by  $(-k^2)$ . The case of the first derivative is a little bit more "complex", for we need to use the basis  $\{e^{ikx}\}$ , since the first derivative of an even function is odd.

With the above technique to compute second derivatives, let us solve the heat equation below

$$\begin{aligned} u_t - \nu u_{xx} &= 0, & x \in [-\pi, \pi], t > 0, \\ u(x, 0) &= e^{-x^2}, & x \in [-\pi, \pi], \end{aligned} \tag{4}$$

where  $u$  is the temperature and  $\nu$  is the diffusion coefficient. Using the approximation

$$u_N(x) = \sum_{k=0}^N \hat{u}_k \cos kx \tag{5}$$

and substituting into the above equation one obtains

$$\sum_{k=0}^N \left( \frac{\partial \hat{u}_k}{\partial t} + \nu k^2 \hat{u}_k \right) \cos kx = 0. \tag{6}$$

Since the functions  $\{\cos kx\}_{k=0}^N$  are orthogonal, which means that  $\int_{-\pi}^{\pi} \cos kx \cos lx dx = \delta_{kl}$ , we can reduce equation (6) to the system of  $N + 1$  ordinary differential equations

$$\begin{aligned} \frac{\partial \hat{u}_k}{\partial t} + \nu k^2 \hat{u}_k &= 0, & \text{for } k = 0, \dots, N, \\ \hat{u}_k(0) &= (u_0)_k, & \text{for } k = 0, \dots, N. \end{aligned} \tag{7}$$

The example above shows the essential steps to solve a PDE with spectral methods. First, is necessary to choose an appropriate basis for the problem to be solved, for instance, periodic problems are solved with the Fourier basis and nonperiodic ones with a polynomial basis. Next, we need the formula relating the spectral coefficients of  $f$  with those of its derivative. In this way, a system of ODEs analogous to the one in (7) can be formed and solved. Note that one does not work with the physical values of the solution, but with its set of spectral coefficients. In fact, the physical values will be necessary to approximate the integrals (2) through quadrature formulae, as we shall see later.

We now show why the error  $|f - f_N|$  decays so fast with the increase of number of points, as seen in Figure 1 above. If  $f$  is such that we can substitute it by its expansion series  $\sum_{k=0}^{\infty} \hat{f}_k \varphi_k(x)$ , the error in approximating  $f$  by  $f_N$  can be measured

by the size of the tail of the above series, given by  $\varepsilon_N = |f - f_N| = \left| \sum_{k>N}^{\infty} \hat{f}_k \varphi_k(x) \right|$ .

Considering the Fourier basis  $\{e^{ikx}\}$ , we may write

$$\begin{aligned} \hat{f}_k &= \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ikx} dx \\ &= \frac{1}{2\pi ik} [f(2\pi) - f(0)] + \frac{1}{2\pi ik} \int_0^{2\pi} f'(x) e^{-ikx} dx. \end{aligned}$$

Therefore, we can see that  $|\hat{f}_k| = O\left(\frac{1}{k}\right)$  and that  $\varepsilon_N = O\left(\frac{1}{N}\right)$ . However, if  $\hat{f}$  is periodic, the boundary term vanishes and we can integrate by parts once again to obtain  $|\hat{f}_k| = O\left(\frac{1}{k^2}\right)$  or  $\varepsilon_N = O\left(\frac{1}{N^2}\right)$ . Further iteration of this argument shows that the decay of the spectral coefficients of  $f$  depend on the differentiability of  $f$ , together with the periodicity of its derivatives. In particular, if  $f$  is  $C^\infty$ , we can iterate the integration by parts above indefinitely to affirm that the decay of the spectral coefficients is faster than any negative power of  $N$ , or that

$$\varepsilon_N = O\left(e^{-cN}\right), \text{ for some } c > 0.$$

This is what is known as exponential or spectral convergence and it is this property that has given its name to spectral methods.

## 2.2 Nonlinear Problems and Pseudospectral Methods

Mounting the system of ODEs as in (7) gets an extra degree of complexity in the presence of nonlinear terms in the PDE. For instance, consider the same heat equation above with a diffusion coefficient that now depends on the temperature itself, say,  $\nu(u) = cu$ , where  $c$  is a constant. Thus, the equation to be considered is  $u_t - cuu_{xx} = 0$ . This time, let us use the complex form of the Fourier expansion and substitute the truncated series (5) into the equation to obtain:

$$\begin{aligned} & \sum_{k=0}^N \frac{\partial \hat{u}_k}{\partial t} e^{ikx} + \left( c \sum_{l=0}^N \hat{u}_l e^{ilx} \right) \sum_{k=0}^N k^2 \hat{u}_k e^{ikx} \\ = & \sum_{n=0}^N \frac{\partial \hat{u}_k}{\partial t} e^{inx} + c \sum_{n=0}^N \sum_{k+l=n}^N (k^2 \hat{u}_l \hat{u}_k) e^{inx} \\ = & \sum_{n=0}^N \frac{\partial \hat{u}_k}{\partial t} e^{inx} + c \sum_{n=0}^N \hat{w}_n e^{inx} = 0. \end{aligned}$$

The computation of the coefficients  $\hat{w}_n$  takes  $O(N^2)$  operations, which is too expensive if compared to the  $O(N)$  cost of a finite difference algorithm to treat the same nonlinear term. In 3 dimensions, this cost rises up to  $O(N^4)$  in the spectral method and  $O(N^3)$  for finite differences.

It was this single fact that historically kept spectral methods from being used on applications involving a large number of data. This situation changed when transform methods between the spectral coefficients and the physical values were developed independently by Orszag() and Eliassen, Machenhauer and Rasmussen (). These transforms, known by the general name of Fast Fourier Transforms (FFT), allow the computation of the coefficients of the nonlinear term to be performed in  $O(N \log N)$  operations.

The general idea of the transform technique to compute the spectral coefficients of a general nonlinear term as  $f(x)g(x)$  is as follows: Start by computing the physical

values of  $f$  and  $g$  through FFTs ( $O(N \log N)$  operations), perform the pointwise product in physical space ( $O(N)$  operations) and go back to spectral space, applying the inverse transformation to generate the coefficients  $(\hat{f}g)_k$  ( $O(N \log N)$ ). Orszag (1971) termed the conjugation of the Galerkin spectral method with the transform technique to compute nonlinear terms as a pseudospectral method. This term is also used in the literature to denote collocation methods, which we will introduce in the next section. This happens, because depending on the way the collocation method is designed, it becomes equivalent to a pseudospectral method. For the interested reader, we advance that if the collocation points are chosen as nodes of the quadrature formulae used in the pseudospectral method, than this equivalence occurs (see Canuto and Hesthaven).

### 2.3 Non-periodic problems

For non-periodic problems, the sets of functions chosen as bases are solutions of certain differential equations, the Sturm-Liouville problems. For instance, the set of Chebyshev polynomials defined as  $T_k(x) = \cos(k \arccos(x))$  are solutions to the following differential equation in the interval  $[-1, 1]$ :

$$-\left(\sqrt{1-x^2}T'_k(x)\right)' = \frac{k^2}{\sqrt{1-x^2}}T_k(x). \quad (8)$$

It is easy to prove that these functions are indeed polynomials, since  $T_0 \equiv 1, T_1(x) = x$  and by using trigonometric identities one can arrive to the recurrence relation  $T_{k+1} = 2xT_k - T_{k-1}$ .

