

Accurate and efficient Jones-Worland spectral transforms for planetary applications

Conference Paper**Author(s):**

Marti, Philippe David ; Jackson, Andrew

Publication date:

2021-07

Permanent link:

<https://doi.org/https://doi.org/10.3929/ethz-b-000505302>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

<https://doi.org/10.1145/3468267.3470620>

Funding acknowledgement:

- Unravelling Earth's magnetic history and processes UEMHP ()
- Understanding planetary magnetic fields from theoretical, numerical and analogue models ()

Accurate and efficient Jones-Worland spectral transforms for planetary applications

Philippe Marti

Andrew Jackson

philippe.marti@erdw.ethz.ch

ajackson@ethz.ch

ETH Zurich

Zurich, Switzerland

ABSTRACT

Spectral transforms between physical space and spectral space are needed for fluid dynamical calculations in the whole sphere, representative of a planetary core. In order to construct a representation that is everywhere smooth, regular and differentiable, special polynomials called Jones-Worland polynomials, based on a type of Jacobi polynomial, are used for the radial expansion, coupled to spherical harmonics in angular variables. We present an exact, efficient transform that is partly based on the FFT and which remains accurate in finite precision. Application is to high-resolution solutions of the Navier-Stokes equation, possibly coupled to the heat transfer and induction equations. Expected implementations would be in simulations with P^3 degrees of freedom, where P may be greater than 10^3 . Memory use remains modest at high spatial resolution, indeed typically P times lower than competing algorithms based on quadrature.

CCS CONCEPTS

• **Applied computing** → **Earth and atmospheric sciences**; • **Mathematics of computing** → *Mathematical software performance*; • **Computing methodologies** → *Massively parallel and high-performance simulations*.

KEYWORDS

spectral method, computational fluid dynamics, HPC

ACM Reference Format:

Philippe Marti and Andrew Jackson. . Accurate and efficient Jones-Worland spectral transforms for planetary applications. In . ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

A goal of geophysics and planetary science is to understand the evolution and dynamics of planetary cores, in particular its influence on heat transfer through convection, their response to mechanical forcing through precession and tides, and their ability to generate magnetic fields through the dynamo action that can be supported

by fluid motions therein. The fluid dynamics of planetary interiors takes place in such extreme conditions that the field has been greatly influenced by numerical computation. A requirement to be able to reach the regime in which fluid motions are complex in both space and time is the clever implementation of time-marching routines together with efficient spatial discretisation of the fluid domain, with the flexibility to reach a high number of degrees of freedom (DOF); typical implementations presently use up to $O(10^8)$ DOF, with the prospect of calculations using $O(10^{10})$ DOF on the near horizon. In the canonical setting of the whole sphere, applicable to many planets and probably 80-90% of Earth history, spectral methods are unsurpassed in accuracy and efficiency, subject to a proper implementation. This is the subject of the present paper.

Spectral methods have a long tradition in geophysical applications. While complex geometries are the domain of local methods (finite difference/element/volume), the undisputed king of simple geometries such as spheres, cubes and spherical shells is the fully spectral method. This is because the high accuracy they provide is unsurpassed compared to local methods. Even the issue of parallelisation of fully spectral methods has been overcome, such that their prominence in the field is assured. While most studies are concerned with spherical shells [3, 5, 20], our focus will entirely be on the full sphere geometry. This geometry is applicable to gas giant planets or the Early Earth [21]. It is also an important tool in the study of the fundamental processes involved in magnetic field generation [2, 8] as it preserves the general spherical geometry but removes secondary geometrical effects due to the presence of an inner core.

Exploiting the smooth and simple geometry, this paper will focus on methods for the spectral expansion of functions in a spherical geometry with polar coordinates (r, θ, ϕ) , with $0 \leq r \leq 1$. As is well known, the spherical harmonics $Y_l^m(\theta, \varphi)$ are designed to gracefully handle behaviour in (θ, φ) , and the remaining question is the description of the radial behaviour for 3D fully spectral simulations. As presented in [12], if the angular components of a scalar are expanded using spherical harmonics of degree l and order m , the remaining radial part must behave as

$$f_l(r) \propto r^l \left(a_0 + a_1 r^2 + a_2 r^4 + \dots + a_n r^{2n} + \dots \right), \quad (1)$$

where $r \in [0, 1]$; the expansion arises from the requirement for infinite differentiability of the function at all points, including the origin, which is a singularity of the coordinate system. Of course, actual computations do not use monomials. Indeed, it is better to use expansions that exhibit orthogonality under a certain inner-product, and while several alternatives have been studied [22, 23],

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org/permissions).

the Jones-Worland polynomials [11, 24] have been shown to provide an accurate expansion for functions behaving as (1). The orthogonality is used constantly in numerical computations to move between the spectral domain and the physical domain, using grid points and quadrature rules. Such transforms are critical to the efficient treatment of non-linear terms which are treated explicitly in the time-marching of the Navier-Stokes equation. While alternative expansions are possible, as for example based on a double Fourier expansion [1] in place of the spherical harmonics or the use of finite differences in the radial discretization, these approaches provide computational advantages but cannot satisfy the regularity condition (1).

In this paper, we present an entirely new algorithm for the Jones-Worland spectral transform (spectral-physical and vice versa) based on the Fast Fourier Transform (FFT) that efficiently solves the problem of transforms. It does not require any explicit evaluation of the polynomials and is an exact algorithm in infinite precision. Since the algorithm is understood to have good conditioning, even in finite precision, the algorithm is accurate to machine precision as expected, with an operation count and memory footprint that are very favourable.

The methods lend themselves to future implementation on multi-core GPU enabled compute nodes, such as the present-day provision of *Piz Daint* at the Swiss National Supercomputer Centre. They provide an advantageous replacement for the quadrature based transforms, suitable for high resolution 3D simulations in the unit ball.

1.1 Jones-Worland polynomials and Gauss-Chebyshev quadrature

The Jones-Worland polynomials have been developed in order to provide a stable and accurate spectral basis for the behaviour in Equation (1) [11]. We recall their main features in this section. Considering a scalar field $f(r, \theta, \varphi)$ in spherical coordinates, its expansion in the unit ball on the spherical harmonics can be written

$$f(r, \theta, \varphi) = \sum_{l,m} f_l^m(r) Y_l^m(\theta, \varphi), \quad (2)$$

where $Y_l^m(\theta, \varphi)$ are the spherical harmonic of degree l and order m . The radial function f_l^m has then the behaviour given in Equation (1). The Jones-Worland polynomials are defined as a product of r^l with a Jacobi Polynomial $P_n^{(\alpha, \beta)}(x)$. The flexibility in the choice of (α, β) is guided by a WKBJ analysis [11] that shows that outside a boundary layer close to $r = 0$, the envelope of oscillations of the function can be made to be constant when $\alpha = -1/2$ and $\beta = l - 1/2$; the polynomials are then almost equi-ripple, and have a deep connection to the ordinary Chebyshev polynomials. We express the Jones-Worland polynomials as follows

$$W_n^l(r) = r^l P_n^{(-\frac{1}{2}, l-\frac{1}{2})}(2r^2 - 1) \quad (3)$$

While other choices of α and β have been used successfully (e.g. [9, 22]), this work focuses on $\alpha = -\frac{1}{2}$ and $\beta = l - \frac{1}{2}$ as it will provide a connection to the Fast Fourier Transform (FFT). Jones-Worland polynomials with the same l are orthogonal with respect

