



Tensor calculus in polar coordinates using Jacobi polynomials



Geoffrey M. Vasil^{a,*}, Keaton J. Burns^b, Daniel Lecoanet^c, Sheehan Olver^a,
Benjamin P. Brown^d, Jeffrey S. Oishi^e

^a School of Mathematics & Statistics, University of Sydney, NSW 2006, Australia

^b Dept. Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

^c Dept. Physics and Theoretical Astrophysics Center, University of California, Berkeley, CA 94720, USA

^d LASP and Dept. Astrophysical & Planetary Sciences, University of Colorado, Boulder, CO 80309, USA

^e Dept. Physics & Astronomy, Bates College, Lewiston, ME 04240, USA

ARTICLE INFO

Article history:

Received 21 October 2015

Received in revised form 17 June 2016

Accepted 11 August 2016

Available online 17 August 2016

Keywords:

Numerical analysis

Partial differential equations

Orthogonal polynomials

Jacobi polynomials

Fluid mechanics

Pipe flow

ABSTRACT

Spectral methods are an efficient way to solve partial differential equations on domains possessing certain symmetries. The utility of a method depends strongly on the choice of spectral basis. In this paper we describe a set of bases built out of Jacobi polynomials, and associated operators for solving scalar, vector, and tensor partial differential equations in polar coordinates on a unit disk. By construction, the bases satisfy regularity conditions at $r = 0$ for any tensorial field. The coordinate singularity in a disk is a prototypical case for many coordinate singularities. The work presented here extends to other geometries. The operators represent covariant derivatives, multiplication by azimuthally symmetric functions, and the tensorial relationship between fields. These arise naturally from relations between classical orthogonal polynomials, and form a Heisenberg algebra. Other past work uses more specific polynomial bases for solving equations in polar coordinates. The main innovation in this paper is to use a larger set of possible bases to achieve maximum bandedness of linear operations. We provide a series of applications of the methods, illustrating their ease-of-use and accuracy.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Cylindrical polar coordinates find applications in countless areas of science and engineering. Important applications include pipe flow, laboratory studies of thermal convection, astrophysical accretion disks, electromagnetic waveguides, elastic deformation of rods, astronomical instrumentation, and plasma tokamaks. Many applications require the accurate and efficient solution of systems of partial differential equations (PDEs). Pseudospectral methods of different types prove useful for this task in many different geometries. In polar coordinates, the periodic nature of the azimuth angle allows the effective use of Fourier series, where

$$f(r, \theta) = \sum_{m=-\infty}^{\infty} f_m(r) e^{im\theta}, \quad f_m(r) = \frac{1}{2\pi} \int_0^{2\pi} f(r, \theta) e^{-im\theta} d\theta. \quad (1)$$

* Corresponding author.

E-mail address: geoffrey.vasil@sydney.edu.au (G.M. Vasil).

After the Fourier transform, differentiation in θ becomes multiplication by im . While Fourier analysis easily dispatches the azimuthal coordinate for functions on a disk, the radial coordinate presents difficulty for the following reason. For functions analytic everywhere on the disk, including the origin,

$$f_m(r) \sim r^m F(r^2) \quad \text{as } r \rightarrow 0, \quad (2)$$

where $F(r^2)$ is an even function of r that is analytic at the origin.

The coordinate singularity at the disk centre requires an m -th order zero for infinite differentiability [26,3]. Enforcing this condition in numerical calculations presents challenges; especially for $m \gg 1$. Many authors address this challenge with equally as many different techniques. Even considering regularity at the origin, the disk geometry allows a large number of possible orthogonal-polynomial bases [8,15]. Zernike (1934) [36] produced the first practical set of polynomials for expanding functions on the unit disk. This basis proves particularly useful in optical applications. Bhatia and Wolf (1954) [1] pointed out that this set is the only out of a possible infinity that contains “*simple properties strictly analogous to that of Legendre polynomials.*”

Boyd and Yu (2011) [3] provide a comprehensive review of the history and contemporary methods used to solve Poisson’s equation in a disk. In particular, the paper reviews bases using Zernike-type polynomials, as well as the more common Chebyshev polynomials. The results for Chebyshev series range from acceptable to untenable. The diversity of Chebyshev methods results from different ways to represent the pole condition and/or the reflectional symmetry near the origin. A minimalist approach happens to produce the best option. This option expands even/odd- m modes in terms of an even/odd-degree Chebyshev series. This approach double wraps the disk using a Chebyshev series over $-1 \leq r \leq 1$ [34,10]. Compared to other Chebyshev options, simple even–odd matching works well with no other special intervention [17]. Even–odd matching and/or double-covering can satisfy equation (2) with good-to-moderate accuracy. These schemes however do not enforce the analytic condition explicitly. This implies that singularities can still arise in higher-order derivatives; also see [17]. Even weak singularities can produce instabilities at the origin when performing time-evolution simulations. As a third option, the Roberts basis combines an even Chebyshev series with an explicit r^m prefactor. In spite of initial attractiveness (e.g., possessing a fast transform), this basis suffers from extreme numerical ill conditioning, and is not recommended [3].

Regarding the Zernike-type bases, Boyd and Yu point out that they are “*More accurate for large m .*” They also discuss the less-fortunate fact that Zernike bases do not admit a fast transform in the radial direction. But that for various reasons “*the advantages of ‘FFT-ability’ is not huge.*” They conclude that “*It is difficult to definitely endorse one particular method for the disk because of the vast diversity of solutions to interesting engineering and science problems.*” Furthermore, Slevinsky (2016) has recently made significant progress toward designing an effective fast transform from values at Chebyshev points to Jacobi coefficients that would work with the Zernike basis [30]. For these reasons and more, we believe that polynomial bases that satisfy equation (2) are very useful in many applications and are worthy of more detailed understanding.

In addition to scalar-valued functions, many situations also require vector and tensor fields. Vectors introduce additional complications near the coordinate singularity. Much less work exists addressing these issues. In particular, the m -th Fourier components of a vector field behave such that

$$v_m(r) \sim r^{m-1} V(r^2) \quad r \rightarrow 0, \quad (3)$$

where, like $F(r^2)$ in equation (2), $V(r^2)$ is an even function of r that is analytic at the origin. We can (for example) see the necessity of equation (3) by differentiating equation (2) with respect to r .

Sakai and Redekopp (2009) [31] circumvented this issue by working with rescaled variables of the form $rv_m(r)$; which behaves like equation (2). Li, Livermore and Jackson (2010) [16] use a poloidal–toroidal formulation to create a genuine (higher-order) scalar system out of a specific vector system. Using a technique equivalent to the r rescaling, Matsushima and Marcus (1995) [18] show (and Boyd and Yu [3] reiterate) that Zernike polynomials produce *pentadiagonal* matrices for the solution of the radial portion of Poisson’s equation. Lastly, Townsend, Wilber and Wright (2016) develop an efficient low-rank approximation of scalar and vector functions on the disk that preserve regularity [33]. These methods work well for data analysis. Their application to time evolving systems remains less clear.

We show in this paper the non-necessity of radial rescaling and/or equation reformulation. Previous works found recursion relationships for elementary operators r^2 and rd/dr . Our calculus finds simpler factorisations of these operations in terms of r , d/dr and m/r . These are all of the elementary operators needed for full tensor calculus. This not only makes calculations easier to formulate, but also more numerically efficient and stable. In the process, we also show how to construct solutions to Poisson’s equation on the disk using only *tridiagonal* (as opposed to pentadiagonal) matrices; see the discussion in Example 1 in §6 for more details. The foundation of the new results rests on exploiting a more general class of orthogonal polynomials. That is, we choose different bases to represent domain and range spaces of operators, so that coupling becomes banded. This mirrors using ultra-spherical polynomials for solving equations on the unit interval [5,7,23]. Moreover, we incorporate azimuthally symmetric variable coefficients without destroying bandedness. This occurs via approximating non-constant coefficients with finite-degree polynomials similar to [23].

A central theme of this paper demonstrates that increasing the collection of available bases can increase (i) the simplicity of a calculation’s numerical implementation; (ii) the speed to compute a solution; and (iii) the accuracy of the result. We outline our following results: §2 derives properties for useful bases for polar coordinates (using properties of Jacobi polynomials). §3 shows how these bases respond to the covariant derivative operator in polar coordinates. §4 discusses multiplication by radial functions. §5 shows how the different bases relate to each other, and how the different operators

form a Heisenberg Lie algebra. §2–5 build the necessary tools to represent and manipulate scalars, vectors, and tensors in different calculations. §6 applies these tools to a series of four example problems. §7 gives concluding remarks.

2. Calculus of Jacobi polynomials

Throughout this paper, we use the definition

$$z \equiv 2r^2 - 1, \quad (4)$$

where $0 \leq r \leq 1$, implies $-1 \leq z \leq +1$. The two following formulae provide the foundation for a simple calculus of functions on the unit disk. For any differentiable function $F(z)$,

$$\left[\frac{d}{dr} - \frac{m}{r} \right] r^m F(z) = 4r^{m+1} \frac{d}{dz} F(z), \quad (5)$$

$$\left[\frac{d}{dr} + \frac{m}{r} \right] r^m F(z) = 2r^{m-1} \left[m + (1+z) \frac{d}{dz} \right] F(z). \quad (6)$$

The z -differential operators on the right-hand side of equations (5) and (6) act on Jacobi polynomials in a particularly simple way.

Jacobi polynomials $P_n^{(a,b)}(z)$ are the two-parameter set of classical orthogonal polynomials on $-1 \leq z \leq 1$ under the weight $(1-z)^a(1+z)^b$, where $-1 < a, b$:

$$\int_{-1}^1 P_n^{(a,b)}(z) P_{n'}^{(a,b)}(z) (1-z)^a (1+z)^b dz = W_n^{a,b} \delta_{n,n'}, \quad (7)$$

and

$$W_n^{a,b} \equiv \frac{2^{a+b+1}}{2n+a+b+1} \frac{\Gamma(n+a+1)\Gamma(n+b+1)}{\Gamma(n+a+b+1)n!}, \quad (8)$$

where $\delta_{n,n'}$ represents the Kronecker delta. Jacobi polynomials satisfy a Sturm–Liouville differential equation, and a three-term recursion relation. They also satisfy many other useful formulae¹ [22]. For the purposes of this paper we use:

For differential operators,

$$2 \frac{d}{dz} P_n^{(a,b)}(z) = (n+a+b+1) P_{n-1}^{(a+1,b+1)}(z), \quad (9)$$

$$\left[b + (1+z) \frac{d}{dz} \right] P_n^{(a,b)}(z) = (n+b) P_n^{(a+1,b-1)}(z). \quad (10)$$

For algebraic operators,

$$(1+z) P_n^{(a,b)}(z) = \frac{2(n+1)}{2n+a+b+1} P_{n+1}^{(a,b-1)}(z) + \frac{2(n+b)}{2n+a+b+1} P_n^{(a,b-1)}(z), \quad (11)$$

$$P_n^{(a,b)}(z) = \frac{n+a+b+1}{2n+a+b+1} P_n^{(a+1,b)}(z) - \frac{n+b}{2n+a+b+1} P_{n-1}^{(a+1,b)}(z). \quad (12)$$

For restriction operators,

$$P_n^{(a,b)}(z=1) = \binom{a+n}{n}. \quad (13)$$

Exchanging the roles of a and b , we can produce similar relations as equations (9)–(13), using

$$P_n^{(a,b)}(-z) = (-1)^n P_n^{(b,a)}(z). \quad (14)$$

Doha and Bhrawy (2006) advocate for using Jacobi polynomials of varying degree to solve a number of problems in one and two dimensions. They specifically state that “The key to the efficiency of these algorithms is to construct appropriate base functions, which lead to systems with specially structured matrices that can be efficiently inverted” [6]. We expand on this philosophy and construct a hierarchy of basis functions that conform particularly well to vector calculus in circular geometry.