The expansion assumes the form

$$f_N(x) = \sum_{k=0}^N \hat{f}_k T_k(x) \text{ with } \hat{f}_k = \int_{-1}^1 f(x) T_k(x) w(x) dx \text{ and } w(x) = (1-x^2)^{-\frac{1}{2}}.$$

Note that, after the change of variable  $x = \cos(\theta)$ , the Chebyshev polynomials become cosine functions, enabling many theoretical results to be transported from the Fourier system, including spectral convergence.

If a function  $f$  is expanded on a Chebyshev series as  $f = \sum_{k=0}^{\infty} \hat{f}_k T_k$ , then its derivative can also be expressed on a Chebyshev series as  $f' = \sum_{k=0}^{\infty} \hat{f}_k^{(1)} T_k$ , whose coefficients  $\hat{f}_k^{(1)}$  are computed with the help of the recurrence relation

$$2T_k(x) = \frac{1}{k+1}T'_{k+1}(x) - \frac{1}{k-1}T'_{k-1}(x), \quad k \geq 1. \quad (9)$$

For instance, when considering an approximation  $f_N$  as above, we have from (9) that  $2k\hat{f}_k^{(1)} = c_{k-1}\hat{f}_{k-1}^{(1)} - \hat{f}_{k+1}^{(1)}$ , for  $k \geq 1$ , with  $c_k = 2$ , for  $j = 0, N$  and 1 otherwise. We

also have  $\hat{f}_k^{(1)} = 0$ , for  $k \geq N$ , and the remaining coefficients can be computed in decreasing order by

$$2k \hat{f}_k^{(1)} = c_{k-1} \hat{f}_{k-1}^{(1)} - \hat{f}_{k+1}^{(1)}, \text{ for } 0 \leq k \leq N-1.$$

This relation can be easily generalized to obtain derivatives of higher order (see [8]).

Another property inherited from the Fourier system is the possibility of applying fast transforms between spectral and physical spaces. This makes the Chebyshev polynomials the preferred system among the many possibilities of nonperiodic bases. Much more information on the Chebyshev polynomials can be found at the books of Fox and Parker (1968) and Rivlin (1974).

Another set of great use is the set of Legendre polynomials, defined as the solutions of the Sturm-Liouville problem

$$-((1-x^2)L_k'(x))' = k(k+1)L_k(x). \quad (10)$$

Although there is no fast transform for these polynomials, the spectral coefficients of the expansion  $f = \sum_{k=0}^{\infty} \hat{f}_k L_k$  are defined as

$$\hat{f}_k = \int_{-1}^1 f(x) L_k(x) dx,$$

where the weight function  $w(x) = 1$  makes much easier to prove theorems for this set than for the Chebyshev polynomials. Here, the book of reference is the one by Szegő (1939).

Note that both functions in the left hand side of equations (8) and (10) vanish at  $\pm 1$ . It is this property that allows the vanishing of the boundary terms on the integration by parts process, yielding spectral convergence to the Chebyshev and Legendre bases. Sturm-Liouville problems with this property are named singular.

Infinite domains can also be treated by polynomials. Hermite and Laguerre polynomials are the common bases used for infinite and doubly infinite domains respectively. It is also common to map infinite domains to finite ones and use the Chebyshev polynomials (see [8]).

### 3 Polynomial Interpolation

In the collocation version of spectral methods, one requires the numerical approximation to be exact on a set of pre-defined points, the collocation points. The general idea is to interpolate the solution on the collocation points and approximate its derivatives by the derivatives of the interpolating polynomial. As we shall see below, collocation spectral methods treat with equal cost problems with linear, variable coefficients or non-linear terms. All products are performed pointwise at the physical space and this makes collocation schemes very similar to a high order finite differences scheme (see [18]).



For the sake of clarity, let us start with the step of computing the numerical approximation of the derivative  $f_x$  of a real function  $f$ , from which we only know its values  $\{f(x_j)\}_{j=0}^N$  at a set of collocation points  $\{x_j\}_{j=0}^N$ . We need to find a polynomial of degree  $N$  that interpolates  $f$  at these  $N + 1$  points. Note that if the degree is less than  $N$ , such polynomial might not exist (not all sets of 3 points are colinear). It is also easy to show that we do not have uniqueness (monic uniqueness) if the degree is greater than  $N$ .

A general formula for such a polynomial depending only on the values  $\{f(x_j)\}_{j=0}^N$  is given by

$$(I_N f)(x) = \sum_{j=0}^N f(x_j)g_j(x), \tag{11}$$

where  $g_j(x)$  is a  $N$ th-degree polynomial satisfying

$$g_j(x_k) = \delta_{jk} = \begin{cases} 1 & \text{if } j = k, \\ 0, & \text{otherwise.} \end{cases}$$

The polynomials  $\{g_j(x)\}_{j=0}^N$  are called cardinal functions. It is easily seen that  $I_N f(x)$  is a  $N$ th-degree polynomial interpolating  $f$ , since it is a linear combination of the  $N$ th-degree polynomials  $g_j(x)$ , and it satisfies

$$(I_N f)(x_k) = f(x_k), \quad k = 0, \dots, N. \tag{12}$$

In this way, given a set of collocation points, we only need to find the  $N + 1$   $N$ th-degree polynomials  $g_j(x)$ , which can be easily computed by means of Lagrangian interpolation through the formula

$$g_j(x) = \frac{P(x)}{P'(x_j)(x - x_j)},$$

where  $P(x) = \prod_{j=0}^N (x - x_j)$ .

The derivative of the interpolating polynomial is simply given by the  $(N - 1)$ th-degree polynomial

$$(I_N f)'(x) = \sum_{j=0}^N f(x_j)g'_j(x), \tag{13}$$

providing an easy way to obtain the approximated values  $(I_N f)'(x_i)$ : we only need to multiply the vector composed of the function values  $\{f(x_j)\}_{j=0}^N$  by the differentiation matrix  $D$ , whose elements are given by  $D_{i,j} = g'_j(x_i)$ . In this way, numerical differentiation through collocation spectral methods can be accomplished through matrix-vector multiplication.

Once the collocation points are provided and the differentiation matrix is obtained, the solution of a PDE is very close. For instance, given an initial temperature distribution  $\vec{u}^0 = [u_0^0, u_1^0, \dots, u_N^0]$ , with  $u_j^0 = u(x_j, t = 0)$ , one can obtain numerical approximations to future temperature profiles through a first order finite difference temporal

discretization of the heat equation  $u_t - \nu u_{xx} = 0$ , conjugated with the collocation spatial derivative:

$$\frac{\bar{u}^{n+1} - \bar{u}^n}{\Delta t} - \nu D_{xx} \bar{u}^n = 0, \quad \text{or} \quad \bar{u}^{n+1} = \bar{u}^n + \Delta t D_{xx} \bar{u}^n, \quad (14)$$

where  $D_{xx}$  is the second derivative pseudospectral matrix that can be obtained by squaring the differentiation matrix  $D$ . If we suppose, as in Section 3.2, that the diffusion coefficient is given by  $\nu(u) = cu$ . Thus, scheme (14) becomes

$$\bar{u}^{n+1} = \bar{u}^n + c\Delta t \bar{u}^n \cdot D_{xx} \bar{u}^n,$$

where the product  $\cdot$  means pointwise product.

Note that at equation (14), all values of  $u$ , including boundary values, are computed simultaneously. In Section 5, we will see that some of the boundary values must be discarded in order to respect the physics of the problem being solved.