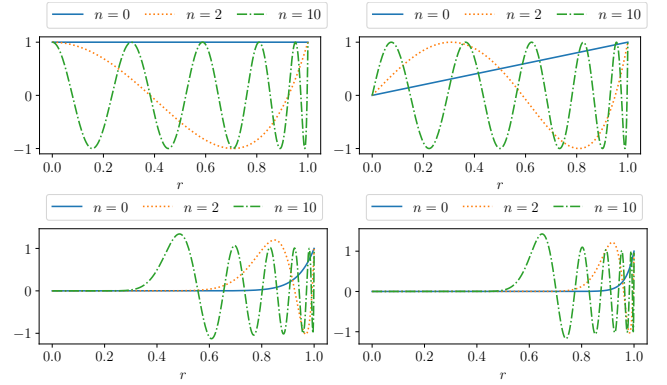


Figure 1: Jones-Worland polynomials for increasing $n = 0, 2, 10$ for (upper left) $l = 0$, (upper right) $l = 1$, (lower left) $l = 16$ and (lower right) $l = 32$.

to a Chebyshev weight

$$\int_0^1 \frac{1}{\sqrt{1-r^2}} W_i^l(r) W_j^l(r) dr = h_i \delta_{ij} \quad (4)$$

$$h_i = \frac{1}{2(2i + \alpha + \beta + 1)} \frac{\Gamma(i + \alpha + 1) \Gamma(i + \beta + 1)}{\Gamma(i + \alpha + \beta + 1) \Gamma(i + 1)}$$

The radial behaviour of $W_n^l(r)$ is illustrated on a few examples in Figure 1. As the harmonic degree l increases the range over which the polynomials are essentially zero extends outwards from the origin. Higher polynomial order n for fixed l is then required to penetrate deeper into the sphere. Finally, the spectral expansion of the scalar field $f(r, \theta, \varphi)$ is obtained as

$$f(r, \theta, \varphi) = \sum_{n=0}^N \sum_{l=0}^L \sum_{m=0}^l f_{n,l}^m W_n^l(r) Y_l^m(\theta, \varphi). \quad (5)$$

with truncation limits $l = L$ and $n = N$.

In order to simplify the notation, the spherical harmonic indices will be dropped as only the radial behaviour is studied in this paper. The focus is then on computing for a fixed harmonic degree l the expansion

$$f_l^m(r) \equiv f(r) = \sum_n f_n W_n^l(r) \quad (6)$$

If one is interested in quadrature rules with the least number of quadrature points, the computation of the expansion coefficients f_n can be performed using the Gauss-Jacobi quadrature rule. But given the dependence of the polynomials on spherical harmonic degree l , this requires a different quadrature grid for each harmonic degree l [12], and is not amenable to 3D simulations where a single grid is needed, on which all nonlinear terms are computed in physical space. In light of (4), a Gauss-Chebyshev quadrature rule (GCQ) is used instead, requiring a larger number of nodes but which has simple grid points and weights. This approach has been used successfully in several studies including [9, 10, 13]. The GCQ approach allows to write the projection integral as a sum,

$$f_n = \int_0^1 \frac{1}{\sqrt{1-r^2}} f(r) W_n^l(r) dr = \sum_i \omega_i f(r_i) W_n^l(r_i), \quad (7)$$

where r_i and ω_i are the Gauss-Chebyshev grid points and weights respectively. The evaluation of the Jones-Worland polynomials on the grid is obtained by using the recurrence relation for the Jacobi polynomials, as shown in Equation (8) but using r^l as a starting value.

$$\begin{aligned} a_n P_n^{(\alpha,\beta)}(x) &= (b_n x + c_n) P_{n-1}^{(\alpha,\beta)}(x) + d_n P_{n-2}^{(\alpha,\beta)}(x) \\ a_n &= 2n(n + \alpha + \beta)(2n + \alpha + \beta - 2) \\ b_n &= (2n + \alpha + \beta - 1)(2n + \alpha + \beta)(2n + \alpha + \beta - 2) \\ c_n &= (2n + \alpha + \beta - 1)(\alpha^2 - \beta^2) \\ d_n &= -2(n + \alpha - 1)(n + \beta - 1)(2n + \alpha + \beta) \end{aligned} \quad (8)$$

For best performance, the values of the $W_n^l(r)$ on the quadrature grid are computed and stored in a matrix \mathcal{W} . The spectral transforms are then computed as a matrix product. The forward transform is obtained as

$$\vec{f}(r) = \mathcal{W} \hat{f}, \quad (9)$$

and the backward transform as

$$\hat{f} = \mathcal{W}^T \omega \vec{f}(r), \quad (10)$$

where $\vec{f}(r)$ are the values of $f(r)$ on the grid, \hat{f} the vector of spectral coefficients, ω a diagonal matrix of the Chebyshev weights and column j of \mathcal{W} contains $W_j(r_i)$. While this is a good approach at lower resolutions, it does have important shortcomings at high resolution discussed in the next section.

1.2 Shortcomings of GCQ

It is time to address the elephant in the room: how does one stably compute polynomials (3) for large l , essential for the use of GCQ? It is precisely the competition of two ingredients that leads to the beautifully behaved product depicted in Figure 1. A WKBJ analysis [11] shows that the amplitude of the oscillations in $P_n^{(-1/2, l-1/2)}(2r^2 - 1) \rightarrow r^{-l}$ while this effect is, of course, offset by the r^l behaviour of the prefactor. At small values of r this can lead to underflow in r^l and overflow in $P_n^{(-1/2, l-1/2)}(2r^2 - 1)$.

As an example, when $\alpha = -1/2$ we use Gauss-Chebyshev quadrature to calculate required integrals. The first quadrature point is $\cos(\pi/(4N_r))$ for a general radial expansion with $N_r = 3N/2 + 3L/4 + 1$ radial quadrature points. The first point is at $\sim (2N + L)^{-1}$ and so will be at $r_1 \sim 10^{-2}$ if $L = 50$ and $r_1 \sim 10^{-3}$ if $L = 512$. In double precision r^l results in an underflow if the value is $< 10^{-300}$, which occurs for the latter value which gives $r^l = \mathcal{O}(10^{-1500})$, well below values representable by standard 64 bit floating point values. One can easily discover the breaking point: it is $L = 125$ for a quadrature point at $r = 0.004^1$. The device of starting the recurrence at r^l discussed in (8) does nothing to ameliorate the problem. The situation becomes worse as more complicated operators are considered which involve derivatives as is generally the case for full-scale 3D computational fluid dynamics (CFD) simulations.

A second shortcoming is that the storage requirements for the GCQ transform matrices \mathcal{W} becomes problematic at high resolutions. Indeed for a full spectral transform, there are $L + 1$ matrices with $\mathcal{O}(N^2)$ elements. In full-scale 3D CFD simulations, not only projection operators are required but also derivatives. Ten different

¹The use of extended precision, advocated by [7], fails similarly at $L = 1500$

operators are required and need to be stored. For a planetary dynamo simulation with $N = 1000$, $\mathcal{O}(100\text{GB})$ are required to store the matrices. While the parallelisation allows us to distribute the memory across nodes, load balancing considerations [12] still lead to a large memory footprint for the GCQ method. When the implementation on accelerators (e.g. GPU) is considered, the memory footprint become even more important as they generally offer less memory than CPU-based machines.