We define the following radial basis elements in terms of Jacobi polynomials,

$$Q_n^{k,m}(r) \equiv \frac{r^m P_n^{(k,m)}(z)}{\sqrt{N_n^{k,m}}}; \quad N_n^{k,m} \equiv \frac{W_n^{k,m}}{2^{2+k+m}}, \quad (15)$$

¹ All relevant formulae are found in Digital Library of Mathematical Functions, Chapter on Orthogonal Polynomials, §18.3–§18.9: <http://dlmf.nist.gov/18>.

which implies an orthonormal basis

$$\int_0^1 Q_n^{k,m}(r) Q_{n'}^{k,m}(r) (1-r^2)^k r dr = \delta_{n,n'}. \quad (16)$$

From now on, the m index corresponds to the Fourier mode number in equation (1). Factorials replace Gamma functions in equation (15) for integer values of k . Each element of the radial basis satisfies the analytic property in equation (2). We could consider a non-integer shift in the starting value of m in equation (15); e.g., this could correspond to starting with Worland polynomials rather than Zernike [17]. The following analysis would work with such an alteration, albeit with a large sacrifice to simplicity. We therefore only discuss integer- m values.

The derivative operators for Jacobi polynomials imply the derivative operators on the radial basis

$$\left[\frac{d}{dr} - \frac{m}{r} \right] Q_n^{k,m}(r) = 2\sqrt{n(n+k+m+1)} Q_{n-1}^{k+1,m+1}(r), \quad (17)$$

$$\left[\frac{d}{dr} + \frac{m}{r} \right] Q_n^{k,m}(r) = 2\sqrt{(n+m)(n+k+1)} Q_n^{k+1,m-1}(r). \quad (18)$$

Equations (17) and (18) form the foundation of a compact calculus of fields expanded in terms of $Q_n^{k,m}(r)$.

Translating explicit statements like equations (17) and (18) into more abstract operators acting on vector spaces allows easier interpretation of many of the results and derivations. Therefore, we use Dirac's bra-ket notation² for the following analysis [9]. The following row-vector definition forms the foundation for our algebra,

$$\langle k, m, r | \equiv [Q_0^{k,m}(r), Q_1^{k,m}(r), \dots]. \quad (19)$$

The k, m indices parameterise different bases. The corresponding ket is the transpose $|k, m, r\rangle = \langle k, m, r|^T$. In the bra-ket notation equations (17) and (18) become

$$\frac{1}{\sqrt{2}} \left[\frac{d}{dr} \mp \frac{m}{r} \right] \langle k, m, r | \equiv \langle k+1, m \pm 1, r | D_{k,m}^{\pm}, \quad (20)$$

where $D_{k,m}^{\pm}$ represent single-band matrices. The operators D^{\pm} depend on the k, m indices, but from now on we omit these labels when the context makes their values clear. D^- is a diagonal matrix, and D^+ contains only a single band on the first super diagonal. The special case $m=0$ permits only the raising operation, and $D^- \equiv D^+$. In equation (20), we choose the convention that r -space actions operate on the left, and matrix actions operate from the right.

We see that the derivative operators on equations (5) and (6) provide m -raising and lowering operators on the radial basis. Equations (17) and (18) in many ways resemble the same operators applied to Bessel functions,

$$\left[\frac{d}{dr} - \frac{m}{r} \right] J_m(r) = -J_{m+1}(r), \quad \left[\frac{d}{dr} + \frac{m}{r} \right] J_m(r) = J_{m-1}(r). \quad (21)$$

This is no accident. Both Bessel functions and the scaled Jacobi polynomials behave similarly near the origin. Alternatively, for many reasons Bessel functions are not as useful for representing the solutions to arbitrary PDEs [3]. In general, Bessel-function solutions achieve only algebraic convergence rates. For example, $f = r^2 - 1$ satisfies $\nabla^2 f = 4$, $f(r=1) = 0$. This exact polynomial solution only decays as $\mathcal{O}(n^{-5/2})$ in a Bessel-function series expansion. This behaviour results from a mismatch between the higher-order derivatives of an exact solution and a given approximating expansion; e.g., the third derivative of $r^2 - 1$ vanishes identically, which does not happen for any finite Bessel function series. The same principle favours orthogonal polynomials over special functions in many other cases [25].

3. Tensor calculus

We start with a scalar function

$$f(r, \theta) = \sum_{m=-\infty}^{\infty} f_m(r) e^{im\theta}. \quad (22)$$

Using the bra-ket notation, we expand the radial dependence of the Fourier components in terms of the weighted Jacobi polynomials

$$f_m(r) = \langle 0, m, r | f_m \rangle = \sum_{n=0}^{\infty} f_{m,n} Q_n^{0,m}(r). \quad (23)$$

² See Appendix B for an introduction to the relevant aspects of this notation.

We expand all scalars, vector, and tensor components in terms of the $k=0$, and appropriate m basis. As we differentiate a given component, the basis changes its m index, and the k index increments upward. A later section defines operators to convert between different k, m bases.

The covariant derivative does not couple the azimuthal Fourier modes. We can therefore write the action of its radial and azimuthal components on an individual Fourier mode as

$$\nabla_r f_m(r) = \frac{d}{dr} f_m(r), \quad \nabla_\theta f_m(r) = \frac{im}{r} f_m(r). \quad (24)$$

Using equations (17) and (18),

$$\nabla_r f_m(r) = \frac{1}{\sqrt{2}} \langle 1, m-1, r | D^- | f_m \rangle + \frac{1}{\sqrt{2}} \langle 1, m+1, r | D^+ | f_m \rangle, \quad (25)$$

$$\nabla_\theta f_m(r) = \frac{i}{\sqrt{2}} \langle 1, m-1, r | D^- | f_m \rangle - \frac{i}{\sqrt{2}} \langle 1, m+1, r | D^+ | f_m \rangle. \quad (26)$$

Introducing a spinor basis simplifies the computations of vectors and tensors. The spinor basis elements are

$$e_\pm = \frac{1}{\sqrt{2}} (e_r \mp i e_\theta), \quad (27)$$

where e_r, e_θ represent the unit vectors in the r, θ coordinate directions respectively. A 2×2 unitary matrix transforms between the components of a vector in the r, θ basis and the \pm basis. In the spinor basis, the covariant derivative becomes

$$\nabla = e_r \nabla_r + e_\theta \nabla_\theta = e_- \nabla_- + e_+ \nabla_+, \quad (28)$$

where acting on individual scalar functions,

$$\nabla_\sigma \equiv \frac{1}{\sqrt{2}} \left[\frac{d}{dr} - \frac{\sigma m}{r} \right], \quad (29)$$

with $\sigma = \pm 1$. From standard vector calculus,

$$\nabla_r e_r = \nabla_r e_\theta = 0, \quad \nabla_\theta e_r = \frac{1}{r} e_\theta, \quad \nabla_\theta e_\theta = -\frac{1}{r} e_r. \quad (30)$$

The spin basis diagonalises the connection coefficients such that

$$\nabla_\sigma e_\mu = -\frac{\sigma \mu}{\sqrt{2} r} e_\mu, \quad (31)$$

where $\sigma, \mu = \pm 1$. From the standard three-dimensional vector algebra of e_r, e_θ, e_3 , we can easily deduce

$$e_+ \times e_- = i e_3, \quad e_+ \cdot e_- = 1, \quad e_+ \cdot e_+ = e_- \cdot e_- = 0, \quad (32)$$

where the dot product represents simple component-wise contraction, i.e., *not* the proper complex-valued inner product. Further,

$$e_3 \times e_\pm = \pm i e_\pm. \quad (33)$$

We note that e_3 denotes any locally orthogonal third unit vector. For example, it could represent the axial direction in cylindrical coordinates, or an additional angle in a torus.

Equations (29) and (31) imply the following formula for the covariant derivative of an arbitrary tensor component

$$\begin{aligned} \nabla_\sigma e_\mu \langle k, m + \mu, r | &= -\frac{\sigma \mu}{\sqrt{2} r} e_\mu \langle k, m + \mu, r | + e_\mu \nabla_\sigma \langle k, m + \mu, r | \\ &= \frac{e_\mu}{\sqrt{2}} \left[\frac{d}{dr} - \frac{\sigma(m + \mu)}{r} \right] \langle k, m + \mu, r | \\ &= e_\mu \langle k + 1, m + \mu + \sigma, r | D^\sigma. \end{aligned} \quad (34)$$

Moreover, this formula holds if μ represents a multi-index such that

$$e_\mu = \prod_{i=1}^s e_{\mu_i}, \quad \text{and} \quad \bar{\mu} = \sum_{i=1}^s \mu_i, \quad (35)$$

where $\mu_i = \pm 1$ for each i . If $\mu \in S = [-1, +1]^s$ represents an s -component multi-index, then

$$T_m(r) = \sum_{\mu \in S} e_\mu \langle k, m + \bar{\mu}, r | T_m^\mu \rangle \quad (36)$$

represents a given rank- s tensor. The covariant derivative becomes the rank- $(s + 1)$ tensor

$$\nabla T_m(r) = \sum_{\sigma=\pm 1} \sum_{\mu \in S} e_{\sigma} e_{\mu} \langle k+1, m+\bar{\mu}+\sigma, r | D^{\sigma} | T_m^{\mu} \rangle. \quad (37)$$

In equation (37), k represents whichever basis the tensor began with. It needs not equal the tensor rank, s . In most cases $k = 0$.

Computing the gradient of a scalar

$$\nabla f_m(r) = e_- \langle 1, m-1, r | D^- | f_m \rangle + e_+ \langle 1, m+1, r | D^+ | f_m \rangle. \quad (38)$$

We can further compute the second covariant derivative of a scalar

$$\begin{aligned} \nabla \nabla f_m(r) = & e_- e_- \langle 2, m-2, r | D^- D^- | f_m \rangle + \\ & e_+ e_- \langle 2, m, r | D^+ D^- | f_m \rangle + \\ & e_- e_+ \langle 2, m, r | D^- D^+ | f_m \rangle + \\ & e_+ e_+ \langle 2, m+2, r | D^+ D^+ | f_m \rangle. \end{aligned} \quad (39)$$

The operators D^{\pm} commute in the sense that

$$D_{k+1, m+1}^- D_{k, m}^+ = D_{k+1, m-1}^+ D_{k, m}^-, \quad (40)$$

where the k, m indices show that operators take on different meaning depending on their order. This implies the covariant derivative components commute; as they must for a flat space. Taking the trace of the second derivative gives the Laplacian of a scalar

$$\nabla^2 f_m(r) = 2 \langle 2, m, r | D^+ D^- | f_m \rangle. \quad (41)$$

4. Multiplication and non-constant coefficients

Many applications require multiplying scalar and vectors function by a non-constant, axisymmetric function of r . Here, we present a pair of operators representing multiplication without spoiling the bandedness of the matrix. In their discussion of vector formulations on disks, Sakai and Redekopp (2009) used vector components of the form $rv_m(r)$ [31]. These components behave like scalars. Two additional Jacobi polynomial recursions reconcile this formulation with our current analysis. Equation (11), and equations (12) and (14) together imply two distinct r -multiplication operators

$$r \langle k, m, r | = \langle k, m \pm 1, r | R_{k, m}^{\pm}, \quad (42)$$

where the dependency on k and m is dropped when inferred from context. R^+ has entries on the diagonal and first super-diagonal, while R^- contains entries on the diagonal and first sub-diagonal. For $m = 0$, $R^- \equiv R^+$. Equations (A.6) and (A.9) show the individual elements associated with the operators in equation (42).

