Ocean Modelling 182 (2023) 102160

Contents lists available at ScienceDirect

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

Scalable self attraction and loading calculations for unstructured ocean tide models

Steven R. Brus [a,*], Kristin N. Barton [b], Nairita Pal [c], Andrew F. Roberts [c], Darren Engwirda [c], Mark R. Petersen [d], Brian K. Arbic [e], Damrongsak Wirasaet [f], Joannes J. Westerink [f], Michael Schindelegger [g]

[a] Mathematics and Computer Science, Argonne National Laboratory, Lemont, IL, USA
[b] Department of Physics, University of Michigan, Ann Arbor, MI, USA
[c] Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, USA
[d] Computer, Computational and Statistical Science Division, Los Alamos National Laboratory, Los Alamos, NM, USA
[e] Department of Earth and Environmental Sciences, University of Michigan, Ann Arbor, MI, USA
[f] Department of Civil and Environmental Engineering and Earth Sciences, University of Notre Dame, Notre Dame, IN, USA
[g] Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany

ARTICLE INFO

Keywords:
Self attraction and loading
Tidal modeling
Unstructured ocean model
High performance computing

ABSTRACT

Self attraction and earth-loading effects are important for accurately modeling global tides. A common approach of handling this forcing is to expand mass anomalies into spherical harmonics, which are scaled by load Love numbers to account for elastic earth deformation. We investigate two different approaches to perform these calculations for ocean models that employ unstructured meshes and distributed memory parallelization. The first approach leverages a highly efficient spherical harmonics library, but requires all-to-one and one-to-all communications and interpolation operations between the unstructured and a structured mesh. This approach is compared to a parallel algorithm that computes the spherical harmonic transformations directly on the unstructured mesh with an all-reduce communication. Our results show that although the unstructured mesh calculations are more expensive, the scalability of the unstructured mesh approach allows for more efficient spherical harmonics transforms for high-resolution meshes and large processor counts. This methodology enables the efficient inclusion of tidal dynamics large-scale Earth system model simulations.

1. Introduction

Movement and exchanges of mass across Earth's geophysical fluids have both short-term and long-term effects on sea levels (Stammer et al., 2013). Ocean tides and other long period waves, such as storm surges and tsunamis, represent variations in mass distribution in the ocean on the scale of hours to days. The dynamically relevant effects of these mass redistributions are typically collected in a term known as the self attraction and loading (SAL) term (Hendershott, 1972). The SAL terms account for the Earth's elastic deformation in response to oceanic motion (e.g., seafloor depression under a high tide) and the change in gravitational potential due to the self-gravity of the so-deformed Earth as well as of the tidal mass field itself. Sea levels also respond to changes in land-based water storage (Tamisiea et al., 2010) and glacial isostatic adjustment (GIA) (Farrell and Clark, 1976) on seasonal and centurial-timescales, respectively. At global scales, inclusion of SAL forcing is necessary for accurate tidal model results and can change tidal amplitudes by up to 20% (Gordeev et al., 1977). Ocean bottom pressure variations in response to broadband atmospheric forcing also cause changes in SAL (Vinogradova et al., 2015). One method of calculating the SAL terms involves scaling the spherical harmonic decomposition of the mass anomaly, i.e., the surface elevation in the case of barotropic tides, by load Love numbers (Ray, 1998).

Spherical harmonic transforms (SHTs) are often used as a global basis for solving partial differential equations. Due to their global nature, they are most often computed on regular meshes, which are designed to optimize their calculation via fast transformation methods (Driscoll and Healy, 1994; Mohlenkamp, 1999; Suda and Takami, 2001; Healy et al., 2003). Therefore, the consideration of SAL effects in unstructured parallel ocean models presents a unique use-case that has not been addressed in the literature. Typically, SAL has been incorporated in tidal models by either a scalar approximation (Stepanov and Hughes, 2004) or by taking pertinent expansions at individual spectral lines from other highly-accurate tidal models (Pringle et al., 2021). The former approach is an approximation that is much less accurate than more formal "in-line" treatments of SAL (Barton et al.,

