

Article

Algorithmic Aspects of Simulation of Magnetic Field Generation by Thermal Convection in a Plane Layer of Fluid

Daniil Tolmachev ¹, Roman Chertovskih ^{2,*}  and Vladislav Zheligovsky ¹

¹ Institute of Earthquake Prediction Theory and Mathematical Geophysics, Russian Academy of Sciences, 84/32 Profsoyuznaya St., 117997 Moscow, Russia

² Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

* Correspondence: roman@fe.up.pt

Abstract: We present an algorithm for numerical solution of the equations of magnetohydrodynamics describing the convective dynamo in a plane horizontal layer rotating about an arbitrary axis under geophysically sound boundary conditions. While in many respects we pursue the general approach that was followed by other authors, our main focus is on the accuracy of simulations, especially in the small scales. We employ the Galerkin method. We use products of linear combinations (each involving two to five terms) of Chebyshev polynomials in the vertical Cartesian space variable and Fourier harmonics in the horizontal variables for space discretisation of the toroidal and poloidal potentials of the flow (satisfying the no-slip conditions on the horizontal boundaries) and magnetic field (for which the boundary conditions mimic the presence of a dielectric over the fluid layer and an electrically conducting bottom boundary), and of the deviation of temperature from the steady-state linear profile. For the chosen coefficients in the linear combinations, the products satisfy the respective boundary conditions and constitute non-orthogonal bases in the weighted Lebesgue space. Determining coefficients in the expansion of a given function in such a basis (for instance, for computing the time derivatives of these coefficients) requires solving linear systems of equations for band matrices. Several algorithms for determining the coefficients, which are exploiting algebraically precise relations, have been developed, and their efficiency and accuracy have been numerically investigated for exponentially decaying solutions (encountered when simulating convective regimes which are spatially resolved sufficiently well). For the boundary conditions satisfied by the toroidal component of the flow, our algorithm outperforms the shuttle method, but the latter proves superior when solving the problem for the conditions characterising the poloidal component. While the conditions for the magnetic field on the horizontal boundaries are quite specific, our algorithms for the no-slip boundary conditions are general-purpose and can be applied for treating other boundary-value problems in which the zero value must be admitted on the boundary.

Keywords: thermal convection; convective magnetic dynamo; Galerkin method; Chebyshev polynomials; shuttle algorithm; numerical accuracy

MSC: 65M60; 76D05; 76W05; 76R10; 76M22



Citation: Tolmachev, D.; Chertovskih, R.; Zheligovsky, V. Algorithmic Aspects of Simulation of Magnetic Field Generation by Thermal Convection in a Plane Layer of Fluid. *Mathematics* **2023**, *11*, 808. <https://doi.org/10.3390/math11040808>

Academic Editor: Xiang Liu

Received: 6 December 2022

Revised: 16 January 2023

Accepted: 21 January 2023

Published: 5 February 2023

Corrected: 22 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

We present an algorithm for numerical study of the fully nonlinear processes of magnetic field generation by thermal convection in a plane layer of electrically conducting fluid rotating about an inclined axis. This is required, for instance, for investigating magnetic reversals and their precursors, which is at present a hot topic in geophysics.

The dynamo model must comply with the intention to apply it for the purposes of this study. The equations describing the geodynamo have two essential properties: they have a quadratic nonlinearity and they are symmetric with respect to reversing the orientation of the magnetic field [1]. Many other specific features of the natural Earth's dynamo are of a

lesser importance and can be disregarded in order to reduce the numerical complexity of computations. For instance, we neglect the processes of sedimentation of the heavy fraction, crystallisation and other phase transitions, as well as chemical reactions in the melt, the complex character of the rheology of substance at high pressures and temperatures, and the small-scale geometric irregularity of the core–mantle boundary.

The choice of the geometry of the fluid volume is again a compromise between the numerical efficiency of the algorithm and its subsequent application for solving the geophysical problems. We consider a rectangular fluid cell that is supposed to represent a small segment of the melted outer Earth's core, having the form of a spherical layer. Computing in a fluid parallelepiped is numerically more efficient than in a part of a spherical layer, and for a relatively small cell we expect the rudimentary sphericity not to affect the flow and dynamo behaviour significantly. Since the cell can reside at an arbitrary latitude, the axis of rotation can have an arbitrary direction. The assumed boundary conditions emulate the conditions under which the Earth's dynamo operates: there is a largely dielectric mantle over the outer core and a predominantly iron solid inner core, which is a good electric conductor. Consequently, we assume that there is a dielectric over the upper horizontal boundary of the fluid layer, the lower horizontal boundary is electrically conducting, and the flow satisfies the no-slip boundary conditions (see the illustrative Figure 1 reflecting the physical process with a coordinate system). All the unknown fields (the flow velocity, magnetic field and temperature) are supposed to be periodic in the horizontal Cartesian variables. While convective dynamos with rotation about the vertical axis are studied in many papers (see, e.g., [2]), rotation about an inclined axis has not attracted the attention of many investigators, Refs. [3–5] being notable exceptions.

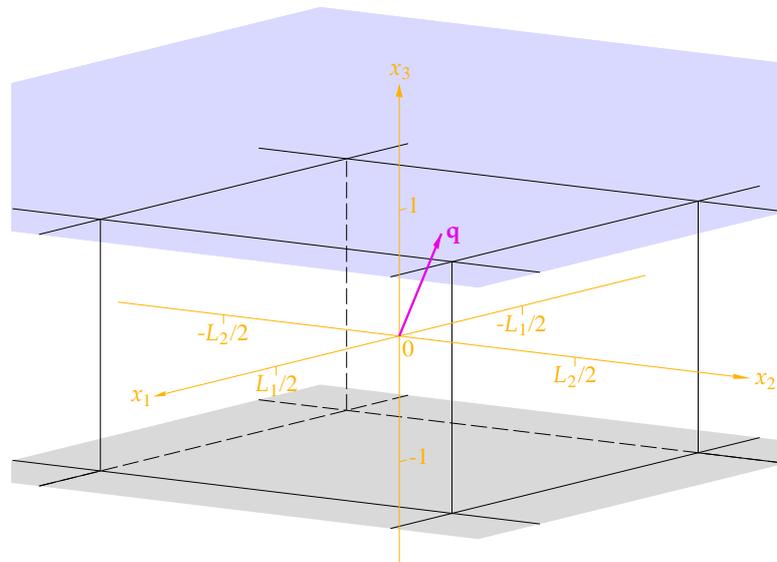


Figure 1. An illustrative sketch of the physical process: there is a dielectric over the upper horizontal boundary (blue) of the fluid layer (white), the lower horizontal boundary (gray) is electrically conducting, and the flow satisfies the no-slip boundary conditions at the boundaries. The fluid layer is rotating about the axis, whose arbitrary direction is shown by the unit vector \mathbf{q} (magenta), the Cartesian coordinate system (orange colour, respectively) is co-rotating with the fluid.

We know from the analysis of the equations of magnetohydrodynamics [6] that the behaviour of the small-scale (spatial scales are meant here) component of the solution is consequential: it can provoke the hypothetical break-ups of the spatial smoothness and analyticity of solutions, and it is also likely to carry the signature of the developing magnetic reversals. We therefore pay special attention to the accuracy of the numerical solution throughout the entire spatial spectrum.

We apply a spatial discretisation for the partial differential equations similar to that employed in [3,4,7], in particular, Chebyshev polynomials are employed *ibid.* to approxi-

mate the behaviour of the unknown fields in the vertical (i.e., perpendicular to the layer) direction. Chebyshev polynomials are widely used in numerical analysis [8,9]. Methods relying on them for computing solenoidal flows satisfying the no-slip boundary conditions were developed, e.g., for a horizontal layer of fluids in [9–12] and in the cylindrical geometry in [13]. Among recent applications, let us mention the use of Chebyshev polynomials in [14] for expansion of marginally stable modes in the quasilinear reduction of equations of the Rayleigh–Bénard convection and in [15] for a numerical study of the complex dynamics in convection in an inclined fluid layer, where a trajectory in a phase space follows a sequence of connections between unstable invariant states before advancing towards a stable attracting state. Employing Chebyshev polynomials for discretising the fields in the vertical (or radial) direction can be advantageous for two reasons. First, the standard pseudospectral methods can then be employed together with the fast Chebyshev transform, which is essentially the usual fast cosine transform. Second, zeroes of these polynomials increasingly accumulate near the ends of the interval $[-1, 1]$, this suggesting that they are well-suited for representing functions that feature a complex behaviour near the end points, e.g., when boundary layers emerge in convection for low Ekman numbers. This clustering of roots of Chebyshev polynomials proved useful for exploring the steady two-dimensional amagnetic convection in the plane layer at high Rayleigh numbers by Chebyshev collocations [16,17] and for tackling the magnetic dynamo problem [18] for a uniform shear flow between a solid plate sliding over another one, both having anisotropic electrical conductivity. (It can be exploited directly: for instance, finite differences in the radial direction were used in [19] with the nodes located at the roots of Chebyshev polynomials.)

When applying the Galerkin method, one faces the dilemma: either the orthogonality of the basis must be sacrificed (if the numerical complexity of the algorithm must be kept minimal) or the basic functions do not satisfy the boundary conditions. For instance, the so-called tau method [12] (used, e.g., in [20] to perform the linear stability analysis of convection in an infinite rotating tilted cavity), which is a particular case of the Petrov–Galerkin method, follows the latter path and can suffer from numerical instability [21]. We apply the Galerkin method because it guarantees the minimisation of the discrepancy due to the spatial discretisation at each time step.

Let us note that for the Navier–Stokes equation, the issue of the boundary conditions is intricate. For instance, the flow being solenoidal, the no-slip boundary condition (three scalar conditions) on a horizontal boundary implies the fourth independent scalar condition for the normal derivative of the vertical component of the flow, $\partial v_3 / \partial x_3 = 0$, on this boundary. This additional condition serves, in fact, as an implicit boundary condition for the pressure; while it is difficult to elicit the latter in the explicit form, the strategy of solving the Poisson equation for the pressure when solving the hydrodynamic boundary value problems was explored by many authors (see [22] and references therein).

Our approach to the treatment of the boundary conditions is original. We split the flow and magnetic field into the toroidal, poloidal and mean-field components, identify the (distinct) conditions on the horizontal boundaries for the potentials of these components, and choose the coefficients of linear combinations of Chebyshev polynomials in the vertical variable (each involving two to five terms) that satisfy these conditions. The sought vector potentials are expanded into finite series of products of these linear combinations and Fourier harmonics in the horizontal variables. Expansion in the basic functions that satisfy the required boundary conditions is a hallmark of the Galerkin method. The linear combinations constitute a non-orthogonal basis in the functional weighted Lebesgue space (the weight required for the orthogonality of the Chebyshev polynomials is assumed). As a result, for determining the time derivatives of the coefficients in the expansion of the fields, we need to solve linear systems of equations for band matrices. The matrices emerging when tackling the problems for toroidal parts of the flow and magnetic field, as well as for the poloidal part of the flow, have special properties: for instance, the sum of the entries is zero in all rows except for the upper two and the lower two. We propose specialised

(sub)algorithms for determining the coefficients in the expansion of a given function in the basis of these polynomials, where such properties of the problem are taken into account.

These algorithms are based on exact formulae and are therefore algebraically equivalent, but in applications with a finite-precision arithmetic the equivalence is lost. We have analysed the efficiency and accuracy of the proposed algorithms, and identified the algorithms yielding the most (in the statistical sense) accurate solutions that have an exponentially decaying energy spectrum, as happens when a sufficiently high spatial resolution is used. The algorithms are applicable for a wide range of linear problems arising when Chebyshev polynomials are used for the spatial discretisation of a solution to a boundary-value problem for a partial differential equation. Notably, the choice of algorithms is very sensitive to the problem to be solved: we find that while our original algorithm is optimal for no-slip boundaries, a similar algorithm is inferior to the shuttle method when applied to an analogous problem involving a five-diagonal band matrix that arises for somewhat different boundary conditions.

To offset the influence of the stiffness of the Fourier–Galerkin system of equations, which we integrate numerically, we apply the third-order Runge–Kutta scheme with exponential differencing [23] for integration in time, which has proved useful in similar computations [24–26].

The statement of the problem is presented in the next section. Discretisation of the flow and magnetic field, and algorithms for computing their time derivatives are discussed in Sections 3 and 4, respectively. The algorithms considered in Section 3.1 for the toroidal component of the flow are general-purpose; they are applicable for all functions vanishing at the boundary and are also employed, in particular, for the deviation of the temperature from the linear profile. Our concluding remarks are summarised in the last section.

2. The Governing Equations

Equations governing the geodynamo in the Earth’s outer core were discussed in detail in [27]. We focus on the nonlinear magnetic dynamo powered by the Rayleigh–Bénard convection in a plane layer of rotating fluid heated from below, the axis of rotation being inclined at an arbitrary constant angle. The Boussinesq approximation is considered: the buoyancy is assumed to depend linearly on temperature, density variation is neglected and the flow is deemed incompressible. Electrically insulating horizontal boundaries of the layer are supposed to be held at constant temperatures; the flow satisfies the no-slip conditions at the boundary and periodicity in horizontal directions is assumed. In contrast with [3,4,27], we neglect neither the time derivative of the flow, nor the advective term in the momentum equation.