Given the gradient vector in equation (38),

$$r \nabla f_m(r) = e_- \langle 1, m, r | R^+ D^- | f_m \rangle + e_+ \langle 1, m, r | R^- D^+ | f_m \rangle. \quad (43)$$

Both components of equation (43) contain the same m dependence as the scalar $f_m(r)$.

Considering equation (4) and both \pm parts of equation (42),

$$z \langle k, m, r | = \langle k, m, r | (2R^- R^+ - I), \quad (44)$$

which leaves k, m unchanged. In the same style as equation (40), radial multiplication commutes with itself,

$$R_{k, m+1}^- R_{k, m}^+ = R_{k, m-1}^+ R_{k, m}^-, \quad (45)$$

where we note that the \pm operators act on different m -bases depending on their order.

The standard three-term recursion relation for Jacobi polynomials implies the symmetric three-term recursion for the radial functions

$$B_n^{k, m} Q_{n-1}^{k, m}(r) + A_n^{k, m} Q_n^{k, m}(r) + B_{n+1}^{k, m} Q_{n+1}^{k, m}(r) = z Q_n^{k, m}(r), \quad (46)$$

where

$$A_n^{k, m} \equiv \frac{m^2 - k^2}{(2n + k + m)(2n + k + m + 2)} \quad (47)$$

$$B_n^{k, m} \equiv \frac{2}{2n + k + m} \sqrt{\frac{n(n+k)(n+m)(n+k+m)}{(2n+k+m)^2 - 1}}. \quad (48)$$

Therefore,

$$Z \equiv 2R^-R^+ - I \quad (49)$$

represents the standard symmetric tridiagonal Jacobi matrix for Jacobi polynomials. The operation of multiplication by any purely radial function occurs via

$$F(2r^2 - 1) \langle k, m, r | = \langle k, m, r | F(Z). \quad (50)$$

We expand $F(z)$ in terms of a truncated series of any polynomials,

$$F(Z) = \sum_{n=0}^{N_F} F_n P_n(Z) \quad (51)$$

An orthogonal polynomial basis allows the advantage of a matrix-valued three-term recursion of the form,

$$B_{n+1} P_{n+1}(Z) + A_n P_n(Z) + B_n P_{n-1}(Z) = Z \cdot P_n(Z) \quad (52)$$

Equation (52) needs not represent any of the recursions in equation (46), but it can. For example, if we choose to expand in terms the same basis as the axisymmetric $k = m = 0$ modes, then

$$A_n = 0, \quad B_n \equiv \frac{n}{\sqrt{4n^2 - 1}} \quad (53)$$

But this is not an essential restriction. We could use Chebyshev polynomials for $P_n(Z)$ if the series in equation (51) converges more rapidly in that basis. In that case, $A_n = 0$, $B_n = 1/2$ for $n > 0$, and $B_0 = 1/\sqrt{2}$.

Reducing the truncation degree, N_F , reduces the bandwidth of the multiplication operator matrices and can save significant computation time even for modest differences in the truncation degree. Z is a tridiagonal matrix, therefore the bandwidth of $F(Z)$ is $2N_F + 1$, which is typically much smaller than the required size of the matrix.

Given any choice in equation (52) and taking $P_{-1} \equiv 0$, we can use Clenshaw's algorithm to compute $F(Z)$ directly via

$$K_{N_F+1} = K_{N_F+2} = 0 \quad (54)$$

$$K_n = F_n I + \frac{1}{B_{n+1}} (Z - A_n I) \cdot K_{n+1} - \frac{B_{n+1}}{B_{n+2}} K_{n+2} \quad (55)$$

$$F(Z) = P_0(Z) K_0, \quad (56)$$

where I represents the identity matrix, and $P_0(Z) = P_0 I$ is a constant times the identity matrix. Each iteration of Clenshaw's algorithm increases the bandwidth by one, via multiplication by the tridiagonal matrix Z . If the coefficients vary strongly with latitude, then the multiplication matrices are dense. But this happens in a gradual way. Most of the time, the coefficients do not vary nearly as rapidly as the solution itself.

Multiplication by non-azimuthally symmetric functions must obey more complicated selection rules for the different m indices. This would create systems that couple both the θ and r directions. We do not consider linear operators with $m > 0$ dependence.

5. Basis conversion

Equation (41) has a problem: the Laplacian of a scalar with $k = 0$ results in a scalar with $k = 2$. We must convert between the k bases in order to compare the input and output.

Equation (12) implies a 2-band conversion operator such that

$$\langle k, m, r | \equiv \langle k + 1, m, r | C_{k,m}, \quad (57)$$

where the dependence on k and m is again dropped when inferred from context. C has entries on the diagonal and first super-diagonal. Like the D^\pm operators, we omit the k, m indices when the context makes it clear. Equation (A.12) shows the individual elements associated with the operators in equation (57).

Raising the k index allows converting to the basis that most naturally represents derivatives. This is the same principle that underlies sparse representations with Chebyshev polynomials, i.e.,

$$\frac{d}{dz} T_n(z) = n U_{n-1}(z), \quad 2T_n(z) = U_n(z) - U_{n-2}(z). \quad (58)$$

Many papers reinventing Chebyshev numerical schemes rely on this simple fact [37,4,2,12,5,14,7,21,23,35,32]. Chebyshev polynomials are special cases of Jacobi polynomials with $a = b = -1/2$ and $a = b = 1/2$ representing the first and second kind, respectively. Using greater flexibility in the a, b indices allows for an increased ability to construct sparse operators for a particular application.

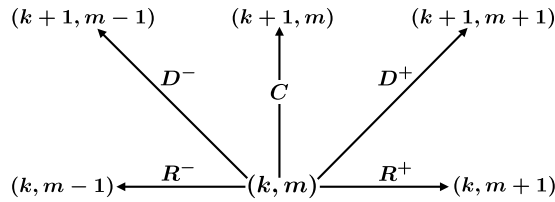


Fig. 1. Diagram of operator structure for the (k, m) indices. Every other node in the graph contains its own set of operators mapping to new nodes in a similar fashion. Only the right half of the diagram exists for $m = 0$.

5.1. Heisenberg algebra

Radial multiplication, differentiation, and basis conversion satisfy simple algebraic compatibility conditions. Differentiation and multiplication satisfy standard commutation relations, but with the added element of conversion.

The commutation relations satisfy the two-dimensional Heisenberg Lie algebra, \mathfrak{h}_2 . That is,

$$[D^+, R^-] = [D^-, R^+] = \sqrt{2}C. \quad (59)$$

Keeping track of the different k, m indices,

$$D_{k,m\mp 1}^\pm R_{k,m}^\mp - R_{k\pm 1,m\pm 1}^\mp D_{k,m}^\pm = \sqrt{2}C_{k,m}. \quad (60)$$

Otherwise all other \pm operators commute with each other and with C . Fig. 1 shows the structure of the different elements of \mathfrak{h}_2 acting on the $|k, m, r\rangle$ state. The diagram omits the equivalent operators acting on the other nodes in the infinite (k, m) lattice. Traditional representation theory considers the Lie algebra of operators acting on a single Hilbert space. Here it appears that we are considering the unconventional situation of a Lie algebra of operators acting *between* Hilbert spaces; each with different k, m . Considering a total Hilbert space comprising the direct sum of all individual k, m spaces rectifies this issue.

The operators also satisfy conditions apart from commutation relations. Taking the θ component of equation (43) implies

$$mC = \frac{1}{\sqrt{2}}(R^+D^- - R^-D^+). \quad (61)$$

Taking the r component of equation (43) implies,

$$r\frac{d}{dr}C = \frac{1}{\sqrt{2}}(R^+D^- + R^-D^+). \quad (62)$$

Equation (62) is equivalent to equation (15) from Matsushima and Marcus (1995) [18].

The Heisenberg algebra allows the automatic reformulation of equations at the numerical level. Equation (59) gives a simple and reliable recipe for commuting radial non-constant coefficients past derivatives using the numerical operators.

5.2. Boundary restriction

Solving PDEs on the unit disk usually requires appropriate boundary conditions at $r = 1$. No boundary at $r = 0$ is needed, nor allowed. To impose the boundary condition at $r = 1$ we use

$$Q_n^{k,m}(r=1) = \sqrt{2(2n+m+k+1)} \binom{n+k}{k} \binom{n+m+k}{k}, \quad (63)$$

which follows from equations (13) and (15). We can build higher-order boundary restriction operators out of equation (63) and various differentiation formulae. For example,

$$\frac{d}{dr}Q_n^{k,m}\Big|_{r=1} = \frac{2n(n+k+1) + m(2n+k+1)}{k+1} Q_n^{k,m}(r=1). \quad (64)$$

It is convenient to modify the basis so that the boundary condition only appears in the first element. This leads to banded systems, as opposed to almost banded systems, which may have dense rows corresponding to the boundary conditions. Many black-box linear algebra packages can solve banded systems very efficiently, while almost-banded systems require specially designed schemes, e.g., the adaptive QR method [23], or use of the Woodbury formula.

We describe how to modify the basis to accommodate Dirichlet boundary conditions, corresponding to (13). The adjoint of the conversion operators gives a k -lowering operator.

$$(1-r^2)\langle k, m, r | = \langle k-1, m, r | C^\dagger \quad (65)$$

where C^\dagger has entries on the diagonal and first sub-diagonal. This follows from a straightforward application of equations (11) and (14). Equation (A.15) gives the explicit indices of equation (65). The k -lowering operator satisfies the commutation relations,

$$[C^\dagger, D^\pm] = \sqrt{2}R^\pm. \quad (66)$$

Otherwise C^\dagger commutes with all other relevant operators. The R^\pm operators are the adjoints of each other. Adjoint of the D^\pm operators exist, but provide little apparent use for the types of problems we consider here.

Define

$$B \equiv C^\dagger + \alpha^{-1}|0\rangle\langle 0|, \quad \text{where} \quad \alpha \equiv \langle k, m, r=1 | 0 \rangle, \quad (67)$$

where $|0\rangle$ represents the unit basis element corresponding to the $n=0$ mode. Equation (67) defines a two-band lower-diagonal operator.

