Scalability of pseudospectral methods for geodynamo simulations

Christopher J. Davies[1,*,†], David Gubbins[1] and Peter K. Jimack[2]

[1]*School of Earth and Environment, University of Leeds, Leeds LS2 9JT, U.K.*
[2]*School of Computing, University of Leeds, Leeds LS2 9JT, U.K.*

SUMMARY

The problem of understanding how Earth's magnetic field is generated is one of the foremost challenges in modern science. It is believed to be generated by a dynamo process, where the complex motions of an electrically conducting fluid provide the inductive action to sustain the field against the effects of dissipation. Current dynamo simulations, based on the numerical approximation to the governing equations of magnetohydrodynamics, cannot reach the very rapid rotation rates and low viscosities (i.e. low Ekman number) of Earth due to limitations in available computing power. Using a pseudospectral method, the most widely used method for simulating the geodynamo, computational requirements needed to run simulations in an 'Earth-like' parameter regime are explored theoretically by approximating operation counts, memory requirements and communication costs in the asymptotic limit of large problem size. Theoretical scalings are tested using numerical calculations. For asymptotically large problems the spherical transform is shown to be the limiting step within the pseudospectral method; memory requirements and communication costs are asymptotically negligible. Another limitation comes from the parallel implementation, however, this is unlikely to be threatened soon and we conclude that the pseudospectral method will remain competitive for the next decade. Extrapolating numerical results based upon the code analysis shows that simulating a problem characterizing the Earth with Ekman number $E = 10^{-9}$ would require at least 13 000 days per magnetic diffusion time with 54 000 available processors, a formidable computational challenge. At $E = 10^{-8}$ an allocation of around 350 million CPU hours would compute a single diffusion time, many more CPU hours than are available in current supercomputing allocations but potentially reachable in the next decade. Exploration of the $10^{-6} \leq E \leq 10^{-7}$ regime could be performed at the present time using a substantial share of national supercomputing facilities or a dedicated cluster. Copyright © 2010 John Wiley & Sons, Ltd.

1. INTRODUCTION

Geodynamo theory asserts that the Earth's magnetic field is continually generated and destroyed by motions of a vigorously convecting, electrically conducting fluid (liquid iron plus a small percentage of lighter elements) confined to the outer core, a 2260 km thick spherical shell some 2800 km below Earth's surface. In this dynamo process, the movement of fluid across magnetic field lines induces electric currents that sustain the magnetic field against the effects of electrical resistance. Not all flows act as dynamos and the highly nonlinear magnetohydrodynamic equations that govern the problem make understanding the physics of dynamo action a formidable task.

---

*Correspondence to: Christopher J. Davies, Institute of Geophysics and Planetary Physics, Scripps Institution of Oceanography, La Jolla, CA 92093-0225, U.S.A.
†E-mail: cjdavies@ucsd.edu

The equations must be solved numerically, which has led to many successful dynamo solutions (e.g. [1–9]). These solutions reproduce certain features of the observed field: dipole-dominance, westward drift of magnetic features and complete polarity reversals. However, simulations operate in parameter regimes far removed from that believed to be appropriate for the Earth, which remains inaccessible using the current computing power.

The mathematical problem consists of solving three advection–diffusion partial differential equations in a spherical shell representing the Earth's liquid core. The dependent variables are the fluid flow, magnetic field and density. Two of these equations describe the conservation of momentum and heat; the third, a combination of Maxwell's laws of electromagnetism and Ohm's law, governs the evolution of the magnetic field.

For any choice of parameter values, boundary conditions and basic state, the computational task involves representing dependent variables with sufficient spatial resolution to obtain converged solutions (solutions that do not change upon increasing resolution) while using sufficient temporal resolution to resolve the intrinsic timescales of the system. Great disparity of scales causes the main numerical difficulties: intrinsic timescales range from the rotation period, one day, to the reversal timescale, a few hundred thousand years. The numerical timestep must, therefore, be small enough to resolve the shortest timescales, but the equations must be integrated for at least one magnetic diffusion time (≈ 25000 years, the time taken for the dipole field to decay in the absence of a generation mechanism) to demonstrate a successful dynamo. Polarity reversals [10] require an even longer integration time. Turbulence in the core will likely lead to a broad spectrum of lengthscales; numerical simulations will never be able to resolve the smallest lengthscales without some simplifying assumptions about the turbulence. The spatial resolution for any particular model must be sufficient for that model to have converged, which puts 'Earth-like' parameters out of reach at present. As simulations move towards more 'Earth-like' parameter regimes, spatial resolution must increase whereas the timestep must decrease.

This paper assesses the scalability of a parallel, distributed memory pseudospectral code, by far the most common numerical implementation for solving the dynamo equations [5]. In particular, the following questions will be addressed:

1. How does the method scale with increasing problem size?
2. What are the limits on the size of problem the method can address using current available hardware?
3. How big a computer is needed to perform an 'ideal' calculation that runs in an 'Earth-like' parameter regime?

These objectives are achieved by first specifying the simplest physical problem that might be representative of the Earth. This problem will be called the *ideal* problem as it represents a goal not achieved by any geodynamo simulation to date. Despite its relative simplicity this problem remains well beyond the capacity of current computers. Next, estimates of the spatial and temporal resolutions required to undertake the ideal problem are determined. Naïve operation counts, memory requirements and communication costs for asymptotically large problems are then obtained for the pseudospectral method. Numerical calculations are then presented for comparison with theoretical scalings. Finally, theoretical scalings are used to extrapolate numerical calculations to Earth-like values, obtaining computational requirements for simulating the ideal problem.

Numerical calculations are conducted on a parallel computer cluster located at the University of Leeds. The cluster comprises 128 dual-core nodes connected by a Myrinet 2000 network. Each core has a clock speed of 2.4 Ghz and each node has 2 GB of memory. As we show later, the computational costs outweigh memory and communication requirements and hence this computer cluster is satisfactory for the calculations pursued here.

The numerical solutions reproduce Case 1 of the dynamo benchmark [5]. This solution is chosen because it incorporates all the essential physics and reaches a final state that is precisely defined in term of energies, drift rate and local properties of the dependent variables, making it accurately reproducible by any dynamo code. Moreover, Case 1 of the dynamo benchmark converges at modest

resolutions, allowing a wide range of resolutions to be explored with the computing resources available. The parameter values defining Case 1 are many orders of magnitude removed from those of the Earth and therefore from the ideal problem discussed in this paper, but this is true of any parameter set that is tractable on current computers.

Scalability of the pseudospectral method will be assessed numerically in two ways. First, for a fixed problem size the number of processors, N_p, will be incrementally increased from a serial calculation with $N_p = 1$. This type of scalability will be called *strong* scalability, assessed by measuring the *speedup*, $S = T_1/T_p$, where T_1 is the time in seconds to complete a timestep on one processor and T_p is the time in seconds to complete a timestep on p processors. Ideal strong scalability occurs when S doubles as N_p doubles. Second, the problem size and the number of processors will be proportionally increased from a serial calculation in such a way so as to keep the work per processor constant. This type of scalability will be called *weak* scalability, assessed using asymptotic scaling laws (derived in Section 3 below) for the total work, i.e. number of computations for a simulation, and the work per timestep. Weak scalability is the more relevant to dynamo modellers because larger simulations generally run on larger clusters; the goal is primarily to increase the problem size, a necessity as parameter values are pushed to more realistic values, rather than solve existing problems faster.

The physical and mathematical problem is described in Section 2, where the required spatial and temporal resolutions for simulating the ideal problem are determined. Approximate operation counts, memory requirements and communication costs for asymptotically large problems are obtained in Section 3. Section 4 shows the results from numerical simulations. Section 5 uses the results of numerical simulations and the operation counts to extrapolate to large clusters and more 'realistic' parameter values. Summary and discussion are presented in Section 6.

2. PROBLEM FORMULATION

2.1. The ideal problem

We seek to define what might be deemed an 'ideal' geodynamo simulation: one that includes the most important effects, i.e. those that are physically essential and/or may lead to predictions of existing observations. We consider an electrically conducting fluid confined to a rapidly rotating spherical shell of radial extent $d = r_o - r_i$ and aspect ratio $r_i/r_o = 0.35$. Here, r_i corresponds to the inner boundary and r_o to the outer boundary. The fluid rotates about the vertical z-axis with angular velocity Ω. Gravity, denoted g, is assumed to vary linearly with radius, a close approximation to the Earth [11]. The fluid is assumed to be incompressible and Boussinesq (e.g. [12]): density variations are neglected other than when they modify gravity.