* Corresponding author.
E-mail address: sbrus@anl.gov (S.R. Brus).

https://doi.org/10.1016/j.ocemod.2023.102160
Received 12 May 2022; Received in revised form 10 December 2022; Accepted 2 January 2023
Available online 4 January 2023
1463-5003/

2022; Shihora et al., 2022), while the latter approach neglects the effect of non-periodic processes. The formal "in-line" treatment of SAL effects requires computationally expensive spherical harmonic calculations. Currently, there are no existing packages that use distributed memory parallelism to perform these calculations efficiently on unstructured meshes.

Unstructured ocean models such as the Model for Prediction Across Scales-Ocean (MPAS-Ocean) (Ringler et al., 2013; Petersen et al., 2015) have been developed to take advantage of localized mesh resolution (Hoch et al., 2020). The flexibility of variable resolution meshes is particularly advantageous for tidal modeling because they can selectively resolve coastal tidal dynamics and relevant bathymetric gradients without requiring global high-resolution (Pringle et al., 2021). For example, mesh resolution can be specified via an *a priori* estimate of the tidal wavelength (Conroy et al., 2012; Roberts et al., 2019):

$$h = \frac{T}{N_{WL}}\sqrt{gb}, \tag{1}$$

where $T = 12.5$ h is the period of the M_2 tide, N_{WL} is the number of grid cells per wavelength, b is the bathymetric depth, and g is the acceleration due to gravity. This means that the mesh resolution can be made to vary with the relevant tidal wavelength scales. In shallow coastal regions, tidal wave speeds become slower, leading to a shortening of the wavelength. In general, higher mesh resolution is employed in coastal areas to resolve these smaller scales, while coarser resolution can be used in the deep ocean where wavelengths are larger.

Many packages are available to compute spherical harmonics on structured (Gaussian) meshes. One example is the `SHTns` library used in this work (Schaeffer, 2013). The library can be readily adopted for ocean models that are based on structured meshes (Schindelegger et al., 2018). However, in order for global unstructured ocean models to use these packages, interpolations to and from a Gaussian mesh are required. This step introduces interpolation errors, thus degrading the accuracy of the SHT calculations and requiring higher-order transformations to achieve a given level of accuracy. A possible remedy is employing algorithms for spherical harmonic estimation using non-uniform data, one of which (Kunis and Potts, 2003) has been implemented in the `NFFT3` library (Keiner et al., 2009). While this has proved useful for analysis of climate data from unstructured meshes (Cavanaugh et al., 2017), it has not been used in computing SAL for tides. In addition, many spherical harmonics packages (including `SHTns` and `NFFT3`) lack distributed parallelism, which is widely employed in both structured and unstructured ocean models.

This paper demonstrates that for unstructured global ocean models, calculating SAL directly on the unstructured mesh has advantages in terms of both accuracy and performance. In particular, computing the SHT directly on the unstructured mesh avoids mesh interpolation errors and improves convergence rates. In addition, direct unstructured mesh SHT computations can make use of the domain decompositions and distributed memory parallelism already employed in solving the ocean model governing equations. This capability provides improved performance and scalability at large processor counts. The efficiency of the SAL calculation is an important consideration for the inclusion of tides in coupled Earth system models (ESMs), such as the Energy Exascale Earth System Model (E3SM) (Golaz et al., 2019). In these models, the scalability of the ocean model relative to other component models impacts the overall throughput of climate simulations.

The remainder of the paper is organized as follows. First, we give an overview of SHTs and the scaling factors involved in the SAL calculation, followed by a description of the serial and parallel versions we use to compute the SAL term for an unstructured ocean model. Next, we test the two algorithms on two different idealized cases: one with a uniform mesh and another with a variable resolution mesh. We then assess the accuracy and performance improvements for the parallel approach. Afterwards, we show validation and performance results for tidal simulations using a series of uniform resolution (icosahedral) meshes and a variable resolution mesh. Finally, we provide a discussion of the results and offer conclusions.