As before,

$$f_m(r) = \langle k, m, r | f_m \rangle \quad (68)$$

If we change variables such that,

$$|f_m\rangle \equiv B|g_m\rangle \quad (69)$$

then

$$\begin{aligned} f_m(r) &= \langle k, m, r | C^\dagger | g_m \rangle + \frac{\langle k, m, r | 0 \rangle}{\langle k, m, 1 | 0 \rangle} \langle 0 | g_m \rangle \\ &= (1-r^2)\langle k+1, m, r | g_m \rangle + \frac{\langle k, m, r | 0 \rangle}{\langle k, m, 1 | 0 \rangle} \langle 0 | g_m \rangle. \end{aligned} \quad (70)$$

Therefore

$$f_m(r=1) = \langle 0 | g_m \rangle. \quad (71)$$

The B operator localises the boundary-restriction operator to the first mode of g . This contrasts with a dense row operation for the original variable. This comes at the expense of a single band in the bulk equation operators, but produces an entirely banded system.

For the derivative, because for every individual element $\langle k, m, r=1 | > 0$, there exists a diagonal operator Λ such that

$$f'_m(r=1) = \langle k, m, 1 | \Lambda | f_m \rangle \quad (72)$$

This also extends to more general left-hand side restriction operators. In the case of the derivative, equation (64) implies the diagonal elements

$$\langle n | \Lambda | n \rangle = \frac{2n(n+k+1) + m(2n+k+1)}{k+1}. \quad (73)$$

Therefore, if

$$|f_m\rangle \equiv \Lambda^{-1}B|g_m\rangle, \quad (74)$$

then

$$f'_m(r=1) = \langle k, m, 1 | B | g_m \rangle = \langle 0 | g_m \rangle. \quad (75)$$

The same principle applies for any Λ , not only for the derivative. Julien and Watson (2009) demonstrate many of the advantages of a sparse transformation to a Galerkin basis for general boundary operators [14]. Equation (73) shows that $\Lambda_{n=0} = 0$ for $m=0$, otherwise $\Lambda_n > 0$. This precludes forming equation (74) for $n=m=0$. Neumann boundary conditions fail to determine this particular mode, which requires additional information. In many applications, setting this mode requires fixing a physically irrelevant (but nevertheless mathematically essential) gauge condition.

5.3. Quadrature

All the above analysis considers field variables in the spectral coefficient domain. This allows for efficient differentiation operations. Given a set of spectral coefficients, we need the option to construct the function on a radial grid. We also need to compute spectral coefficients from data on a grid. In general, we need each m mode on the same radial grid. This allows constructing solutions on the full r, θ disk. We consider an expansion in terms of a finite number of radial modes,

$$f_m(r) = \sum_{n=0}^{N_{k,m}} Q_n^{k,m}(r) \langle k, m, n | f_m \rangle, \quad (76)$$

Table 1

Timings for Fast Chebyshev/Cosine Transform (FCT) versus a Gauss–Legendre Matrix–Multiply Transform for different resolutions. The timings use SciPy (FCT) and NumPy (MMT) on a Late-2013 Mac Pro 2.7 GHz 12-Core Intel Xeon E5. Each timing represents the average over 30,000 forward and inverse transforms of random data; only the actual transforms factor into the average.

N_r	32	64	128	256	512	1024	2048
FCT (μsec)	0.94	1.97	4.42	8.56	17.8	40.7	81.3
MMT (μsec)	0.43	1.75	9.19	30.1	117	462	1905
MMT/FCT	0.45	0.89	2.08	3.51	6.59	11.3	23.4

where $N_{k,m}$ depends on both k and m . We use Gauss–Legendre quadrature on the grid $z_i = 2r_i^2 - 1$ such that

$$\langle k, m, n | f_m \rangle = \sum_{i=1}^{N_r} Q_n^{k,m}(r_i) (1 - r_i^2)^k w_i f_m(r_i) \quad (77)$$

for some N_r . Gauss quadrature is exact for all z polynomials less than degree $2N_r$, or r polynomials less than degree $4N_r$. In the r variable,

$$\deg \left[Q_n^{k,m}(r) (1 - r^2)^k f_m(r) \right] = 2(n + m + k + N_{k,m}) < 4N_r \quad (78)$$

where $n \leq N_{k,m}$. Therefore,

$$N_{k,m} \equiv N_r - 1 - \text{floor} \left[\frac{k + m}{2} \right] \quad (79)$$

yields exact quadrature on a Gauss–Legendre grid. Gauss–Legendre quadrature is based on the $k = 0, m = 0$ case. For non-integer starting k values, w_i, r_i would result from a more general Gauss–Jacobi quadrature, but the basic principle remains the same. In practice, we never need to compute equation (77) with $k > 0$. Instead, we can project any function on r_i to the $k = 0$ spectral coefficients and convert the result up the desired k using the C conversion operator.

The absence of a fast transform is the largest downside to the Jacobi-based method compared to a Chebyshev scheme. In extreme cases of large N_r , where a fast transform would show significant gains, Chebyshev schemes struggle to represent the severe analytic behaviour of the solution near $r = 0$. For example, if m is an even integer, with $p \equiv m/2$, and $k = 0$, then

$$r^m P_n^{0,m}(2r^2 - 1) = 2 \sum_{j=0}^{n+p} T_{2j}(r) \sum_{i=\max\{j,p\}}^{n+p} \frac{\binom{2i}{i+j} \binom{i}{p} \binom{n+p}{i} \binom{i+n+p}{n} (-1)^{i+n+p}}{\binom{n+p}{p} 4^i}, \quad (80)$$

where $T_{2j}(r)$ represents a Chebyshev polynomial of even degree $2j$. The primed-summation notation in equation (80) represents multiplying the $j = 0$ term of the series by $1/2$. The series in equation (80) contains Chebyshev modes up to degree $2n + m$. A similar relation exists for odd m values. Given a relative error tolerance ε , in some cases these series coefficient will become smaller than the allowed error before the end of the exact series. However, even this does not gain much advantage. In the regime $m \sim n$, a Chebyshev series expansion of $Q_n^{0,m}(r)$ requires nearly all of the $2n + m$ modes to achieve an accurate representation. We note that these considerations ignore the possibility of a low-rank approximation to specific functions, which can also exploit a fast transform; see [33].

Also, because $N_{k,m}$ decreases with larger m , a Jacobi-based scheme requires fewer total modes to obtain uniform resolution over the entire disk. That is, after summing equation (79) over $|m| \leq N_\theta/2$ we find

$$N_{\text{tot.}} \approx \left(N_r - \frac{N_\theta}{8} \right) N_\theta, \quad (81)$$

compared to simply $\approx N_r N_\theta$ for a Chebyshev radial scheme. The savings are maximised for $N_\theta \approx 4N_r$, with $N_{\text{tot.}} \approx N_r N_\theta/2$.

Two final mitigating factors arise from (i) the somewhat-large cost of fast transforms for moderate sizes, and (ii) the easy parallelisation of matrix transforms (i.e., quadrature). Table 1 compares the timing of a fast Chebyshev transform (FCT, based on the type-II fast cosine transform), versus quadrature computed with a matrix–vector multiply (MMT). The tests show that while the FCT gives better asymptotic scaling, the MMT does better at moderate sizes. The speed crossover occurs somewhere between 64 and 128 modes; a reasonable size for many applications. The timing difference remains less than an order of magnitude all the way up to 1024 modes. Lastly, a quadrature-based transform uses the orthogonality of the eigenfunctions to compute the inverse transform from the transpose of the forward transform. That is, both the forward and inverse transform represent $N_r N_{k,m}$ operations. However, because of the matrix–vector structure, the operations actually entail N_r totally independent vector–vector dot products. This parallelises quite easily.

6. Examples

This section considers a series of practical examples. [Example 1](#) demonstrates the operator methods on a simple scalar equation. [Example 2](#) demonstrates the same for a vector set of equations. [Example 3](#) concerns non-constant coefficients in a vector system. [Example 4](#) considers a problem with full θ, r -dependence (i.e., with a large number of m values) and considers computational performance.

The first three examples entail finding the generalised eigenvalues and eigenvectors for an analytically tractable, and/or previously published numerical test. We compute eigenvalues because of the demanding nature of these problems. Eigenvalues and eigenvectors give a significant fraction of all the useful information about a given system. For example, the eigenvalue problem determines the stability of general time-stepping schemes. In some cases, defective numerical schemes can produce large spurious eigenvalues in the wrong part of the complex plane [37,11,19]. Such methods can give accurate solutions for a single linear solve with specific right-hand side terms. But rogue eigenvalues in linear operators can render a time stepper useless after a small number of iterations. A small random projection onto a bad eigenvector will diverge exponentially upon iteration. Also, eigenvalues in the wrong part of the complex plane can prevent otherwise conserved quantities from remaining constant.

In each case we build the operator matrices by inspection of the original equation and use standard linear algebra libraries for the actual calculations. Apart from [Example 4](#), we code all the examples in Python, and take advantage of the features within NumPy and/or SciPy. We produce Jupyter notebooks, and scripts for each case and make them available online within the Dedalus project.³

[Example 4](#) is coded in the ApproxFun.jl Julia package [23,24], to utilise its implementation of the adaptive QR method.

Example 1 (*Bessel function eigenvalue problem*). We solve the following eigenvalue problem

$$\frac{1}{r} \frac{d}{dr} \left[r \frac{d}{dr} J_m(\kappa r) \right] - \frac{m^2}{r^2} J_m(\kappa r) = -\kappa^2 J_m(\kappa r), \quad (82)$$

with the Dirichlet boundary condition,

$$J_m(\kappa) = 0. \quad (83)$$

Many available software packages compute Bessel functions and their corresponding zeros. We therefore use this example as a numerical test.

We solve the generalised eigenvalue problem,

$$2D^- D^+ |f_m\rangle = -\kappa^2 CC |f_m\rangle, \quad (84)$$

where CC denotes applying two (different) conversion operators. The left-hand side operator only contains a single off-diagonal band. Each C operator only contains a single off-diagonal band. The product of the two is tridiagonal. This contrasts with the penta-diagonal system for this problem in [18].

The boundary restriction operator in equation (63) replaces the last matrix row on the left-hand side. A row of zeros replace the last matrix row on the right-hand side. After constructing the appropriate right- and left-hand side matrices, we use the NumPy eigenvalue solver to find the eigenvalues and eigenvectors in coefficient space. We sort the results from lowest-to-highest in eigenvalue, and transform the coefficient-space vector to a radial grid; see §5.3 for details regarding the spectral-to-spatial transforms.

[Fig. 2a](#) shows the relative error in the eigenvalues for an $m = 50$ solution set with $N_r = 500$ radial modes. The plot is typical of the behaviour for other m values. We choose this case to examine closely because it gives a good illustration of the ability of the method to capture the large dynamic range near $r = 0$. However the method's performance near the origin is independent of the m value. The behaviour of the numerical eigenvalues is typical of orthogonal polynomial solutions of boundary-value problems, including those using Chebyshev polynomials. Typically, given a value of N_r , approximately the first 1/2 to 2/3 of the spectrum will contain values with virtually no error; apart from machine floating-point error, and a little from the eigenvalue solver routine. In [Fig. 2a](#) this region spans from mode 0 to about mode 300 out of 500 modes. After this point, the exact solutions begin to outstrip the inherent finite-degree polynomial approximation. This error increases rapidly, with the largest eigenvalues scaling such that

$$|\Delta \kappa_n| \sim \kappa_n^2. \quad (85)$$

Like Chebyshev schemes, this behaviour is typical for orthogonal polynomial solutions of two-point boundary eigenvalue problems. [Fig. 2](#) implies that obtaining accurate solutions to Poisson's equation,

$$\nabla^2 f_m(r) = s_m(r) \quad \longleftrightarrow \quad 2D^- D^+ |f_m\rangle = CC |s_m\rangle \quad (86)$$

³ dedalus-project.org.