In the Cartesian coordinate system, co-rotating with the fluid layer, the governing equations for the nondimensionalised quantities are as follows: the Navier–Stokes equation

$$\frac{\partial \mathbf{v}}{\partial t} = \mathbf{v} \times (\nabla \times \mathbf{v}) + P \nabla^2 \mathbf{v} - \nabla p + \sigma \theta \mathbf{e}_3 + \tau \mathbf{v} \times \mathbf{q} - \mathbf{b} \times (\nabla \times \mathbf{b}) \tag{1a}$$

(the last three terms in the r.h.s. of (1a) express the Archimedes buoyancy, Coriolis and Lorentz forces, respectively), the magnetic induction equation

$$\frac{\partial \mathbf{b}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{b}) + \eta \nabla^2 \mathbf{b} \tag{1b}$$

and the heat transfer equation

$$\frac{\partial \theta}{\partial t} = -\mathbf{v} \cdot (\nabla \theta - \mathbf{e}_3) + \nabla^2 \theta. \tag{1c}$$

The flow and magnetic field are solenoidal:

$$\nabla \cdot \mathbf{v} = \nabla \cdot \mathbf{b} = 0. \tag{1d}$$

Here \mathbf{v} denotes the flow velocity, t time, p the modified pressure, \mathbf{b} the magnetic field, θ the difference between the temperature of fluid T and the linear temperature profile (if the horizontal boundaries $x_3 = -1$ and $x_3 = 1$ are kept at constant temperatures T_{-1} and T_1 , respectively, then $\theta = T - (T_1 + T_{-1} + (T_1 - T_{-1})x_3)/2$), a unit vector \mathbf{q} is aligned with the axis of rotation of the fluid layer, and \mathbf{e}_n are the unit vectors of the Cartesian coordinate system $\mathbf{x} = (x_1, x_2, x_3)$, \mathbf{e}_3 being vertical. Thermal convection is often characterised by the Rayleigh number, Ra , reflecting the magnitude of the buoyancy force; the Prandtl number, P , the ratio of kinematic viscosity to thermal diffusivity; the magnetic Prandtl number, P_m , the ratio of kinematic viscosity to magnetic diffusivity; and the Taylor number, Ta , measuring the speed of rotation. We have denoted $\sigma = PRa$, $\tau/P = \sqrt{Ta}$ is proportional to the angular speed of rotation of the fluid, and $\eta = P/P_m$.

We assume no-slip conditions for the flow velocity at the electrically insulating upper and perfectly electrically conducting lower horizontal boundaries. The magnetic field in the insulator over the upper boundary is a gradient of a harmonic function and the field is continuous at the boundary:

$$\mathbf{v}|_{x_3=\pm 1} = 0, \tag{2a}$$

$$\mathbf{b} = \nabla h, \quad \nabla^2 h = 0 \text{ for } x_3 \geq 1, \quad h \rightarrow 0 \text{ for } x_3 \rightarrow \infty, \tag{2b}$$

$$\frac{\partial b^1}{\partial x_3} \Big|_{x_3=-1} = \frac{\partial b^2}{\partial x_3} \Big|_{x_3=-1} = b^3 \Big|_{x_3=-1} = 0, \tag{2c}$$

and by construction

$$\theta|_{x_3=\pm 1} = 0. \tag{2d}$$

The fields have periods $L_i = 2\pi/\alpha_i$ in the horizontal directions: for all integer n_1 and n_2 ,

$$\begin{aligned} \mathbf{v}(\mathbf{x}) &= \mathbf{v}(x_1 + n_1 L_1, x_2 + n_2 L_2, x_3), \\ \mathbf{b}(\mathbf{x}) &= \mathbf{b}(x_1 + n_1 L_1, x_2 + n_2 L_2, x_3), \\ \theta(\mathbf{x}) &= \theta(x_1 + n_1 L_1, x_2 + n_2 L_2, x_3). \end{aligned} \tag{3}$$

The equations are solved numerically by the pseudospectral methods [28]. However, care must be taken in satisfying the boundary conditions (2).

3. Discretisation of the Flow

Let us consider the implications of the boundary conditions (2a) for the flow. Solenoidality of $\mathbf{v}(\mathbf{x}, t)$ (1d) and the no-slip conditions for the horizontal components of the flow imply that in addition to (2a), the condition

$$\frac{\partial v^3}{\partial x_3} \Big|_{x_3=\pm 1} = 0 \tag{4}$$

holds (which is a boundary condition for the pressure in an implicit form).

The flow can be decomposed into the sum of the toroidal, $\mathbf{T}^v(\mathbf{x}, t)$, poloidal, $\mathbf{P}^v(\mathbf{x}, t)$, and mean-field, $\mathbf{M}^v(\mathbf{x}, t)$, (understood here in the sense of averaging over the horizontal variables x_1 and x_2) components (see, e.g., [29]):

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{T}^v(\mathbf{x}, t) + \mathbf{P}^v(\mathbf{x}, t) + \mathbf{M}^v(x_3, t), \tag{5a}$$

$$\mathbf{T}^v(\mathbf{x}, t) = \nabla \times (T^v(\mathbf{x}, t)\mathbf{e}_3), \tag{5b}$$

$$\mathbf{P}^v(\mathbf{x}, t) = \nabla \times \nabla \times (P^v(\mathbf{x}, t)\mathbf{e}_3), \tag{5c}$$

$$\mathbf{M}^v(x_3, t) = \left\langle (v^1(\mathbf{x}, t), v^2(\mathbf{x}, t), 0) \right\rangle_h, \tag{5d}$$

where

$$\langle \mathbf{f} \rangle_h = (L_1 L_2)^{-1} \int_0^{L_2} \int_0^{L_1} \mathbf{f}(\mathbf{x}) dx_1 dx_2.$$

In view of the boundary conditions (2a) for the flow, the potentials T^v and P^v , and the mean-field component satisfy the boundary conditions

$$T^v \Big|_{x_3=\pm 1} = 0, \tag{6a}$$

$$P^v \Big|_{x_3=\pm 1} = \frac{\partial P^v}{\partial x_3} \Big|_{x_3=\pm 1} = 0, \tag{6b}$$

$$\mathbf{M}^v \Big|_{x_3=\pm 1} = 0. \tag{6c}$$

Adding an arbitrary function of x_3 (and time) to the toroidal or poloidal potential does not alter the respective components of a vector field; thus, we assume

$$\langle T^v \rangle_h = \langle P^v \rangle_h = 0.$$

The equation for the toroidal potential is obtained by taking the vertical component of the curl of (1a):

$$\begin{aligned} \frac{\partial \nabla_h^2 T^v}{\partial t} &= P \nabla^2 \nabla_h^2 T^v + F^{tv}, \tag{7} \\ F^{tv} &= -\tau(\mathbf{q} \cdot \nabla)v^3 - \mathbf{e}_3 \cdot (\nabla \times (\mathbf{v} \times (\nabla \times \mathbf{v}) - \mathbf{b} \times (\nabla \times \mathbf{b}))), \end{aligned}$$

where $\nabla_h^2 = \partial^2/\partial x_1^2 + \partial^2/\partial x_2^2$ is the Laplacian in horizontal variables. The inverse operator $(\nabla_h^2)^{-1}$ acts in the subspace of vector fields, whose mean $\langle \cdot \rangle_h$ vanishes. Equation (7) is equivalent to the parabolic equation

$$\frac{\partial T^v}{\partial t} = P \nabla^2 T^v + (\nabla_h^2)^{-1} F^{tv} \tag{8}$$

(the mean over horizontal variables of the vertical component of the curl of a field periodic in horizontal directions is zero; therefore, applying the inverse of ∇_h^2 is legitimate here).

Following the Galerkin method, we expand the potentials in series of functions satisfying the respective boundary conditions and consider ordinary differential equations in time for the coefficients of the expansions that express the orthogonality of the resultant discrepancies in the partial differential equations to certain test functions. The full set of functions used for expansion of approximate solutions should constitute a basis in the suitable functional space, as well as the full set of the test functions. The number of the test functions employed for the orthogonal projection when constructing an approximate solution should be equal to the dimension of the functional subspace, where the approximate solution is sought. The traditional Galerkin method, in which the same functional subspaces are employed for approximating the solution and for the orthogonal projection, is usually advantageous (e.g., one can derive energy estimates useful for controlling the convergence of the approximate solutions to the precise ones on increasing the dimension of the subspace). (In the Petrov–Galerkin method, distinct functional subspaces are used for expanding solutions and for the projection [10].) By contrast, it was proposed in [7] to use the collocation method (this can be regarded as “orthogonalisation” to δ -functions, see [28]) to expand solutions in Chebyshev polynomials and to employ two higher-degree Chebyshev polynomials to satisfy the boundary conditions. This procedure is reminiscent of the tau method due to Lancos [30] (see also [31]). However, then the coefficients of the correcting Chebyshev polynomials depend on the coefficients of the low-degree Chebyshev polynomials in such a way that, in principle, the former coefficients can remain large instead of tending to zero on increasing the number of the polynomials used for approximation.

We follow the traditional Galerkin method and use the polynomials (10) for both expanding and projecting. Expansion in the vertical coordinate, x_3 , is not straightforward. In view of (6a), it may seem natural to expand the toroidal potential in the sine Fourier series. However, this basis is not particularly suitable for resolving the boundary layers

emerging at high Ekman numbers. Moreover, every zero of the sine is also a zero of its second derivative; if solutions to the system of equations (1) do not feature this property at the horizontal boundaries (and it is unlikely that they do), then sine series are guaranteed to converge poorly. We therefore resort to the next simplest basis in the Lebesgue space, consisting of Chebyshev polynomials, $T_n(x_3) = \cos(n \arccos x_3)$ for $n \geq 0$ and $-1 \leq x_3 \leq 1$ [32]. They are orthogonal on this interval with the weight $(1 - x_3^2)^{-1/2}$ and we always carry out orthogonalisation in x_3 with this weight function. The relations

$$T_n(x_3) = x_3^n, \quad \frac{dT_n}{dx_3}(x_3) = x_3^{n+1}n^2, \quad \frac{d^2T_n}{dx_3^2}(x_3) = x_3^n(n^4 - n^2)/3 \quad \text{for } x_3 = \pm 1 \quad (9)$$

(see, e.g., [32]) are useful in treating the boundary conditions. They imply that a linear combination of Chebyshev polynomials vanishes at the end points as long as the sum of the coefficients of the even-degree polynomials vanishes, as well as the sum of the coefficients of the odd-degree polynomials. We expand functions satisfying the conditions (6a) in the series of the polynomials

$$T_n^*(x_3) = T_{n+2}(x_3) - T_n(x_3), \quad n \geq 0, \quad (10)$$

as recommended in [33] because of the relative simplicity of the matrices resulting from discretisation of the second- and fourth-order differential operators.

The Gram–Schmidt process yields orthogonal polynomials

$$T_n^\perp(x_3) = T_{n+2}(x_3) - T_n(x_3) + \beta_n T_{n-2}^\perp(x_3), \quad n \geq 0, \quad (11a)$$

$$\beta_0 = \beta_1 = 0, \quad \beta_n = \pi / (4 \|T_{n-2}^\perp\|^2), \quad n \geq 2, \quad (11b)$$

$$\|T_0^\perp\|^2 = 3\pi/4, \quad \|T_1^\perp\|^2 = \pi/2, \quad \|T_n^\perp\|^2 = \pi/2 - (\pi / (4 \|T_{n-2}^\perp\|))^2, \quad n \geq 2. \quad (11c)$$

However, when the polynomials (10) are used, computation of the time derivatives of their coefficients requires solving a system of linear equations for a band matrix; as a result, this approach is computationally more efficient than the use of the orthogonal polynomials T_n^\perp .

3.1. Algorithms for Determining the Coefficients of a Linear Combination of $T_n^*(x_3)$

Consequently, we approximate the toroidal potential by finite series

$$T^v(\mathbf{x}, t) = \sum_{|n_1| \leq N, |n_2| \leq N, 0 \leq n \leq N-3} t_n^v(t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} T_n^*(x_3),$$

where $\mathbf{n} = (n_1, n_2, n)$. The equation for the mean-field component of the flow is derived by averaging over horizontal variables of the horizontal component of (1a). This component is also expanded in polynomials (10),

$$\mathbf{M}^v(x_3, t) = \sum_{0 \leq n \leq N-3} \mathbf{M}_n^v(t) T_n^*(x_3),$$

and equations for the coefficients in the expansion are also obtained by orthogonal projection on these polynomials. In view of the boundary conditions (2d), following the same approach for discretisation of temperature we expand

$$\theta(\mathbf{x}, t) = \sum_{|n_1| \leq N, |n_2| \leq N, 0 \leq n \leq N-3} \theta_n(t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} T_n^*(x_3).$$

The vector \mathbf{d} of the time derivatives of the coefficients of the expansion is a solution to the linear system of equations of the form $G\mathbf{d} = \mathbf{r}$. Here \mathbf{r} is the vector (of length $M = N - 2$) of the dot products of the r.h.s. of (8) with T_m^* , and G is the Gram matrix for the polynomials T_m^* , both (G and \mathbf{r}) divided by $L_1 L_2 \pi / 2$. The matrix has the following non-zero entries: $G_{1,1} = 3$, $G_{k,k} = 2$ for all $k \geq 2$ on the diagonal, and $G_{k,k+2} = G_{k+2,k} = -1$ on two subdiagonals. The linear system of equations can be solved by seven algorithms.