2. Methods

In this section we outline the calculation of the SAL term, ζ_{SAL}. For tides, the calculation involves a forward SHT of the sea surface height, ζ. The resulting spherical harmonic coefficients are then scaled by load Love numbers and an inverse SHT is performed to get the ζ_{SAL} field. The SAL term appears in the pressure gradient term of the momentum balance in the shallow water equations:

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (H\mathbf{u}) = 0, \tag{2}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + f\mathbf{k} \times \mathbf{u} = -g\nabla(\zeta - \zeta_{EQ} - \zeta_{SAL}) + F, \tag{3}$$

where $\mathbf{u}$ is the depth-averaged velocity vector, H is the total water column height, f is the Coriolis parameter, g is the gravitational acceleration, ζ is the sea surface height, ζ_{EQ} is the equilibrium tidal forcing term, and F are additional stress terms.

In unstructured ocean models that use distributed memory parallelization (i.e. MPI), the global mesh is decomposed into "subdomains" (Karypis and Kumar, 1998). Each subdomain contains a partition of mesh cells that is assigned to a given computational core and is computed in parallel alongside the other subdomains from the global mesh. Typically, only point-to-point communication between cores is necessary to update cells along subdomain interfaces in order to compute local gradients, for example, in Eq. (3). However, the global nature of computing SHTs requires a collective communication among all processors. The following subsections describe how the ζ_{SAL} term can be computed under this parallelization paradigm using two different strategies. The first, referred to throughout as the "serial" algorithm, uses a fast SHT library but requires interpolations between the unstructured mesh and a Gaussian mesh. Since the fast SHT library does not use distributed memory parallelism, global information must be gathered to and scattered from a single core for the SHT calculation to take place. The second approach we describe is the "parallel" algorithm, which computes the SHT directly on unstructured meshes using a collective all-to-all communication.

2.1. Spherical harmonics

In a barotropic model, the mass anomaly due to tidal motion can be assessed through the sea surface height variable. The sea surface height can be written as a spherical harmonic expansion:

$$\zeta(\theta, \phi) \approx \sum_{n=0}^{N} \sum_{m=-n}^{n} \zeta^{(n,m)} Y^{(n,m)}(\theta, \phi). \tag{4}$$

Here, θ is the co-latitude, ϕ is the longitude, $\zeta^{(n,m)}$ are the expansion coefficients and $Y^{(n,m)}$ are the spherical harmonic functions of degree n and order m defined as

$$Y^{(n,m)}(\theta, \phi) = P^{(n,m)}(\cos(\theta))e^{im\phi}. \tag{5}$$

The $P^{(n,m)}$ functions are the associated Legendre polynomials and can be computed with the recursion relationship:

$$P^{(m,m)}(\cos(\theta)) = a^{(m,m)} \sin^m(\theta), \tag{6}$$

$$P^{(m+1,m)}(\cos(\theta)) = a^{(m+1,m)} \cos(\theta) P^{(m,m)}(\cos(\theta)), \tag{7}$$

$$P^{(n,m)}(\cos(\theta)) = a^{(n,m)} \cos(\theta) P^{(n-1,m)}(\cos(\theta)) + b^{(n,m)} P^{(n-2,m)}(\cos(\theta)). \tag{8}$$

The coefficients in the recursion relationship are defined as

$$a^{(m,m)} = \sqrt{\frac{1}{4\pi} \prod_{k=1}^{m} \frac{2k+1}{2k}}, \tag{9}$$

$$a^{(n,m)} = \sqrt{\frac{(2n-1)(2n+1)}{(n-m)(n+m)}}, \tag{10}$$

$$b^{(n,m)} = \sqrt{\frac{(2n+1)(n+m-1)(n-m-1)}{(n-m)(n+m)(2n-3)}}. \tag{11}$$

Special treatment to prevent overflow issues near the poles is required when computing the associated Legendre polynomials for high orders ($N > 2000$) (Holmes and Featherstone, 2002). However, for the purposes of computing SAL, we are only concerned with much lower orders $N = O(100)$. In Section 3.3, we show that the accuracy of our tidal simulations does not increase for SHT orders beyond $N = 100$.