### 3.1 The Chebyshev Collocation Method

We will now present the Chebyshev collocation method starting by its most used set of collocation points, the Chebyshev-Gauss-Lobatto points:

$$x_i = \cos\left(\frac{\pi i}{N}\right) \quad i = 0, \dots, N, \quad (15)$$

which take this name because they are the nodes of the Gauss-Lobatto quadrature formula for the Chebyshev polynomials (see [8]). They are also the extrema of the  $N$ -th order Chebyshev polynomial  $T_N(x)$  and can be seen as the projection on the  $x$ -axis of the equispaced set of points  $\theta_i = \frac{\pi i}{N}$ , in the semicircle, as in Figure (2) below

Figure 2: The Chebyshev points as projections on the  $x$ -axis of an equispaced set of points in the semicircle. Note that points are concentrated close to the boundaries.

The  $N$ -th order cardinal function  $g_j(x)$  is given in this case by

$$g_j(x) = \frac{(-1)^{j+1}(1-x^2)T'_N(x_j)}{c_j N^2(x-x_j)} \quad c_j = 1 + \delta_{j,0} + \delta_{j,N}, \quad (16)$$

and the differentiation matrix in closed form is given by

$$D_{kj} = \frac{c_k (-1)^{j+k}}{c_j x_k - x_j}, \quad k \neq j \quad (17)$$

$$D_{kk} = \frac{-1}{2} \frac{x_k}{1-x_k^2}, \quad k \neq 0, N \quad (18)$$

$$D_{00} = -D_{NN} = \frac{2N^2 + 1}{6}. \quad (19)$$

We solved the linear heat equation (4) in the interval  $[-1, 1]$ , using the Chebyshev collocation method, see Figure (3). Since we have imposed temperature zero at both boundaries, we see that the initial condition  $e^{-cx^2}$  converges to a uniform zero temperature on the whole interval, as it should be if we heat a metal bar in the center and immerse its ends on ice.

Figure 3: Solution of the Heat equation by the Chebyshev pseudospectral method.

We end these two introductory sections on spectral methods pointing out that the value of the derivative of a function  $f$  at a point  $x_j$  is a linear combination of all values of  $f$  at the remaining points, as it can be seen from (13). This is why spectral methods are called global methods and also means that the differentiation matrices in collocation spectral methods are full, making more expensive in general the numerical differentiation when compared to Finite Differences or Finite elements, where derivatives are computed at each point using only a small number of adjacent points, a characteristic of local methods.

## 4 Mapping

The Chebyshev and Legendre bases of polynomials are the most used sets of functions for the spectral expansions of solutions of Partial Differential Equations. However, being eigenfunctions of Sturm-Liouville problems, the number of domains which can be modelled with them is limited. In this section we will see how to make use of mappings between two-dimensional domains in order to overcome this barrier and still be able to use valuable numerical tools such as the Fast Fourier Transform.

Mappings also have different applications other than only conforming physical to computational domains. For instance, concentration of points on a region of large gradients might improve resolution. On the other hand, increasing the minimum distance of a set of collocation points increases the stability of temporal integration schemes. Below, we deal with such questions when developing a general framework for 1D mappings, aiming the integration of mappings to differentiation operators in order to solve equations on more complex domains without including the metrics terms into the equations.

### 4.1 Grid Transformation in 1D

Consider a set of collocation points  $\{y_j\}$   $j = 0, \dots, N$ , and a mapped set  $x_j = g(y_j)$ . Given a function  $u$ , with values  $u(x_j)$  at the new points  $\{x_j\}$ , the derivative of  $u$  can be evaluated as

$$\frac{du}{dx} = \frac{du}{dy} \frac{dy}{dx}, \quad (20)$$

where  $\frac{dy}{dx} = \frac{1}{g'(y)}$ . In matricial form, (20) can be rewritten as

$$\vec{u}' = MD\vec{u}, \quad (21)$$

where  $M$  is the diagonal matrix with elements  $M_{jj} = \frac{1}{g'(y_j)}$  and  $D$  is the differentiation matrix with respect to the set of points  $\{y_j\}$ . Note that we need only  $O(N)$  more operations to apply  $M$ , after the differentiation process  $D\vec{u}$  is concluded.

Since fast transforms are only applicable to the Chebyshev-Gauss-Lobato points (15), we see that the above algorithm keeps the possibility of using the FFT for differentiation on sets of collocation points which are mapped from the Chebyshev ones. If we take the example of the most simple and common 1D mapping, which is used to conform the Chebyshev computational domain  $[-1, 1]$  to a physical domain  $[a, b]$  of interest:

$$x = g(y) = a + \frac{(y+1)}{2} (b-a). \quad (22)$$

Then, the matrix  $M$  is the constant diagonal matrix  $M_{jj} = \frac{b-a}{2}$ .

In general, 2D or 3D grids are cross-products of 1D sets of points. For instance, one can simulate a periodic channel by considering a Chebyshev grid in the vertical and a Fourier grid in the horizontal direction. One can also define a grid on the circle by using this same arrangement in the radial and angular directions, respectively.

This is a natural way to proceed since the derivative operators work in each direction separately. Thus, characteristics of the 1D distribution are transported to the 2D or 3D grid and some properties of the multidimensional method are determined by individual properties of each onedimensional grid component.

A relevant practical issue when numerically solving evolutionary PDEs is the size of the maximum timestep allowed by the numerical method. It is empirically shown that this size is in inverse proportion to the minimum separation between the points, which is directly proportional to the number of points. The Chebyshev points are naturally agglomerated at the boundaries and this causes the timestep to be exceedingly small in comparison to other methods. For instance, Finite Differences schemes allow a  $O(N^{-1})$  time step when solving the first order wave equation, while a  $O(N^{-2})$  size must be respected to obtain stability with spectral methods. In [26], Kosloff and Tal-Ezer proposed the following mapping in order to mitigate this problem:

$$x = g(y, \alpha) = \frac{\arcsin(\alpha y)}{\arcsin(\alpha)}, \quad (23)$$

The mapping above stretches the grid spacing on the boundary pushing the points to the center of the domain, generating a quasi-uniform grid, where the strength of the endpoints separation depends on  $\alpha \in (0, 1)$ . Kosloff and Tal-Ezer claimed that the mapping decreases the spectral radius of the derivative operator from  $O(N^2)$  to  $O(N)$ , increasing the allowed time step from  $O(N^{-2})$  to  $O(N^{-1})$ , in the case of the one-way wave equation. However, it can be easily seen that the mapping has singularities at  $y = \pm \frac{1}{\alpha}$ , which introduce some approximation error and make necessary some expertise on choosing the correct value of  $\alpha$ . See [26] and [17] for detailed theoretical and numerical information on the choice of  $\alpha$ , which might depend on  $N$ .

Besides the increase of the timestep, the mapping (23) also increases the resolution of the high modes because the maximum grid spacing  $\Delta x_{\max}$  is diminished. Resolution of a mode is attained when exponential decay is observed on the spectral coefficients. Thus, the improvement of resolution can be measured by looking to the number of points necessary to achieve this order of decay for a fixed frequency. Figure(4) shows the size of the spectral coefficients of  $\cos(mx)$  for varying  $m$  and  $N$ , where we see that resolution is attained exactly at the point where

$$N > mg'(0, \alpha) = \frac{m\alpha}{\arcsin \alpha},$$

where it should be noticed that  $\Delta x_{\max} = g'(0, \alpha) \Delta y_{\max}$  increases when  $\alpha \rightarrow 1$ . For a more up to date discussion on the Tal-Ezer mapping see [4],[9], [13],[30],[31] and [28].

Figure 4: Decay of the spectral coefficients of  $\cos(mx)$  for several values of  $m$  and number of gridpoints  $N$  showing the dependence of spatial resolution on the parameter  $\alpha$ .