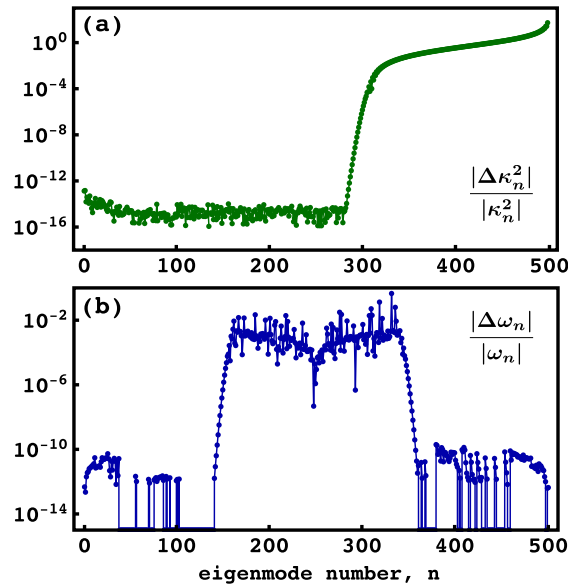


Fig. 2. Plots of relative eigenvalue error from Examples 1 and 2. Each plot compares the numerical eigenvalue solution with the analytically computed value for 500 individual sorted modes. Panel (a) corresponds to Bessel function Example 1 for $m = 50$. Panel (b) corresponds to the inertial waves Example 2 $m = 1$. Each case contains roughly 70% “good” values with high accuracy, and 30% “bad” values with significant error. In both cases, the bad values still lie in the same region on the complex plane as the true solutions. The bad values correspond to small-scale modes that outstrip the fixed resolution. In the Bessel function example, small-scale modes correspond to large κ . Hence the error become large for large mode numbers. For inertial waves, small-scale modes correspond to slow frequencies with $|\omega| \approx 0$. The more accurate large-scale modes correspond to $|\omega| \approx 1$. Therefore, the significant errors on panel (b) occur for intermediate mode numbers; see Table 2.

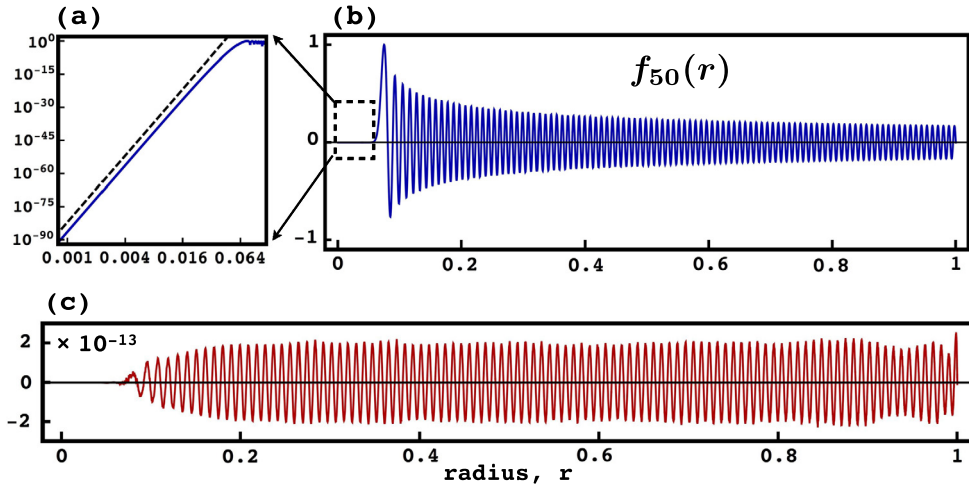


Fig. 3. Plots for the 200th eigenfunction from Example 1 for $m = 50$, and $N_r = 500$. Panel (a) shows a log–log plot of radial dependence near the origin. The solid blue line shows the computed eigenfunction. The eigenfunction should scale as $\sim r^{50}$. The dashed black line denotes this scaling. Panel (b) shows the eigenfunction across the entire radius. The solution vanishes at $r = 1$. Panel (c) shows the error between the computed solution and an analytically computed Bessel function. The characteristic error is $\approx 2 \times 10^{-13}$, and mostly uniform over the domain. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

requires that $s_m(r)$ does not project onto the inaccurate part of the spectrum for a given resolution. This implies that the resolution needed to compute $f_m(r)$ is roughly 3/2 to 2 times larger than needed to resolve $s_m(r)$. But after reaching the critical number of modes, the error is effectively machine precision. This same principle holds for most types of equations.

Fig. 3 shows the eigenfunction behaviour for the $n = 200$ eigenvector for $m = 50$. This corresponds to $\kappa_{200} \approx 707.447066905$. The highly oscillatory solution is accurate to roughly 13 digits mostly uniformly throughout the domain. Fig. 4 shows the spectral coefficients for the same eigenfunction. The magnitude of the coefficients decrease dramatically after approximately 300 modes. This accords with the similar behaviour in the eigenvalues. Trying to represent this mode with fewer than 300 to 400 modes would incur significant error. Beyond this number, more resolution contributes little to the fidelity of the solution.

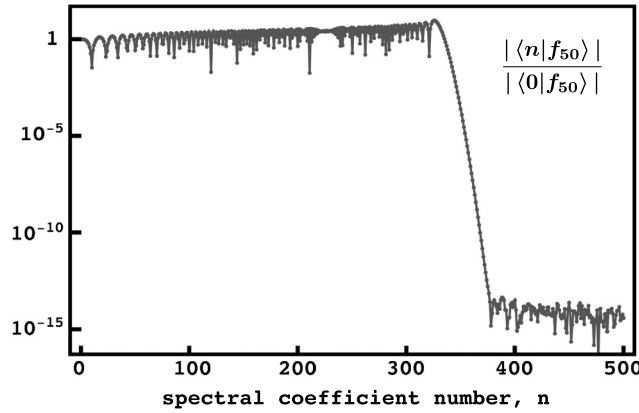


Fig. 4. Plot of the spectral coefficient amplitude of the 200th eigenfunction from Example 1 for $m = 50$. In general, the n th eigenfunction requires $\approx 2n$ spectral coefficients to resolve.

Table 2

Approximate analytically computed eigenvalues versus mode number. The approximate values are intended to give scale to the bottom axes on Fig. 2. κ_n increases monotonically from large-scale modes to small scale. For ω_n , small-scale modes correspond to the middle of the sorted eigenvalue spectrum, and large-scale modes correspond to the values closer to unity. Inertial waves break reflection symmetry. Hence the lack of symmetry $\omega \rightarrow -\omega$.

n	0	100	200	300	400	500
$\kappa_n (m = 50)$	5.71e1	3.91e2	7.07e2	1.02e3	1.33e3	1.64e3
$\omega_n (m = 1)$	-2.14e-1	-3.17e-3	-1.23e-3	1.20e-3	3.12e-3	3.19e-1

Finally, we note that the matrices for radial component in polar coordinates are sparser than the Cartesian Chebyshev analogue in one dimension. Chebyshev discretisation applied in polar coordinates must account for the geometric terms with non-constant coefficients. Using operators that conform to the disk geometry renders systems such as equation (82) effectively constant coefficient problems.

Example 2 (*Inviscid inertial waves in a upright rotating cylinder*). The following example highlights vector–scalar coupled system with a non-trivial analytical solution. It therefore makes a good system for numerical comparison. We consider

$$i\omega \mathbf{v}_m(r) + \mathbf{e}_3 \times \mathbf{v}_m(r) + \nabla p_m(r) = 0, \quad (87)$$

$$i\omega \nabla \cdot \mathbf{v}_m(r) + \alpha^2 p_m(r) = 0. \quad (88)$$

The two-dimensional velocity vector and pressure, $\mathbf{v}_m(r) = v_m^+(r)\mathbf{e}_+ + v_m^-(r)\mathbf{e}_-$ and $p_m(r)$ represent a full eigenstate. The parameter ω represents a frequency eigenvalue. The parameter α represents a mode aspect ratio in the third dimension. We solve equations (87) and (88) with the normal-velocity boundary condition

$$\mathbf{e}_r \cdot \mathbf{v}_m(r=1) = 0. \quad (89)$$

Greenspan (1968) [13] discusses at length the derivation and physical interpretation of equations (87) and (88). We take the system as given. Inertial waves make a good test problem because of the non-trivial frequency constraint

$$-1 < \omega < 1. \quad (90)$$

As with the Bessel function problem, numerical eigenvalue solvers often produce a subset of values that differ significantly from their true solution. It is, however, a sign of a good scheme if all values are maintained in the same region of the complex plane as the true result. As discussed above, a poorly designed scheme can produce very large spurious eigenvalues with a different complex character.

An analytical solution to equations (87)–(89) results from collapsing the vector system to the scalar problem

$$\nabla^2 p_m(r) = -\frac{(1 - \omega^2)\alpha^2}{\omega^2} p_m(r) \quad (91)$$

with

$$\omega \frac{d}{dr} p_m(r) + \frac{m}{r} p_m(r) = 0 \quad \text{at } r = 1. \quad (92)$$

Equation (91) requires a solution in terms of Bessel functions, $p_m(r) = J_m(\kappa r)$. Equation (92) requires

$$\kappa^2 = \frac{(1 - \omega^2)\alpha^2}{\omega^2}, \quad \kappa \omega J'_m(\kappa) + m J_m(\kappa) = 0. \quad (93)$$

We solve equation (93) with a standard numerical root finder.

Recall equation (33), $e_3 \times e_{\pm} = \pm i e_{\pm}$. This implies the following block eigensystem

$$\begin{bmatrix} C & 0 & D^+ \\ 0 & -C & D^- \\ 0 & 0 & \alpha^2 C \end{bmatrix} |\Xi_m\rangle = -\omega \begin{bmatrix} C & 0 & 0 \\ 0 & C & 0 \\ D^- & D^+ & 0 \end{bmatrix} |\Xi_m\rangle, \quad (94)$$

where the eigenstate is the vectorisation of the velocity and pressure,

$$|\Xi_m\rangle \equiv [iv_m^+ \mid iv_m^- \mid p_m]^T. \quad (95)$$

Note that absorbing a factor of $\sqrt{-1}$ into the velocity field renders equation (94) purely real. The boundary conditions are

$$[\langle 0, m+1, 1 \mid \quad \langle 0, m-1, 1 \mid \quad 0 \mid \Xi \rangle = 0 \quad (96)$$

This boundary-restriction operator replaces the last matrix row on top blocks of the left-hand side of equation (94); the blocks corresponding to v^+ . A row of zeros replace the last matrix row of the top blocks on the right-hand side. We solve the numerical discretisation of equation (94) with the same eigenvalue software as Example 1.

Like Example 1, the inertial waves problem only allows one boundary condition for a system with (apparently) two radial derivatives. This results from the coordinate singularity causing a reduction in the differential order at $r = 0$. This manifests in the matrix system because D^- is a non-singular diagonal matrix, and hence requires no additional conditions to invert. Alternatively, the D^+ matrix contains a null space, and requires a boundary condition. We insert the radial-velocity boundary condition in the block corresponding to the $|v_m^+ \rangle$ variable.