Algorithms 1 and 1' are the standard shuttle (also known as Thomas) algorithms for a three-diagonal matrix [34] applied in the direct and reverse direction.

Algorithms 2 and 2'. We observe that all the equations except for the first two and the last two involve the coefficients $-1, 2, -1$ that are also encountered in the simplest finite differences approximation of the second derivative. Consequently, the l.h.s. of these equations do not change when the solution d_n is altered by any linear function of n , i.e., the "intermediate" equations are invariant under the transformation $d_n \mapsto d_n + \beta_1 n + \beta_2$, where β_1 and β_2 are arbitrary constant numbers. Following this observation, we set, in the spirit of the shuttle algorithm,

$$b_1 = b_2 = 0, \quad b_3 = -r_1, \quad b_4 = -r_2, \quad b_n = 2b_{n-2} - b_{n-4} - r_{n-2} \quad \text{for } 5 \leq n \leq N$$

and find (N is assumed to be even)

$$d_1 = -b_{N-1}/(N-1), \quad d_n = nd_1 + b_n \quad \text{for odd } n, \quad 3 \leq n \leq M-1;$$

$$d_2 = -b_N/(N/2), \quad d_n = (n/2)d_2 + b_n \quad \text{for even } n, \quad 4 \leq n \leq M.$$

This is algorithm 2. While here all unknown coefficients d_n are expressed in terms of d_1 and d_2 , in algorithm 2' d_{M-1} and d_M play the role of such "basic" variables: setting

$$b'_M = -r_M, \quad b'_{M-1} = -r_{M-1}, \quad b'_{M-2} = -2r_M - r_{M-2}, \quad b'_{M-3} = -2r_{M-1} - r_{M-3}, \quad (12a)$$

$$b'_n = 2b'_{n+2} - b'_{n+4} - r_n \quad \text{for } 1 \leq n \leq M-4, \quad (12b)$$

we obtain

$$d_{M-1} = -\frac{b'_1 + b'_3}{N-1}, \quad d_n = \frac{N-n-1}{2}d_{M-1} + b'_{n+2} \quad \text{for odd } n, \quad 1 \leq n \leq M-3, \quad (12c)$$

$$d_M = -\frac{b'_2}{N/2}, \quad d_n = \frac{N-n}{2}d_M + b'_{n+2} \quad \text{for even } n, \quad 2 \leq n \leq M-2. \quad (12d)$$

Both versions of this algorithm have the numerical complexity $O(N)$; by contrast, using the orthogonal polynomials T_n^\perp results in the $O(N^2)$ complexity of computations, which is prohibitively large. Given that the Chebyshev series coefficients r_n tend to zero for large n , the recurrence relation (12b), where computations of b'_n proceed from smaller in magnitude values to larger ones and all d_n involve products of large numbers $N-n$ with small factors d_{M-1} and d_M , is less affected by round-off errors. Thus, algorithm 2' (12) apparently suits our purposes better than algorithm 2.

The analogy between the simplest finite-difference approximation of second-order derivatives of a fictitious function in the l.h.s. of the system at hand is more closely exploited in derivation of algorithms 3 and 3'. This gives an opportunity to find explicit solutions for d_k .

In terms of the variables $z_k = d_{k+2} - d_k$ for $1 \leq k \leq M-2$ ("the first-order derivatives" approximated by the Euler scheme for the unit mesh size), the system of equations takes the form

$$2d_1 - z_1 = r_1, \quad (13a)$$

$$d_2 - z_2 = r_2, \quad (13b)$$

...

$$z_{k-2} - z_k = r_k \quad \text{for } 3 \leq k \leq M-2, \quad (13c)$$

...

$$d_{M-1} + z_{M-3} = r_{M-1}, \quad (13d)$$

$$d_M + z_{M-2} = r_M. \quad (13e)$$

We will now "twice integrate" the fictitious function. The system (13) splits into separate systems for even-index and odd-index variables. In the odd case, we find sequentially from (13a), (13c) and (13d) ("the first integration")

$$\begin{aligned}
 z_1 &= 2d_1 - r_1, \\
 z_3 &= 2d_1 - r_1 - r_3, \\
 z_5 &= 2d_1 - r_1 - r_3 - r_5, \\
 &\dots \\
 z_{2k-1} &= 2d_1 - r_1 - r_3 - \dots - r_{2k-1}, \\
 &\dots \\
 -d_{M-1} &= 2d_1 - r_1 - r_3 - \dots - r_{M-1}.
 \end{aligned}$$

Summing up all these $M/2$ relations (“the second integration”) yields

$$-d_1 = z_1 + \dots + z_{M-3} - d_{M-1} = Md_1 - (M/2)r_1 - (M/2 - 1)r_3 - \dots - 1r_{M-1},$$

whereby

$$d_1 = \frac{1}{M+1} \sum_{j=1}^{M/2} \left(\frac{N}{2} - j\right) r_{2j-1}. \tag{14}$$

Summing up the relations from the first up to the k th one now yields

$$d_{2k-1} = (2k - 1)d_1 - \sum_{j=1}^{k-1} (k - j)r_{2j-1} \text{ for } 1 < k \leq M/2. \tag{15}$$

We find the same way from the system in the even-index unknowns

$$\begin{aligned}
 d_2 &= \frac{2}{N} \sum_{j=1}^{M/2} \left(\frac{N}{2} - j\right) r_{2j}, \\
 d_{2k} &= kd_2 - \sum_{j=1}^{k-1} (k - j)r_{2j} \text{ for } 1 < k \leq M/2.
 \end{aligned} \tag{16}$$

In algorithm 3’ the variables d_{M-1} and d_M play the role of the basic variables d_1 and d_2 in algorithm 3, resulting in the relations

$$\begin{aligned}
 d_{M-1} &= \frac{1}{M+1} \sum_{j=1}^{M/2} (2j - 1)r_{2j-1}, \\
 d_{M-1-2k} &= (k + 1)d_{M-1} + \sum_{j=N/2-k}^{M/2} \left(\frac{M}{2} - k - j\right) r_{2j-1} \text{ for } 1 \leq k \leq M/2 - 1, \\
 d_M &= \frac{2}{N} \sum_{j=1}^{M/2} jr_{2j}, \\
 d_{M-2k} &= (k + 1)d_M + \sum_{j=N/2-k}^{M/2} \left(\frac{M}{2} - k - j\right) r_{2j} \text{ for } 1 \leq k \leq M/2 - 1.
 \end{aligned}$$

Rearranging sums in (15) and (16), we obtain alternative equivalent expressions

$$\begin{aligned}
 d_{2k+1} &= \frac{M - 2k}{M} \left(d_1 + \sum_{j=2}^k (j - 1)r_{2j-1}\right) + \frac{k}{M} \sum_{j=k+1}^{M/2} (N - 2j)r_{2j-1}, \\
 d_{2k+2} &= \frac{M - 2k}{M} \left(d_2 + \sum_{j=2}^k (j - 1)r_{2j}\right) + \frac{k}{M} \sum_{j=k+1}^{M/2} (N - 2j)r_{2j}.
 \end{aligned}$$

Algorithm 4 exploits them instead of (15) and (16).

In order to assess the performance of the seven algorithms, we have conducted preliminary numerical experiments, tracking their efficiency (execution times) and the quality of the output (numerical errors originating from rounding-off in the course of computations). Regarding a set of numbers $d_n = p_n e^{-\gamma_n n/M}$, where p_n and γ_n are pseudorandom

numbers uniformly distributed in the intervals $[-1, 1]$ and $[5, 8]$, respectively, as a sample test solution to the problem $Gd = r$, we compute the respective r.h.s., r , and analyse the discrepancies $q_n = \tilde{d}_n - d_n$ for the approximate solutions \tilde{d}_n obtained by each algorithm. This procedure has been performed for $M = 126$ (mimicking a poor resolution of the convective dynamo problem discretization), $M = 510$ (intermediate), and $M = 2046$ (high resolution) for 10^6 sets of d_n , comprising 3×10^6 test problems in total. The results are summarised in Tables 1 and 2, and Figures 2 and 3.

Table 1. Minimum and maximum error norms detected among solutions to the 10^6 test problems by the seven algorithms for determination of coefficients of a linear combination of the polynomials T_n^* .

M	Algorithm	$\ q\ _{\max}$		$\ q\ _{en}$	
		Min. Error	Max. Error	Min. Error	Max. Error
126	1	3.38×10^{-17}	3.26×10^{-15}	1.29×10^{-16}	1.73×10^{-14}
	1'	3.21×10^{-17}	3.44×10^{-15}	1.16×10^{-16}	1.70×10^{-14}
	2	5.97×10^{-16}	1.19×10^{-12}	3.17×10^{-15}	6.53×10^{-12}
	2'	3.12×10^{-17}	3.44×10^{-15}	1.48×10^{-16}	1.72×10^{-14}
	3	2.22×10^{-16}	3.82×10^{-14}	6.06×10^{-16}	1.01×10^{-13}
	3'	1.11×10^{-16}	3.61×10^{-15}	3.22×10^{-16}	1.62×10^{-14}
	4	8.33×10^{-17}	3.77×10^{-15}	3.34×10^{-16}	1.83×10^{-14}
510	1	2.78×10^{-14}	2.20×10^{-14}	1.75×10^{-15}	2.35×10^{-13}
	1'	2.43×10^{-14}	2.28×10^{-14}	1.69×10^{-15}	2.25×10^{-13}
	2	7.75×10^{-11}	3.47×10^{-11}	8.98×10^{-14}	3.63×10^{-10}
	2'	2.78×10^{-14}	2.03×10^{-14}	1.43×10^{-15}	2.14×10^{-13}
	3	1.42×10^{-13}	1.59×10^{-13}	8.70×10^{-15}	7.48×10^{-13}
	3'	1.11×10^{-14}	2.30×10^{-14}	5.52×10^{-15}	2.28×10^{-13}
	4	1.19×10^{-14}	2.36×10^{-14}	5.65×10^{-15}	2.24×10^{-13}
2046	1	2.22×10^{-15}	1.64×10^{-13}	2.74×10^{-14}	3.50×10^{-12}
	1'	2.35×10^{-15}	1.75×10^{-13}	2.63×10^{-14}	3.57×10^{-12}
	2	2.34×10^{-13}	1.16×10^{-9}	5.23×10^{-12}	2.39×10^{-8}
	2'	2.61×10^{-15}	1.72×10^{-13}	2.64×10^{-14}	3.32×10^{-12}
	3	1.43×10^{-14}	6.29×10^{-13}	1.53×10^{-13}	6.11×10^{-12}
	3'	9.24×10^{-15}	1.76×10^{-13}	8.31×10^{-14}	3.89×10^{-12}
	4	9.71×10^{-15}	1.77×10^{-13}	9.37×10^{-14}	3.51×10^{-12}

Table 2. Mean execution times (seconds) of 1000 applications of the seven algorithms for computing coefficients of a linear combination of T_n^* .