The standard model for driving convective fluid motions in the geodynamo is via a mix of thermal and compositional buoyancy sources. The Earth's solid inner core grows slowly over time as the heavy component of the outer core fluid alloy freezes onto it, providing a source of thermal buoyancy [13–15]. Thermal sources may also be internal, perhaps due to the presence of radioactive elements in the core (e.g. [16–18]). Inner core freezing can also release light elements, which rise and provide a source of compositional buoyancy to drive the convection [19]. We assume that a single energy source drives the convection as this is unlikely to produce a drastic increase in the computational task.

We follow common practice by assuming constant thermal diffusivity, κ, constant coefficient of thermal expansion, α, constant viscosity, v, and constant electrical conductivity, σ. The magnetic diffusivity, defined as $1/(\mu_0\sigma)$, where μ_0 is the permeability of free space, is also assumed constant. Temperature is fixed at $T_o + \Delta T$ and T_o on the inner and outer boundaries, respectively. With the aforementioned assumptions we scale the magnetohydrodynamic equations by the shell depth, d, the magnetic diffusion time, d^2/η, ΔT as the unit of temperature and $\mathscr{B} = (2\Omega\rho\mu_0\eta)^{\frac{1}{2}}$ as a measure of the magnetic field strength to obtain the following equations for modelling the dynamo process

in spherical polar coordinates, (r, θ, ϕ):

$$\frac{E}{qPr}\left(\frac{\partial \mathbf{u}}{\partial t} - \mathbf{u} \times (\nabla \times \mathbf{u})\right) + \mathbf{z} \times \mathbf{u} = -\nabla P + q Ra T\mathbf{r} + (\nabla \times \mathbf{B}) \times \mathbf{B} + E\nabla^2\mathbf{u}, \qquad (1)$$

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla)T = q\nabla^2 T, \qquad (2)$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \nabla^2\mathbf{B}, \qquad (3)$$

$$\nabla \cdot \mathbf{u} = 0, \qquad (4)$$

$$\nabla \cdot \mathbf{B} = 0, \qquad (5)$$

where $\mathbf{u}$ is the fluid velocity, $\mathbf{B}$ is the magnetic field, T is the temperature deviation from the basic state temperature, P is the modified pressure, and ρ is the density (see [12] for details). Equation (1) is an expression of Newton's second law of motion. The term $(\partial/\partial t - \mathbf{u} \times \nabla \times)\mathbf{u}$ is the motional derivative (the time derivative of momentum following the fluid) and the term $\mathbf{z} \times \mathbf{u}$ is the Coriolis force. The right-hand side is the sum of the applied forces that cause changes in momentum; from left to right these are the pressure gradient, buoyancy, Lorentz and viscous forces. Equation (2) is an energy conservation equation for the thermal energy source that drives the convection. Equation (3) governs the magnetic field evolution and is nonlinearly coupled to Equation (1) through the induction term $\nabla \times (\mathbf{u} \times \mathbf{B})$. The final two solenoidal equations describe respectively conservation of mass for an incompressible fluid and the fact that no magnetic monopoles have been observed.

The inner and outer boundaries are assumed to co-rotate in our simulations. We assume that the velocity vanishes at both boundaries (the no-slip condition), which are both electrical insulators. The latter condition is essentially non-local and requires the solution of Laplace's equation outside the spherical shell. The pseudospectral method involves an expansion in spherical harmonics which are themselves solutions of Laplace's equation, and so the non-local boundary condition is converted to a local condition on each radial function [20]. The inner boundary is assumed to be held at a fixed temperature. The thermal boundary condition on r_o is likely to be inhomogeneous (e.g. [21]) but we neglect this effect, as it is unlikely to increase the computational task, and prescribe a fixed temperature on r_o.

In Equations (1)–(3) the Ekman number, E, measuring the rotation rate, the Prandtl number, Pr, the ratio of viscous and thermal diffusivities, the Rayleigh number, Ra, measuring the strength of the applied temperature difference across the shell, and Roberts number, q, the ratio of thermal and magnetic diffusivities, are given respectively by

$$E = \frac{v}{2\Omega d^2}, \quad Pr = \frac{v}{\kappa},$$
$$Ra = \frac{g\alpha\Delta T d^3}{2\Omega\kappa}, \quad q = \frac{\kappa}{\eta}. \qquad (6)$$

We complete our specification of the ideal problem by assigning geophysical values to these parameters.

Recent theoretical and experimental work on iron at high temperatures and pressures (see [22] and references therein) has yielded an understanding of the molecular, thermal and chemical properties of the core fluid, although the electrical and thermal conductivities are still uncertain by a factor of up to 3 [23]. The Prandtl number, Pr, is $O(1)$, but the Roberts number and the magnetic Prandtl number, $Pm = qPr$, are 10^{-6} or less; such extreme values will be impossible to incorporate into numerical simulations for the indefinite future. This problem is commonly addressed [24] by assuming a simple turbulence model that brings the thermal and viscous diffusivities up to the value of the magnetic diffusivity (so-called 'turbulent' diffusivity values), implying Prandtl numbers of order unity. We believe that the problem of core turbulence is better studied outside

the context of the full geodynamo problem and hence adopt Prandtl numbers equal to unity for the ideal problem.

The Ekman number is the major computational challenge. Large-scale motions are controlled by the spherical geometry [24]; however, the smallest scales, where energy is dissipated, are determined by the low viscosity [25, 26]. Hence the Ekman number controls the lengthscale of the flow and, in turn, the spatial resolution that is required in numerical simulations. The lowest value of E achieved in any simulation is $\sim 5 \times 10^{-7}$ [8]; very little work has been done in this regime and the calculations are computationally intensive. This is still significantly higher than the 'turbulent' Ekman number for the core, $E = 10^{-9}$ (e.g. [27]), obtained when turbulent diffusivities are invoked. Using molecular values for the diffusivities gives $E = 10^{-16}$, an enormous challenge. To combat the problem some authors have used 'hyperdiffusivity' (e.g. [1, 2, 28]) a form of scale-dependent viscosity that artificially damps high wavenumbers. This allows a low 'headline' Ekman number, applicable to the large scales, without the need to resolve very small scales [29]. Hyperdiffusivity has been shown to prevent formation of small scale structures that may play a crucial role in the dynamics [30]. Here we restrict ourselves to constant turbulent viscosity.

Thermal buoyancy forces in the core are determined by the Rayleigh number. The thermal Rayleigh number (Ra in Equation (1)) may not be too large to simulate [31, 32], perhaps less than 1000 times the critical value for onset of non-magnetic convection. Current dynamo simulations have used Ra up to $O(100)$ times supercritical [7–9]; the highly supercritical regime is very poorly understood in both physical and numerical respects. A low Ra will therefore be relevant. For the ideal problem we therefore assume that Ra is not too far above the critical value for the onset of non-magnetic convection.

From the foregoing discussion, we suggest an 'ideal' geodynamo simulation having $E = 10^{-9}$, Prandtl numbers equal to unity, and a Rayleigh number not too far above critical. A run length of one magnetic diffusion time is required to demonstrate dynamo action. Achieving such a calculation would represent a huge advance. Many other complications could be explored with relatively little additional computational expense.

2.2. Numerical representation

The pseudospectral method used in this work is described in [33]; it is similar to other pseudospectral methods [1, 28]. Vector dependent variables **u** and **B** are expanded in toroidal and poloidal scalars, $\mathscr{T}$ and $\mathscr{P}$,

$$\{\mathbf{u}, \mathbf{B}\} = \nabla \times (\mathscr{T}\mathbf{r}) + \nabla \times \nabla \times (\mathscr{P}\mathbf{r}), \tag{7}$$

(e.g. [20]), thus satisfying the solenoidal conditions (4), (5) exactly. Scalars are then expanded as

$$\{\mathscr{T}(r, \theta, \phi, t), \mathscr{P}(r, \theta, \phi, t)\} = \sum_{l=1}^{N} \sum_{m=0}^{l} S_l^m(r, t) Y_l^m(\theta, \phi), \tag{8}$$

where N is a suitably chosen truncation point for the infinite series, the $S_l^m(r, t)$ are coefficients to be determined and the fully normalized spherical harmonics, $Y_l^m(\theta, \phi)$, are given by

$$Y_l^m(\theta, \phi) = P_l^m(\cos\theta) \exp^{im\phi}, \tag{9}$$

where $P_l^m(\cos\theta)$ are associated Legendre functions of degree l and order m [34]. In our code the radial dependence of the $S_l^m(r, t)$ is represented on a grid of N_r points with variable spacing. Grid points are clustered near the boundaries, allowing finer resolution of the small-scale structures that appear in the boundary layers. Angular derivatives of dependent variables are found using the corresponding derivatives of the spherical harmonics; radial derivatives are computed using high-order finite differences.