Fig. 2b shows the relative eigenvalue error for the inertial wave problem with $m = 1$, $\alpha = 1$. Like the Bessel function example, there is a significant region with very high accuracy, and a region of the spectrum with significant error. Both of these regions occupy roughly the same proportions as the respective regions in the Bessel function problem. The major distinction is that the error occurs in the middle of the spectrum, rather than on one end. This results from the fact that small-horizontal-scale inertial wave oscillate slower than domain-filling modes. The inaccurate part of the spectrum results from under resolving small-scale modes in all cases. In both Example 1 and 2, all the eigenvalues exist in the correct part of the complex plane; on the positive real axis for Example 1, and on the real axis between ± 1 for Example 2. While it is not possible to avoid inaccurate values for some modes, remaining on the real axis is not a guaranteed feature of a numerical approximation. Our scheme succeeds well in this respect.

We discuss $m = 1$ modes in this second example, but note that all modes perform well, just like in Example 1. In the higher- m cases the spectrum becomes squeezed to a smaller region between ± 1 , otherwise there is little difference. The eigenfunction error corresponds to the error in the eigenvalues.

Example 3 (*Linear perturbations of Hagen–Poiseuille pipe flow*). Hagen and Poiseuille first studied pipe flow experimentally in the 1830s [29]. This problem has guided studies investigating the onset of turbulence since the pioneering experiments of Reynolds in the 1880s [28]. A lot of work continues today regarding the nonlinear dynamics of this system [27]. After much investigation, a large sector of the turbulence-modelling community believes all linear perturbations in Poiseuille pipe flow eventually decay exponentially in time. Finite-amplitude perturbations likely produce the rich turbulent behaviour observed in laboratory experiments. This problem interests us because it poses a difficult numerical challenge; independent of the physical significance of the linear dynamics. Meseguer and Trefethen (2003) [20] discuss the linear problem at length. The eigenvalue problem for linear perturbations is:

$$\lambda v_m(r) + \nabla p_m(r) + L v_m(r) = 0 \quad (97)$$

$$\lambda w_m(r) + W'(r) e_r \cdot v_m(r) + i\alpha p_m(r) + L w_m(r) = 0 \quad (98)$$

$$\nabla \cdot v_m(r) + i\alpha w_m(r) = 0, \quad (99)$$

where

$$L \equiv i\alpha W(r) - \nu \nabla^2 + \nu \alpha^2. \quad (100)$$

For the background flow in the stream-wise direction, we choose the pressure-driven parabolic profile

$$W(r) = 1 - r^2, \quad \text{with} \quad W'(r) = -2r. \quad (101)$$

The field $v_m(r)$ represents the two-dimensional vector velocity in the pipe cross section. The scalar fields $w_m(r)$, and $p_m(r)$ represent the velocity in the stream-wise direction, and pressure respectively. The Laplacian implicit in equation (100) represents the scalar Laplacian in equation (98), or the (more complicated) vector operator with curvature terms in equation (97). The advantage of the spinor approach is that these present the same level of complication in the numerical implementation.

The parameters α, λ represent the mode wavenumber along the pipe axis, and the complex-valued growth rate respectively. The viscosity parameter $\nu = 1/\text{Re}$, where Re represents the flow Reynolds number. We require no-slip boundary conditions on the outer wall

$$v_m(r=1) = w_m(r=1) = 0. \quad (102)$$

The full eigenstate now contains four field components

$$|\Xi_m\rangle = [|v_m^+\rangle \quad |v_m^-\rangle \quad |w_m\rangle \quad |p_m\rangle]^T. \quad (103)$$

We solve the generalised eigenvalue problem for the complex-valued growth rate, λ ,

$$\mathcal{L}|\Xi_m\rangle = -\lambda \mathcal{R}|\Xi_m\rangle, \quad (104)$$

where

$$\mathcal{L} \equiv \begin{bmatrix} L & 0 & 0 & CD^+ \\ 0 & L & 0 & CD^- \\ -\sqrt{2}CCR^- & -\sqrt{2}CCR^+ & L & i\alpha CC \\ D^- & D^+ & i\alpha C & 0 \end{bmatrix}, \quad (105)$$

$$\mathcal{R} \equiv \begin{bmatrix} CC & 0 & 0 & 0 \\ 0 & CC & 0 & 0 \\ 0 & 0 & CC & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (106)$$

and $L \equiv i\alpha CC(I - R^+R^-) - 2\nu D^+D^- + \nu\alpha^2 CC$. The boundary conditions become

$$\begin{bmatrix} \langle 0, m+1, 1 | & 0 & 0 & 0 \\ 0 & \langle 0, m-1, 1 | & 0 & 0 \\ 0 & 0 & \langle 0, m, 1 | & 0 \end{bmatrix} |\Xi_m\rangle = 0. \quad (107)$$

As with all other examples, we note that the operators in the different blocks of equations (105) and (106) acquire different (k, m) values depending on the field components.

In addition to the eigenvalue problem, we solve for the parallel-component of the vorticity, ω_3 , and stream function, ψ .

$$\nabla^2 \psi_m = \nabla \cdot (e_3 \times v_m) = i(\nabla_- v_m^+ - \nabla_+ v_m^-) \equiv \omega_3, \quad (108)$$

which translates to

$$2D^-D^+|\psi_m\rangle = iCD^-|v_m^+\rangle - iCD^+|v_m^-\rangle. \quad (109)$$

Often for an incompressible flow, the parallel velocity and stream function provide the dynamical variables in a higher-order-derivative scalar formulation of the original vector system. This example requires no special treatment to enforce the incompressibility constraint in equation (99), which stands on equal footing with the other equations in the system.

Fig. 5 shows full-disk eigenmodes for the first- and second-slowest decaying modes for $\text{Re} = 10^{7/2}$, $\alpha = 1$, and $m = 1$. The plots show the four primitive dynamical variables p, v_r, v_θ, w along with the two *ex post facto* derived fields ω_3, ψ . Below each full-disk image ($x^2 + y^2 \leq 1$), we show a zoomed-in square image on the region $\max(|x|, |y|) \leq 0.2$. The zoom-in plots show no signs of numerical singularities for at the origin. The scalar fields, p, w, ω_3, ψ remain smooth, and the vector quantities show the correct parity. Fig. 6 shows additional full-disk eigenmodes for $m = 5$, and $m = 12$. These plots show that the higher- m still behave as required near the origin. Traditional methods (e.g., Chebyshev bases) can cope with low- m values, but fail for $m \gg 1$. Nonlinear simulations at high Reynolds number require a large range of azimuthal wavenumber to resolve the resulting turbulent states. For these situations, a scheme with a correct accounting for the origin is essential.

Table 3 shows a list of computed growth rates for different m values for $\alpha = 1$, and two different Reynolds numbers. In each case, we report the fastest growing (equivalently slowest decaying) modes of the centre- and wall-localised branches. The $m = 1$ values for both Reynolds numbers match the values reported in Meseguer and Trefethen (2003) [20] to the previously reported 11 digits. We tested all other values reported in [20]. We match all cases. We report additional (higher m) values for potential future numerical comparison. The individual modes reported in Table 3 correspond to the same eigenmodes shown in Figs. 5 and 6. For most cases, the first centre mode decays slower than the first wall mode. But this is not always the case. These modes can lie immediately adjacent to each other in the spectrum, but not necessarily so. For $\text{Re} = 10^4$, we converge to the reported solution after ≈ 50 radial basis elements. For $\text{Re} = 10^7$ we converge after ≈ 200 radial basis elements. In both cases the convergences is independent of m .

Example 4 (Forced Helmholtz equation). We consider the forced Helmholtz equation

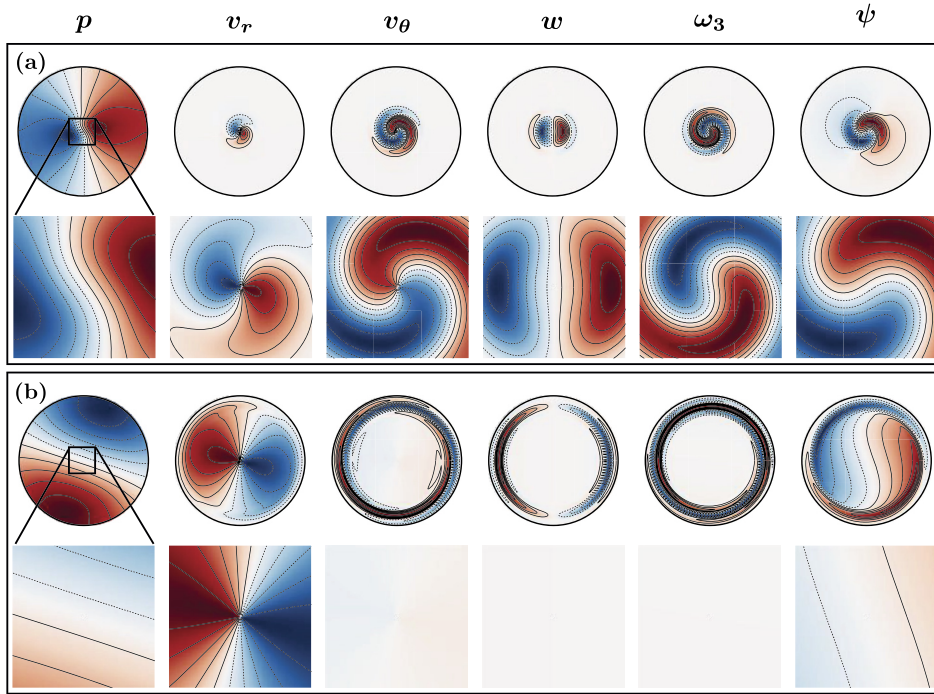


Fig. 5. Full-disk images for the first two modes for pipe-flow with $m = 1$ and $\text{Re} = 3162$, and $N_r = 200$ radial points. Panels (a) and (b) show the first- and second-slowest-decaying mode respectively. In each panel, the square plots show zoom-in plots for $|x|, |y| \leq 0.2$. The behaviour at the origin remains regular for each field.

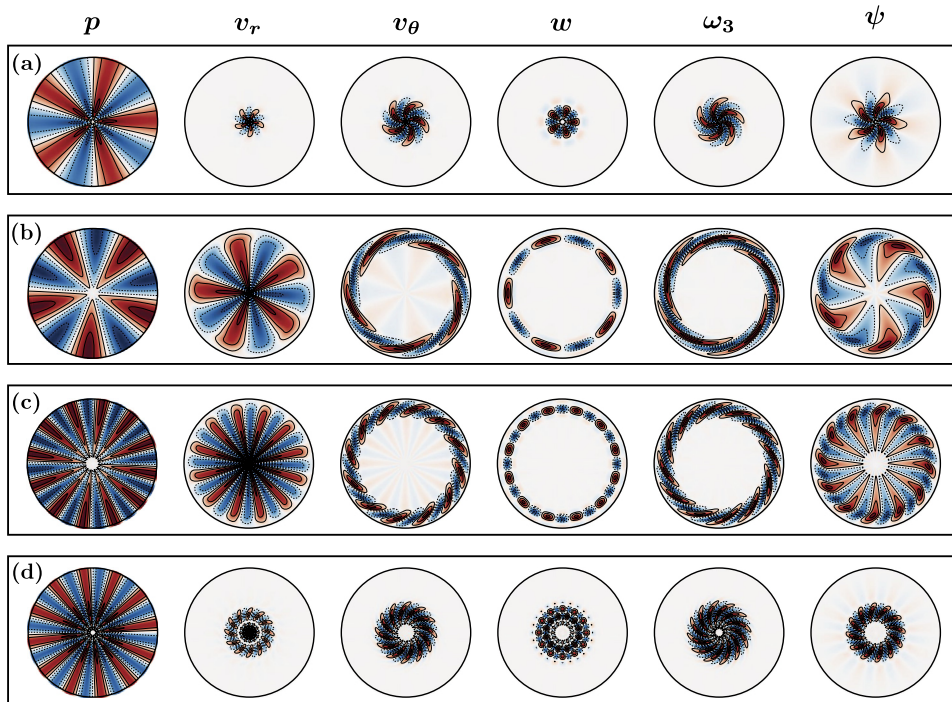


Fig. 6. Additional full-disk images for different pipe-flow modes with $\text{Re} = 3162$, and $N_r = 200$ radial points. (a) corresponds to $m = 5, n = 1$, (b) corresponds to $m = 5, n = 2$, (c) corresponds to $m = 12, n = 1$, (d) corresponds to $m = 12, n = 5$. Zoom-in plots are omitted, but the behaviour at the origin remains smooth for each field.