Algorithm	1	1'	2	2'
$M = 126$	1.32×10^{-3}	1.39×10^{-3}	8.94×10^{-4}	9.98×10^{-4}
$M = 510$	5.42×10^{-3}	5.36×10^{-3}	3.81×10^{-3}	3.83×10^{-3}
$M = 2046$	2.171×10^{-2}	2.162×10^{-2}	1.506×10^{-2}	1.542×10^{-2}
Algorithm	3	3'	4	
$M = 126$	1.45×10^{-3}	1.34×10^{-3}	1.90×10^{-3}	
$M = 510$	5.68×10^{-3}	5.26×10^{-3}	7.97×10^{-3}	
$M = 2046$	2.262×10^{-2}	2.109×10^{-2}	3.197×10^{-2}	

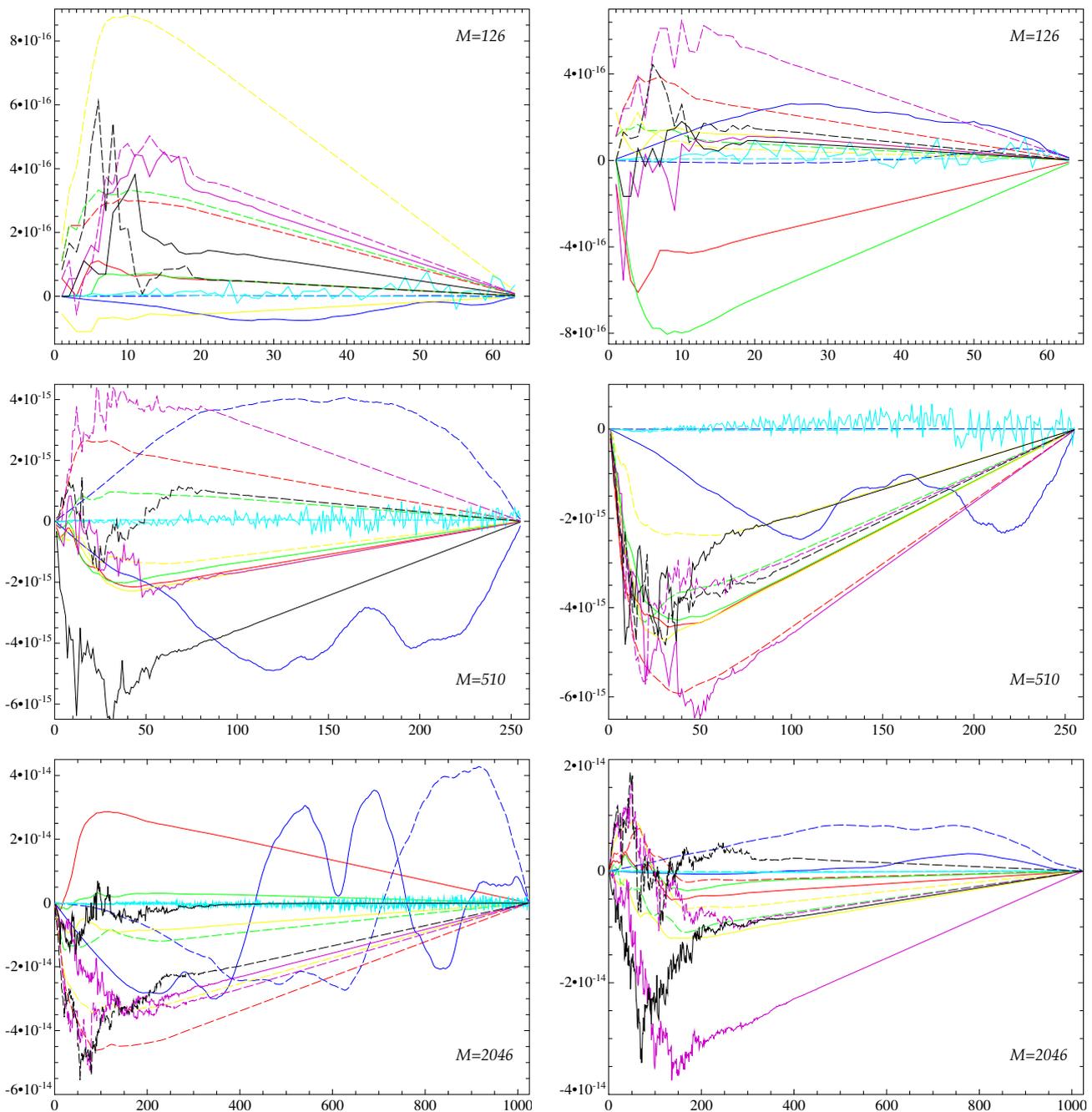


Figure 2. Errors (vertical axes) of approximate solutions obtained for sample test problems by algorithms 1 (red), 1' (green), 2 (blue), 2' (yellow), 3 (cyan), 3' (magenta) and 4 (black colour, respectively) for determining the coefficients of a linear combination of $T_n^*(x_3)$ vs. the coefficient index number (horizontal axes). The panels show the plots for six randomly chosen sample runs for the three values of the resolution parameter M . Continuous line: odd-index coefficients; dashed line: even-index coefficients. In order to fit the panels, the error values have been divided by 10^3 for algorithm 2 and by 10^2 for algorithm 3.

For each resolution parameter (the number M of the sought coefficients in the expansion at hand) and each algorithm, Table 1 presents the ranges (i.e., the maximum and minimum values) of the error norms for 10^6 approximate solutions to the test problems. Two norms are considered: the maximum norm $\|\mathbf{q}\|_{\max} = \max_n |q_n|$ and the so-called “energy” norm $\|\mathbf{q}\|_{\text{en}} = \sqrt{\sum_n |q_n|^2}$.

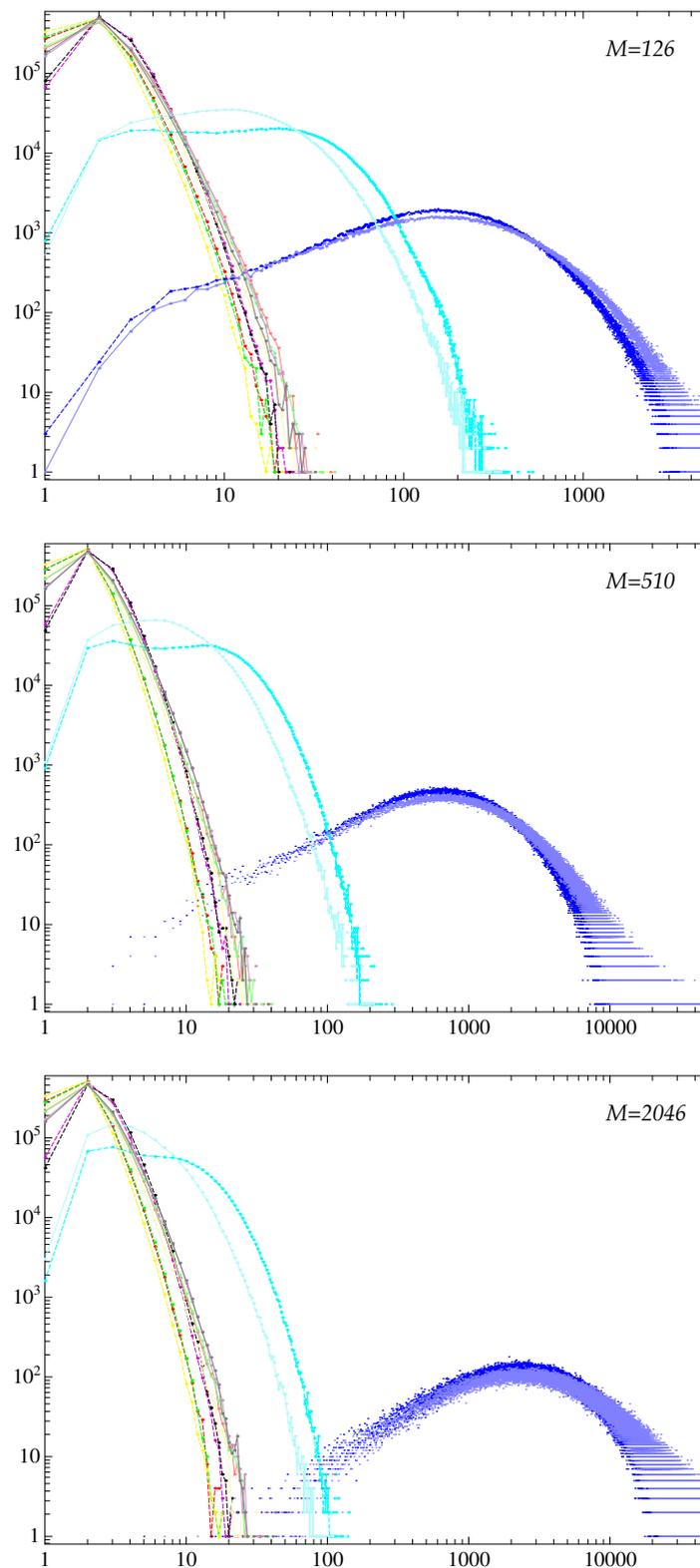


Figure 3. Distribution of error norms of approximate solutions obtained for 10^6 test problems by algorithms 1 (red), 1' (green), 2 (blue), 2' (yellow), 3 (cyan), 3' (magenta) and 4 (black colour, respectively) for determining the coefficients of a linear combination of $T_n^*(x_3)$. Horizontal axis: the error norms relative to the smallest error norm obtained for the test problem by the seven algorithms, m ; vertical axis: the number of cases, $N_m^{(k)}$, for the k th algorithm and each error norm. Dashed lines, saturated colours: the maximum error norm; continuous lines, diluted colours: the energy error norm. The panels show the plots for the three values of the resolution parameter M .

Table 2 illustrates the execution times for each algorithm. Since each individual computation of a sample solution is very fast, the individual durations of computations by the same algorithm for the same M significantly vary from run to run. To reduce the influence of this noise, we have employed the following procedure: for a sample problem, we measure the CPU time required for 1000 (identical) applications of the algorithm under consideration; we make 1000 measurements of this time (for test problems for the same resolution parameter M solved by the same algorithm); the average of the obtained numbers is reported in Table 2. These mean CPU execution times are reproduced to at least 3 significant digits for all the 21 test problems used for measuring the times; thus, the influence of noise in these values is reduced as desirable. In principle, algorithm 1 execution times should not differ from those of algorithm 1'; similarly for the algorithm pairs 2 and 2', and 3 and 3'. The nature of the variation of these values is related to the intimate details of application of the compiler optimisation techniques (in this experiment we have used the maximum optimisation). For algorithms 3 and 3' the variation also reflects the difference in the programming of computation of the sums involved aimed at enhancing the accuracy.

Figure 2 illustrates the error distributions among the approximate solutions \tilde{d}_n to the sample test problems. Plots for two test problems for each of the three considered values of M are shown. The problem splits into two independent subproblems for odd- and even-number variables d_k ; errors for odd-index coefficients are shown by continuous lines and for even-index coefficients by dashed lines. Graph colours indicate by which algorithms the curves have been obtained. We observe that algorithms 2 and 3 yield blatantly high errors; algorithms 2 and 2' are also imperfect in that they produce higher errors for higher-index unknowns \tilde{d}_n . The errors experience a wild spiky behaviour for low-index coefficients d_n , but, except for algorithm 2', it becomes more ordered roughly for $n > M/6$.

To assess how statistically relatively poor or good the considered algorithms are, we present in Figure 3 the distributions of error norms measured in the units of the smallest error norm for the given test problem. More precisely, the plots have been constructed by the following procedure: A set of 10^6 test problems with solutions, pseudorandomly generated as discussed above, have been solved by each of the seven algorithms. For the p th problem and the k th algorithm, the error norms $\|\mathbf{q}^{(p;k)}\|$ have been determined. We denote the smallest of these seven norms by $\omega^{(p)} = \min_k \|\mathbf{q}^{(p;k)}\|$ and define the quantities $N_m^{(k)}$ as the number of cases (among the considered 10^6 test problems), in which the error norm obtained by algorithm k relative to the smallest norm $\omega^{(p)}$ falls into the m th bin, or in other words, the ratio $\|\mathbf{q}^{(p;k)}\|/\omega^{(p)}$ satisfies the inequality $m - 1 < \|\mathbf{q}^{(p;k)}\|/\omega^{(p)} \leq m$. In particular, $N_1^{(k)}$ is the number of test problems, for which algorithm k delivers the smallest, over all the seven algorithms, norm, i.e., $\omega^{(p)} = \|\mathbf{q}^{(p;k)}\|$. These computations have been performed for both norms, the maximum norm $\|\mathbf{q}\|_{\max}$ and the energy norm $\|\mathbf{q}\|_{\text{en}}$.

For each algorithm and both error norms, in Figure 3 we show the quantities $N_m^{(k)}$ versus the bin number m . The data points for six algorithms are joined into plots up to the smallest m such that $N_m^{(k)} = 0$ (the use of the logarithmic scale for $N_m^{(k)}$ forces us to break the plot at this point). The outlier values $N_m^{(k)} > 0$ for larger m are shown as disjoint points; all these values are small. The data distribution for algorithm 2 has a different nature, more resembling a cloud; we therefore render the data as two plots (for odd- and even-index numbers) only for the small-dimensional problem ($M = 126$) for $N_m^{(3)} < 100$. The outliers $N_m^{(k)} > 0$ for $m > 5 \times 10^3$ for $M = 126$, and for $m > 5 \times 10^4$ for $M = 510$ and 2046 are ignored in Figure 3. There are 211 and 1529 such outliers for the odd- and even-index test problems, respectively, for algorithm 3 for $N = 128$, 18 for the even-index problems for $M = 510$, and 2265 and 8831 for the odd- and even-index test problems for $M = 2046$. These statistics demonstrate that the larger the number of unknown coefficients M , the more the output of algorithm 2 is affected by the numerical noise.