The toroidal and poloidal parts of the governing equations are timestepped in the spectral domain (i.e. by operating on spherical harmonic coefficients); the right-hand sides of Equations (1)–(3) must therefore be evaluated in spectral space. Our code uses a semi-implicit predictor-corrector method: diffusion terms are treated implicitly whereas all nonlinear terms and the Coriolis term

are treated explicitly to reduce the size of the resulting matrices. The spherical harmonics satisfy the differential equation

$$\nabla^2 S_l^m(r,t)Y_l^m(\theta,\phi) = \mathscr{D}_l S_l^m(r,t)Y_l^m(\theta,\phi), \tag{10}$$

(e.g. [12]), where the $\mathscr{D}_l$ operator is given by

$$\mathscr{D}_l = \frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\right) - \frac{l(l+1)}{r^2}, \tag{11}$$

and hence the Laplacian does not couple spherical harmonic modes. Hence these terms in Equations (1)–(3) can be treated separately for each harmonic and the timestepping equations take the form

$$\frac{\partial S_l^m(r,t)}{\partial t} + \mathscr{D}_l S_l^m(r,t) = N_l^m(r,t), \tag{12}$$

where N_l^m represents the result of evaluating the nonlinear terms in any of Equations (1)–(3). Curls and gradients are evaluated in spectral space (i.e. by operating on spherical harmonic coefficients). As with the Laplacians, these operations do not couple harmonic modes. Computing nonlinear terms in spectral space is very slow and hence the *spherical transform method* is used [35] in which the nonlinear terms are formed by multiplication in physical space (i.e. evaluated on gridpoints in a spherical coordinate system).

In this paper we will focus on the parallel decomposition used in our code; discussion of alternative decompositions is deferred to Section 5.1. Our code uses two parallelization strategies depending on the operations being performed. For linear operations, harmonic coefficients are split across processors with each processor having access to the whole radial grid. Before undertaking the spherical transform the logical grid is reorganized so that all harmonics for a given radial grid point are on the same processor, and hence radial shells are split across processors. The vector transpose that accomplishes this change in data distribution is the major communication step in our code. Because the spherical transforms are not split across processors the number of processors that our code can use is limited to be no greater than the number of radial points. This matter is revisited in Section 5.

2.3. Anticipated resolution

Fluid motions at the onset of non-magnetic convection in a rapidly rotating spherical shell take the form of columns with an $O(E^{1/3})$ azimuthal lengthscale [36, 37], much smaller than their radial and axial lengthscales e.g. [38]. The required spatial resolution will therefore scale as $\Delta x = E^{1/3}$. Magnetic forces and low Prandtl numbers are known to increase the azimuthal lengthscale near onset (e.g. [16, 39]); these effects are therefore unlikely to increase the required resolution. Higher Rayleigh numbers increase the turbulence, changing the lengthscale in all directions, however, the smallest scales are ultimately dictated by the viscosity, i.e. E (Equation (6)). Thin ($O(E^{1/2})$) structures may appear in boundary layers, however, these can be accommodated in the pseudospectral code by using a non-uniform grid (e.g. [33]). Detached shear layers, such as Stewartson layers [40] may form, but these are likely thicker than $E^{1/3}$ [41].

We, therefore, propose that $E^{1/3}$ is the smallest lengthscale to be resolved. In practice, a suite of calculations at progressively lower E is required. If any important structures with lengthscales smaller than $E^{1/3}$ exist they should become evident before any calculations are performed with insufficient resolution. With this assumption, spatial resolution requires $N = kN_r = K_s(E^{-1/3})$ points, or spherical harmonics, for each dimension, where k is a number that allows N to differ from N_r and $K_s = O(1)$ cannot be determined more accurately without undertaking simulations. Asymptotically, K_s is assumed not to be of primary significance.

Temporal resolution is determined from the Courant–Friedrichs–Lewy (CFL) conditions [42] arising from the use of explicit timestepping to advance the nonlinear terms $(\mathbf{u}\cdot\nabla)T$, $\mathbf{u}\times(\nabla\times\mathbf{u})$, $\mathbf{B}\times(\nabla\times\mathbf{B})$ and $\nabla\times(\mathbf{u}\times\mathbf{B})$ in Equations (1)–(3). The time restriction for all terms except the

Lorentz force is $\Delta t < \Delta x / |u|$, where $|u|$ is the local amplitude of fluid motions. The Lorentz force is quadratic in **B** and errors can grow due to the coupling between Equations (1) and (3). These equations support many different types of magnetohydrodynamic waves [25], the fastest of which travel at a speed V. A reasonable estimate for V will be the Alfvén velocity $V_A = \mathscr{B} / \sqrt{\mu_0 \rho}$, where $\mathscr{B}$ is a characteristic amplitude of the magnetic field (e.g. [43]). The timestep is assumed here to be determined by the smaller of $\Delta x / |u|$ or $\Delta x / |V_A|$. The important point is that an increase in spatial resolution results in a proportional decrease in the maximum stable timestep.

The fluid velocity must be great enough to regenerate magnetic field, which is usually measured by the magnetic Reynolds number, $R_m = |u| d / \eta$, which must exceed a critical value for dynamo action. R_m is an output parameter from a numerical simulation because it depends on the solution. Formal bounds on the minimum R_m for dynamo action are $O(10)$ [44, 45], but these are only lower bounds. Taking $|u| = 10^{-4}\,\mathrm{m\,s^{-1}}$ (e.g. [46]), $d = 2260\,\mathrm{km}$ and $\eta = 1.6\,\mathrm{m^2\,s^{-1}}$ for the Earth [22] gives $Rm = O(100)$. Using the non-dimensionalization of Section 2.1, R_m is a dimensionless measure of the fluid velocity and hence $|u|$ is likely to be $O(100)$.

Estimates of V_A depend on the field strength, $\mathscr{B}$, which itself depends on many factors, most notably the Rayleigh number. Hence $\mathscr{B}$ cannot be estimated *a priori*. In dimensionless units the ratio $V_A / |u| = d\sqrt{2\Omega / \eta} \approx 8 \times 10^4$, so that the timestep restriction will be determined by the magnetic field rather than by the fluid velocity. A dimensional argument based on Earth-like values gives a similar result: core fluid velocities inferred from time variations in the geomagnetic field are $O(10^{-4})\,\mathrm{m\,s^{-1}}$ whereas Alfvén speeds for magnetic field strengths of 1 mT are $O(10^{-2})\,\mathrm{m\,s^{-1}}$. Faster magnetohydrodynamic waves than the Alfvén waves may exist in the system but this does not alter the conclusion that the timestep restriction comes from the magnetic field.

Another timestep restriction can come from the Coriolis force. This restriction is not significant at the values of E currently employed in geodynamo simulations, but will become overwhelming when E is reduced to geophysical values, say 10^{-9}. The term is linear in **u** and can be treated implicitly, as already undertaken by some authors (e.g. [47, 48]), which removes the timestep restriction at the expense of solving larger matrix systems [29]. We do not, therefore, consider the Coriolis term's impact on the choice of Δt in this paper.

Because $\Delta t < \Delta x / \max[|u|, |V_A|]$ and $\Delta x = E^{1/3}$ has been assumed, the number of timesteps, N_t, for a given run must scale as $N_t = K_t(E^{-1/3})$, where K_t depends on $|u|$ and V_A and cannot be determined more accurately without undertaking simulations. We assume that K_t is not of primary significance for asymptotically small E, i.e. asymptotically large problem size. In practice, our code uses a semi-implicit timestepping algorithm that ensures stability of the timestep. By changing the spatial resolution it is therefore possible to measure the scaling of Δt in the computationally accessible parameter regime, which can be compared to the theoretical scaling.

In the following sections scalings will be derived per timestep, with the extra cost associated with the decrease in Δt with Δx accommodated empirically when extrapolations are undertaken.

3. APPROXIMATE SCALINGS FOR THE PSEUDOSPECTRAL METHOD

3.1. Operation count

The evaluation of nonlinear terms by the spherical transform is the rate-determining part of the pseudospectral method; to obtain an approximate operation count this section therefore focuses solely on calculation of the nonlinear terms.