Table 3

Pipe-flow eigenvalues computed in Example 3. In each case, the “centre” and “wall” modes represent the slowest-decaying solution with eigenfunctions localised near the origin or near the boundary, respectively. The number n represents where a given mode occurs in a sorted list of growth rates.

m	type	n	Re	Real(λ)	Imag(λ)
1	centre	0	10^4	−0.0227049145535	0.951481194735
1	wall	1	10^4	−0.0472321995947	0.273788709331
5	centre	0	10^4	−0.0725274157946	0.898561158159
5	wall	1	10^4	−0.0793504734563	0.247410847332
12	wall	0	10^4	−0.0948648867252	0.144951983763
12	centre	3	10^4	−0.170456145014	0.800901547889
1	centre	0	10^7	−0.000721091206991	0.998464685977
1	wall	15	10^7	−0.00748956875998	0.0303389812102
5	centre	0	10^7	−0.00229096203822	0.996790918537
5	wall	15	10^7	−0.00855398926555	0.0148836399355
12	centre	0	10^7	−0.00538731680888	0.993703412087
12	wall	5	10^7	−0.00784725003139	0.0296167267785

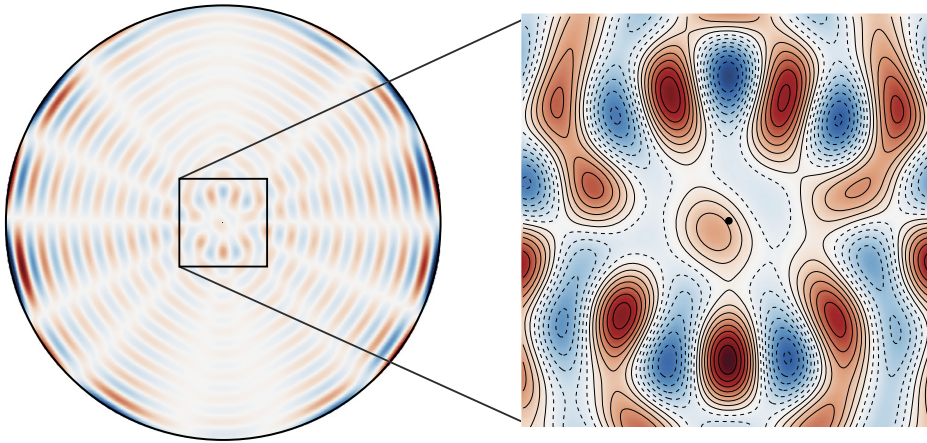


Fig. 7. Contour plot of the calculated solution to the forced Helmholtz equation for $\kappa = 60$, with $s(x, y) = e^{-(x-0.4)^2 - (y-0.3)^2}$ and $g(x, y) = y \cos(10x)$. No artificial singularity appears near the origin (marked with small dot in the zoomed-in image).

$$(\nabla^2 + \kappa^2)f = s \quad \text{and} \quad f(r=1, \theta) = g(\theta) \quad (110)$$

with $s(x, y) = e^{-(x-0.4)^2 - (y-0.3)^2}$ and $g(x, y) = y \cos(10x)$. When κ is very large, the solutions become increasingly oscillatory; see Fig. 7. The zoom-in plot on the right demonstrates that no artificial singularity arises at the origin, despite the complicated nature of the solution. We choose this problem to demonstrate the performance of the methods on a challenging and natural problem requiring simultaneous r, θ dependence.

The bandedness of our representation implies the complexity of the solution is $O(MN)$ operations, where M is the number of θ modes needed to resolve $g(\theta)$ and $s(r, \theta)$ in the θ direction, and N is the optimal number of polynomial coefficients needed to solve to a given tolerance in radius. In Fig. 8 we demonstrate the linear complexity for two different resolved right-hand sides. We approximate, $e^{-(x-0.4)^2 - (y-0.3)^2}$ with 25 θ modes and 11 radial polynomial coefficients per mode, while $\cos(40 \sin(3xy) + 1)$ requires 149 θ modes and 60 polynomial coefficients per mode. We use the adaptive QR method implemented in the ApproxFun.jl Julia package [23,24]. This adaptively determines the optimal number, N , of polynomial coefficients needed to resolve the solution for each θ mode. Equation (110) presents a challenge because the solution can become much more structured than the forcing for $\kappa \gg 1$. At the high range of κ , we solve an equation with over 25 million unknowns in less than 25 seconds.⁴

We also solve a screened Poisson equation similar in form the equation (110) only with $\kappa^2 < 0$. Rather than small-scale oscillations, the solutions tend to form sharp boundary layers near $r = 1$. The performance efficiency and accuracy of the methods treats this case equally as well as the Helmholtz problem.

⁴ Timings on a 2012 MacBook Pro, with an 2.7 GHz Intel Core i7.

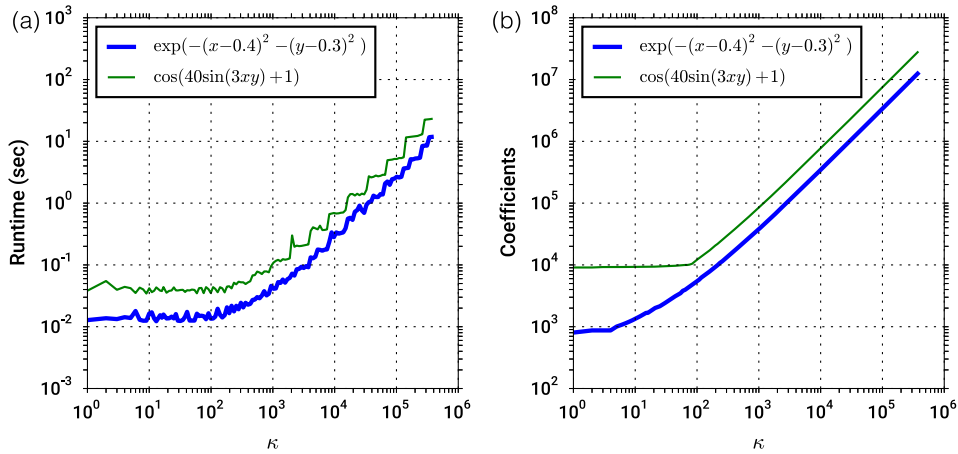


Fig. 8. (a) Time taken to calculate solutions to the Helmholtz equation for two different forcing terms (green versus blue) with $g(x, y) = y \cos(10x)$. (b) Total number of coefficients needed to resolve the solutions. The asymptotic slope becomes linear in κ after an initial fixed costs needed to resolve the right-hand side forcing, as indicated by the background dashed grid. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7. Conclusion

This paper produces a new set of algorithms for numerical computations on the unit disk. Choosing a hierarchy of bases built out of Jacobi polynomials allows for solving a range of important differential equations. We show a range of time-independent examples. Extending to time-evolving system only requires conducting the type of calculations presented here in the iterative fashion.

The methods described here explicitly construct basis elements that respect the coordinate singularity in polar coordinates at $r = 0$. Beyond the disk, the type of calculus presented here will exist for other geometries with coordinate singularities. The two-dimensional disk represents only a prototypical example.

The Laguerre class of polynomials allows representing functions in polar coordinates for the whole two-dimensional plane. In particular,

$$\lim_{R \rightarrow \infty} r^m P_n^{(R^2, m)}(2(r/R)^2 - 1)(1 - (r/R)^2)^{R^2} = (-1)^n r^m L_n^{(m)}(r^2) e^{-r^2}. \quad (111)$$

Many of the formulae present in this paper extend to the infinite plane for $k \rightarrow \infty$ with only minor adjustments for the Gaussian factor. In this limit, the conversion matrix becomes the identity, $C \rightarrow I$.

Two-dimensional spheres and solid balls also display similar behaviour as the unit disk. The sphere effectively contains a local disk at both poles. Spherical harmonic functions naturally resolve the coordinate singularities at the poles for scalar functions. However, a wider class of Jacobi polynomials allows the use of sparse vector calculus for higher-order tensor fields. In addition to the poles, the centre of a solid sphere requires relations completely analogous to equations (2) and (3) only for the spherical harmonic degree [26]. In an upcoming publication, we show how to organise sparse tensor calculus in three-dimensional spheres. This includes algebraic structures similar to those found here for the disk.

The algebraic-analytic link is an important theme this current paper. As a general principle, the singularities of a PDE, either coordinate-based or physical, dictate what types of bases are needed to represent meaningful solutions. When followed carefully, this guiding principle should allow the construction of numerical methods more efficient and straightforward to implement than more traditional schemes. Spherical geometry exhausts the parameter variation found in Jacobi polynomials. However, many more lesser-explored sets of orthogonal polynomials exist, with a large variety of symmetry properties. This includes discrete sets. We encourage the future development of numerical methods based on the guiding algebraic-geometric properties of a given general problem class.

Acknowledgements

GMV acknowledges support from the Australian Research Council, project number DE140101960. KJB is supported by a National Science Foundation Graduate Research Fellowship under Grant No. 1122374. DL is supported by the Hertz Foundation. SO acknowledges support from the Australian Research Council, project number DE130100333

Appendix A. Matrix entries

In this appendix $\delta_{i,j}$ represents the Kronecker delta. We show all matrix coefficients as they act on columns of spectral coefficients from the left.