## 4.2 Transfinite Mapping

In this section we define a simple and useful bidimensional mapping which consists of mapping the square  $[0, 1]^2$  onto any quadrilateral with curved boundaries. The Transfinite mapping is defined by four functions from the segment  $[0, 1]$  onto the curves defining the sides of the quadrilateral.

Let  $\vec{\pi}_i(\xi) = (x_i(\xi), y_i(\xi))$ ,  $\xi \in [0, 1]$ ,  $i = 1, 3$  and  $\vec{\pi}_j(\eta) = (x_j(\eta), y_j(\eta))$ ,  $\eta \in [0, 1]$ ,  $j = 2, 4$  be the functions defining the sides of the quadrilateral, then the transfinite mapping is expressed by

$$\begin{aligned} \vec{\Omega}(\xi, \eta) = & \frac{1-\eta}{2}\vec{\pi}_1(\xi) + \frac{1+\eta}{2}\vec{\pi}_3(\xi) \\ & + \frac{1+\xi}{2}\left(\vec{\pi}_2(\eta) - \frac{1+\eta}{2}\vec{\pi}_2(1) - \frac{1-\eta}{2}\vec{\pi}_2(-1)\right) \\ & + \frac{1+\xi}{2}\left(\vec{\pi}_4(\eta) - \frac{1+\eta}{2}\vec{\pi}_4(1) - \frac{1-\eta}{2}\vec{\pi}_4(-1)\right). \end{aligned} \quad (24)$$

This mapping was first presented in [19] and in the figures below we show some domains which can be generated. Two very interesting cases are the ones of the circle and the ellipse. Below, in Figure (5), we show a circular grid which is obtained by dividing the boundary of the circle in four curves and mapping them to the interval  $[0, 1]$  as above. This is an easy way to avoid the singularities of polar coordinates which appears at the center of the circle due to the vanishing of the radius.

Figure 5: Circular grid generated by the Transfinite Mapping.

Figure (6) below shows the solution of the second-order wave equation

$$u_{tt} - c^2 \Delta u = 0,$$

on an elliptical domain, generated in a similar way as the circular one above, simulating the travelling of sound waves between the foci of an ellipse.

Figure 6: Acoustic wave equation simulation on the elliptical grid generated by the transfinite mapping.

### 4.3 Two-Dimensional Mappings

Consider the classical case of the polar coordinates transformation

$$(x(r, \theta), y(r, \theta)) \mapsto (r \cos \theta, r \sin \theta), (r, \theta) \in [r_1, r_2] \times [0, 2\pi],$$

where the radial direction is discretized with the Chebyshev collocation points and the angular, with the Fourier points. If we need to solve a problem in the annular domain of Figure (7), this is clearly the transformation to be used.

Figure 7: Grid obtained from the polar mapping.

For instance, the problem one wants to solve might be the case of a fluid between two concentric and rotating cylinders. Suppose we are only interested in the dynamics on a cross section perpendicular to the axis and, to simplify even more, let us look only to the diffusion of heat on the space between the cylinders. Thus, the equation to be discretized is the two-dimensional heat equation

$$u_t - \nu (u_{xx} + u_{yy}) = 0. \quad (25)$$

We have two alternatives: The first one is to apply the change of variables to the equation and solve it in the new variables  $r$  and  $\theta$ . For instance, the new equation would become

$$u_t - \nu \left( u_{rr} + \frac{u_r}{r} + \frac{u_{\theta\theta}}{r^2} \right) = 0. \quad (26)$$

This approach has the drawback of requiring some analytical work that sometimes can get cumbersome if the mapping is a little bit more involved. Take a look at the transfinite mapping (24) above.

The second approach is to express only the derivatives in  $(x, y)$  of the original equation in terms of the computational variables  $(r, \theta)$ . For instance, differentiation with respect to  $x$  is carried out using the chain rule:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial u}{\partial \theta} \frac{\partial \theta}{\partial x}.$$

The derivatives with respect to  $r$  and  $\theta$  are respectively the Chebyshev and Fourier derivatives, but we still need to spend some work on the computation of the quantities



$\frac{\partial r}{\partial x}, \frac{\partial r}{\partial y}, \frac{\partial \theta}{\partial x}$  and  $\frac{\partial \theta}{\partial y}$ . No problem in the polar mapping case, however, look again to the transfinite mapping and see that it can get very troublesome.

Thus, it would be handy to have a formula for  $\frac{\partial u}{\partial x}$  and  $\frac{\partial u}{\partial y}$  not involving the metrics  $\frac{\partial r}{\partial x}, \frac{\partial r}{\partial y}, \frac{\partial \theta}{\partial x}$  and  $\frac{\partial \theta}{\partial y}$ , but the inverse metrics  $\frac{\partial x}{\partial r}, \frac{\partial x}{\partial \theta}, \frac{\partial y}{\partial r}$  and  $\frac{\partial y}{\partial \theta}$ , which can be computed on a straightforward way. We can resort to some very simple linear algebra and write

$$\begin{pmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial \theta} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial \theta} & \frac{\partial y}{\partial \theta} \end{bmatrix} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix}.$$

We solve the system above for  $\left[ \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right]$  by simply using the straightforward formula for the inverse of a  $2 \times 2$  matrix and obtain:

$$\begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} = \frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial \theta} & -\frac{\partial y}{\partial r} \\ -\frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial r} \end{bmatrix} \begin{pmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial \theta} \end{pmatrix}, \tag{27}$$

where  $J = \left( \frac{\partial x}{\partial r} \frac{\partial y}{\partial \theta} - \frac{\partial y}{\partial r} \frac{\partial x}{\partial \theta} \right)$  is the Jacobian of the mapping.

At this point, we would like to advocate a computational procedure which makes simpler the solution of PDEs on mapped domains. Looking at formulae, we see that after computing the inverse metrics and the Jacobian, which can be done at the begining of the computations, we can define the operators  $\frac{\partial}{\partial x}$  and  $\frac{\partial}{\partial y}$  and code the equation in its original form, with no need to include the metrics. The library PseudoPack contains routines for the computation of the metrics and the Jacobian, which makes the computation of a term like  $u_{xx}$  as simpler as

```
Call PS-Diff (D t, D r, u, uxx, 2, radx, rady, thx, thy, Jacob)
```

We will see below, in Section 5, that this procedure is of great help on clarifying the coding and implementation of more involved equations as the Navier-Stokes and enables the clear use of more complex differential operators, like the ones of Vector Calculus.

## 5 Applications

### 5.1 The Wave Equation

The first-order wave equation:

$$\begin{cases} u_t + cu_x = 0, & t > 0, \quad x \in [-1, 1], \\ u(x, 0) = u_0(x), & x \in [-1, 1], \end{cases} \tag{28}$$

provides us with a typical situation on the study of boundary conditions: The one where some quantity is carried through the domain towards its boundary. It can be seen as the modelling of a travelling wave on a unidimensional medium as a material pulse on a rope or an acoustic wave on a piece of wire. The constant  $c$  is the speed of the wave. One can easily check that

$$u(x, t) = u_0(x - ct) \tag{29}$$

is the solution to equation (28), and if  $c > 0$ , it means that information is being propagated to the right, or that values of the solution at a point  $x$  depends on values of the function at points to the left of  $x$  and at previous times.

If one is interested in the propagation of the initial pulse  $u_0(x)$  with no other influence than the pure advection phenomenon, no boundary condition must be set at the point  $x = 1$ , since the solution values are going to be "brought" there by the numerical integration itself (remember that spectral differentiation also computes values for the boundary points, see Section 3). The imposition of a boundary condition such as  $u(1) = 0$ , as it might be naively proposed, would simulate a numerical barrier on which the travelling pulse would hit and reflect backwards with inverse values. Just think about swinging a rope with one end fixed at a wall. On the other hand, we necessarily have to impose a boundary condition at  $x = -1$  and discard the value computed by the spectral differentiation, otherwise, we would be going against the restriction expressed by equation (29) that information must always come from the left.