Given spherical harmonic coefficients, f_l^m and g_l^m, of two scalar functions, $f(\theta, \phi)$ and $g(\theta, \phi)$, the task is to find the spherical harmonic coefficients of the product fg. The procedure is to individually forward transform the functions from the spectral domain to the physical space domain, carry out the multiplications, and inverse transform the result back. In the physical domain the scalar functions are represented on a lateral grid (θ_i, ϕ_j) at each radial grid point. The θ-points are chosen to lie at the Gauss points, i.e. the zeros of a high-order polynomial that cluster near the poles [49]. The ϕ-points are equally spaced. Using Gaussian quadrature (e.g. [49]) for the integration gives an exact discrete transform between physical and spectral domains with the

minimum number of θ-points for a given truncation in spectral space [35, 50]. The transform pair takes the form

$$f(\theta_i, \phi_j) = \sum_{l=1}^{N} \sum_{m=0}^{l} f_l^m Y_l^m(\theta_i, \phi_j) = \sum_{l=1}^{N} \sum_{m=0}^{l} f_l^m P_l^m(\theta_i) \exp^{im\phi_j}, \qquad (13)$$

$$f_l^m = \sum_{i=1}^{3N/2} \sum_{j=1}^{N} f(\theta_i, \phi_j) w_i Y_l^{m*}(\theta_i, \phi_j) = \sum_{i=1}^{3N/2} \sum_{j=1}^{N} f(\theta_i, \phi_j) w_i P_l^m(\theta_i) \exp^{-im\phi_j}, \quad (14)$$

where w_i are Gaussian weights and $*$ denotes complex conjugation. Because physical quantities are real, f_l^{-m} is the complex conjugate of f_l^m, a fact that is used to reduce storage and computational requirements for the complex coefficients.

The m- and j-sums in the transform pair constitute discrete Fourier transforms and hence the FFT algorithm can be used. Separating the sums as

$$f(\theta_i, \phi_j) = \sum_{m=0}^{l} \left(\sum_{l=1}^{N} f_l^m P_l^m(\theta_i) \right) \exp^{im\phi_j}, \qquad (15)$$

$$f_l^m = \sum_{i=1}^{3N/2} w_i P_l^m(\theta_i) \left(\sum_{j=1}^{N} f(\theta_i, \phi_j) \exp^{-im\phi_j} \right), \qquad (16)$$

shows that Equation (15) is a Fourier transform of the l-sum whereas Equation (16) is an i-sum of the Fourier transform of the $f(\theta_i, \phi_j)$ over j. The l-sum in (15) requires $O(N^2)$ operations for each i and all m, whereas the i-sum in (16) requires $O(N^2)$ operations for each l and all m. Here an operation is taken to be a multiplication followed by an addition. The subsequent and embedded Fourier transforms take only $O(N \log_2 N)$ operations [49], which is negligible for asymptotically large N. Each sum must be computed for all θ-points and all radial points, N_r, giving an asymptotic operation count of $O(N^3)$ per radial point or, assuming that $N = kN_r$, $O(N^4)$ per scalar transform. Note that there may be scope for improving this asymptotic estimate very slightly (e.g. using algorithms such as that of Strassen [51], which may become competitive when N is sufficiently large), however, we do not consider such improvements in this work.

The total number of scalar transforms required at each timestep is obtained by considering Equations (1)–(3). For the forward transform it is necessary to transform $\mathbf{u}$ and $\mathbf{B}$ to calculate the term $\nabla \times (\mathbf{u} \times \mathbf{B})$, but it is also necessary to transform $\nabla \times \mathbf{u}$ and $\nabla \times \mathbf{B}$ so that the nonlinear terms $\mathbf{u} \times (\nabla \times \mathbf{u})$ and $\mathbf{B} \times (\nabla \times \mathbf{B})$ can be computed. T must be transformed and the final nonlinear term, $(\mathbf{u} \cdot \nabla)T$, requires that ∇T is also transformed. For the inverse transform the nonlinear terms $\mathbf{u} \times (\nabla \times \mathbf{u})$ and $\mathbf{B} \times (\nabla \times \mathbf{B})$ in the momentum equation can be added together in physical space and hence only the result is transformed back. The terms $\nabla \times (\mathbf{u} \times \mathbf{B})$ and $(\mathbf{u} \cdot \nabla)T$ must also be transformed back to spectral space. Hence, the forward transform requires 4 vector transforms and 2 scalar transforms, whereas the inverse transform requires 2 vector transforms and 1 scalar transform.

The number of scalar transforms comprising a vector transform can be identified from the components of the toroidal and poloidal scalars. When expanded in spherical harmonics, these components are

$$A_r(r, \theta, \phi) = \sum_{l,m} \frac{l(l+1)}{r} \mathscr{P}_l^m(r) P_l^m(\theta) \exp^{im\phi},$$

$$A_\theta(r, \theta, \phi) = \sum_{l,m} \left(\frac{im}{\sin\theta} \mathscr{T}_l^m(r) + \frac{1}{r} \frac{dP_l^m(\theta)}{d\theta} \frac{d}{dr}[r\mathscr{P}_l^m(r)] \right) \exp^{im\phi},$$

$$A_\phi(r, \theta, \phi) = \sum_{l,m} \left(-\mathscr{T}_l^m(r) \frac{dP_l^m(\theta)}{d\theta} + \frac{im}{r\sin\theta} \frac{d}{dr}[r\mathscr{P}_l^m(r)] P_l^m(\theta) \right) \exp^{im\phi},$$

where $\mathbf{A} \in \{\mathbf{u}, \mathbf{B}\}$. A_r requires a single scalar transform, A_θ a further two, and A_ϕ only a further one because the last term in the sum involves the same l-sum as that computed for A_r; the additional radial differentiations and multiplications do not add to the leading term in the asymptotic operation count. Hence each vector transform requires 4 scalar transforms. Thus the forward transform requires 18 scalar transforms whereas the inverse transform requires 9 scalar transforms, giving a total of 27 scalar transforms per timestep. Transforms must be computed at each radial point yielding an approximate operation count, O, of

$$O = 27 C_1 N^4, \qquad (17)$$

for a fixed time unit, where C_1 is a constant reflecting the leading asymptotic term for a single scalar transform. As previously discussed, the timestep is assumed to scale with N and hence the total operation count is $O(N^5)$.

An estimate of the operation count per processor involves replacing a factor of N in (17) by N/N_p, thus accounting for parallel decomposition in the radial direction only. This gives an operation count of

$$O = \frac{27 C_1}{N_p} N^4 \qquad (18)$$

per processor per timestep.

3.2. Memory requirements

Our code uses dynamic array allocation so that memory is only allocated when needed. Hence the amount of memory required at any stage during a given timestep may vary. In the current implementation of the code the point of maximum memory usage occurs when integrating the solution forward in time as this requires storage of all matrices and vectors for the inversions. Releasing memory once an inversion has been calculated may enable a reduction in memory requirements but this has not been attempted.

The timestepping equations take the form of a set of linear algebraic equations

$$\mathbf{A}\mathbf{x} = \mathbf{N}, \qquad (19)$$

where the vector $\mathbf{N}$ represents the spherical harmonic expansion of the nonlinear terms, which have been evaluated at each radial point by the spherical transform method. In addition, $\mathbf{x}$ is an unknown vector of spherical harmonic coefficients at the new timestep, and $\mathbf{A}$ is a matrix of known coefficients. Equation (19) can be considered separately for each harmonic (see Section 2.2). When timestepping in spectral space, this also allows harmonics to be split across processors in the parallel implementation, with all radial points for a given harmonic located on a single processor.

Radial derivatives are calculated using fourth order finite differences with a stencil width of 10. For N_r radial points Equation (19) is a system of N_r equations with the banded matrix $\mathbf{A}$ having dimensions $(10 \times N_r)$. Equation (19) is solved for each of the $N_h = O(N^2)$ harmonic coefficients, which are split across N_p processors. All numbers are of double precision; each vector or matrix element occupies 8 bytes in memory. The number of bytes of memory, M, needed for one inversion is then approximated by

$$M = 8(2N_r + 10N_r)\frac{N_h}{N_p}. \qquad (20)$$

Here the factor $2N_r$ is the storage required for the two vectors, $\mathbf{N}$ and $\mathbf{x}$, whereas the factor $10N_r$ is the storage required for the matrix A.

Equations of the form (19) are inverted at each timestep for each of the scalar variables: u_T, u_P, T, B_T, B_P, where subscripts T and P represent, respectively, the toroidal and poloidal scalars for the velocity **u** and magnetic field **B**. The equation for poloidal velocity is solved using a Green's function method [33, 50], which adds another matrix inversion. Furthermore, because a complex representation of the spherical harmonics is used, each scalar requires two inversions; one for the real part and one for the imaginary part. Hence there are 12 matrix inversions per timestep. The final memory estimate is therefore

$$M = 1152 N_r \times \frac{N_h}{N_p}. \tag{21}$$

Assuming once more that $N = kN_r$, this gives $M = O(N^3)$.