The numerical results lead us to the following conclusions:

- i. Except for algorithm 2, in applications for solving the 3×10^6 test problems, for any M and for both error norms each algorithm delivers approximate solutions \tilde{d}_n of any possible accuracy from the best to the worst one, i.e., for each algorithm, solutions to some test problems obtained by this algorithm have the smallest, the largest, or any differently placed intermediate error norm among the seven error norms obtained for this problem.
- ii. Algorithm 2 yields solutions of blantly poor accuracy. For $M = 126$, its output still includes solutions occupying any possible place between the best-accuracy and worst-accuracy solutions. However, for $M = 126$, it yields just three most accurate solutions in terms of the maximum error norm and just one, if the energy error norm is used. These numbers gradually increase via 46 and 15 penultimately worst-accuracy solutions to 999,934 and 999,975 least accurate solutions. By contrast, for $M = 510$ and 2046 all its outputs are the least accurate solutions for both error norms, except for one case of penultimately worst-accuracy solutions for the maximum norm.
- iii. The error norms (see Table 1) are compatible with the standard “double” (in the Fortran speak; 64 bit words) computer precision (the 10^{-15} “machine epsilon”). The poor performance of algorithm 2 and the second-worst (accuracy-wise) algorithm 3 stems from the involvement in the expressions used in their formulation of the products of d_1 and d_2 with the numbers proportional to the indices of the unknown coefficients that increase up to the large numbers M .
- iv. We can regard the accuracy results at a different angle. In the worst-case scenario, the norm of a discrepancy in the r.h.s. of the equation $G\mathbf{d} = \mathbf{r}$ is multiplied by the condition number of the matrix G , which is defined as the product of norms of the matrices G and G^{-1} . The former norm is obviously order 1; to estimate the latter, we use (14) and (15) for the r.h.s. where all $r_k = 1$, and find $\|G^{-1}\| \geq \|\mathbf{d}\|_{\text{en}} / \|\mathbf{r}\|_{\text{en}} = O(M^2)$. Hence, the condition number of G is (at least) order M^2 , which is compatible with the accuracy results for algorithm 2 (see maximum errors in Table 1). We observe that numerical errors are amplified by an algorithm-specific effective condition number [35], which can be much smaller than the worst-case theoretical condition number, provided a specialised algorithm takes into account particular properties of the problem,
- v. For all the three M values used, most frequently the smallest-error approximate solutions \tilde{d}_n are provided by algorithm 2' for whichever error norm: 315,174 and 236,707 times out of 10^6 for $M = 126$, 325,841 and 236,544 for $M = 510$, and 333,062 and 239,009 for $M = 2046$ (the first number in a pair is for the maximum error norm and the second one for the energy norm). Accuracy-wise, the shuttle algorithms 1 and 1' are mutually close and not significantly inferior to algorithm 2'.
- vi. The quantities $N_m^{(k)}$ have the maxima at $k = 2$ (see Figure 3), i.e., for each algorithm and each M , the error norms fall into the bin $1 < \|\mathbf{q}^{(p;k)}\| / \omega^{(p)} \leq 2$ with the highest probability.
- vii. Solutions \tilde{d}_n obtained by all algorithms except for two and three have significantly larger errors q_n for small n (say, for $n < M/6$) than for larger n (see Figure 3). By contrast, algorithms 2 and 3 yield maximum errors for intermediate and high n . In addition, the errors in solutions computed by algorithm 3 wildly oscillate, which is not typical for the behaviour of errors generated by the other six algorithms. Consequently, algorithms 2 and 3 yield approximate solutions in which \tilde{d}_n for intermediate and high n have exceptionally high relative errors.
- viii. Algorithms 2 and 2' are significantly faster than the other five algorithms (see Table 2). Their execution CPU times are mutually close and differ much more from the execution times registered for the other ones.

Properties iii and vii render algorithms 2 and 3 inapplicable. None of the seven algorithms is “perfect”, but the compromise between the highest efficiency and accuracy reveals the optimality of algorithm 2'.

3.2. Algorithms for Determining the Coefficients of a Linear Combination of $T_n^{**}(x_3)$

The equation for the poloidal potential is obtained by applying twice the curl to (1a) and taking the vertical component of the result:

$$\frac{\partial \nabla_h^2 \nabla^2 P^v}{\partial t} = P \nabla_h^2 (\nabla^2)^2 P^v + F^{pv},$$

$$F^{pv} = -\sigma \nabla_h^2 \theta + \mathbf{e}_3 \cdot (\tau(\mathbf{q} \cdot \nabla) \nabla \times \mathbf{v} - (\nabla \times)^2 (\mathbf{v} \times (\nabla \times \mathbf{v}) - \mathbf{b} \times (\nabla \times \mathbf{b}))).$$

The above equation is equivalent to the parabolic equation

$$\frac{\partial \nabla^2 P^v}{\partial t} = P (\nabla^2)^2 P^v + (\nabla_h^2)^{-1} F^{pv}. \tag{17}$$

To apply the inverse Laplacian $(\nabla^2)^{-1}$ is now tempting but unfeasible, since the conditions (6b) do not imply suitable boundary conditions for $\nabla^2 P^v$. By (9), the polynomials

$$T_n^{**}(x_3) = (n + 1)(T_{n+4}(x_3) - T_{n+2}(x_3)) - (n + 3)(T_{n+2}(x_3) - T_n(x_3)), \quad n \geq 0, \tag{18}$$

satisfy (6b), and the poloidal potentials can be approximated by finite series

$$P^v(\mathbf{x}, t) = \sum_{|n_1| \leq N, |n_2| \leq N, 0 \leq n \leq N-5} p_n^v(t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} T_n^{**}(x_3),$$

where $\mathbf{n} = (n_1, n_2, n)$.

Substituting the series into (17), we obtain an equation, whose l.h.s. is of the form

$$\sum_{|n_1| \leq N, |n_2| \leq N, 0 \leq n \leq N-1} \frac{dp_n^v}{dt} e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} \tilde{p}_n(x_3),$$

where \tilde{p}_n is a polynomial of degree $n + 4$. As usual, we apply the Fourier transform in the horizontal directions to deduce the equation for each pair of indices n_1 and n_2 , and orthogonally project \tilde{p}_n on the polynomials $T_m^{**}(x_3)$ with the weight $(1 - x_3^2)^{-1/2}$. Computing dp_n^v/dt requires inverting the matrices acting on the columnar vectors consisting of the time derivatives dp_n^v/dt for $0 \leq n \leq N - 1$ and n_1, n_2 fixed. As in the case of the toroidal potential of the flow, these are band matrices. To see this, we integrate twice by parts the integral $\int_{-1}^1 d^2 T_n^{**}/dx_3^2 (1 - x_3^2)^{-1/2} T_m^{**} dx_3$ encountered in the projection of the l.h.s. of (17) (noting that the test polynomial T_m^{**} satisfies the boundary conditions (6b)) and apply the relation

$$\frac{d^2}{dx_3^2} ((1 - x_3^2)^{-1/2} T_m^{**}) = 4(m + 1)(m + 2)(m + 3)(1 - x_3^2)^{-1/2} T_{m+2}. \tag{19}$$

It can be proven by using the identities

$$2(x_3^2 - 1) \frac{dT_m}{dx_3} = m(T_{m+1} - T_{m-1}), \quad \frac{d}{dx_3} \left((1 - x_3^2)^{1/2} \frac{dT_m}{dx_3} \right) = -m^2(1 - x_3^2)^{-1/2} T_m.$$

Relation (19) is also useful for computing

$$\begin{aligned} & \int_{-1}^1 (\nabla^2)^2 \left(e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} T_n^{**} \right) (1 - x_3^2)^{-1/2} T_m^{**} dx_3 \\ &= e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} \left[\frac{\pi \Gamma^4}{2} \left((m + 3)(m - 3) \delta_{n+4}^m - 2((m + 3)m + (m + 2)(m - 1)) \delta_{n+2}^m \right. \right. \\ & \quad \left. \left. + ((m + 1)^2 + (2m + 4)^2 + (m + 3)^2 (1 + \delta_0^m)) \delta_n^m \right. \right. \\ & \quad \left. \left. - 2((m + 1)(m + 4) + (m + 2)(m + 5)) \delta_{n-2}^m + (m + 1)(m + 7) \delta_{n-4}^m \right) \right. \\ & \quad \left. + 4(m + 1)(m + 2)(m + 3) \left(-\pi \Gamma^2 ((m - 1) \delta_{n+2}^m - 2(m + 2) \delta_n^m + (m + 5) \delta_{n-2}^m) \right) \right. \\ & \quad \left. + \int_{-1}^1 \frac{d^2 T_n^{**}}{dx_3^2} (1 - x_3^2)^{-1/2} T_{m+2} dx_3 \right], \end{aligned}$$

where $\Gamma = (\alpha_1^2 n_1^2 + \alpha_2^2 n_2^2)^{1/2}$ and δ_n^m is the Kronecker symbol. Differentiation in x_3 can now be performed ([9], see also [36]) using the recurrence relation

$$\frac{d}{dx_3} T_{k+1} = \frac{k+1}{k-1} \frac{d}{dx_3} T_{k-1} + 2(k+1)T_k.$$

Thus, the vector \mathbf{d} of the time derivatives of the expansion coefficients is a solution to the linear system of equations of the form

$$G\mathbf{d} = \mathbf{r}, \tag{20a}$$

where \mathbf{r} is the vector (of length $M' = N - 4$) of the dot products of the r.h.s. of (17) with T_m^{**} ; both G and \mathbf{r} are divided by $\pi L_1 L_2 \Gamma^2 / 2$. The matrix has non-zero entries on the diagonal:

$$G_{k,k} = -6k^2 - 12k - 8 - 9\delta_1^k - 2\Gamma'_k(k+1) \tag{20b}$$

and on four subdiagonals:

$$G_{k,k-4} = -(k-4)(k+2), \quad G_{k,k-2} = 4k^2 - 8 + \Gamma'_k(k-2), \tag{20c}$$

$$G_{k,k+2} = 4(k+2)^2 - 8 + \Gamma'_k(k+4), \quad G_{k,k+4} = -k(k+6), \tag{20d}$$

where

$$\Gamma'_k = 4k(k+1)(k+2)/\Gamma^2. \tag{20e}$$

Algorithm 1 is the standard shuttle algorithm for a pentadiagonal matrix [37]. We note that the odd-index subproblem is separated from the even-index one and use the ansatz

$$d_n = A_n d_{n+2} + B_n d_{n+4} + C_n. \tag{21}$$

The first four equations imply

$$\begin{aligned} A_1 &= -G_{1,3}/G_{1,1}, & B_1 &= -G_{1,5}/G_{1,1}, & C_1 &= r_1/G_{1,1}, \\ A_2 &= -G_{2,4}/G_{2,2}, & B_2 &= -G_{2,6}/G_{2,2}, & C_2 &= r_2/G_{2,2}, \\ A_3 &= -\frac{B_1 G_{3,1} + G_{3,5}}{A_1 G_{3,1} + G_{3,3}}, & B_3 &= -\frac{G_{3,7}}{A_1 G_{3,1} + G_{3,3}}, & C_3 &= \frac{r_3 - G_{3,1} C_1}{A_1 G_{3,1} + G_{3,3}}, \\ A_4 &= -\frac{B_2 G_{4,2} + G_{4,6}}{A_2 G_{4,2} + G_{4,4}}, & B_4 &= -\frac{G_{4,8}}{A_2 G_{4,2} + G_{4,4}}, & C_4 &= \frac{r_4 - G_{4,2} C_2}{A_2 G_{4,2} + G_{4,4}}. \end{aligned}$$

The $M - 4$ equations for $5 \leq n \leq M$ yield the recurrence relations

$$\begin{aligned} A_n &= -\frac{(G_{n,n-4} A_{n-4} + G_{n,n-2}) B_{n-2} + G_{n,n+2}}{(G_{n,n-4} A_{n-4} + G_{n,n-2}) A_{n-2} + G_{n,n-4} B_{n-4} + G_{n,n}}, \\ B_n &= -\frac{G_{n,n+4}}{(G_{n,n-4} A_{n-4} + G_{n,n-2}) A_{n-2} + G_{n,n-4} B_{n-4} + G_{n,n}}, \\ C_n &= \frac{r_n - (G_{n,n-4} A_{n-4} + G_{n,n-2}) C_{n-2} - G_{n,n-4} C_{n-4}}{(G_{n,n-4} A_{n-4} + G_{n,n-2}) A_{n-2} + G_{n,n-4} B_{n-4} + G_{n,n}}, \end{aligned}$$

where it is assumed $G_{n,m} = 0$ for $m > M$. This yields $d_n = C_n$ for $n = M - 1$ and M , and we use the recurrence relation (21) to compute sequentially all the unknowns d_n for decreasing n . (Note $B_n = 0$ for $n = M - 1$ and M .)

Algorithm 1' is the same shuttle procedure applied to the system of equations with the reverse numbering of equations and variables.