If we use the Gauss-Lobato set of points  $\{x_j\}$ , then  $x_0 = 1$  and  $x_N = -1$  are boundary points and both are updated at the temporal integration process, since the differentiation matrix includes lines for these points. However, the value of  $u$  at  $-1$  has to be thrown out and substituted by a convenient boundary condition, such as

$$u(-1) = 0, \quad (30)$$

at the end of every step (at the end of every stage, when using a Runge-Kutta scheme, see [11]).

**Remark 5.1** *In the case above, the inclusion of the boundary point  $x = 1$  is not essential for the simulation. In these situations one could use instead the Gauss-Radau set of points*

$$x_j = \cos \frac{2\pi j}{2N+1}, \quad 1 \leq j \leq N.$$

*In the case of the second-order wave equation, however, both endpoints are needed, since we will have waves moving in both directions, as we shall see below.*

**Remark 5.2** *Statement (30) is equivalent to zeroing the last row of the differentiation matrix. This can be generalized to the case of a second order PDE with homogeneous boundary conditions, where one would fill the first and last lines with zeroes. This does not apply if one has nonzero Dirichlet boundary conditions.*

The typical situation provided by the one-way wave equation (28) will help us to design boundary conditions for problems of greater interest like the Navier-Stokes equations. Equations whose solutions involve propagation of information at a finite speed are called hyperbolic equations and (28) is their very simplest example. In what follows, we will try to reduce more complicated hyperbolic equations to a form which is similar to the one-way wave equation:

$$U_t + AU_x = 0, \quad (31)$$

where  $U$  is a vector with the dependent variables and  $A$  is a matrix that does not depend on the derivatives of  $U$ , but may depend of  $U$  itself. For this reason, we denote (31) a quasilinear system of partial differential equations.

Another characteristic of hyperbolic equations in the form of the system (31), is that all the eigenvalues of  $A$  are nonzero real numbers, in this way we are able to diagonalize  $A = \Gamma\Lambda\Gamma^{-1}$  (for the sake of simplicity, let us disregard the case of defective eigenvalues), where  $\Gamma$  and  $\Lambda$  are the matrices of eigenvectors and eigenvalues of  $A$ , respectively. Applying the variables transformation  $V = \Gamma^{-1}U$  to equation (31) yields the uncoupled system of equations

$$V_t + \Lambda V_x = 0. \tag{32}$$

Since each equation of the above system is similar to the one-way wave equation, we are able to construct boundary conditions for all the transformed variables.

The eigenvalues of  $A$  are the velocities with which the new dependent variables in  $V$  are being propagated. For practical applications, the diagonalization of  $A$  is performed ahead of the temporal integration process and the system (32) is used only to impose the boundary conditions to the transformed variables. The process ends by the computation of the boundary values of the original variables by an inverse transformation.

Below, we illustrate this technique with the case of the second order wave equation  $u_{tt} - c^2u_{xx} = 0$  by reducing it to the first order linear system  $U_t = AU_x$ , where

$$U = \begin{pmatrix} u \\ v \end{pmatrix} \text{ and } A = \begin{bmatrix} 0 & c \\ c & 0 \end{bmatrix}.$$

The diagonalization of the matrix  $A = \Gamma\Lambda\Gamma^{-1}$  makes possible to write the system above in its characteristic form:

$$\begin{pmatrix} u + v \\ u - v \end{pmatrix}_t = c \begin{pmatrix} u + v \\ u - v \end{pmatrix}_x, \tag{33}$$

since

$$\Gamma = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} c & 0 \\ 0 & -c \end{bmatrix}.$$

Defining the characteristic functions  $R_1 = u + v$  and  $R_2 = u - v$ , we easily obtain the solution of (33) as

$$R_1(x, t) = R_1(x - ct, 0) \text{ and } R_2 = R_2(x + ct, 0).$$

Thus,  $R_1$  has to have its right boundary condition imposed and  $R_2$ , its left one.

This reasoning is particularly useful when simulating infinite domains, since they impose no barrier for the travelling quantity to leave the computational domain. Due to its derivation, these are called characteristic boundary conditions. Another name for the characteristic functions above is "Riemann Quantities" and they are the main subject of study when designing numerical schemes for hyperbolic conservation laws.

For us, they will be of great use in Section 6 when performing this same characteristic decomposition for the more complex (and more interesting) system of the Navier Stokes equations. They will also be useful for the setting of interface conditions for multi-domain implementation, in Section 6.

Figure 8: Second-order wave equation with characteristic boundary conditions.

In Figure (8) above, we show the solution of the second-order wave equation using characteristic boundary conditions with an initial pulse at the middle of the domain. This initial pulse is decomposed into two smaller ones travelling to opposite directions. Note that when passing through the boundaries, little wiggles start to appear, but nothing is reflected inside the domain and once the waves are gone, nothing is "sounding".

## 5.2 Nonlinearity: Burgers Equation

The viscous Burgers equation in 1D is given by:

$$u_t + uu_x - \nu u_{xx} = 0, \quad (34)$$

and is the simplest equation where the phenomena of diffusion and nonlinear advection are present. It is therefore very used on testing numerical methods for fluid dynamics and their proposed improvements.

Another important feature of the Burgers equation is the development of discontinuities, or shocks, in finite time. In Figure (9) below, we used the Fourier collocation method with 64 points in order to show the numerical solution of (34), with a Gaussian function centered at the middle of the periodic domain as the initial condition. The advective speed is given by  $u$  itself, which means that the initial profile is going to move to the right, however, since the velocity start to decrease at center of the domain, particles departing from the center will colide with slower particles on their right, causing the discontinuity observed in the numerical solution. The value of the diffusion constant  $\nu$  is the measure of how much the solution is going to behave as the heat equation. Thus, smaller the value of  $\nu$ , sharper is the discontinuity. In this example, a value of  $\nu = 0.001$  was used.

The occurrence of a discontinuity in the solution poses an extra difficulty for spectral methods. The oscilations emanating from the discontinuity are known as the Gibbs phenomenon and are caused by the fact that a discontinuous solution is being approximated by an oscilatory set of smooth functions.

Filtering is an effective tool to get rid of the oscillations and recover exponential accuracy away from the discontinuity. The central idea of using filters is to increase the order of convergence away from the discontinuity by attenuating the high order coefficients, which decay only linearly. Care must be taken in order to maintain the important information contained in these coefficients, otherwise we might obtain a strongly smeared function.

The Gibbs phenomenon is a relevant issue due to the contamination of the solution with the spurious oscillations. The use of filters is recommended in these situations and in Figure (5.2) below, we see the attenuation of the oscillations after the use of a filter of the exponential type, also paying the price of an extra smoothing of the discontinuity. More information on the several types of filters and their properties can be found in [8].

Figure 9: Burgers Equation in 1D.