2 JONES-WORLAND TRANSFORM ALGORITHM

There has been extensive research on fast transforms for orthogonal polynomials and specifically Jacobi polynomials (e.g. [6, 16, 17, 19]). The presence of the regularizing r^l factor in the Jones-Worland polynomial leads to its own challenges and requires a specialized solution. Furthermore, fast Jacobi transforms relying on asymptotic expressions like the ones in [19] are not applicable to the large values of $\beta = \mathcal{O}(10^3)$ which are required here. This transform algorithm is based on the connection relations that exist between different families of Jacobi polynomials. We will follow a similar approach to [14, 15, 18] and adapt it to the Jones-Worland polynomials. The algorithm heavily relies on two recurrence relations for the Jacobi polynomials. The first is

$$\tilde{P}_n^{(\alpha,\beta)}(x) = \gamma_n^{(\alpha,\beta)} \tilde{P}_n^{(\alpha,\beta+1)}(x) + \zeta_n^{(\alpha,\beta)} \tilde{P}_{n-1}^{(\alpha,\beta+1)}(x), \quad (11)$$

where $\tilde{P}_n^{(\alpha,\beta)}(x)$ are the normalized Jacobi polynomials and the coefficients γ_n and ζ_n are given by

$$\gamma_n^{(\alpha,\beta)} = \sqrt{\frac{2(n + \beta + 1)(n + \alpha + \beta + 1)}{(2n + \alpha + \beta + 1)(2n + \alpha + \beta + 2)}} \quad (12)$$

$$\zeta_n^{(\alpha,\beta)} = \sqrt{\frac{2n(n + \alpha)}{(2n + \alpha + \beta)(2n + \alpha + \beta + 1)}}. \quad (13)$$

This relation allows to promote the β parameter to $\beta + 1$. The second relation that is needed is

$$(1 + x) \tilde{P}_n^{(\alpha,\beta)}(x) = \mu_n^{(\alpha,\beta)} \tilde{P}_n^{(\alpha,\beta-1)}(x) + \nu_n^{(\alpha,\beta)} \tilde{P}_{n+1}^{(\alpha,\beta-1)}(x) \quad (14)$$

where the coefficients μ_n and ν_n are given by

$$\mu_n^{(\alpha,\beta)} = \sqrt{\frac{2(n + \beta)(n + \alpha + \beta)}{(2n + \alpha + \beta)(2n + \alpha + \beta + 1)}} \quad (15)$$

$$\nu_n^{(\alpha,\beta)} = \sqrt{\frac{2(n + 1)(n + \alpha + 1)}{(2n + \alpha + \beta + 1)(2n + \alpha + \beta + 2)}}. \quad (16)$$

While also promoting β , the second relation will be used to introduce the r^l factor of the Jones-Worland polynomials, since $r = \sqrt{(x + 1)/2}$.

In order to simplify the notation, from now on we will assume

$$\alpha = -\frac{1}{2}, \quad (17)$$

$$\beta = l - \frac{1}{2}, \quad (18)$$

which is the relevant (α, β) pair for the Jones-Worland polynomials (Eq. (3)). To formulate the algorithm, we cast recurrence relation (11) as an upper bidiagonal matrix \mathbf{V}_l where the diagonal elements are given by $\gamma_n^{(\alpha,\beta-1)}$ and the superdiagonal elements by $\zeta_{n+1}^{(\alpha,\beta-1)}$ for

row n . Matrix \mathbf{V}_l can then be used to move from an expansion of the form $\sum_{i=0} c_i^{(l-1)} \tilde{P}_i^{(-\frac{1}{2}, l-1-\frac{1}{2})}$ to an expansion $\sum_{i=0} c_i^{(l)} \tilde{P}_i^{(-\frac{1}{2}, l-\frac{1}{2})}$ with

$$\begin{pmatrix} \vdots \\ c_i^{(l)} \\ \vdots \end{pmatrix} = \mathbf{V}_l \begin{pmatrix} \vdots \\ c_i^{(l-1)} \\ \vdots \end{pmatrix}. \quad (19)$$

The recurrence relation (14) is written as a lower bidiagonal matrix \mathbf{M}_l such that it allows to move from an expansion $\sum_{i=0} c_i^{(l-1)} \tilde{P}_i^{(-\frac{1}{2}, l-1-\frac{1}{2})}$ to an expansion $\sum_{i=0} d_i^{(l)} (1+x) \tilde{P}_i^{(-\frac{1}{2}, l-\frac{1}{2})}$

$$\begin{pmatrix} \vdots \\ d_i^{(l)} \\ \vdots \end{pmatrix} = \mathbf{M}_l^{-1} \begin{pmatrix} \vdots \\ c_i^{(l-1)} \\ \vdots \end{pmatrix}. \quad (20)$$

In this form and, upon closer inspection of the two recurrence relations, the two matrices \mathbf{V}_l and \mathbf{M}_l are related by

$$\mathbf{M}_l = \mathbf{V}_l^T. \quad (21)$$

Consider a smooth function f_l with the behaviour given in Equation (1) evaluated on the N_g Gauss-Chebyshev quadrature nodes

$$x_i = \cos\left(\frac{2i+1}{2N_g}\pi\right), \quad (22)$$

$$r_i = \sqrt{\frac{x_i+1}{2}}. \quad (23)$$

Initially, we assume l to be an even integer. Odd l will be treated subsequently as it requires a different first stage in the algorithm. As l is even and therefore the function is an even polynomial in r , when evaluated for $x \in [-1, 1]$ the function f_l remains a polynomial with the order reduced by a factor of 2. The Chebyshev expansion of f can then be computed using the Discrete Cosine Transform (DCT) type II formally defined as

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right), \quad k = 0, \dots, N-1. \quad (24)$$

Up to a normalization, this allows us to compute the expansion of f_l on $\tilde{P}_n^{(-\frac{1}{2}, -\frac{1}{2})}(x)$. Truncating the series at N_T gives

$$f(x) \approx \sum_{i=0}^{N_T} c_i^{(0)} \tilde{P}_i^{(-\frac{1}{2}, -\frac{1}{2})}(x). \quad (25)$$

For $l = 0$ this is the only step in the algorithm. The goal is now to use the recurrence relations from Equations (11) and (14) to convert the expansion in Equation (25) into the coefficients for higher harmonic degrees. Repeatedly applying \mathbf{V}_l in order to obtain the expansion for $l/2$

$$\begin{pmatrix} \vdots \\ c_i^{(l/2)} \\ \vdots \end{pmatrix} = \mathbf{V}_{\frac{l}{2}} \cdots \mathbf{V}_2 \mathbf{V}_1 \begin{pmatrix} \vdots \\ c_i^{(0)} \\ \vdots \end{pmatrix}, \quad (26)$$

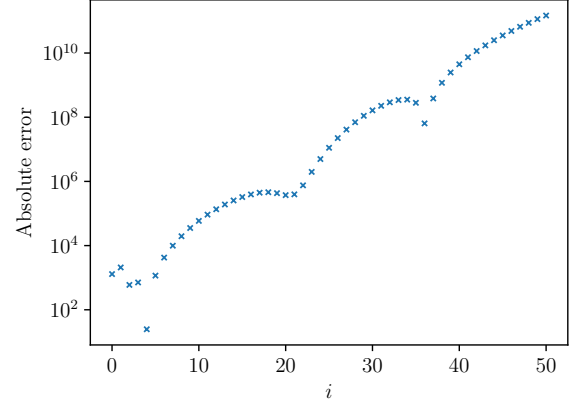


Figure 2: Absolute error on the spectral coefficients f_i in backward-forward transform loop with full unit spectrum with $L = l = 100$ and $N = 50$ by simply applying the recurrence relations repeatedly. Computations are performed using double precision.