Algorithms 2 and 2' are modifications of the shuttle algorithm based on the observation that the sum of coefficients in the equation number k for $4 < k < M - 3$ is zero. Hence,

in new variables $d'_k = d_k - d_{k-2}$, $k \geq 3$, the system reduces to a problem with a matrix G' involving three subdiagonals for $4 < k < M - 3$:

$$G'_{k,k-2} = (k - 1)^2 - 9, \quad G'_{k,k} = -3k^2 - 2k - \Gamma'_k(k - 2),$$

$$G'_{k,k+2} = 3k^2 + 10k + 8 + \Gamma'_k(k + 4), \quad G'_{k,k+4} = -k(k + 6).$$

The problem splits into two independent subproblems for odd- and even-number variables d_k . We formulate algorithm 2 for solving the subproblem for even-number variables; it is straightforward to reformulate it for the subproblem for odd-number variables. We introduce $M' + 1 = M/2 + 1$ variables $z_1 = d_2$, $z_k = d_{2k} - d_{2k-2}$ for $2 \leq k \leq M'$ and $z_{M'+1} = d_{2M}$. They satisfy M' original equations and the gauge equation

$$z_{M'+1} - \sum_{k=1}^{M'} z_k = 0. \tag{22}$$

The first equation is equivalent to $z_1 = A_1 z_2 + B_1 z_3 + C_1$. Equations for $2 \leq k \leq M'$ take the form

$$G'_{2k,2k-2} z_{k-1} + G'_{2k,2k} z_k + G'_{2k,2k+2} z_{k+1} + G'_{2k,2k+4} z_{k+2} = r_{2k},$$

whereby

$$z_k = A_k z_{k+1} + B_k z_{k+2} + C_k, \tag{23}$$

$$A_k = -\frac{G'_{2k,2k+2} + B_{k-1} G'_{2k,2k-2}}{G'_{2k,2k} + A_{k-1} G'_{2k,2k-2}}, \quad B_k = \frac{-G'_{2k,2k+4}}{G'_{2k,2k} + A_{k-1} G'_{2k,2k-2}}, \quad C_k = \frac{r_{2k} - C_{k-1} G'_{2k,2k-2}}{G'_{2k,2k} + A_{k-1} G'_{2k,2k-2}},$$

which defines the first, “direct run of the shuttle”. Here, $z_{M'+2} = 0$. For $k = M'$, the relation (23) reduces to just

$$z_k = D_k z_{M'+1} + E_k, \tag{24}$$

where $D_{M'} = A_{M'}$ and $E_{M'} = C_{M'}$. We can now use (23) for $k = M' - 1, \dots, 1$ to obtain the coefficients in (24) recursively,

$$D_k = A_k D_{k+1} + B_k D_{k+2}, \quad E_k = A_k E_{k+1} + B_k E_{k+2} + C_k$$

(the second, “reverse run of the shuttle”). Simultaneously, we compute the coefficients F_k and R_k in the partial sums

$$z_{M'+1} - \sum_{n=k}^{M'} z_n = F_k z_{M'+1} + R_k.$$

By (22), we then find $z_{M'+1} = -R_1 / F_1$. Now, the third “run of the shuttle” yields

$$d_{2n} = \sum_{k=1}^n z_k = z_{M'+1} - \sum_{k=n+1}^{M'} z_k - \left(z_{M'+1} - \sum_{k=1}^{M'} z_k \right) = F_{n+1} z_{M'+1} + R_{n+1}. \tag{25}$$

This algorithm is apparently more involved than the standard shuttle algorithm. However, since it works with the differences $z_k = d_{2k} - d_{2k-2}$, it may be advantageous in yielding more accurate values of d_{2k} .

Algorithm 2' amounts to the same computational procedure applied to the system of equations, where the numbering of equations and variables is reversed. Thus, we obtain the relations, similar to (25), in which all d_{2n} are expressed in terms of z_1 instead of $z_{M'+1}$. Normally, the coefficients d_k tend to zero, whereby z_1 is a priori much larger than $z_{M'+1}$; for a given precision of computations (in our case, the standard real*8 “double” precision of the floating point arithmetics), this is a tighter constraint on the number of correct digits after the decimal point. Consequently, we may expect algorithm 2' to yield less accurate values of d_k than algorithm 2.

We have investigated the performance of the four algorithms following the same approach as for the algorithms for determining the coefficients of linear combinations of the polynomials T_n^* : we have synthesized 10^6 pseudorandom sample test solutions d_n by the same procedure, outlined in Section 3.1, as for the polynomials T_n^* , and analysed the errors $q_n = \tilde{d}_n - d_n$ for the approximate solutions obtained by each of the four algorithms for $M = 124, 508$ and 2044 terms in linear combinations of T_n^{**} .

For each resolution parameter M and each algorithm, Table 3 shows the ranges (i.e., the maximum and minimum values) of the maximum and energy norms of the approximate solution errors. Table 4 illustrates the execution times for each algorithm, measured by the same procedure (see Section 3.1) as for the polynomials T_n^* . The obtained mean execution times are again accurate to at least three significant digits. We expect the algorithm 1 execution times to coincide with those for algorithm 1', similarly for algorithms 2 and 2'. However, the times presented in Table 4 for algorithms 1 and 2 are slightly smaller than those for algorithms 1' and 2', respectively. This difference is insignificant, it just reflects the way we have programmed the latter algorithms: in our implementation, the orders of the equations and unknown variables are reversed at run time, which can be avoided by a more careful programming of the algorithms using index reversion.

Table 3. Minimum and maximum error norms detected among solutions to 10^6 test problems (20a) by the four algorithms.

M	Algorithm	$\ \mathbf{q}\ _{\max}$		$\ \mathbf{q}\ _{\text{en}}$	
		Min. Error	Max. Error	Min. Error	Max. Error
124	1	4.51×10^{-17}	3.22×10^{-15}	1.10×10^{-16}	8.47×10^{-15}
	1'	5.55×10^{-17}	2.83×10^{-15}	1.49×10^{-16}	7.83×10^{-15}
	2	5.55×10^{-17}	3.55×10^{-15}	1.42×10^{-16}	1.02×10^{-14}
	2'	8.33×10^{-17}	4.22×10^{-15}	2.68×10^{-16}	1.54×10^{-14}
508	1	4.30×10^{-16}	1.95×10^{-14}	1.81×10^{-15}	1.12×10^{-13}
	1'	2.78×10^{-16}	2.03×10^{-14}	1.98×10^{-15}	1.10×10^{-13}
	2	3.61×10^{-16}	2.21×10^{-14}	2.19×10^{-15}	1.29×10^{-13}
	2'	5.00×10^{-16}	2.07×10^{-14}	2.47×10^{-15}	1.27×10^{-13}
2044	1	2.78×10^{-15}	1.35×10^{-13}	2.53×10^{-14}	1.69×10^{-12}
	1'	2.53×10^{-15}	1.33×10^{-13}	2.95×10^{-14}	1.62×10^{-12}
	2	2.89×10^{-15}	2.55×10^{-13}	3.61×10^{-14}	2.10×10^{-12}
	2'	3.16×10^{-15}	1.53×10^{-13}	3.32×10^{-14}	1.88×10^{-12}

Table 4. Mean execution times (seconds) of 1000 applications of the four algorithms for computing coefficients of a linear combination of T_n^{**} .

Algorithm	1	1'	2	2'
$M = 124$	1.41×10^{-3}	1.51×10^{-3}	1.43×10^{-3}	1.51×10^{-3}
$M = 508$	6.04×10^{-3}	6.30×10^{-3}	5.88×10^{-3}	6.18×10^{-3}
$M = 2044$	2.506×10^{-2}	2.632×10^{-2}	2.464×10^{-2}	2.595×10^{-2}

Figure 4 illustrates the error distributions among the unknowns detected for solutions to six sample test problems (plots for two problems for all the considered $M = 124, 508$ and 2044 are shown). Figure 5 presents the distributions of error norms measured by the same procedure, discussed in Section 3.1, as for the polynomials T_n^* . For the p th problem and the k th algorithm, we find the error norms $\|\mathbf{q}^{(p;k)}\|$ and the smallest norm for this problem, $\omega^{(p)} = \min_k \|\mathbf{q}^{(p;k)}\|$. We again denote by $N_m^{(k)}$ the number of cases, among the 10^6 test problems, in which the error norm produced by algorithm k , measured in the units of $\omega^{(p)}$,

falls into the m th bin, i.e., m satisfies the inequality $m - 1 < \|\mathbf{q}^{(p;k)}\|/\omega^{(p)} \leq m$. For each algorithm and both error norms (the maximum norm and the energy norm), Figure 5 shows the histogram $N_m^{(k)}$ versus the bin number m . The data points are joined up to the smallest m such that $N_m^{(k)} = 0$, and the outlier values $N_m^{(k)} > 0$ (always small) for larger m are shown as disjoint points; unlike in Figure 3, all outliers are shown.

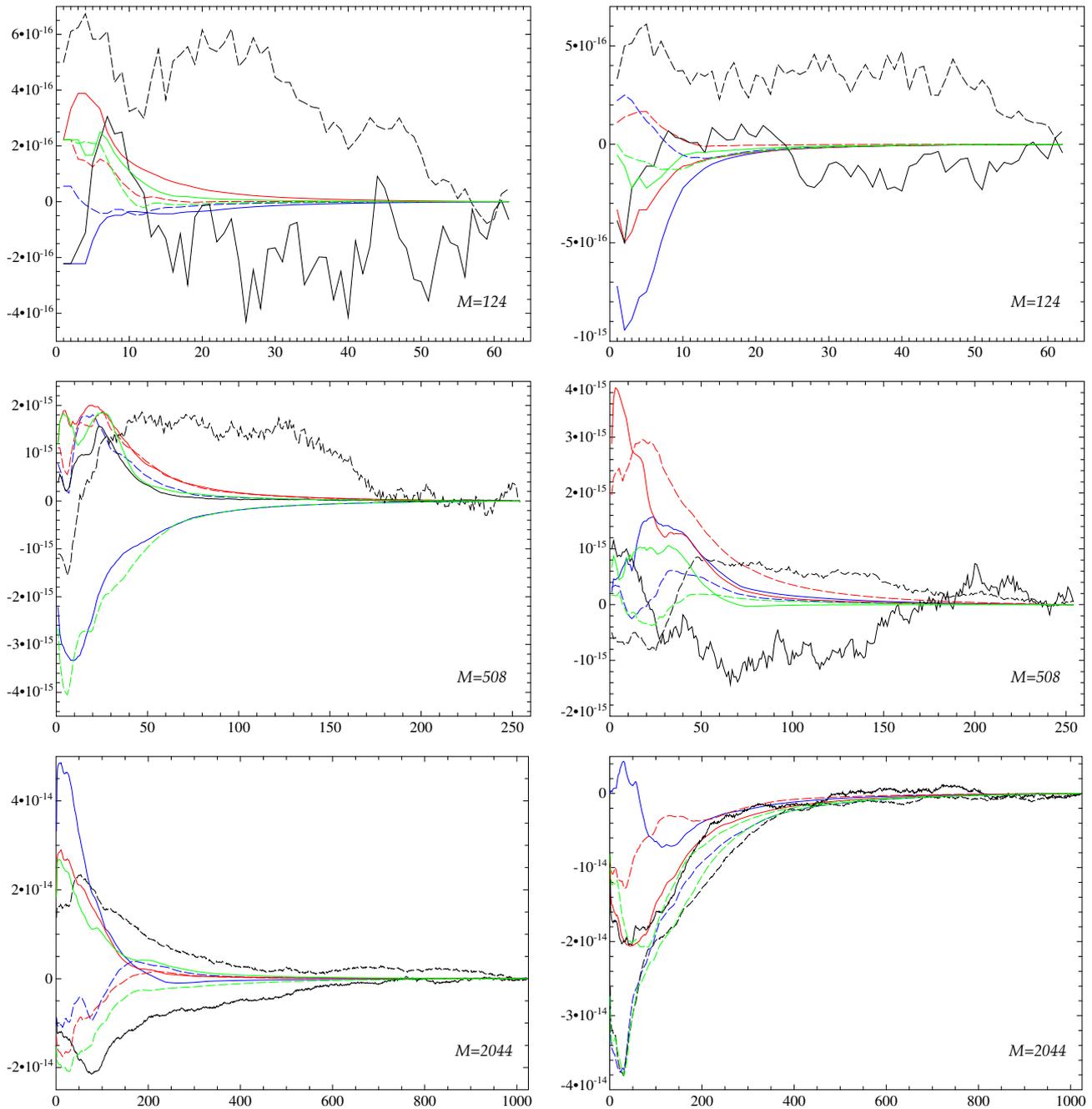


Figure 4. Errors (vertical axes) of approximate solutions obtained for sample test problems by algorithms 1 (red), 1' (green), 2 (blue) and 2' (black colour, respectively) for determining the coefficients of a linear combination of $T_n^{**}(x_3)$ vs. the coefficient index number (horizontal axes). The panels show the plots for six randomly chosen sample runs for the three values of the resolution parameter M . Continuous line: odd-index coefficients; dashed line: even-index coefficients.