3.3. Communication costs

A simple model for the time, T, taken to send a single message of length L across a network can be written as

$$T = F + \beta L, \tag{22}$$

where F is proportional to the network latency and β is inversely proportional to the bandwidth. Note that such a model assumes that the switching capability of the network scales sufficiently well that packet collisions do not dominate as the number and length of messages grow. Furthermore, F and β are properties of the network that can be estimated on a specific parallel computer. A theoretical estimate of communication costs, therefore, requires estimating the number of messages sent per timestep and the length of each message.

Spherical harmonic coefficients are split across processors in spectral space. Before undertaking the spherical transform the logical grid is reorganized so that all harmonic coefficients corresponding to a given radial point are located on the same processor; parallelization is achieved in radius with each processor holding all harmonics for a subset of the radial grid. The change between these two parallel decompositions is accomplished by a transpose, which is the main communication step. For the forward transpose each processor initially holds a subset of the total number of harmonics on all radial levels. After the forward transpose each processor holds all harmonics on a subset of the radial grid. Hence each processor must send all its harmonics, N_h/N_p, for a given number of radial points, N_r/N_p. The reverse is true of the inverse transpose. The length of a single message for forward or inverse transposes in bytes is

$$L = 16 \left(\frac{N_r \times N_h}{N_p^2} \right), \tag{23}$$

where the prefactor contains a factor 2 because the coefficients are complex and a factor 8 to give the expression in bytes.

The total communication time depends on how many messages of length L are sent. Source and destination processors must be chosen with care because the destination processor only requires harmonics that correspond to radial points within that processor's radial subdomain. Hence the communication is undertaken in $N_p - 1$ steps. At each step, each processor sends and receives a message of length L.

The number of variables that need to be transposed is the same as the number that must be transformed using the spherical transform method. Hence there are 18 forward transposes and

9 inverse transposes per timestep. Each of these transposes is accomplished in $N_p - 1$ steps. The final estimated communication time is therefore

$$T = 27(N_p - 1)\left(F + 16\beta\left(\frac{N_r N_h}{N_p^2}\right)\right).$$ (24)

Assuming $N = kN_r$ therefore gives $T = O(N^3)$.

For asymptotically large problems the limiting factor is, therefore, the computational work of the spherical transform, which scales as $O(N^4)$ per timestep, whereas communication scales only as $O(N^3)$. Memory requirements also scale as $O(N^3)$.

4. NUMERICAL TESTS

The non-dimensional parameter values for the dynamo benchmark Case 1 are: $Ra = 32.5$, $E = 5 \times 10^{-4}$, $Pr = 1$ and $q = 5$. Numerical integration from the initial condition prescribed in [5] using 1 processor and $N = N_r = 48$ shows that kinetic and magnetic energy reach stationary values in under 1 magnetic diffusion time (Figure 1). The solution in this final state has dimensionless kinetic energy (KE) 30.77 and dimensionless magnetic energy (ME) 626.3, which are defined as

$$\text{KE} = \frac{1}{2Vq^2}\int_V \mathbf{u}^2 dV,$$ (25)

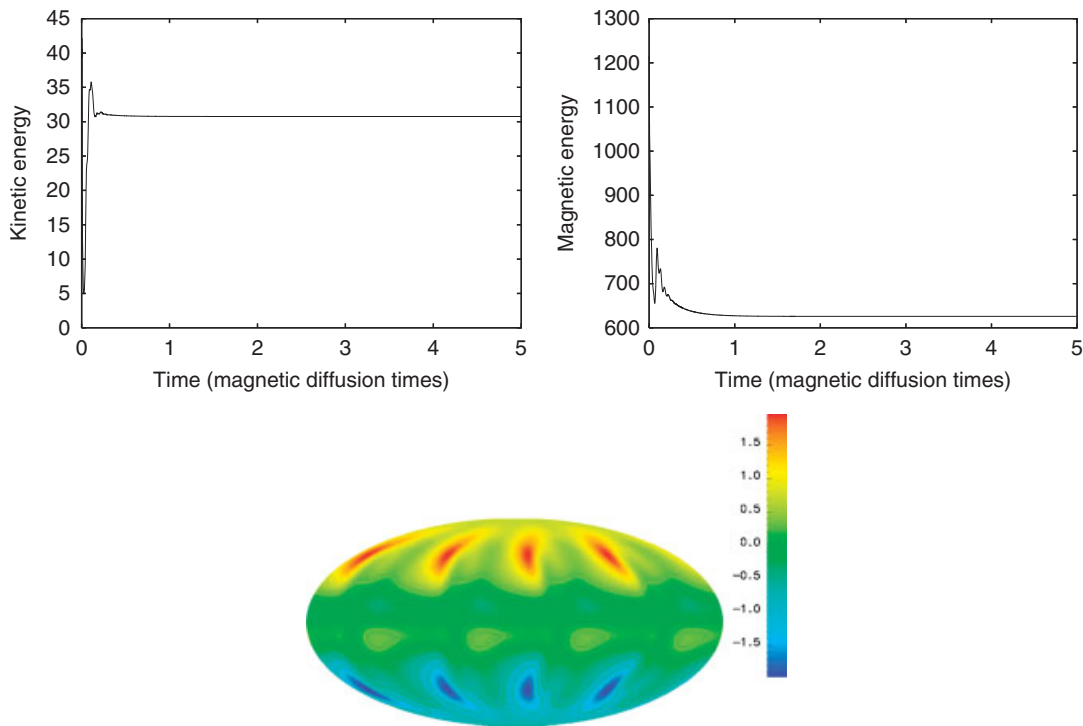$$\text{ME} = \frac{1}{2Vq^2}\int_V \mathbf{B}^2 dV,$$ (26)

Figure 1. Solution for the dynamo benchmark Case 1. Top left: Kinetic energy time series; top right magnetic energy time series; bottom: snapshot of the radial component of magnetic field, B_r, in Mollweide projection at the outer boundary at $t = 5$. The solution displays a fourfold azimuthal symmetry and magnetic features drift with a constant frequency in the westward direction.

Table I. Global properties of the solution for the dynamo benchmark Case 1 using different numerical resolutions.

N	Δt	KE	ME
40	4.08×10^{-5}	32.892	730.26
48	3.94×10^{-5}	30.779	626.34
72	2.91×10^{-5}	30.774	626.15
96	2.39×10^{-5}	30.781	626.37
120	2.09×10^{-5}	30.781	626.34
144	1.92×10^{-5}	30.781	626.29

The timestep increment, Δt, is determined by an adaptive timestepping algorithm.

where V is the volume of the shell. Table I lists the global properties of the benchmark solution at different numerical resolutions, showing that the solution has converged. A snapshot of B_r at the outer boundary is also displayed in Figure 1. The pattern has a fourfold symmetry in longitude and drifts westward at a constant rate.

Figure 2 shows the strong scalability curves for Case 1 at different numerical resolutions. The top curve shows speedup as a function of the number of processors. The time taken to complete one timestep in serial mode is 3.9 s for $N=48$, 48.2 s for $N=96$ and 213.4 s for $N=144$. All curves depart from the ideal linear scaling as N_p is increased because of communication overhead; for a fixed problem size the ratio of computational work to communication overhead decreases as N_p increases. Larger problems (higher N) follow the linear trend to higher N_p because the computational work is greater for a serial calculation[‡].

Figure 2 also shows the scaling of memory requirements with N_p, which approaches a constant value for each resolution. This is because, for N_p large and a fixed problem size, significant memory requirements come from information held locally by each processor. For fixed problem size the arrays that are split across processors contribute a smaller amount to the total memory requirement as N_p increases. The difference between observed and theoretical memory requirements [Equation(21)] is also shown in Figure 2. The differences also asymptote to constant values as N_p increases. This is again due to the static parts of the code that are not split across processors. Arrays that are not split across processors grow in size as the problem size is increased and hence each curve asymptotes to a different value; larger problem sizes asymptote at higher memory.