followed by repeated backsolving against \mathbf{V}_l^T

$$\mathbf{V}_{\frac{l}{2}+1}^T \cdots \mathbf{V}_{l-1}^T \mathbf{V}_l^T \begin{pmatrix} \vdots \\ c_i^{(l)} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ c_i^{(l/2)} \\ \vdots \end{pmatrix}, \quad (27)$$

produces an unstable algorithm for which accuracy is lost in finite precision arithmetic even for low resolution. After a backward-forward transform loop of a full unit spectrum the solution is completely lost as is shown in Figure 2 for $l = 100$ and $N = 50$. All of our computations are performed in double precision. \mathbf{V}_i is the upper bidiagonal matrix obtained by truncating the recurrence relation at N_T . Therefore the last coefficient obtained by the matrix product with \mathbf{V}_i is not exact as the recurrence relation involves the coefficient $N_T + 1$. Through repeated application of \mathbf{V}_i and \mathbf{V}_j^T the whole spectrum is corrupted. The application of \mathbf{V}_j^T does not present any problems as it is a lower bidiagonal matrix. The final Jones-Worland expansion is truncated at N which implies that at least $N_T = N + \frac{l}{2}$ Chebyshev modes are required for an exact transform. In order to take into account additional normalization that depend on the implementation of the DCT-II and the choice of normalization for the Jones-Worland polynomials, a diagonal normalization matrix \mathcal{D} is introduced in the algorithms.

There are no restrictions on the order in which the \mathbf{V}_i and \mathbf{M}_j matrices should be applied. A well conditioned algorithm is obtained by building pairs of \mathbf{V}_i and \mathbf{M}_{i+1}^{-1} leading to the following expression

$$\hat{\mathbf{f}} = \left(\mathbf{M}_l^{-1} \mathbf{V}_{l-1}\right) \cdots \left(\mathbf{M}_4^{-1} \mathbf{V}_3\right) \left(\mathbf{M}_2^{-1} \mathbf{V}_1\right) \mathbf{c}^{(0)}. \quad (28)$$

The banded lower/upper bidiagonal matrices \mathbf{M}_{i+1} and \mathbf{V}_i have size

$$(n_s - \lfloor i/2 \rfloor) \times (n_s - \lfloor i/2 \rfloor).$$

The pseudocode for the resulting algorithm is then given by Algorithm 1. Intuitively, it is clear that the order (28) provides a better

algorithm as each pair $C_i = (M_{i+1}^{-1}V_i)$ (with condition number $O(1)$) translates the coefficients between two well behaved expansions with the same weight function.

Algorithm 1: Forward Jones-Worland transform of function f for even l

Result: Jones-Worland expansion w
 # Compute Jacobi-Chebyshev expansion through DCT-II
 $t = \text{DCT}_{\text{II}}(f)$
 $N_{\frac{l}{2}} = N + l/2$
 # Extract $N + \frac{l}{2}$ first Chebyshev modes
 $p = t[:N_{\frac{l}{2}}]$
 # Increase β in steps of 2
for i **in** $\text{range}(1, l, 2)$ **do**
 $v = V_i * p$
 $p = \text{solve}(M_{i+1}, v)$
 # Drop last mode
 $p = p[: -1]$
end
 # Normalize and extract N first modes
 $w = \mathcal{D}^{-1} * p[:N]$

The backward transform corresponding to Algorithm 1 is obtained by reversing the operations. Consider the following algorithm with t being the Jacobi-Chebyshev expansion with at least $N_{\frac{l}{2}} = N + l/2$ modes and w the Jones-Worland expansion of size N . The banded lower bidiagonal matrices M_{i+1} have size

$$\left(N_{\frac{l}{2}} - \lfloor i/2 \rfloor\right) \times \left(N_{\frac{l}{2}} - \lfloor i/2 \rfloor - 1\right)$$

and the upper bidiagonal matrices V_i have size

$$\left(N_{\frac{l}{2}} - \lfloor i/2 \rfloor\right) \times \left(N_{\frac{l}{2}} - \lfloor i/2 \rfloor\right)$$

Algorithms 1 and 2 are only applicable to even l . A modified approach is required to handle odd harmonic degree l . The problem arises as the change of variable to $x \in [-1, 1]$ does not produce a polynomial in x , making it impossible to write an algorithm that is exact for a polynomial function f . Therefore we remain in r space but extend the definition for $r \in [0, 1]$ to $\hat{r} \in [-1, 1]$. In this way

$$W_n^l(\hat{r}) \equiv \hat{r}^l P_n^{(-1/2, l-1/2)}(2\hat{r}^2 - 1), \quad (29)$$

is a polynomial for odd l on the interval $[-1, 1]$. Furthermore each $W_n^l(\hat{r})$ has a parity controlled by l . The function f on $r \in [0, 1]$ is extended as function F on $\hat{r} \in [-1, 1]$ as follows

$$F(\hat{r}) \equiv \begin{cases} (-1)^l f(|\hat{r}|), & \hat{r} \in [-1, 0) \\ f(\hat{r}), & \hat{r} \in [0, 1] \end{cases}, \quad (30)$$

which has an odd symmetry for odd l . The Chebyshev expansion of F , obtained through a standard DCT-II, will therefore only contain odd polynomials thanks to the parity

$$F(\hat{r}) = \sum_{i=0} c_{2i+1}^{(0)} \hat{P}_{2i+1}^{(-\frac{1}{2}, -\frac{1}{2})}(\hat{r}). \quad (31)$$

Algorithm 2: Backward Jones-Worland transform of w for even l

Result: Evaluation of function f on Chebyshev grid
 # Pad Jones-Worland with zeros and normalize
 $N_{\frac{l}{2}} = N + l/2$
 $p = \text{zeros}(N_{\frac{l}{2}})$
 $p[:N] = \mathcal{D} * w$
 # Lower β in steps of 2
for i **in** $\text{range}(l-1, 0, -2)$ **do**
 $N_p = N_{\frac{l}{2}} - i/2$
 # Multiplication by M_i adds a mode
 $v[:N_p] = M_{i+1} * p[:N_p - 1]$
 $p[:N_p] = \text{solve}(V_i, v[:N_p])$
end
 # Pad Jacobi-Chebyshev for DCT-II
 $t = \text{zeros}(N_g)$
 $t[:N_{\frac{l}{2}}] = p$
 # Evaluate Jacobi-Chebyshev expansion on grid
 $f = \text{DCT}_{\text{II}}(t)$

While this produces accurate results the grid size needs to be twice as large as for even l . A better approach appears upon closer inspection of the orthogonality relation for the Jones-Worland polynomial with $l = 1$ and upon a change in variables back to $x = 2r^2 - 1$ to obtain Equation (32).

$$\int_0^1 \frac{1}{\sqrt{1-r^2}} W_n^1(r) W_m^2(r) dr = \frac{1}{4} \int_{-1}^1 (1-x)^{-\frac{1}{2}} (1+x)^{\frac{1}{2}} P_n^{(-\frac{1}{2}, \frac{1}{2})}(x) P_m^{(-\frac{1}{2}, \frac{1}{2})}(x) dx \quad (32)$$

$$= \frac{1}{4} \int_{-1}^1 (1-x)^{\frac{1}{2}} (1+x)^{-\frac{1}{2}} P_n^{(\frac{1}{2}, -\frac{1}{2})}(x) P_m^{(\frac{1}{2}, -\frac{1}{2})}(x) dx \quad (33)$$