In the two-dimensional case, the viscous Burgers equation assumes the form:

$$\begin{cases} U_t - \nu \Delta U + (U \cdot \nabla U) = f, & \text{in } (0, 1)^2, \quad t \text{ in } [0, T], \\ U = g & \text{on } \partial\Omega, \\ U(x, y, 0) = U_0(x, y), & \forall (x, y) \in \Omega, \end{cases} \quad (35)$$

where  $U = (u, v)$ . The initial condition  $U_0(x, y) = (\cos(\pi x) \cos(\pi y), (\frac{x+y}{4}))$  and forcing function  $f(x, y) =$

$$(\sin(3\pi x) \sin(4\pi y + 3 \sin(16\pi x) \sin(8\pi y)), \sin(4\pi x) \sin(2\pi y + 2 \sin(10\pi x) \sin(16\pi y)))$$

are chosen in such a way to promote enough dynamics in the fluid, forcing an interaction between the several frequencies involved, and generate some "turbulence". As it can be seen in Figure (10), we have a discontinuity front close to  $x = 0.6$ . In this example, taken from [7], we used 64 points in each direction and a timestep of size  $\Delta t = 10^{-4}$ . Without the use of the exponential filter, the code became unstable.

Figure 10:

### 5.3 2D Flow through an Obstacle -The Compressible Navier Stokes Equation

We now have all the necessary ingredients (mapping, filtering, boundary conditions) to simulate a more realistic example. We chose the classical case of the flow past a circular cylinder. We will use the two-dimensional compressible viscous Navier–Stokes equations in strong conservation form (see [2] and [32]). These equations describe the laws of conservation of mass, momentum and energy when the system is isolated from the influence of external forces. They are much more involved equations than the ones we have been dealing, nevertheless, their derivation can be found in many basic books on Fluid Dynamics, for instance, Anderson [2] presents a clear and thorough construction of these equations in two and three dimensions.

The non-dimensional form of these equations in the Cartesian coordinates  $(x, y)$  is

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = \frac{1}{\text{Re}} \left( \frac{\partial F_\nu}{\partial x} + \frac{\partial G_\nu}{\partial y} \right) \quad (36)$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix},$$

and

$$\begin{aligned}
 F &= \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (E + P)u \end{bmatrix}, & F_\nu &= \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \gamma\kappa \text{Pr}^{-1} \frac{\partial T}{\partial x} + u\tau_{xx} + v\tau_{xy} \end{bmatrix}, \\
 G &= \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (E + P)v \end{bmatrix}, & G_\nu &= \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \gamma\kappa \text{Pr}^{-1} \frac{\partial T}{\partial y} + u\tau_{xy} + v\tau_{yy} \end{bmatrix}.
 \end{aligned} \tag{37}$$

The variables are the density  $\rho$ , the velocity field  $(u, v)$  and the total energy  $E$ . The parameter  $\gamma = \frac{c_p}{c_v}$  is the ratio of specific heats at constant density and volume,  $\text{Pr}$  is the Prandtl number and  $\kappa$  is the coefficient of thermal conductivity.

The stress tensor elements are given by

$$\tau_{xx} = \frac{2}{3} \mu \left( 2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right), \quad \tau_{xy} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \quad \tau_{yy} = \frac{2}{3} \mu \left( 2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right).$$

and the viscosity  $\mu$  is related to the temperature by the Sutherland viscosity law

$$\mu = \frac{(1 + S)T^{\frac{3}{2}}}{S + T}, \tag{38}$$

where  $S$  is a thermodynamic non-dimensionalized constant.

The Navier Stokes equations are coupled with the non-dimensional equations of state and internal energy, respectively:

$$P = (\gamma - 1) \rho T, \quad E = \rho \left[ T + \frac{1}{2}(u^2 + v^2) \right].$$

The Navier–Stokes equations (36) and their necessary supplementary relations such as the Sutherland viscosity law are characterized by four parameters [32]. These parameters are the free–stream Mach number  $M_\infty$ , the free–stream Reynolds number  $\text{Re}$ , the diameter of the cylinder  $D$ , and the dimensional temperature  $T_0$ . The free–stream Reynolds number  $\text{Re}_\infty = \rho_\infty U_\infty D / \mu_\infty$  of the flow is based on the diameter of the cylinder  $D$ , the free–stream velocity  $U_\infty$ , the free–stream density  $\rho_\infty$  and the dynamic viscosity  $\mu_\infty$ , which are the values of these quantities away from the cylinder.

The physical domain will be represented by the ring-shaped region shown in Figure (7) below, which is the image of the computational domain  $(\xi, \eta) \in [0, 2\pi] \times [-1, 1]$  through the polar mapping.

The Navier–Stokes equations (36) written in the coordinates  $(\xi, \eta)$  assume the form

$$\frac{\partial \bar{Q}}{\partial t} + \frac{\partial \bar{F}}{\partial \xi} + \frac{\partial \bar{G}}{\partial \eta} = \frac{1}{\text{Re}} \left( \frac{\partial \bar{F}_\nu}{\partial \xi} + \frac{\partial \bar{G}_\nu}{\partial \eta} \right), \tag{39}$$

where

$$\bar{Q} = \frac{1}{J} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix},$$

$$\begin{aligned} \bar{F} &= \frac{F}{J} \frac{\partial \xi}{\partial x} + \frac{G}{J} \frac{\partial \xi}{\partial y}, & \bar{F}_\nu &= \frac{F_\nu}{J} \frac{\partial \xi}{\partial x} + \frac{G_\nu}{J} \frac{\partial \xi}{\partial y}, \\ \bar{G} &= \frac{F}{J} \frac{\partial \eta}{\partial x} + \frac{G}{J} \frac{\partial \eta}{\partial y}, & \bar{G}_\nu &= \frac{F_\nu}{J} \frac{\partial \eta}{\partial x} + \frac{G_\nu}{J} \frac{\partial \eta}{\partial y}; \end{aligned} \quad (40)$$

and  $J$  is the Jacobian of the transformation  $(\xi, \eta) \rightarrow (x, y)$ , given by  $J = \xi_x \eta_y - \xi_y \eta_x$ .

All the derivative terms in (37) have to be computed with respect to the coordinates  $(\xi, \eta)$ . Therefore, the elements of the stress tensor are given by

$$\begin{aligned} \tau_{xx} &= \frac{2}{3} \mu \left[ 2 \left( \frac{\partial u}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \eta} \frac{\partial \eta}{\partial x} \right) - \left( \frac{\partial v}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial v}{\partial \eta} \frac{\partial \eta}{\partial y} \right) \right], \\ \tau_{xy} &= \mu \left[ \left( \frac{\partial u}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial u}{\partial \eta} \frac{\partial \eta}{\partial y} \right) + \left( \frac{\partial v}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial v}{\partial \eta} \frac{\partial \eta}{\partial x} \right) \right], \\ \tau_{yy} &= \frac{2}{3} \mu \left[ 2 \left( \frac{\partial v}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial v}{\partial \eta} \frac{\partial \eta}{\partial y} \right) - \left( \frac{\partial u}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \eta} \frac{\partial \eta}{\partial x} \right) \right] \end{aligned}$$

and the gradient of the temperature  $T$ :

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial T}{\partial \eta} \frac{\partial \eta}{\partial x}, \quad \frac{\partial T}{\partial y} = \frac{\partial T}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial T}{\partial \eta} \frac{\partial \eta}{\partial y}.$$

In this problem, we take  $\gamma = 1.4$ . and  $\text{Pr} = 0.72$

Figure (11) shows a contour plot of the density function  $\rho$  after the trail of vortices started to develop, as the streamlines of the velocity field show. Go to [www.labma.ufrj.br/~bcosta](http://www.labma.ufrj.br/~bcosta) to see the movie.

We used 64 points in both directions and applied the Kosloff-Tal-Ezer mapping in the radial direction. An exponential filter of order 16 was also applied. The boundary conditions were imposed through the characteristic decomposition as above, in order to simulate a free boundary. For more information on the characteristic variables decomposition and further results on the flow through an obstacle problem, the interested reader can check [22].