Weak scalability can be investigated in two ways, both of which require the use of an approximate asymptotic scaling. The first is to assume an operation count scaling as $O(N^5)$, meaning that the total work per processor over the course of a simulation is constant as the problem size is increased. This is probably the most useful for practical applications. Doubling the resolution in each direction would then require $2^5=32$ times more processors, which is impractical given the number of processors available. The second option is to assume an N^4 scaling; work per processor per timestep is constant, however the run will require more timesteps to reach the same final state. An N^4 scaling for weak scalability is used here, but can be easily extended to an N^5 scaling as shown below.

Figure 3 shows weak scalability for Case 1 using an N^4 operation count, i.e. asymptotically fixed work per processor per timestep. The time to compute a single timestep is relatively unaffected as N^4 and N_p are proportionally increased and hence communication costs are seen to be unimportant so long as the problem size is large enough. Furthermore, this demonstrates that the pseudospectral method does indeed follow the $O(N^4)$ scaling predicted by Equation (17). Figure 3 also shows that the memory requirement per processor increases as N^4 and N_p are proportionally increased.

[‡]The rather poor strong scaling performance shown in Figure 2 (top) is a consequence of the size of the fixed problem not being sufficiently large for perfect strong scalability to occur. The current implementation of the code may be improved by passing fewer, but longer, messages, however, we use this original version here primarily because it illustrates that good weak scalability can be achieved even when the work per processor is not sufficient to allow such good strong scalability to be observed.
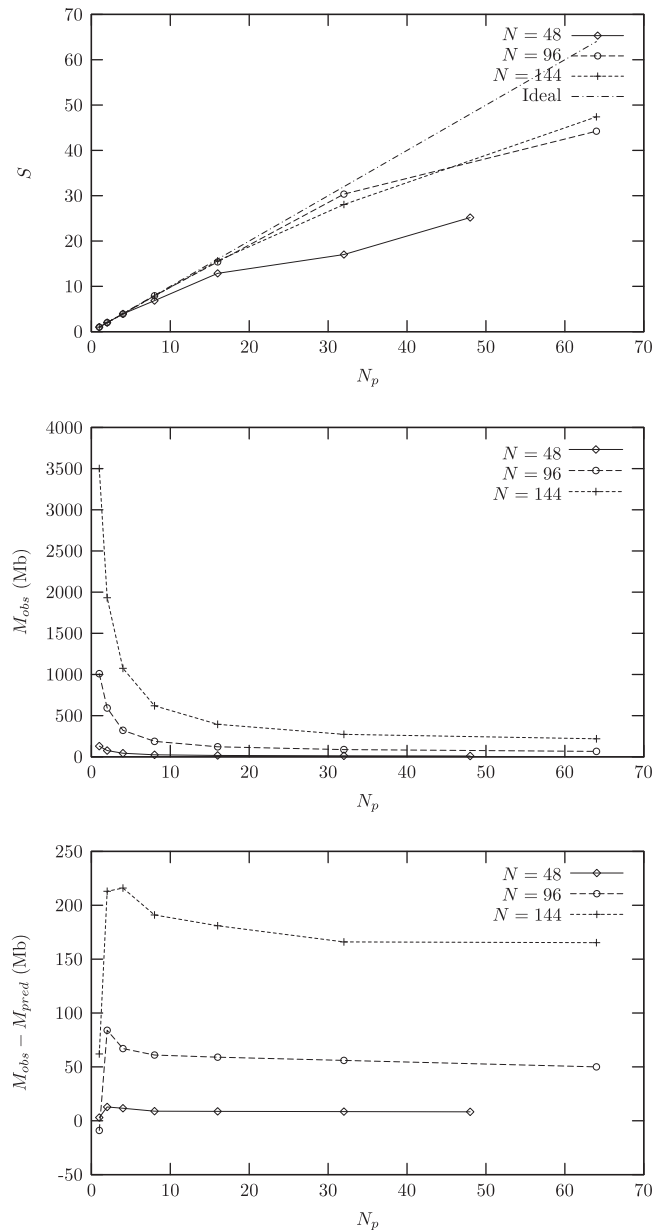
Figure 2. Strong scalability curves for the dynamo benchmark Case 1. Top: Speedup, S, plotted against number of processors. The dot-dashed line shows the ideal linear scaling. Middle: Observed memory requirement per core, M_{obs}, as a function of the number of processors. Bottom: Difference between M_{obs} and the predicted memory requirements, M_{pred}, as a function of the number of processors. The theoretical prediction uses Equation (21).

This observation disagrees with our $O(N^3)$ asymptotic memory scaling and will be discussed in Section 5.1.

Allowing N_p to grow in proportion to N^4 does not account for changes in the time increment, Δt, as the spatial resolution changes. Figure 4 shows the estimated number of timesteps, N_t, as a function of N. The observed number of timesteps required to simulate one magnetic diffusion time has a shallower gradient than the prediction as resolution is increased; the predicted scaling represents an overestimate. It is not entirely clear why this should be, however we note that the variable timestep selection algorithm is based upon a simple local truncation error estimate and
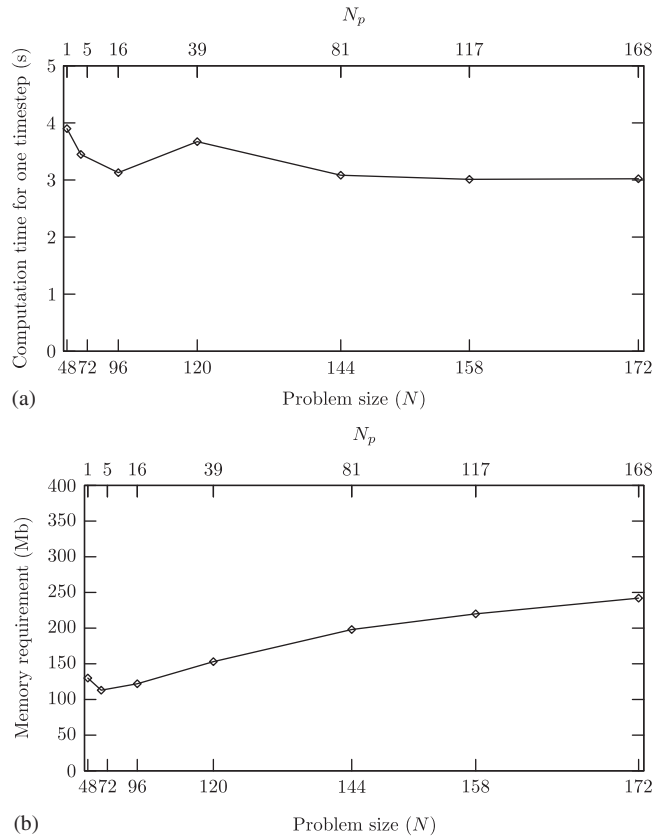
Figure 3. Weak scalability for the dynamo benchmark Case 1 where N_p is chosen to be proportional to N^4. Top: Time taken to complete one timestep as a function of the problem size (bottom x-axis) and number of processors (top x-axis). Bottom: Memory requirement per processor as a function of the problem size (bottom x-axis) and number of processors (top x-axis).
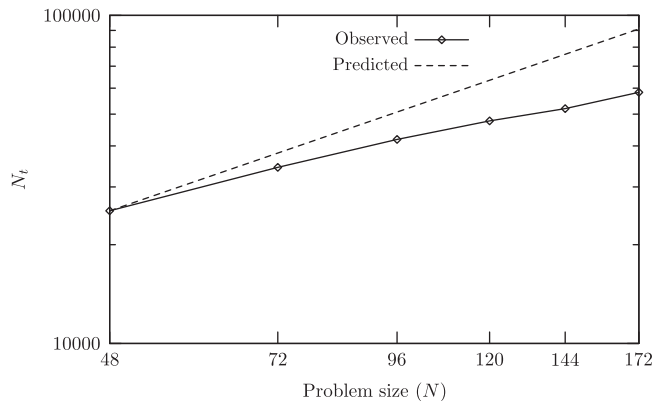

Figure 4. Number of timesteps required to simulate one magnetic diffusion time, N_t, plotted against problem size, N, on a log–log scale. The two lines represent the observed scaling using Case 1 of the dynamo benchmark and the predicted scaling that the timestep decreases with increasing resolution proportional to N (Section 2.3).

hence for large Δx (small N) it is very possible that the timestep is less then the maximum stable step size. Whatever the explanation, assuming an $O(N^5)$ scaling for the total operation count certainly represents a conservative estimate.

5. EXTRAPOLATION OF NUMERICAL RESULTS

5.1. Alternative domain decompositions

Our weak scalability analysis applies to a one-dimensional parallel decomposition of the pseudospectral method that does not split spherical transforms across processors. This is the strategy implemented in our parallel solver and it enables us to (1) derive simple expressions for operation counts, memory requirements and communication costs that do not depend critically on the underlying network architecture and (2) determine the impact of communication costs by testing our asymptotic scalings using a weak scalability analysis. Moreover, simple choices of the truncation points in each dimension ensured a balanced load per timestep.