A further change of variable from $x = -x$, allows us to exchange the α and β parameters and leads to Equation (33). Up to a normalization factor, this last expression is the orthogonality relation for the Chebyshev polynomial of the fourth kind [4, Table 18.3.1]. The DCT-IV, formally defined below,

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) \left(k + \frac{1}{2}\right)\right), \quad k = 0, \dots, N-1, \quad (34)$$

allows for a fast computation of Equation (33). Therefore replacing the DCT-II with a DCT-IV renders the algorithm for odd l as efficient as the even case. Indeed, applying $\lfloor \frac{l}{2} \rfloor$ times the promotion operators, produces the required Jones-Worland expansion. Algorithm 3 shows the full process for odd l . The banded bidiagonal matrices M_i and V_i require a slightly larger truncation of $N_{\lfloor \frac{l}{2} \rfloor} = N + \lceil \frac{l}{2} \rceil = N + \lfloor \frac{l}{2} \rfloor + 1$ and thus have size

$$\left(N_{\lfloor \frac{l}{2} \rfloor} - \lfloor (i-1)/2 \rfloor\right) \times \left(N_{\lfloor \frac{l}{2} \rfloor} - \lfloor (i-1)/2 \rfloor\right)$$

The backward transform is obtained by reversing the operations. The banded lower bidiagonal M_{i+1} have size

$$\left(N_{\lfloor \frac{l}{2} \rfloor} - \lfloor (i-1)/2 \rfloor\right) \times \left(N_{\lfloor \frac{l}{2} \rfloor} - \lfloor (i-1)/2 \rfloor - 1\right)$$

Algorithm 3: Forward Jones-Worland transform of function f for odd l

Result: Jones-Worland expansion w

```

# Compute Jacobi-Chebyshev expansion through DCT-IV
t = DCTIV(f)
N[ $\frac{l}{2}$ ] = N + l/2 + 1
# Extract  $rP_n^{(-1/2,1/2)}$  expansion
p = t[: N[ $\frac{l}{2}$ ]]
# Increase  $\beta$  in steps of 2
for i in range(2, l, 2) do
    v = Vi * p
    p = solve(Mi+1, v)
    # Drop last mode
    p = p[: -1]
end
# Normalize and extract n first modes
w =  $\mathcal{D}^{-1}$  * p[: N]

```

and the upper bidiagonal V_i have size

$$\left(N_{\lceil \frac{l}{2} \rceil} - \lfloor (i-1)/2 \rfloor\right) \times \left(N_{\lceil \frac{l}{2} \rceil} - \lfloor (i-1)/2 \rfloor\right)$$

Algorithm 4: Backward Jones-Worland transform of w for odd l

Result: Evaluation of function f on Chebyshev grid

```

# Pad Jones-Worland with zeros and normalize
N[ $\frac{l}{2}$ ] = n + l/2 + 1
p = zeros(N + l/2)
p[: N] =  $\mathcal{D}$  * w
# Lower  $\beta$  in steps of 2
for i in range(l-1, 1, -2) do
    Np = N[ $\frac{l}{2}$ ] - (i-1)/2
    # Multiplication by Mi adds a mode
    v[: Np] = Mi+1 * p[: Np - 1]
    p[: Np] = solve(Vi, v[: Np])
end
# Pad Jacobi-Chebyshev of fourth kind for DCT-IV
t = zeros(Ng)
t[: N[ $\frac{l}{2}$ ]] = p
# Evaluate Jacobi-Chebyshev expansion f on grid
f = DCTIV(t)

```

3 COMPUTATIONAL COST

In order to assess the performance of the algorithm presented above, the computational cost is compared to the one from a standard Gauss-Chebyshev quadrature. We consider a homogeneous radial truncation of N and a maximal harmonic degree of L . The harmonic degrees of interest are $0 \leq l \leq L$. In order to reduce the parameter space, a standard truncation of $N = \frac{L}{2}$ will be used. In order to compute the quadratic nonlinear interactions required in

3D simulations, the radial grid needs to be of size

$$N_r = \frac{3}{2}(N + L/2 + 1) = \frac{3}{2}(2N + 1) \approx 3N + 2.$$

Furthermore, the size of the connection matrices V_l and M_l depend on l and the following truncations are introduced

$$\begin{aligned} N_l &= N + l \\ N_{l/2} &= N + l/2. \end{aligned}$$

When GCQ is used, a single matrix product with a matrix of size $N_r \times N$ is required and has thus the complexity q_{gcq} is

$$q_{gcq} = \mathcal{O}(N_r N) = \mathcal{O}(3N^2) \quad (35)$$

complexity. Forward and backward transforms have the same complexity if the quadrature matrix is computed and stored in an initialization step as the quadrature weights can be included. Note that q_{gcq} does not depend on l . Our new algorithm will be compared against this reference implementation.

The new algorithm can be split into two stages:

- (1) Chebyshev transform via a Discrete Cosine Transform (DCT-II or DCT-IV)
- (2) Conversion to the Jones-Worland basis by performing pairs comprised of a bidiagonal matrix vector product and a bidiagonal linear solve.

The complexity of the DCT in the first stage is

$$q_{dct} = \mathcal{O}(N_r \log(N_r)) = \mathcal{O}(3N \log(3N)) \quad (36)$$

and is independent of l . The second stage consists of $\frac{l}{2}$ stages of a bidiagonal matrix-vector product of complexity $\mathcal{O}(N_{l/2})$ and a bidiagonal linear solve which has the same complexity $\mathcal{O}(N_{l/2})$. Clearly the current algorithm is not asymptotically faster than GCQ and remains $\mathcal{O}(N^2)$.

In order to obtain a more accurate description, the exact operation count needs to be considered. The bidiagonal matrix-vector product and linear solve for a matrix of size $M \times M$ require $3M$ floating point operations each. The GCQ matrix-vector product requires

$$q_{gcq} = 2N_r N - N = 2N(3N + 1) \quad (37)$$

operations for a forward or backward transform. For a given l applying the series of connection matrices V_l and M_l in the second stage of the algorithm requires

$$q_j = 2 \sum_{i=0}^{l/2} 3(N_{l/2} - \frac{i}{2}) = \frac{3}{8}(l+2)(8N+3l). \quad (38)$$

The DCT requires

$$q_{FFT} = 2N_r \log(N_r) - N_r + 2. \quad (39)$$

And combining both stages gives the cost of the full Jones-Worland transform q_{wt} as

$$q_{wt} = \frac{3}{8}(l+2)(8N+3l) + 2N_r \log(N_r) - N_r + 2 \quad (40)$$

for $0 \leq l \leq 2N$. The worst case is obtained for the highest harmonic degree $l = 2N$ for which

$$\lim_{N \rightarrow \infty} \frac{q_{wt}}{q_{gcq}} = \frac{7}{4}. \quad (41)$$

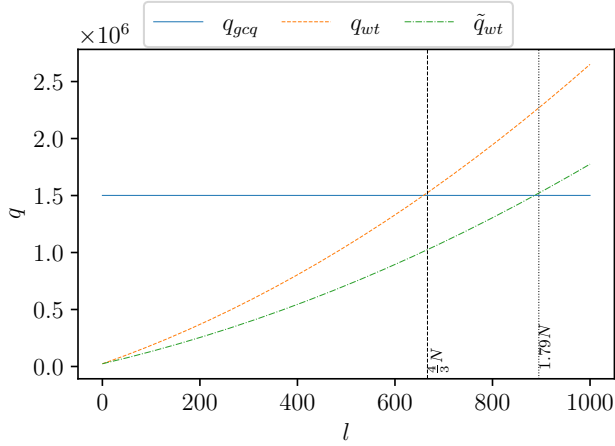


Figure 3: Comparison of the operation count q for $L = 1000$ for (blue,solid) GCQ algorithm, (orange,dashed) this algorithm and (green,dotted) this algorithm with unit diagonals. The black vertical lines show when the algorithms break even as $N \rightarrow \infty$ at $\frac{4}{3}N$ (dashed) and at $1.79N$ (dotted).