Figure 11: Contour density plot for the flow through an obstacle problem.

Equations (36) and (39) can be written in the general form

$$\frac{\partial Q}{\partial t} + \nabla_{x,y} \cdot (F, G) = \frac{1}{\text{Re}} \nabla_{x,y} \cdot (F_v, G_v) \quad (41)$$

and one only has to note that  $\nabla_{\xi,\eta} \cdot (\bar{F}, \bar{G})$  is the conservative form of the Divergence  $\nabla_{x,y} \cdot (F, G)$  in the general Cartesian coordinates  $(x, y)$  (see [2]). It is now worth pointing that the software PseudoPack2000 has all the tools needed for the transformation from a classical set of computational variables like Fourier, Chebyshev or Legendre to a general set of curvilinear coordinates like polar, cylindrical, spherical or user defined, including the computation of the metrics and the Jacobian. Below, we show the one-page code in Fortran 90 necessary to implement the main part of the Navier-Stokes program to generate Figure (11) when using the subroutines of PseudoPack2000.

```

!*****
! Compute the velocities u and v and the Temperature
!*****
u = Q(:, :, 2)/Q(:, :, 1)
v = Q(:, :, 3)/Q(:, :, 1)
T = Q(:, :, 4)/Q(:, :, 1)
T = T - HALF*(u**2+v**2)
!*****
! Compute the viscosity array mu
!*****
mu = (ONE+S)*T*SQRT(T)/(S+T)
!*****
! Compute the derivatives of u and v with respect to x and y
!*****
Call PSGrad(Dt, Dr, u, ux, uy, radx, rady, thx, thy, Jacob)
Call PSGrad(Dt, Dr, v, vx, vy, radx, rady, thx, thy, Jacob)
!*****
! Compute the Stress Terms
!*****
Tauxx = rre*mu*(c43*ux-c23*vy)
Tauxy = rre*mu*( uy+ vx)
Tauyy = rre*mu*(c43*vy-c23*ux)
!*****
! Compute Tx and Ty
!*****
Call PSGrad(Dt, Dr, T, Tx, Ty, radx, rady, thx, thy, Jacob)
Tx = cq*rre*mu*Tx
Ty = cq*rre*mu*Ty
!*****
! Compute the Pressure term
!*****
P = Q(:, :, 1)*T*gm1
!*****
! Compute the Divergence of (F-Fv, G-Gv) in the conservative form
!*****
F(:, :, 1) = Q(:, :, 2)
F(:, :, 2) = Q(:, :, 2) *u - Tauxx + P
F(:, :, 3) = Q(:, :, 3) *u - Tauxy
F(:, :, 4) = (Q(:, :, 4)+P)*u - Tauxx*u - Tauxy*v - Tx
G(:, :, 1) = Q(:, :, 3)
G(:, :, 2) = Q(:, :, 2) *v - Tauxy
G(:, :, 3) = Q(:, :, 3) *v - Tauyy + P
G(:, :, 4) = (Q(:, :, 4)+P)*v - Tauxy*u - Tauyy*v - Ty
do i = 1,4
Call PSDiv(Dt, Dr, F(:, :, i), G(:, :, i), DFlux(:, :, i), radx, rady, thx, thy, Jacob, 1)
enddo

```

## 5.4 Multidomain

As we saw in Section 4, the bases used in spectral methods are sets of eigenfunctions of Sturm-Liouville problems, having limitations with the types of domain that they can be used at. A common way to represent a solution in complex geometries is to use a multidomain setting, where each domain element is of a simple type, or can be mapped to a simple one. For instance, the figure below shows a multidomain representation of the circular obstacle problem with an inner tube through the obstacle.

Figure 12: Multidomain grid for the cylinder with an inner tube.

This setting came from the general problem of Vortex Induced Vibrations (VIV) common to submarine raisers in oil extraction. It was noticed experimentally that positioning the tube in order to allow passage of the fluid through the obstacle would eliminate the resonance of the obstacle induced by the vortex generation. In this simple representation, we are only extending the problem of last section to a multidomain configuration, therefore, the equations are being solved separately in each subdomain. For passing information through the domain interfaces, we are simply averaging the solution, ensuring its continuity. While being a very simple setting, it shows the potentiality of the transfinite mapping associated to a multidomain spectral scheme. It is also worth pointing that heavy filtering is necessary in order to sustain stability on the corners of the tube. In this experiment we applied a global filtering of the exponential type, however, this can be improved by also applying a heavier filter only to the solution at the corners. There is a whole new area of research for the multidomain setting, the spectral element method, where the idea is to use much smaller elements than the ones above ( see [25]).

## 5.5 The Fortran 90 Library PseudoPack2000

While several software tools for the solution of partial differential equations (PDEs) exist in the commercial (e.g. DiffPack) as well as the public domain (e.g. PETSc), they are almost exclusively based on the use of low-order finite difference, finite element or finite volume methods. Geometric flexibility is one of their main advantages.

For most PDE solvers employing pseudospectral (collocation) methods, one major component of the computational kernel is the differentiation operator. The differentiation must be done accurately and efficiently on a given computational platform for a successful numerical simulation. It is not an easy task given the number of choice of algorithms for each new and existing computational platform. Issues involving the complexity of the coding, efficient implementation, geometric restriction and lack of high quality software library tended to discourage the general use of pseudospectral methods in scientific research and practical applications. In particular, the lack of standard high quality library for pseudospectral methods forced individual researchers to build codes that were not optimal in efficiency and accuracy. Furthermore, while pseudospectral methods are at a fairly mature level, many critical issues regarding efficiency and accuracy have only recently been addressed and resolved. The knowledge of these solutions is not widely known and appears to restrict a more general usage of such schemes.

When developing the software PseudoPack2000, the authors aimed to develop a numerical software to make available to the user, in a high performance computing environment, a library of subroutines providing an accurate, versatile, optimal and efficient implementation of the basic components of global pseudospectral methods on which to address a variety of applications of interest to scientists.

This library provides subroutines for computing the derivative of the Fourier collocation methods for periodical domain and Chebyshev and Legendre collocation methods for the non-periodical domain. State-of-the-art numerical techniques such as Even-Odd Decomposition [33], and specialized fast algorithms are employed to increase the efficiency of the library. Kosloff-Tal-Ezer Mapping is used for reducing roundoff error in the Chebyshev and Legendre collocation methods [17]. Moreover, highly accurate Lagrange polynomial interpolation is used when forming the differentiation matrices [12], decreasing solution contamination by roundoff error when dealing with a large number of grid points. Other routines for filtering and grid mapping are also included. Since the user is shielded from any coding errors in the main computational kernels, reliability of the solution is enhanced. PseudoPack will speed up code development, increase productivity and enhance re-usability.

The library contains several simple user callable subroutines that return the derivatives and/or filtering (smoothing) of, possibly multi-dimensional, data sets. In term of flexibility and user interaction, any aspect of the library can be modified by a simple change of a small set of input parameters. All source codes are written in Fortran 90. Using the macro and conditional capability of the C Preprocessor, this software package can be compiled into several versions with several different computational platforms. Several popular computational platforms (IBM RS6000, SGI Cray, SGI, SUN) are supported to take advantages of any existing optimized native library such

as General Matrix-Matrix Multiply (GEMM) from Basic Linear Algebra Level 3 Subroutine (BLAS 3), Fast Fourier Transform (FFT) and Fast Cosine/Sine Transform (CFT/SFT). The Fortran 90 Library PseudoPack2000 is available by request at the webpage of the authors and it comes with a manual.