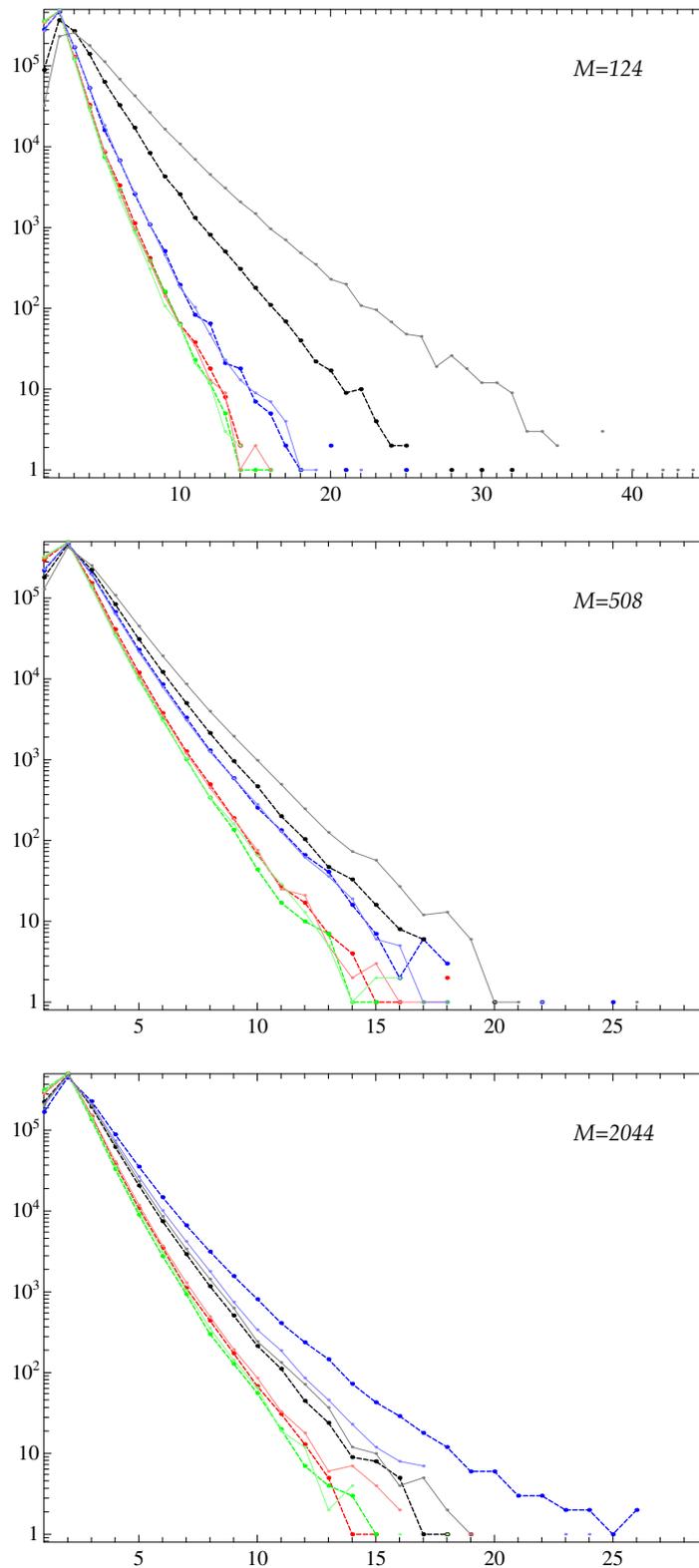


Figure 5. Distribution of error norms of approximate solutions obtained for 10^6 sample test problems by algorithms 1 (red), 1' (green), 2 (blue) and 2' (black colour, respectively) for determining the coefficients of a linear combination of $T_n^{**}(x_3)$. Horizontal axis: the error norms relative to the smallest error norm obtained for the test problem by the four algorithms, m ; vertical axis: the number of cases, $N_m^{(k)}$, for the k th algorithm and each error norm. Dashed lines, saturated colours: the maximum error norm; thin continuous lines, diluted colours: the energy error norm. The panels show the plots for the three values of the resolution parameter M .

The results reveal the following properties of the algorithms:

- i. None of the four algorithms is “perfect”: for all considered M and for both error norms, when solving the 3×10^6 test problems each algorithm delivers solutions of any possible relative accuracy from the best to the worst one, in the sense that solutions to some test problems obtained by any algorithm have the smallest, the largest or any intermediate error norms among the four error norms produced by the four algorithms for this problem.
- ii. The quantities $N_m^{(k)}$ have the maxima at $m = 2$ (see Figure 5), i.e., for each algorithm the error norms fall into the second bin, $1 < \|\mathbf{q}^{(p;k)}\|/\omega^{(p)} \leq 2$, with the highest probability.
- iii. For any M and for both error norms, algorithm 1' delivers the best accuracy solutions (has the maximum $N_1^{(k)}$ over k) more often than the other three. For $M = 124$ and 508 , the least accurate solutions are obtained most frequently by algorithm 2' and for $M = 2044$ by algorithm 2.
- iv. The errors q_n for intermediate and high n (say, for $n > M/6$) in approximate solutions computed by algorithm 2' are significantly larger than in solutions obtained by any of the other three algorithms.
- v. The execution CPU times of algorithms 1 and 1' are slightly larger than those for algorithms 2 and 2'.

Property iv implies that algorithm 2' yields approximate solutions, where \tilde{d}_n for intermediate and high n are polluted by high relative errors, rendering this algorithm inapplicable. The smallest execution time and the highest frequency of yielding the smallest errors distinguish algorithm 1', but algorithms 2 and 2' follow closely.

4. Discretisation of the Magnetic Field

Here we consider the implications of the boundary conditions (2b) for the magnetic field. Since the magnetic field is solenoidal (1d), it can be decomposed into the sum of the toroidal, $\mathbf{T}^b(\mathbf{x}, t)$, poloidal, $\mathbf{P}^b(\mathbf{x}, t)$, and mean-field, $\mathbf{M}^b(\mathbf{x}, t)$, components:

$$\mathbf{b}(\mathbf{x}, t) = \mathbf{T}^b(\mathbf{x}, t) + \mathbf{P}^b(\mathbf{x}, t) + \mathbf{M}^b(x_3, t), \tag{26a}$$

$$\mathbf{T}^b(\mathbf{x}, t) = \nabla \times (T^b(\mathbf{x}, t)\mathbf{e}_3), \tag{26b}$$

$$\mathbf{P}^b(\mathbf{x}, t) = \nabla \times \nabla \times (P^b(\mathbf{x}, t)\mathbf{e}_3), \tag{26c}$$

$$\mathbf{M}^b(x_3, t) = \left\langle (b^1(\mathbf{x}, t), b^2(\mathbf{x}, t), 0) \right\rangle_h \tag{26d}$$

such that

$$\left\langle T^b \right\rangle_h = \left\langle P^b \right\rangle_h = 0.$$

In view of the boundary conditions (2b) and (2c) for the magnetic field, the potentials and the mean-field component satisfy the boundary conditions

$$T^b \Big|_{x_3=1} = \frac{\partial T^b}{\partial x_3} \Big|_{x_3=-1} = 0, \tag{27a}$$

$$\nabla P^b \Big|_{x_3=1} = \nabla h \Big|_{x_3=1'} \tag{27b}$$

$$\frac{\partial^2 P^b}{\partial x_3^2} \Big|_{x_3=-1} = P^b \Big|_{x_3=-1} = 0, \tag{27c}$$

$$\mathbf{M}^b \Big|_{x_3=1} = \frac{\partial \mathbf{M}^b}{\partial x_3} \Big|_{x_3=-1} = 0. \tag{27d}$$

The poloidal potential is determined from the vertical component of (1b), the toroidal one from the vertical component of the curl of (1b),

$$\frac{\partial T^b}{\partial t} = \eta \nabla^2 T^b + (\nabla_h^2)^{-1} F^{tb}, \quad F^{tb} = \mathbf{e}_3 \cdot (\nabla \times)^2 (\mathbf{v} \times \mathbf{b}), \tag{28}$$

and the mean-field component of the magnetic field from the equation derived by averaging over horizontal variables of the horizontal component of (1b).

The harmonic function h referred to in (27a) is defined by (2b). The boundary condition (27b) can be further detailed for individual harmonics in horizontal variables for the poloidal potential expanded as the Fourier series

$$P^b(\mathbf{x}, t) = \sum_{|n_1| \leq N, |n_2| \leq N} P_{n_1, n_2}^b(x_3, t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)}$$

$$\Rightarrow \nabla P^b = \sum_{|n_1| \leq N, |n_2| \leq N} \left(i\alpha_1 n_1 P_{n_1, n_2}^b, i\alpha_2 n_2 P_{n_1, n_2}^b, \frac{\partial P_{n_1, n_2}^b}{\partial x_3} \right) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)}.$$

At the upper boundary $x_3 = 1$, the horizontal components of the gradient of P_{n_1, n_2}^b coincide with the gradient of the harmonic function

$$h(\mathbf{x}, t) = P_{n_1, n_2}^b(1, t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2) - (\alpha_1^2 n_1^2 + \alpha_2^2 n_2^2)^{1/2} (x_3 - 1)}$$

and therefore (27b) is satisfied as long as

$$\frac{\partial P_{n_1, n_2}^b}{\partial x_3} \Big|_{x_3=1} = -(\alpha_1^2 n_1^2 + \alpha_2^2 n_2^2)^{1/2} P_{n_1, n_2}^b \Big|_{x_3=1}. \tag{29}$$

In view of (9), the boundary condition (27c) is satisfied by the polynomials

$$\Pi_n = (n + 1)(2n + 1)T_{n-1} + (4n^2 + 2)T_n + (n - 1)(2n - 1)T_{n+1} \text{ for } n \geq 0$$

and both conditions, (27c) and (29), are satisfied by the polynomials

$$T_n^{***}(x_3) = ((4(n + 1)^4 - 2(n + 1)^2 + 1 + \Gamma(4(n + 1)^2 + 2))\Pi_n - (4n^4 - 2n^2 + 1 + \Gamma(4n^2 + 2))\Pi_{n+1}) / (2n + 1)$$

$$= \kappa_{n+1}T_{n-1} + (\kappa_n + \lambda_n)T_n - (\kappa_{n+1} - \lambda_n)T_{n+1} - \kappa_n T_{n+2}$$
(30)

for $n \geq 1$, where

$$\kappa_n = n(4n^4 - 2n^2 + 1 + \Gamma(4n^2 + 2)), \quad \lambda_n = 16n(n + 1)(n(n + 1) + 1).$$

Thus, $P_{n_1, n_2}^b(x_3, t)$ can be expanded as

$$P^b(\mathbf{x}, t) = \sum_{|n_1| \leq N, |n_2| \leq N, 1 \leq n \leq N-2} p_n^b(t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} T_n^{***}(x_3).$$

The vector \mathbf{d} of the time derivatives of the coefficients of the expansion is a solution to the linear system of equations of the form $G\mathbf{d} = \mathbf{r}$. Here G is the heptadiagonal Gram matrix for the basic functions T_m^{***} and \mathbf{r} is the vector (of length $M'' = N - 3$) of dot products of the r.h.s. with T_m^{***} . Unlike in the two linear problems considered above, there is no separation into odd- and even-index subproblems.

The shuttle process can again be applied for solving this problem. We use the ansatz

$$d_n = A_n d_{n+1} + B_n d_{n+2} + C_n d_{n+3} + D_n. \tag{31}$$

The first equation is equivalent to

$$A_1 = -G_{1,2}/G_{1,1}, \quad B_1 = -G_{1,3}/G_{1,1}, \quad C_1 = -G_{1,4}/G_{1,1}, \quad D_1 = r_1/G_{1,1}.$$

The second equation implies

$$A_2 = -(G_{2,1}B_1 + G_{2,3})/A'_2, \quad B_2 = -(G_{2,1}C_1 + G_{2,4})/A'_2, \\ C_2 = -G_{2,5}/A'_2, \quad D_2 = (r_2 - G_{2,1}D_1)/A'_2,$$

where

$$A'_2 = G_{2,1}A_1 + G_{2,2}.$$

From the third equation,

$$A_3 = -(A'_3B_2 + G_{3,1}C_1 + G_{3,4})/A''_3, \quad B_3 = -(A'_3C_2 + G_{3,5})/A''_3, \quad C_3 = -G_{3,6}/A''_3, \\ D_3 = (r_3 - G_{3,1}D_1 - A'_3D_2)/A''_3,$$

where

$$A'_3 = G_{3,1}A_1 + G_{3,2}, \quad A''_3 = A'_3A_2 + G_{3,1}B_1 + G_{3,3}.$$

For $n \geq 4$, it is convenient to introduce the additional variables

$$A'_n = G_{n,n-3}A_{n-3} + G_{n,n-2}, \quad A''_n = A'_nA_{n-2} + G_{n,n-3}B_{n-3} + G_{n,n-1}, \\ A'''_n = A''_nA_{n-1} + A'_nB_{n-2} + G_{n,n-3}C_{n-3} + G_{n,n}.$$

For $4 \leq n \leq M'' - 3$, in the n th equation we use (21) to make consecutive substitutions of d_{n-2} , d_{n-1} and d_n . This yields the recurrence relations

$$A_n = -(A''_nB_{n-1} + A'_nC_{n-2} + G_{n,n+1})/A'''_n, \quad B_n = -(A''_nC_{n-1} + G_{n,n+2})/A'''_n, \\ C_n = -G_{n,n+3}/A'''_n, \quad D_n = (r_n - G_{n,n-3}D_{n-3} - A'_nD_{n-2} - A''_nD_{n-1})/A'''_n.$$

Finally, the remaining equations for $M'' - 2 \leq n \leq M''$ yield a system of three equations in $d_{M''-2}$, $d_{M''-1}$ and $d_{M''}$. The equations are:

$$(A''_nA_{n-1} + A'_nB_{n-2} + G_{n,n-3}C_{n-3} + G_{n,n})d_{M''-2} + (A''_nB_{n-1} + A'_nC_{n-2} + G_{n,n+1})d_{M''-1} \\ + (A''_nC_{n-1} + G_{n,n+2})d_{M''} = r_{M''-2} - A''_nD_{n-1} - A'_nD_{n-2} - G_{n,n-3}D_{n-3}$$

for $n = M'' - 2$;

$$(A'_nA_{n-2} + G_{n,n-3}B_{n-3} + G_{n,n-1})d_{M''-2} + (A'_nB_{n-2} + G_{n,n-3}C_{n-3} + G_{n,n})d_{M''-1} + \\ (A'_nC_{n-2} + G_{n,n+1})d_{M''} = r_{M''-1} - A'_nD_{n-2} - G_{n,n-3}D_{n-3}$$

for $n = M'' - 1$;

$$(G_{n,n-3}A_{n-3} + G_{n,n-2})d_{M''-2} + (G_{n,n-3}B_{n-3} + G_{n,n-1})d_{M''-1} + \\ (G_{n,n-3}C_{n-3} + G_{n,n})d_{M''} = r_{M''} - G_{n,n-3}D_{n-3}$$

for $n = M''$.