In the worst case, our FFT based transform requires less than twice as many operations. The GCQ and this algorithm break even for $l \approx \frac{4}{3}N$. The evolution of the cost is illustrated in Figure 3.

The operation count can be improved by choosing a different normalization for the matrices V_l and M_l such that they have unit entries on the diagonals. Indeed, the cost of applying a matrix of size $M \times M$ with unit diagonal is reduced to $2M$ operations compared to the $3M$ operations required by a general matrix. The diagonal normalization required to achieve this can readily be integrated into the \mathcal{D} normalization matrix in our algorithm. The computation cost is then given by

$$\tilde{q}_{wt} = \frac{1}{4}(l+2)(8N+3l) + 2N_r \log(N_r) + 2, \quad (42)$$

and the worst case becomes

$$\lim_{N \rightarrow \infty} \frac{q_{wt}}{q_{gcq}} = \frac{7}{6}. \quad (43)$$

The GCQ and the optimized algorithm break even for $l \approx 1.79n$ (see Figure 3).

3.1 Full pseudospectral transform

In most applications, the Jones-Worland transform is a part of a 3D pseudospectral transform and thus not only a single l needs to be transformed but all $0 \leq l \leq L$. The GCQ approach has a fixed cost for each transform and is independent of l . Thus the total cost for all l is simply given by

$$Q_{gcq} = (L+1)q_{gcq} = 2N(2N+1)(3N+1) = 12N^3 + \mathcal{O}(N^2) \quad (44)$$

The FFT based transform behaves differently as the computational cost of each transform increases with increasing l (see Fig.

3). For the optimized version, we have

$$\begin{aligned} Q_{wt} &= \sum_{l=0}^L \tilde{q}_{wt} = \frac{1}{4}(2N+1)(12N^2 + 23N + 8) \\ &\quad + 3(2N+1)^2 \log(3N + \frac{3}{2}) \\ &= 6N^3 + \mathcal{O}(N^2 \log(N)). \end{aligned} \quad (45)$$

Therefore for large N , the comparison to GCQ quadrature if all l are transformed becomes

$$\lim_{N \rightarrow \infty} \frac{Q_{wt}}{Q_{gcq}} = \frac{1}{2}. \quad (46)$$

3.2 Memory requirements

At high resolution, memory requirements for the transforms become an important aspect. Temporary storage required during the transform will not be considered here. The GCQ approach is simple in this respect and requires the $N_r \times N$ entries of the quadrature matrix. As they are different for each harmonic degree l , the memory requirements become

$$S_{gcq} = (L+1)N_r N = 3N^2(2N+1) = \mathcal{O}(N^3) \quad (47)$$

for a full 3D transform. The algorithm presented here requires the V_i and M_i bidiagonal matrices to be stored. For a given l , the memory footprint is then

$$S_{wt} = \sum_{i=2}^L 2 \left(N + \frac{L}{2} - i/2 \right) - 1 = \frac{1}{2}(2N-1)(7N-3) = \mathcal{O}(N^2). \quad (48)$$

As shown in Equations (47) and (48), the FFT based algorithm leads to an extensive saving in memory as only $\mathcal{O}(N^2)$ elements need to be stored.

4 ACCURACY AND NUMERICAL TESTS

As discussed in previous sections, our algorithm requires a lower but comparable operation count as the standard GCQ approach (see Eq. (46)). The accompanying memory savings also provide a solution to the memory footprint considerations. It remains to be demonstrated that the algorithm provides highly accurate solutions. In this respect, a first observation is that the algorithm described above does not require the explicit evaluation of the Jones-Worland polynomials on the physical grid. The conversions are done in spectral space and final evaluation on the grid is obtained through the FFT. Therefore the underflow issues present for the GCQ approach do not appear.

As a first test for the accuracy of the transform, we consider the test functions

$$f_l = r^l (1.0 + r^2 + r^4 + r^8) \quad (49)$$

evaluated for various l . The function is evaluated on the grid and the corresponding Jones-Worland coefficients are computed using the new algorithm. Reference values are obtained using Mathematica to 16 digits of precision. Figure 4 shows the absolute error incurred after a single forward transform for $l = 100$ and $l = 101$ in order to test both the even and odd cases.

In order to test both forward and backward transforms for even and odd l , thus testing all of algorithms 1-4, we show in Figures 5 and 6 the round trip errors in spectral-physical space transforms

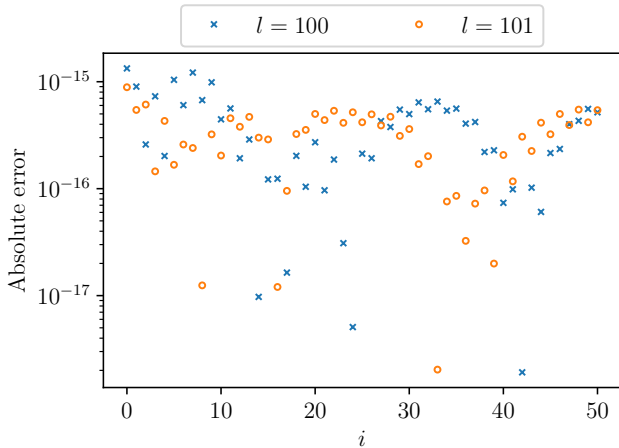


Figure 4: Absolute error observed on the spectral coefficients f_i for the forward transform with $N = 50$ when (blue, cross) $l = 100$ and (orange, circle) $l = 101$, using the test function of equation (49).

and back. In Figure 5 we consider a transform where the input is a single mode with the highest N , whereas 6 considers a transform where the input is a broadband signal with unit power for all modes. One can see that the error incurred is tiny, remaining at less than 10^{-14} even when L is as large as 2001. The scaling of the error with increasing harmonic degree l is shown in Figure 7. The errors for a spectrum mimicking a well converged expansion, with $f_i = U(-1, 1)/n^2$ where $U(-1, 1)$ is a uniform distribution, are shown to scale approximately with $O(\sqrt{N})$.

5 PERSPECTIVES

The algorithm presented here applies to the forward and backward transform of Jones-Worland expansion. For a 3D simulation where, for example, fluid flow \mathbf{u} is based on a Toroidal/Poloidal decomposition

$$\mathbf{u} = \nabla \times (\mathcal{T}\mathbf{r}) + \nabla \times \nabla \times (\mathcal{P}\mathbf{r}), \quad (50)$$

in addition to requiring scalar fields, additional operators need to be computed in spectral space and on the grid. Frequently needed spectral expansions are rf , $\frac{1}{r}f$, or $\frac{1}{r}\frac{\partial}{\partial r}f$, while on the grid evaluations include $\frac{1}{r}f$, $\frac{\partial}{\partial r}f$, $\frac{1}{r}\frac{\partial}{\partial r}(rf)$ and $\nabla^2 f$. Providing a complete description on how those operators are implemented is outside the scope of this study. Nevertheless, we note that differentiation of a Jacobi polynomial can be expressed as a combination of Jacobi polynomials with promoted α and β parameters while factors of r affect the r^l prefactor. The recurrence relations (11) and (14) together with the complementary relations for promoting the α parameter allow adaptation of our algorithms to obtain the required operators. In doing so, spectral differentiation can also be computed without the need to explicitly evaluate the polynomials on the grid.