## 6 Conclusions

The main goal of this article was to provide a more than rough introduction to the subject of spectral methods and their applications in the study of PDEs and the field of the Dynamics of Fluids. We followed the approach of starting from the very beginning and only building the necessary elements to arrive at a meaningful application, like the flow through an obstacle. Thus, many points of relevant interest were ignored and, sometimes, only quickly mentioned through the text. Among them we can cite time integration processes and their consequences towards numerical stability. Nevertheless, the reader with some experience on FD or FE noticed that the mere substitution of the discrete differentiation operator by the spectral one will make most of the temporal integration formulae work with higher precision, with a much smaller time step, however.

The interested reader might now check the cited literature and a very effective starting point is the book of Trefethen [35], containing the essentials of spectral collocation methods and a large range of applications. The book is rich of practical examples using a very powerful tool for scientific computing, the Matlab language. Those interested on a more complete overview will be very pleased. For more extensive insights, the work of Canuto et al [8] is a good encyclopedia in spectral methods directed towards Fluid Dynamics applications, as well as the more recent monography by Fornberg [18], where a close relationship between collocation spectral methods and high order Finite Difference schemes is developed. The book by Boyd [6] also contains plenty information on spectral methods and several areas of research, as well as a substantial discussion on spherical harmonics and other bases for problems in the sphere, with applications to Meteorology. The reader interested in Multidomain spectral methods can found state-of-the-art information in the book by Karniadakis. Finally, the work of Gottlieb and Orszag [20] will satisfy those looking for the mathematical foundations of spectral methods in a modern setting.

## References

- [1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1972.
- [2] Anderson, J. D.; Degrez, G.; Dick, E.; Grundmann, R. *Computational fluid dynamics. An introduction*. Edited and with a preface by John F. Wendt. A von Karman Institute Book. Springer-Verlag, Berlin, 1992.

- [3] J. Augenbaum, *An adaptive pseudospectral method for discontinuous problems*, ICASE report No. 88-54. Appl. Numer. Math. (1988)
- [4] M. R. Abril-Raimundo and B. García-Archilla, *Approximation properties of a mapped Chebyshev method*, Appl. Numer. Math., Vol. 32, (2000) pp. 119-136.
- [5] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw Hill Book, New York (1978)
- [6] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed., Dover, New York, 2000.
- [7] Calgario, C.; Laminie, J.; Temam, R. *Dynamical multilevel schemes for the solution of evolution equations by hierarchical finite element discretization*. Appl. Numer. Math. 23 (1997), no. 4, 403–442.
- [8] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer Series in Computational Physics, Springer, Berlin, 1988.
- [9] J. M. Carcione, *A 2D Chebyshev Differential Operator for the Elastic Wave Equation*, Comput. Meth. Appl. Mech. Engr., Vol. 130, (1996), pp. 33-45.
- [10] J. W. Cooley, P. A. Lewis and P. D. Welch, *The Fast Fourier Transform Algorithm: Programming considerations in the Calculations of Sine, Cosine and Laplace Transforms*, J. Sound Vib, Vol 12, No. 3, pp 315-337 (1970)
- [11] B. Costa, L. Dettori, D. Gottlieb and R. Temam, *Time Marching Techniques for the Nonlinear Galerkin Method*, SIAM Journal of Sci. Comput., Vol. 23 (2001), No. 1, pp. 46-65.
- [12] B. Costa and W. S. Don, *On the Computation of Higher Order Pseudospectral Derivatives*, Appl. Numer. Math., Vol. 33, No. 1, (2000), pp. 151-159.
- [13] B. Costa, W. S. Don and A. Simas, *Spectral Convergence of Mapped Chebyshev Methods*, submitted.
- [14] B. Costa and W.S. Don, *General Mappings for the Chebyshev Collocation Method*, in preparation.
- [15] T. J. Dekker, *A Floating-Point Technique for Extending the Available Precision*, Numer. Math. (18), pp. 224-244 (1971)
- [16] W. S. Don and A. Solomonoff, *Accuracy and speed in computing the Chebyshev collocation derivative*, SIAM J. Sci. Comp, Vol.16, No. 6, pp. 1253-1268
- [17] W. S. Don and A. Solomonoff, *Accuracy Enhancement for Higher Derivatives Using Chebyshev Collocation and a Mapping Technique*, SIAM J. Sci. Comput., Vol. 18, (1997) pp. 1040-1055.

- [18] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, New York, 1996.
- [19] W. J. Gordon and C. A. Hall, *Transfinite Element Methods: Blending-Function Interpolation over Arbitrary Curved Elements Domains*, Numer. Math. 21, pp. 109-129, 1973.
- [20] D. Gottlieb, S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia, PA, 1977.
- [21] D. Gottlieb and W. S. Don, *The Chebyshev-Legendre Method: Implementing Legendre Methods on Chebyshev Points*, SIAM J. Numer. Anal., Vol. 40, No. 5, pp. 1666-1682.
- [22] D. Gottlieb and W. S. Don, *Spectral Simulation of Unsteady Compressible Flow Past a Circular Cylinder*
- [23] D. Gottlieb and J. S. Hesthaven, *Spectral Methods for Hyperbolic Problems*, Journal of Computational and Applied Mathematics, 128 (2001), 83-131.
- [24] J. S. Hesthaven, P. G. Dinesen and J. P. Lynov, *Spectral Collocation Time-Domain Modeling of Diffractive Optical Elements*, J. Comput. Phys., Vol. 155, (1999), pp. 287-306.
- [25] Karniadakis, George Em; Sherwin, Spencer J. *Spectral element methods for CFD*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 1999.
- [26] D. Kosloff and H. Tal-Ezer, *Modified Chebyshev Pseudospectral Method With  $O(N^{-1})$  Time Step Restriction*, J. Comput. Phys., Vol. 104, (1993), pp. 457-469.
- [27] A. I. Markushevich, *Theory of Functions of a Complex Variable*, Chelsea, New York (1977).
- [28] J. L. Mead and R. A. Renaut, *Accuracy, Resolution and Stability Properties of a Modified Chebyshev Method*, SIAM J. Sci. Comput., Vol. 24, No. 1, (2002), pp. 143-160.
- [29] S. A. Orszag, *Comparison of Pseudospectral and Spectral Approximation*, Studies in Applied Mathematics, Vol. LI, No. 3, 1972, pp. 253-259.
- [30] S. C. Reddy, J. A. C. Weideman and G. F. Norris, *On a Modified Chebyshev Pseudospectral Method*, (unpublished, Oregon State University, Corvallis, OR).
- [31] R. A. Renaut and Y. Su, *Evaluation of Chebyshev Pseudospectral Methods for Third Order Differential Equations*, Numer. Algo., Vol. 16, No. 3, (1997), pp. 255-281.
- [32] P. J. Roache, *Computational Fluid Dynamics*, Hermosa Publishers, pp 209-223, 1976.

- [33] Solomonoff, Alex *A fast algorithm for spectral differentiation*. J. Comput. Phys. 98 (1992), no. 1, 174–177.
- [34] P. Swarztrauber, *Symmetric FFT's*, Math. Comp., Vol 47, No. 175, pp. 323-346 (1986)
- [35] L. N. Trefethen, *Spectral Methods in Matlab*, SIAM, Philadelphia, 2000.
- [36] L. N. Trefethen and M. R. Trummer, *An Instability Phenomenon in Spectral Methods*, SIAM J. Numer. Anal., Vol. 24, No. 5, (1987)
- [37] J. A. C. Weideman and L. N. Trefethen, *The Eigenvalues of Second-Order Spectral Differentiation Matrices*, SIAM J. Numer. Anal., Vol. 23, No. 6, (1988)