Alternative parallel decompositions are possible and would allow the number of usable processors to be increased by splitting the Fourier and/or Legendre transforms across processors as well as across the radial domain. In principle it is possible to decompose all three physical dimensions across processors. Communication is required between the individual processors that compute Fourier transforms for a given latitude and then between processors that compute Legendre transforms for a given azimuthal wavenumber. The number of usable processors is then potentially $O(N^3)$ although this is likely to be unachievable in practice because of the amount of interprocessor communication required. Other possibilities involve reorganizing the physical or spectral grid so that the Fourier and Legendre transforms are each performed on a single processor. Communication is then required to reorganize the grid via a vector transpose, but not when computing the transform. For example, each processor performs the Fourier transform for a given latitude and then performs the Legendre transform for a given azimuthal wavenumber, as done in the dynamo context by [52]. In this configuration the number of usable processors is potentially $O(N^2)$.

Many communication strategies based on these approaches are available for the spectral transform (see [53] for a discussion of some possibilities). It is, therefore, possible that an alternative parallel decomposition to the one that we have implemented and studied could demonstrate weak scalability while improving the overall balance between computation and communication. It is of course theoretically possible to derive asymptotic scaling laws for such decomposition strategies; however, without an efficient implementation against which to benchmark, there is no way of either verifying their correctness or of quantifying the multiplicative constants that are implicit in such scalings. This last point is especially important when bandwidth limitations and communication latencies become important due to the increased volume of communications. The resulting communication costs then depend critically on the network architecture and the logical layout of processors, making asymptotic formulae much less useful; empirical scalings based on particular network configurations must be used to gain meaningful predictions of communication overheads [53]. Furthermore, not all domain decompositions will display the ideal weak scalability that we demonstrate for the one-dimensional decomposition (i.e. time scales in proportion to the work per processor). Such weak scalability is essential to our goal of using scaling laws to extrapolate to high resolution. For these reasons the remainder of this section is restricted to further consideration of the one-dimensional parallel decomposition employed in our code as described above.

5.2. Extrapolation for 1D parallelization

The results from weak scalability analysis suggest that the asymptotic scaling for operation counts [Equation (17)] is adequately followed by numerical calculations for the number of processors currently available. Conversely, the theoretical memory scaling [Equation (21)] is not borne out in the numerical calculations. This occurs because certain working arrays, e.g. those required to hold the Legendre polynomials and their derivatives, must be replicated on each processor. This is only problematic if extrapolation of the true results to larger resolutions leads to memory requirements that are beyond the capacity of modern computers.

To extrapolate to higher resolutions we assume $N = K_s E^{-1/3}$ (Section 2.3). The estimate of K_s depends on many factors and is highly uncertain, but we note from Table 1 that the dynamo benchmark solution has converged with $N \approx 40$, giving $K_s = 3$. A dynamo model with similar parameters

Table II. Computing time required for runs with the different of E and N_p using $K_s = 3$.

E	N_p	Time (days)	Time (CPU hours)
5×10^{-7}	6803	3.3	540 000
10^{-7}	11 634	29	8×10^6
10^{-8}	25 064	620	3.8×10^8
10^{-9}	54 000	13 392	1.7×10^{10}

to the benchmark but using $E = 10^{-5}$ gives a converged solution when $K_s = 3.1$. Throughout this section we therefore take $K_s = 3$ and note that an improved estimate of K_s can only be obtained by undertaking many simulations over a wide range of parameters and resolutions. The ideal problem, where $E = 10^{-9}$, then requires $N = 3000$. Extrapolating to $N = 3000$ using Figure 3, which appears to allow linear extrapolation beyond $N = 172$, reveals that for $E = 10^{-9}$ each core would require ≈ 2.5 GB of RAM: many current supercomputing facilities can provide this. It is, therefore, unlikely that memory issues will be important at lower E.

Before proceeding we note a parallel limitation of the current version of our pseudospectral code: the maximum number of processors cannot exceed the number of radial points. This is an unnecessary restriction because at each timestep there are 18 independent forward scalar transforms and 9 independent inverse scalar transforms that could also be split across processors without significant complication or overhead. Consequently, for the purposes of this extrapolation we assume the maximum number of usable processors to be $18N_r$; $27N_r$ processors is not possible because forward and inverse transforms do not occur concurrently.

To find the time T required to complete a simulation of one diffusion time undertaken at resolution N using N_p processors, we extrapolate from a solution undertaken at resolution N_B using N_p^B processors that required a time T_B to compute one diffusion time using the formula

$$T = T_B \times \left(\frac{N}{N^B} \right)^5 \bigg/ \left(\frac{N_p}{N_p^B} \right). \tag{27}$$

The solution with $N_B = 96$ and $N_p^B = 16$ required $T_B = 36.4$ hours to simulate one diffusion time. To simulate $E = 10^{-9}$ requires $N = 3000$ points or spherical harmonics with a maximum number of $18N_r = 54000$ available processors, which, upon substituting into Equation (27) gives $T \approx 13000$ days of computer time, or 1.7×10^{10} CPU hours, a formidable computational task.

Computational requirements for simulating various Ekman numbers based on our N^5 scaling are summarized in Table II. Simulating $E = 10^{-8}$, where $N = 3 \times (10^{-8})^{-1/3} \approx 1390$, gives a maximum usable number of processors $18N \approx 25064$. Using 25 064 processors requires, using (27), over 350 million CPU hours; such an allocation may be available sometime in the next ten years. Simulating $E = 10^{-7}$ is predicted to require 8 million CPU hours; such a simulation is possible at present although it may be out of reach for current allocations of computing time. Simulating $E = 5 \times 10^{-7}$ requires 540 000 hours, hence a few runs could be accomplished within a single allocation of computing time. We therefore conclude that meaningful parameter studies in the Ekman number range $10^{-6} \leq E \leq 10^{-7}$ are possible at the present time.

6. SUMMARY AND DISCUSSION

This paper has analysed the scalability of the pseudospectral method for geodynamo simulations. We have derived scaling laws for operation counts, memory requirements, and communication costs in the asymptotic limit of large problem size and find that the method follows an N^5 scaling. The scalings represent best estimates; if our scalings are optimistic or pessimistic it will become apparent as the simulations proceed to lower values of E.

The excellent efficiency of the pseudospectral method shows that it will compete with any other method that performs comparably in serial mode: only small studies are therefore needed to evaluate relative efficiencies with large clusters. Finite element methods are unlikely to be faster in serial because they require a large number of matrix inversions. Finite difference methods could outperform the pseudospectral method but may require more points for the same resolution everywhere on the sphere. The question of relative accuracy of the various representations is a complex issue that cannot be addressed without a full comparison of the different methods across a wide range of parameters.

For the number of processors presently available, the limiting factor in the pseudospectral method is the spherical transform; memory requirements and communication costs are asymptotically negligible relative to the operation count. Currently memory resources and network interconnects are large/fast enough to demonstrate this on a modest parallel computer. Another issue comes from the parallel implementation: it is currently only possible to use a maximum of $18 N_r$ processors for a given resolution because only the radial domain is split across processors. This limit is unlikely to be threatened in the near future and can be extended by discretizing in the θ direction [52], but this would need to be achieved without compromising the weak scalability.

We have also outlined an ideal geodynamo problem that operates in a parameter regime relevant to the Earth. Using our theoretical and empirical scaling relationships it has been argued that to simulate this problem using the pseudospectral method would require approximately four years of computing time using 54 000 processors. This challenge is so formidable that it is unlikely to be accomplished in the next ten years. Indeed, more aggressive parallelization strategies are likely to be essential in order to permit an increase in the number of processors so as to reduce the elapsed time. Simulations with $10^{-6} \le E \le 10^{-7}$ could be performed at the present time, whereas simulations in the next few years appear capable of moving towards the $E = 10^{-8}$ regime. This would represent a significant advance.