The boundary conditions (27a) and (27d) for the toroidal potential and mean-field components of the magnetic field are satisfied by the polynomials

$$T_n^{****}(x_3) = \mu_n T_{n-1} - 4nT_n - \mu_{n-1}T_{n+1}, \text{ where } \mu_n = 2n(n+1) + 1, \quad (32)$$

and hence we use the expansions

$$T^b(\mathbf{x}, t) = \sum_{|n_1| \leq N, |n_2| \leq N, 1 \leq n \leq N-2} t_n^b(t) e^{i(\alpha_1 n_1 x_1 + \alpha_2 n_2 x_2)} T_n^{*****}(x_3),$$

$$\mathbf{M}^b(x_3, t) = \sum_{0 \leq n \leq N} \mathbf{M}_n^b(t) T_n^{*****}(x_3).$$

The coefficients in the expansion of a function in this basis can be determined by solving a linear system of equations with a pentadiagonal Gram matrix G , whose non-zero entries are as follows:

$$G_{k,k-2} = -\mu_{k-3}\mu_k, \quad G_{k,k-1} = -4(k-1)\mu_k + 4k\mu_{k-2}, \quad G_{k,k} = \mu_k^2(1 + \delta_1^k) + 16k^2 + \mu_{k-1}^2,$$

$$G_{k,k+1} = -4k\mu_{k+1} + 4(k+1)\mu_{k-1}, \quad G_{k,k+2} = -\mu_{k-1}\mu_{k+2}.$$

Apparently, the system has no special properties enabling us to develop specialized methods for solving it numerically; it can be solved by the shuttle method (see Algorithm 1 in Section 3.2).

5. Conclusions

We have presented an original algorithm for numerical solution of the equations of convective dynamo in a plane horizontal layer rotating about an inclined axis under geophysically sound boundary conditions. In many respects, our approach does not differ significantly from the general approach that was followed by other authors. Focusing mainly on the accuracy of computations, especially in the small scales, we have discussed the numerical techniques which we have designed with this purpose in mind.

The Galerkin method is applied for computing the toroidal and poloidal components of physical vector fields and their mean components. We exploit the bases of functions that are products of linear combinations (T_n^* (10), T_n^{**} (18), T_n^{***} (30), and T_n^{*****} (32)) of Chebyshev polynomials in the vertical coordinate and Fourier harmonics in the horizontal coordinates. The basic functions involving the polynomials T_n^* are used for the spatial discretisation of the unknown functions that take zero values on the horizontal boundaries $x_3 = \pm 1$, namely, in the problem under discussion, the toroidal potential of the flow and its mean fields, and the deviation of temperature from the steady-state linear profile. The poloidal potential of the flow, which, for the no-slip boundary conditions, must vanish on the horizontal boundaries together with the first derivative in the vertical direction, is expanded in a series of functions involving the polynomial factors T_n^{**} . The functions involving T_n^{***} are used for discretising the poloidal potential of the magnetic field in the presence of the dielectric over the fluid layer and an electric conductor below it; while the no-slip boundary conditions for a solenoidal field are often considered in different problems, the latter boundary conditions are rather special. Finally, the toroidal potential of the magnetic field, which, for the boundary conditions considered here, is zero on the upper fluid boundary, and whose first derivative in x_3 vanishes on the lower boundary, is expanded in the series of the basic functions involving the polynomials T_n^{*****} .

The basic functions satisfy the boundary conditions, but they are non-orthogonal. Specialized algorithms for determining the coefficients in the expansion of an arbitrary function in a series of the basic functions involving polynomials T_n^* and T_n^{**} are proposed and analyzed. They are general-purpose and are applicable whenever the same boundary conditions are imposed. We find that when the spatial resolution is sufficiently high, our original algorithm 2' is optimal from the point of view of accuracy and efficiency when T_n^* are used, and algorithm 1' (the reverse shuttle algorithm) is optimal for T_n^{**} .

Author Contributions: D.T. designed algorithm 4 for finding coefficients in expansions in T_n^* , developed the codes and performed computations. R.C. wrote auxiliary codes for post-processing and analysis of the results. V.Z. conceived the project, designed algorithms 2, 2', 3 and 3' for finding coefficients in expansions in T_n^* , and algorithms 2 and 2' in expansions in T_n^{**} , prepared the figures,

and wrote the paper. All authors were involved in the discussion of the results and planning the implementation of the project, and provided critical feedback about the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The project was financed by the grant № 22-17-00114 of the Russian Science Foundation, <https://rscf.ru/project/22-17-00114/>.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Podvigina, O.M. A route to magnetic field reversals: an example of an ABC-forced non-linear dynamo. *Geophys. Astrophys. Fluid Dyn.* **2003**, *97*, 149–174. [[CrossRef](#)]
2. Chertovskih, R.; Gama, S.M.A.; Podvigina, O.; Zheligovsky, V. Dependence of magnetic field generation by thermal convection on the rotation rate: A case study. *Physica D* **2010**, *239*, 1188–1209. [[CrossRef](#)]
3. Jones, C.A.; Roberts, P.H. Convection-driven dynamos in a rotating plane layer. *J. Fluid Mech.* **2000**, *404*, 311–343. [[CrossRef](#)]
4. Rotvig, J.; Jones, C.A. Rotating convection-driven dynamos at low Ekman number. *Phys. Rev. E* **2002**, *66*, 056308. [[CrossRef](#)]
5. Eltayeb, I.A.; Rahman, M.M. Model III: Benard convection in the presence of horizontal magnetic field and rotation. *Phys. Earth Planet. Inter.* **2013**, *221*, 38–59. [[CrossRef](#)]
6. Zheligovsky, V. Space analyticity and bounds for derivatives of solutions to the evolutionary equations of diffusive magnetohydrodynamics. *Mathematics* **2021**, *9*, 1789. [[CrossRef](#)]
7. Stellmach, S.; Hansen, U. An efficient spectral method for the simulation of dynamos in Cartesian geometry and its implementation on massively parallel computers. *Geochem. Geophys. Geosyst.* **2008**, *9*, Q05003. [[CrossRef](#)]
8. Fox, L.; Parker, I.B. *Chebyshev Polynomials in Numerical Analysis*; Oxford University Press: London, UK, 1968.
9. Canuto, C.; Hussaini, M.Y.; Quarteroni, A.; Zang, T.A. *Spectral Methods in Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 1988. [[CrossRef](#)]
10. Peyret, R. *Spectral Methods for Incompressible Viscous Flow*; Springer: Berlin/Heidelberg, Germany, 2002. [[CrossRef](#)]
11. Kleiser, L.; Schumann, U. Treatment of incompressibility and boundary conditions in 3-D numerical spectral simulations of plane channel flows. In Proceedings of the Third GAMM—Conference on Numerical Methods in Fluid Mechanics, Cologne, Germany, 10–12 October 1979; Hirschel E.H., Ed.; Notes on Numerical Fluid Mechanics; Vieweg+Teubner Verlag: Wiesbaden, Germany, 1980; Volume 2, pp. 165–173.
12. Werne, J. Incompressibility and no-slip boundaries in the Chebyshev-Tau approximation: correction to Kleiser and Schumann’s influence-matrix solution. *J. Comput. Phys.* **1995**, *120*, 260–265. [[CrossRef](#)]
13. Meseguer, Á.; Trefethen, L.N. Linearized pipe flow to Reynolds number 10^7 . *J. Comput. Phys.* **2003**, *186*, 178–197. [[CrossRef](#)]
14. O’Connor, L.; Lecoanet, D.; Anders, E.H. Marginally stable thermal equilibria of Rayleigh–Bénard convection. *Phys. Rev. Fluids* **2021**, *6*, 093501. [[CrossRef](#)]
15. Reetz, F.; Schneider, T. Invariant states in inclined layer convection. Part 1. Temporal transitions along dynamical connections between invariant states. *J. Fluid Mech.* **2020**, *898*, A22. [[CrossRef](#)]
16. Wen, B.; Goluskin, D.; LeDuc, M.; Chini, G.P.; Doering, C.R. Steady Rayleigh–Bénard convection between stress-free boundaries. *J. Fluid Mech.* **2020**, *905*, R4. [[CrossRef](#)]
17. Wen, B.; Goluskin, D.; Doering, C.R. Steady Rayleigh–Bénard convection between no-slip boundaries. *J. Fluid Mech.* **2022**, *933*, R4. [[CrossRef](#)]
18. Alboussière, T.; Drif, K.; Plunian, F. Dynamo action in sliding plates of anisotropic electrical conductivity. *Phys. Rev. E* **2020**, *101*, 033107. [[CrossRef](#)]
19. Sreenivasan, B.; Gopinath, V. Confinement of rotating convection by a laterally varying magnetic field. *J. Fluid Mech.* **2017**, *822*, 590–616. [[CrossRef](#)]
20. Perez-Espejel, D.; Avila, R. Linear stability analysis of the natural convection in inclined rotating parallel plates. *Phys. Lett. A* **2019**, *383*, 859–866. [[CrossRef](#)]
21. Gottlieb, D.; Orszag, S.A. *Numerical Analysis of Spectral Methods: Theory and Applications*; SIAM-CBMS: Philadelphia, USA, 1977.
22. Tuckerman, L.S. Divergence-free velocity fields in nonperiodic geometries. *J. Comput. Phys.* **1989**, *80*, 403–441. [[CrossRef](#)]
23. Cox, S.M.; Matthews, P.C. Exponential time differencing for stiff systems. *J. Comput. Phys.* **2002**, *176*, 430–455. [[CrossRef](#)]
24. Chertovskih, R.; Rempel, E.L.; Chimanski, E.V. Magnetic field generation by intermittent convection. *Phys. Lett. A* **2017**, *381*, 3300–3306. [[CrossRef](#)]
25. Chertovskih, R.; Chimanski, E.V.; Rempel, E.L. Route to hyperchaos in Rayleigh–Bénard convection. *Europhys. Lett.* **2015**, *112*, 14001. [[CrossRef](#)]

26. Jeyabalan, S.R.; Chertovskii, R.; Gama, S.; Zheligovsky, V. Nonlinear large-scale perturbations of steady thermal convective dynamo regimes in a plane layer of electrically conducting fluid rotating about the vertical axis. *Mathematics* **2022**, *10*, 2957. [[CrossRef](#)]
27. Braginsky, S.I.; Roberts, P.H. Equations governing convection in Earth's core and the geodynamo. *Geophys. Astrophys. Fluid Dynamics* **1995**, *79*, 1–97. [[CrossRef](#)]
28. Boyd, J.P. *Chebyshev and Fourier Spectral Methods*; Dover: Downers Grove, IL, USA, 2001.
29. Schmitt, B.J.; von Wahl, W. Decomposition of solenoidal fields into poloidal fields, toroidal fields and the mean flow. Applications to the Boussinesq-equations. In *The Navier–Stokes Equations II—Theory and Numerical Methods. Proceedings, Oberwolfach 1991*; Lecture Notes in Math.; Heywood, J.G., Masuda, K., Rautmann, R., Solonnikov, S.A., Eds.; Springer: Berlin, Germany, 1992; Volume 1530, pp. 291–305. [[CrossRef](#)]
30. Lanczos, C. *Applied Analysis*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1956.
31. Madabhushi, R.K.; Balachandar, S.; Vanka, S.P. A divergence-free Chebyshev collocation procedure for incompressible flows with two non-periodic directions. *J. Comput. Phys.* **1993**, *105*, 199–206. [[CrossRef](#)]
32. Rivlin, T.J. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*; Wiley-Interscience: Hoboken, NJ, USA, 1974.
33. Shen, J. Efficient spectral-Galerkin method II. Direct solvers of second and fourth order equations by using Chebyshev polynomials. *SIAM J. Sci. Comput.* **1995**, *16*, 74–87. [[CrossRef](#)]
34. Godunov, S.K.; Ryabenkii, V.S. *Difference Schemes. An Introduction to the Underlying Theory*, 1st ed.; Elsevier: Amsterdam, The Netherlands, 1987.
35. Li, Z.-C.; Chien, C.-S.; Huang, H.-T. Effective condition number for finite difference method. *J. Comp. Appl. Math.* **2007**, *198*, 208–235. [[CrossRef](#)]
36. Mason, J.C.; Handscomb, D.C. *Chebyshev Polynomials*; CRC Press: Boca Raton, FL, USA, 2003. [[CrossRef](#)]
37. Il'in, V.P.; Karpov, V.V.; Maslennikov, A.M. *Numerical Methods for Solving Structural Mechanics Problems. A Handbook*; Vyshaiishaya Shkola: Minsk, Russia, 1990.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.