Differentiation matrices.

$$D^+ : \{k, m\} \rightarrow \{k + 1, m + 1\} \quad (\text{A.1})$$

$$D_{n',n}^+ = \sqrt{2n(n+k+m+1)} \delta_{n',n-1} \quad (\text{A.2})$$

$$D^- : \{k, m\} \rightarrow \{k + 1, m - 1\} \quad (\text{A.3})$$

$$D_{n',n}^- = \sqrt{2(n+k+1)(n+m)} \delta_{n',n} \quad (\text{A.4})$$

Multiplication matrices.

$$R^+ : \{k, m\} \rightarrow \{k, m + 1\} \quad (\text{A.5})$$

$$R_{n',n}^+ = \sqrt{\frac{(n+m+1)(n+k+m+1)}{(2n+k+m+1)(2n+k+m+2)}} \delta_{n',n} \quad (\text{A.6})$$

$$+ \sqrt{\frac{n(n+k)}{(2n+k+m)(2n+k+m+1)}} \delta_{n',n-1} \quad (\text{A.7})$$

$$R^- : \{k, m\} \rightarrow \{k, m - 1\} \quad (\text{A.8})$$

$$R_{n',n}^- = \sqrt{\frac{(n+1)(n+k+1)}{(2n+k+m+1)(2n+k+m+2)}} \delta_{n',n+1} \quad (\text{A.9})$$

$$+ \sqrt{\frac{(n+m)(n+k+m)}{(2n+k+m)(2n+k+m+1)}} \delta_{n',n} \quad (\text{A.10})$$

Conversion matrices.

$$C : \{k, m\} \rightarrow \{k + 1, m\} \quad (\text{A.11})$$

$$C_{n',n} = \sqrt{\frac{(n+k+1)(n+k+m+1)}{(2n+k+m+1)(2n+k+m+2)}} \delta_{n',n} \quad (\text{A.12})$$

$$- \sqrt{\frac{n(n+m)}{(2n+k+m)(2n+k+m+1)}} \delta_{n',n-1} \quad (\text{A.13})$$

$$C^\dagger : \{k, m\} \rightarrow \{k - 1, m\} \quad (\text{A.14})$$

$$C_{n',n}^\dagger = \sqrt{\frac{(n+1)(n+m+1)}{(2n+k+m+1)(2n+k+m+2)}} \delta_{n',n+1} \quad (\text{A.15})$$

$$+ \sqrt{\frac{(n+k)(n+k+m)}{(2n+k+m)(2n+k+m+1)}} \delta_{n',n} \quad (\text{A.16})$$

Appendix B. Dirac's bra-ket notation

Dirac notation is uncommon in numerical analysis applications outside of quantum mechanics. We believe the notation provides many advantages with regard to clarity and abstraction. This short appendix reviews the important aspects of the notation as it applies to the bulk of the paper.

In the appendix, $Q_n(r)$ denotes any one of the Jacobi-based orthogonal polynomials in the main section. We drop the k, m indices that parameterise different bases. We represent functions of r such that

$$f(r) = \sum_{n=0}^{\infty} f_n Q_n(r). \quad (\text{B.1})$$

For a given weight function $w(r)$ [e.g., $r(1-r^2)^k$],

$$\int_0^1 Q_n(r) Q_{n'}(r) w(r) dr = \delta_{n,n'}. \quad (\text{B.2})$$

Therefore a function's spectral-expansion coefficients result from projecting against a given polynomial

$$f_n = \int_0^1 f(r) Q_n(r) w(r) dr. \quad (\text{B.3})$$

We define a (discrete) complete orthonormal basis such that

$$\langle n | n' \rangle = \delta_{n,n'}, \quad \sum_{n=0}^{\infty} |n\rangle \langle n| = \text{I}. \quad (\text{B.4})$$

The element $\langle n |$ denotes an infinite row vector with 0's in the first n entries, a 1 in the $(n+1)$ st entry, and 0's after that. For example,

$$\langle 0 | = [1, 0, 0, 0, 0, 0, \dots], \quad \langle 4 | = [0, 0, 0, 0, 1, 0, \dots]. \quad (\text{B.5})$$

In terms of the discrete basis and orthonormal functions, we define the continuous basis element corresponding to evaluation at the point r

$$\langle r | \equiv \sum_{n=0}^{\infty} Q_n(r) \langle n | \quad (\text{B.6})$$

$$\int_0^1 |r\rangle \langle r| w(r) dr = \sum_{n=0}^{\infty} \sum_{n'=0}^{\infty} |n\rangle \langle n'| \int_0^1 Q_{n'}(r) Q_n(r) w(r) dr \quad (\text{B.7})$$

Therefore,

$$\int_0^1 |r\rangle \langle r| w(r) dr = \text{I}. \quad (\text{B.8})$$

Also,

$$\langle r | r' \rangle \equiv \sum_{n=0}^{\infty} Q_n(r) Q_n(r') = \frac{1}{w(r)} \delta(r-r'), \quad (\text{B.9})$$

which follows from the Christoffel–Darboux formula for orthogonal polynomials.

The orthogonal functions represent the coefficients of the r basis in terms of the n basis,

$$\langle r | n \rangle = Q_n(r). \quad (\text{B.10})$$

Now,

$$f(r) = \langle r | f \rangle = \sum_{n=0}^{\infty} Q_n(r) \langle n | f \rangle \quad (\text{B.11})$$

where

$$f_n = \langle n | f \rangle \quad (\text{B.12})$$

represent the coefficients of the state f in the n basis.

The L^2 inner product of functions takes the form

$$\int_0^1 g^*(r) f(r) w(r) dr = \int_0^1 \langle g | r \rangle \langle r | f \rangle w(r) dr \quad (\text{B.13})$$

therefore

$$\int_0^1 g^*(r) f(r) w(r) dr = \langle g | f \rangle. \quad (\text{B.14})$$

References

- [1] A. Bhatia, E. Wolf, On the circle polynomials of Zernike and related orthogonal sets, in: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 50, Cambridge University Press, 1954, pp. 40–48.
- [2] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, Second revised edition, Dover, 2001.
- [3] J.P. Boyd, F. Yu, Comparing seven spectral methods for interpolation and for solving the Poisson equation in a disk: Zernike polynomials, Logan–Shepp ridge polynomials, Chebyshev–Fourier series, cylindrical Robert functions, Bessel–Fourier expansions, square-to-disk conformal mapping and radial basis functions, *J. Comput. Phys.* 230 (2011) 1408–1438.
- [4] E.A. Coutsias, T. Hagstrom, D. Torres, An efficient spectral method for ordinary differential equations with rational function coefficients, *Math. Comput.* 65 (1996) 611–635.
- [5] E.H. Doha, W.M. Abd-Elhameed, Efficient spectral–Galerkin algorithms for direct solution of second-order equations using ultraspherical polynomials, *SIAM J. Sci. Comput.* 24 (2002) 548–571.
- [6] E.H. Doha, A.H. Bhrawy, Efficient spectral–Galerkin algorithms for direct solution for second-order differential equations using Jacobi polynomials, *Numer. Algorithms* 42 (2006) 137–164.
- [7] E. Doha, W. Abd-Elhameed, Efficient spectral ultraspherical–dual-Petrov–Galerkin algorithms for the direct solution of $(2n + 1)$ th-order linear differential equations, *Math. Comput. Simul.* 79 (2009) 3221–3242.
- [8] C.F. Dunkl, Y. Xu, *Orthogonal Polynomials of Several Variables*, Second edition, Cambridge University Press, 2014.
- [9] P.A.M. Dirac, A new notation for quantum mechanics, *Math. Proc. Camb. Philos. Soc.* 35 (1939) 1–4.
- [10] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, vol. 1, Cambridge University Press, 1998.
- [11] D.R. Gardner, S.A. Trogden, R.W. Douglass, A modified tau spectral method that eliminates spurious eigenvalues, *J. Comput. Phys.* 80 (1989) 137–167.
- [12] L. Greengard, Spectral integration and two-point boundary value problems, *SIAM J. Numer. Anal.* 28 (1991) 1071–1080.
- [13] H.P. Greenspan, *The Theory of Rotating Fluids*, vol. 1, Cambridge University Press, 1968.
- [14] K. Julien, M. Watson, Efficient multi-dimensional solution of PDEs using Chebyshev spectral methods, *J. Comput. Phys.* 228 (2009) 1480–1503.
- [15] T. Koornwinder, Two-variable analogues of the classical orthogonal polynomials, in: *Theory and Applications of Special Functions*, 1975, pp. 435–495.
- [16] K. Li, P.W. Livermore, A. Jackson, An optimal Galerkin scheme to solve the kinematic dynamo eigenvalue problem in a full sphere, *J. Comput. Phys.* 229 (2010) 8666–8683.
- [17] P.W. Livermore, C.A. Jones, S.J. Worland, Spectral radial basis functions for full sphere computations, *J. Comput. Phys.* 227 (2007) 1209–1224.
- [18] T. Matsushima, P. Marcus, A spectral method for polar coordinates, *J. Comput. Phys.* 120 (1995) 365–374.
- [19] G.B. McFadden, B.T. Murray, R.F. Boisvert, Elimination of spurious eigenvalues in the Chebyshev tau spectral method, *J. Comput. Phys.* 91 (1990) 228–239.
- [20] A. Meseguer, L.N. Trefethen, Linearized pipe flow to Reynolds number 10^7 , *J. Comput. Phys.* 186 (2003) 178–197.
- [21] B.K. Muite, A numerical comparison of Chebyshev methods for solving fourth order semilinear initial boundary value problems, *J. Comput. Appl. Math.* 234 (2010) 317–342.
- [22] F.W.J. Olver, D.W. Lozier, R.F. Boisvert, C.W. Clark, *NIST Handbook of Mathematical Functions*, Cambridge University Press, 2010.
- [23] S. Olver, A. Townsend, A fast and well-conditioned spectral method, *SIAM Rev.* 55 (2013) 462–489.
- [24] S. Olver, A. Townsend, A practical framework for infinite-dimensional linear algebra, in: *HPTCDL*, 2014, pp. 57–62.
- [25] S.A. Orszag, Accurate solution of the Orr–Sommerfeld stability equation, *J. Fluid Mech.* 50 (1971) 689–703.
- [26] S.A. Orszag, Fourier series on spheres, *Mon. Weather Rev.* 102 (1974) 56–75.
- [27] C.C.T. Pringle, R.R. Kerswell, Using nonlinear transient growth to construct the minimal seed for shear flow turbulence, *Phys. Rev. Lett.* 105 (2010) 154502.
- [28] O. Reynolds, An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels, *Philos. Trans. R. Soc. Lond.* 174 (1883) 935–982.
- [29] S.P. Sutera, R. Skalak, The history of Poiseuille's law, *Annu. Rev. Fluid Mech.* 25 (1993) 1–19.
- [30] R.M. Slevinsky, On the use of Hahn's asymptotic formula and stabilized recurrence for a fast, simple, and stable Chebyshev–Jacobi transform, *arXiv:1602.02618*, 2016.
- [31] T. Sakai, L. Redekopp, An application of one-sided Jacobi polynomials for spectral modelling of vector fields in polar coordinates, *J. Comput. Phys.* 228 (2009) 7069–7085.
- [32] A. Townsend, S. Olver, The automatic solution of partial differential equations using a global spectral method, *J. Comput. Phys.* 299 (2015) 106–123.
- [33] A. Townsend, H. Wilber, G. Wright, Computing with functions in spherical and polar geometries II. The disk, <http://arxiv.org/abs/1604.03061>, 2016.
- [34] L.N. Trefethen, *Spectral Methods in MATLAB*, vol. 10, SIAM, 2000.
- [35] D. Viswanath, Spectral integration of linear boundary value problems, *J. Comput. Appl. Math.* 290 (2015) 159–173.
- [36] F. Zernike, Beugungstheorie des Schneidverfahrens und seiner verbesserten Form, der Phasenkontrast-methode, *Physica* 1 (1934) 689–704.
- [37] A. Zebib, A Chebyshev method for the solution of boundary value problems, *J. Comput. Phys.* 53 (1984) 443–455.