REFERENCES

1. Glatzmaier G, Roberts PH. A three-dimensional convective dynamo simulation with rotating and finitely conducting inner core and mantle. *Physics of the Earth and Planetary Interiors* 1995; **91**:63–75.
2. Sakuraba A, Kono M. Effect of the inner core on the numerical solution of the magnetohydrodynamic dynamo. *Physics of the Earth and Planetary Interiors* 1998; **111**:105–121.
3. Kono M, Sakuraba A, Ishida M. Dynamo simulations and paleosecular variation models. *Philosophical Transactions of the Royal Society A* 2000; **358**:1123–1139.
4. Dormy E, Valet J, Coutillot V. Numerical models of the geodynamo and observational constraints. *Geochemistry, Geophysics, Geosystems* 2000; **1**:1–42.
5. Christensen U, Aubert J, Cardin P, Dormy E, Gibbons S, Glatzmaier GA, Grote E, Honkura Y, Jones C, Kono M, Matsushima M, Sakuraba A, Takahashi F, Tilgner A, Wicht J, Zhang K. A numerical dynamo benchmark. *Physics of the Earth and Planetary Interiors* 2001; **128**:25–34.
6. Takahashi F, Matsushima M, Honkura Y. Simulations of a quasi-Taylor state geomagnetic field including polarity reversals on the Earth simulator. *Science* 2005; **309**:459–461.
7. Christensen U, Aubert J. Scaling properties of convection-driven dynamos in rotating spherical shells and application to planetary magnetic fields. *Geophysical Journal International* 2006; **166**:97–114.
8. Kageyama A, Miyagoshi T, Sato T. Formation of current coils in geodynamo simulations. *Nature* 2008; **454**: 1106–1109.
9. Simitev R, Busse F. Bistability of dipolar dynamos generated by turbulent convection in rotating spherical shells. *Europhysics Letters* 2009; **85**(1):19001.
10. Jacobs JA. *Reversals of the Earth's Magnetic Field* (1st edn). Cambridge University Press: Cambridge, 1984.
11. Dziewonski A, Anderson D. Preliminary reference Earth model. *Physics of the Earth and Planetary Interiors* 1981; **25**:297–356.
12. Gubbins D, Roberts PH. Magnetohydrodynamics of Earth's core. *Geomagnetism*, Jacobs JA (ed.). Academic Press: New York, 1987.

13. Verhoogen J. Heat balance of the Earth's core. *Geophysical Journal of Royal Astronomical Society* 1961;
    4:276–281.
14. Gubbins D. Energetics of the Earth's core. *Journal of Geophysics* 1977; **43**:453–464.
15. Buffett B. Earth's core and the geodynamo. *Science* 2000; **288**:2007–2011.
16. Fearn D. Hydromagnetic flow in planetary cores. *Reports on Progress in Physics* 1998; **61**:175–235.
17. Nimmo F, Price G, Brodholt J, Gubbins D. The influence of potassium on core and geodynamo evolution.
    Geophysical Journal International 2004; **156**:363–376.
18. Costin S, Butler S. Modelling the effects of internal heating in the core and lowermost mantle on the Earth's
    magnetic history. *Physics of the Earth and Planetary Interiors* 2008; **157**:55–71.
19. Braginsky S. Structure of the F layer and reasons for convection in the Earth's core. *Soviet Physics—Doklady*
    1963; **149**:8–10.
20. Bullard E, Gellman H. Homogeneous dynamos and terrestrial magnetism. *Philosophical Transactions of the
    Royal Society A* 1954; **247**:213–278.
21. Bloxham J, Gubbins D. Thermal core-mantle interactions. *Nature* 1987; **325**:511–513.
22. Stacey F. Core properties, physical. *Encyclopedia of Geomagnetism and Paleomagnetism*, Gubbins D, Herrero-
    Bervera E (eds.). Springer: Berlin, 2007; 91–94.
23. Stacey F, Anderson O. Electrical and thermal conductivities of Fe-Ni-Si alloy under core conditions. *Physics of
    the Earth and Planetary Interiors* 2001; **124**:153–162.
24. Buffett B, Matsui H. Core turbulence. *Encyclopedia of Geomagnetism and Paleomagnetism*, Gubbins D, Herrero-
    Bervera E (eds.). Springer: Berlin, 2007; 101–103.
25. Moffatt H. Magnetic field generation in electrically conducting fluids. *Cambridge Monographs on Mechanics
    and Applied Mathematics*. Cambridge University Press: Cambridge, 1978.
26. Vocadlo L, Alfe D, Gillan M, Price G. The properties of iron under core conditions from first principles
    calculations. *Physics of the Earth and Planetary Interiors* 2003; **140**:101–125.
27. Gubbins D. Dimensional analysis and timescales for the geodynamo. *Encyclopedia of Geomagnetism and
    Paleomagnetism*, Gubbins D, Herrero-Bervera E (eds.). Springer: Berlin, 2007; 287–300.
28. Kuang W, Bloxham J. Numerical modelling of magnetohydrodynamic convection in a rapidly rotating spherical
    shell: Weak and strong field dynamo action. *Journal of Computational Physics* 1999; **153**:51–81.
29. Glatzmaier G. Geodynamo simulations—How realistic are they? *Annual Review of Earth and Planetary Sciences*
    2002; **30**:237–257.
30. Zhang K, Jones C. The effect of hyperdiffusion on geodynamo models. *Geophysics Research Letters* 1997;
    24:2869–2872.
31. Gubbins D. The Rayleigh number for convection in the Earth's core. *Physics of the Earth and Planetary Interiors*
    2001; **128**:3–12.
32. Jones C. Convection-driven geodynamo models. *Philosophical Transactions of the Royal Society A* 2000; **358**:
    873–897.
33. Willis A, Sreenivasan B, Gubbins D. Thermal core-mantle interaction: Exploring regimes for 'locked' dynamo
    action. *Physics of the Earth and Planetary Interiors* 2007; **165**:83–92.
34. Abramowitz M, Stegun I. *Handbook of Mathematical Functions* (2nd edn). Dover: New York, 1965.
35. Orszag S. Numerical simulation of incompressible flows within simple boundaries. I. Galerkin (spectral) methods.
    Studies in Applied Mathematics 1971; **50**:293–327.
36. Roberts PH. On the thermal instability of a rotating fluid sphere containing heat sources. *Philosophical Transactions
    of the Royal Society A* 1968; **263**:93–117.
37. Busse F. Thermal instabilities in rapidly rotating systems. *Journal of Fluid Mechanics* 1970; **44**:441–460.
38. Zhang K, Liao X. A new asymptotic method for the analysis of convection in a rapidly rotating sphere. *Journal
    of Fluid Mechanics* 2004; **518**:319–346.
39. Zhang K. On coupling between the Poincaré equation and the heat equation. *Journal of Fluid Mechanics* 1994;
    268:211–229.
40. Stewartson K. On almost rigid rotations, part 2. *Journal of Fluid Mechanics* 1966; **26**:131–144.
41. Hollerbach R. Instabilities of the Stewartson layer Part 1. The dependence on the sign of *Ro*. *Physics of the
    Earth and Planetary Interiors* 1995; **87**:171–181.
42. Courant R, Friedrichs K, Lewy H. On the partial difference equations of mathematical physics. *IBM Journal*
    1928; **100**:32–74.
43. Davidson P. *Introduction to Magnetohydrodynamics*. Cambridge University Press: Cambridge, 2001.
44. Backus G. A class of self-sustaining dissipative spherical dynamos. *Annals of Physics* 1958; **4**:372–447.
45. Childress S. Théorie magnetohydrodynamique de l'effet dynamo. *PhD Thesis*, Paris Départment Méchanique de
    la Faculté des Sciences, 1969.
46. Bloxham J, Jackson A. Time-dependent mapping of the magnetic field at the core-mantle boundary. *Journal of
    Geophysical Research* 1992; **97**:19537–19563.
47. Glatzmaier G, Clune T. Computational aspects of geodynamo simulations. *Computing in Science and Engineering*
    2000; **2**:61–67.
48. Chan K, Li L, Liao X. Modelling the core convection using finite element and finite difference methods. *Physics
    of the Earth and Planetary Interiors* 2006; **157**:124–138.
49. Press W, Teukolshy S, Vetterline W, Flannery B. *Numerical Recipies in Fortran*: *The Art of Scientific Programming*.
    Cambridge University Press: Cambridge, 1992.

50. Young R. Finite amplitude thermal convection in a spherical shell. *Journal of Fluid Mechanics* 1974; **63**: 695–721.
51. Strassen V. Gaussian elimination is not optimal. *Numerische Mathematik* 1969; **13**:354–356.
52. Clune T, Elliott J, Miesch M, Toomre J. Computational aspects of a code to study rotating turbulent convection in spherical shells. *Parallel Computing* 1999; **25**:361–380.
53. Foster I, Worley P. Parallel algorithms for the spectral transform method. *SIAM Journal on Scientific Computing* 1997; **18**:806–837.