Vasil et al. [22] advocate the choice $(\alpha, \beta) = (0, l + 1/2)$, this can be accomplished using the present algorithms by first performing a fast Jacobi transform (e.g. [19]) and by using one additional multiplication by V_{l+1} .

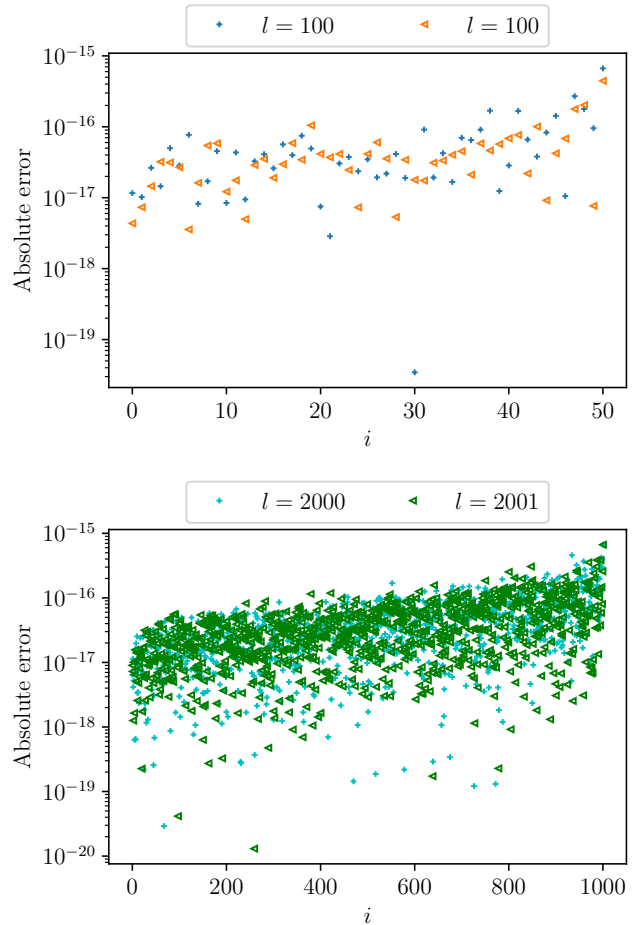


Figure 5: Absolute error on the spectral coefficients f_i in backward-forward transform loop with only highest mode set to 1 for (top) $L = 101$, $N = 50$ and (bottom) $L = 2001$ and $N = 1000$, as a function of i .

The formulation we have presented provides a low-memory implementation of the Jones-Worland transform with a favourable operation count compared to the quadrature based approach. With its lower memory footprint it presents a suitable alternative for an implementation on modern-day GPU-equipped computers. A proof-of-concept implementation for GPU has not revealed any fundamental issues. While only an optimized implementation will give a definite answer on its computational performance, alternatives as the presented algorithm are required as the quadrature approach is not viable anymore for high resolution with $N = O(10^3)$.

We have not described the advantageous features that arise in time-stepping application of this formulation to the Navier-Stokes equation [12]. The graceful retreat of the first zero of the Jones-Worland polynomials from the origin with increasing l is a critical aspect of this methodology that leads to innocuous time step constraints based on the CFL condition that would otherwise be onerous when other types of polynomial are used. The challenges

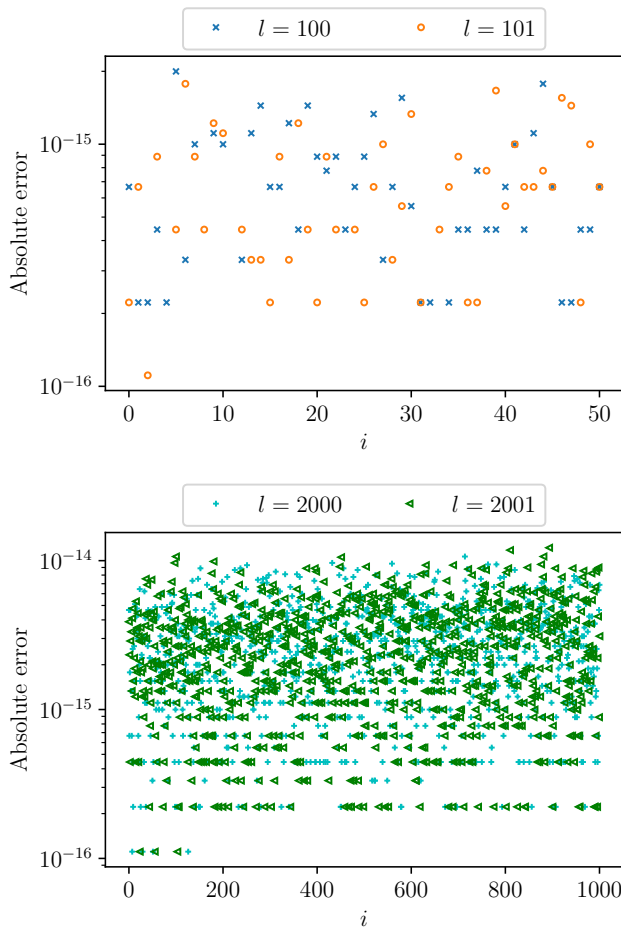


Figure 6: Absolute error on the spectral coefficients f_i in backward-forward transform loop with full unit spectrum for (top) $L = 101$, $N = 50$ and (bottom) $L = 2001$ and $N = 1000$, as a function of i .

that lie ahead are to consider implementations that might lead to even more advantageous operation count whilst retaining accuracy.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme grant agreement No 833848 (<https://cordis.europa.eu/project/id/833848>) (UEMHP), and from SNF grant 200021_165641/1.

REFERENCES

- [1] Nicolas Boullé and Alex Townsend. 2020. Computing with Functions in the Ball. *SIAM Journal on Scientific Computing* 42, 4 (2020), C169–C191. <https://doi.org/10.1137/19M1297063> arXiv:<https://doi.org/10.1137/19M1297063>
- [2] Long Chen, Wietze Herreman, Kuan Li, Philip W Livermore, JW Luo, and Andrew Jackson. 2018. The optimal kinematic dynamo driven by steady flows in a sphere. *Journal of Fluid Mechanics* 839 (2018), 1–32.
- [3] U. Christensen, P. Olson, and G. A. Glatzmaier. 1999. Numerical modelling of the geodynamo: a systematic parameter study. *Geophysical Journal International* 138, 2 (1999), 393–409. <https://doi.org/10.1046/j.1365-246X.1999.00886.x>

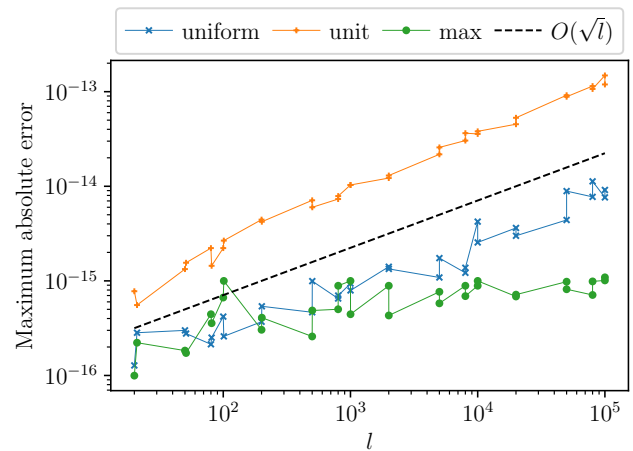


Figure 7: Maximum absolute error for $l = 2N$ on the spectral coefficients f_i in backward-forward transform loop with (blue cross) full unit spectrum, (orange plus) only highest mode set to 1 and (green dot) a converged uniform spectrum $f_i = U(-1, 1)/i^2$. The dashed black line is $O(\sqrt{l})$.

- [4] DLMF [n.d.]. *NIST Digital Library of Mathematical Functions*. <http://dlmf.nist.gov/>, Release 1.0.24 of 2019-09-15. <http://dlmf.nist.gov/> F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.
- [5] Gary A. Glatzmaier and Paul H. Roberts. 1995. A three-dimensional convective dynamo solution with rotating and finitely conducting inner core and mantle. *Physics of the Earth and Planetary Interiors* 91, 1–3 (1995), 63 – 75. [https://doi.org/10.1016/0031-9201\(95\)03049-3](https://doi.org/10.1016/0031-9201(95)03049-3)
- [6] Nicholas Hale and Alex Townsend. 2014. A Fast, Simple, and Stable Chebyshev–Legendre Transform Using an Asymptotic Formula. *SIAM Journal on Scientific Computing* 36, 1 (2014), A148–A167. <https://doi.org/10.1137/130932223> arXiv:<https://doi.org/10.1137/130932223>
- [7] Daniel Lecoanet, Geoffrey M. Vasil, Keaton J. Burns, Benjamin P. Brown, and Jeffrey S. Oishi. 2019. Tensor calculus in spherical coordinates using Jacobi polynomials. Part-II: Implementation and examples. *Journal of Computational Physics: X* 3 (2019), 100012. <https://doi.org/10.1016/j.jcpx.2019.100012>
- [8] Kuan Li, Andrew Jackson, and Philip W. Livermore. 2018. Taylor state dynamos found by optimal control: axisymmetric examples. *Journal of Fluid Mechanics* 853 (2018), 647–697. <https://doi.org/10.1017/jfm.2018.569>
- [9] Kuan Li, Philip W. Livermore, and Andrew Jackson. 2010. An optimal Galerkin scheme to solve the kinematic dynamo eigenvalue problem in a full sphere. *J. Comput. Phys.* 229, 23 (2010), 8666 – 8683. <https://doi.org/10.1016/j.jcp.2010.07.039>
- [10] Yufeng Lin, Philippe Marti, Jerome Noir, and Andrew Jackson. 2016. Precession-driven dynamos in a full sphere and the role of large scale cyclonic vortices. *Physics of Fluids* 28, 6 (2016), 066601. <https://doi.org/10.1063/1.4954295> arXiv:<https://doi.org/10.1063/1.4954295>
- [11] Philip W. Livermore, Chris A. Jones, and Steven J. Worland. 2007. Spectral radial basis functions for full sphere computations. *J. Comput. Phys.* 227, 2 (2007), 1209 – 1224. <https://doi.org/10.1016/j.jcp.2007.08.026>
- [12] Philippe Marti and Andrew Jackson. 2016. A fully spectral methodology for magnetohydrodynamic calculations in a whole sphere. *J. Comput. Phys.* 305 (2016), 403–422. <https://doi.org/10.1016/j.jcp.2015.10.056>
- [13] P. Marti, N. Schaeffer, R. Hollerbach, D. Cébron, C. Nore, F. Luddens, J.-L. Guermond, J. Aubert, S. Takehiro, Y. Sasaki, Y.-Y. Hayashi, R. Simitev, F. Busse, S. Vantieghem, and A. Jackson. 2014. Full sphere hydrodynamic and dynamo benchmarks. *Geophysical Journal International* 197, 1 (2014), 119–134. <https://doi.org/10.1093/gji/ggt518>
- [14] Akil Narayan and Jan S. Hesthaven. 2012. Computation of connection coefficients and measure modifications for orthogonal polynomials. *BIT Numerical Mathematics* 52, 2 (01 Jun 2012), 457–483. <https://doi.org/10.1007/s10543-011-0363-z>
- [15] A. C. Narayan and J. S. Hesthaven. 2009. The Application of the Fast Fourier to Jacobi Polynomial expansions. (2009). <https://www.semanticscholar.org/paper/The-Application-of-the-Fast-Fourier-Transform-to-Narayan-Hesthaven/6044a0e2cbf2712501c13644eaa69c2d788be4bf>

- [16] Steven Alan Orszag. 1986. Fast eigenfunction transforms. In *Science and Computers, Advances in Mathematics, Supplementary Studies, Vol. 10*, G. C. Rota (Ed.). Academic Press, 13–30.
- [17] Daniel Potts. 2003. Fast algorithms for discrete polynomial transforms on arbitrary grids. *Linear Algebra Appl.* 366 (2003), 353–370. [https://doi.org/10.1016/S0024-3795\(02\)00592-X](https://doi.org/10.1016/S0024-3795(02)00592-X) Special issue on Structured Matrices: Analysis, Algorithms and Applications.
- [18] Jie Shen, Yingwei Wang, and Jianlin Xia. 2018. Fast structured Jacobi-Jacobi transforms. *Math. Comp.* 88 (05 2018), 1. <https://doi.org/10.1090/mcom/3377>
- [19] Richard Mikael Slevinsky. 2017. On the use of Hahn’s asymptotic formula and stabilized recurrence for a fast, simple and stable Chebyshev–Jacobi transform. *IMA J. Numer. Anal.* 38, 1 (02 2017), 102–124. <https://doi.org/10.1093/imanum/drw070> arXiv:<https://academic.oup.com/imajna/article-pdf/38/1/102/23651272/drw070.pdf>
- [20] Krista M. Soderlund, Eric M. King, and Jonathan M. Aurnou. 2012. The influence of magnetic fields in planetary dynamo models. *Earth and Planetary Science Letters* 333–334 (2012), 9 – 20. <https://doi.org/10.1016/j.epsl.2012.03.038>
- [21] John A. Tarduno, Rory D. Cottrell, Michael K. Watkeys, Axel Hofmann, Pavel V. Dobrovine, Eric E. Mamajek, Dunji Liu, David G. Sibeck, Levi P. Neukirch, and Yoichi Usui. 2010. Geodynamo, Solar Wind, and Magnetopause 3.4 to 3.45 Billion Years Ago. *Science* 327, 5970 (2010), 1238–1240. <https://doi.org/10.1126/science.1183445> arXiv:<http://www.sciencemag.org/content/327/5970/1238.full.pdf>
- [22] Geoffrey M. Vasil, Daniel Leccoanet, Keaton J. Burns, Jeffrey S. Oishi, and Benjamin P. Brown. 2019. Tensor calculus in spherical coordinates using Jacobi polynomials. Part-I: Mathematical analysis and derivations. *Journal of Computational Physics: X* 3 (2019), 100013. <https://doi.org/10.1016/j.jcpx.2019.100013>
- [23] W.T.M. Verkleij. 1997. A Spectral Model for Two-Dimensional Incompressible Fluid Flow in a Circular Basin. *J. Comput. Phys.* 136, 1 (1997), 100 – 114. <https://doi.org/10.1006/jcph.1997.5747>
- [24] S. J. Worland. 2004. *Magnetoconvection in rapidly rotating spheres*. Ph.D. Dissertation. University of Exeter.