2.2. Forward transformation

The spherical harmonic coefficients can be computed via the forward transformation

$$\zeta^{(n,m)} = \int_0^{2\pi} \int_0^{\pi} \zeta(\theta,\phi) Y^{(n,m)}(\theta,\phi) \sin(\theta) d\theta d\phi. \tag{12}$$

For real-valued functions (such as ζ) the property, $\zeta^{(n,-m)} = \overline{\zeta^{(n,m)}}$, can reduce the number of spherical harmonic coefficients that need to be calculated. On a structured, Gaussian mesh, this is typically computed as a Fourier transform for the ϕ integral

$$\zeta^{(m)}(\theta) = \int_0^{2\pi} \zeta(\theta,\phi) e^{im\phi} d\phi, \tag{13}$$

and a Legendre transform in the θ direction

$$\zeta^{(n,m)} = \int_0^{\pi} \zeta^{(m)}(\theta) P^{(n,m)}(\cos(\theta)) \sin(\theta) d\theta. \tag{14}$$

This approach has several advantages in that a fast Fourier transform algorithm can be used to compute Eq. (13). The latitude points are also positioned at Gauss–Legendre nodes to facilitate the computation of Eq. (14). In addition, due to the symmetric nature of spherical harmonics about $\phi = \pi/2$, the operation count in computing (14) can be halved. However, in the context of unstructured meshes these simplifications are no longer explicitly possible and a different strategy for computing Eq. (12) must be used.

2.3. SAL scaling

Once the spherical harmonic coefficients, $\zeta^{(n,m)}$, are calculated, they are multiplied with a scaling factor composed of the load Love numbers (Munk and MacDonald, 1960), k'_n and h'_n:

$$\zeta_{SAL}^{(n,m)} = \frac{3\rho_w(1 + k'_n - h'_n)}{\rho_e(2n+1)} \zeta^{(n,m)}. \tag{15}$$

In this expression ρ_w is the average density of seawater and ρ_e is the average density of Earth. The factor $(1+k'_n-h'_n)$ combines the three SAL processes: the self-gravitational attraction of the water mass toward the load (1), the solid Earth deformation under tidal loading (h'_n), and the change in potential due to the load deformation (k'_n).

2.4. Inverse transformation

The final SAL field is computed from the scaled $\zeta_{SAL}^{(n,m)}$ coefficients by the inverse spherical harmonic transformation:

$$\zeta_{SAL}(\theta,\phi) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \zeta_{SAL}^{(n,m)} Y^{(n,m)}(\theta,\phi), \tag{16}$$

$$= \sum_{n=0}^{N} \sum_{m=1}^{n} \left(\zeta_{SAL}^{(n,m)} Y^{(n,m)}(\theta,\phi) + \overline{\zeta_{SAL}^{(n,m)}}\,\overline{Y^{(n,m)}}(\theta,\phi) \right) + \sum_{n=0}^{N} \zeta_{SAL}^{(n,0)} Y^{(n,0)}(\theta,\phi), \tag{17}$$

$$= \sum_{n=0}^{N} \sum_{m=0}^{n} \mu \left(\Re\left(\zeta_{SAL}^{(n,m)}\right) \Re\left(Y^{(n,m)}(\theta,\phi)\right) - \Im\left(\zeta_{SAL}^{(n,m)}\right) \Im\left(Y^{(n,m)}(\theta,\phi)\right) \right), \tag{18}$$

$$= \sum_{n=0}^{N} \sum_{m=0}^{n} \mu \left(\Re\left(\zeta_{SAL}^{(n,m)}\right) \cos(m\phi) - \Im\left(\zeta_{SAL}^{(n,m)}\right) \sin(m\phi) \right) P^{(n,m)}(\cos(\theta)), \tag{19}$$

where $\mu = 2$ if $m > 0$, $\mu = 1$ if $m = 0$.

2.5. Serial unstructured mesh calculations

One option for performing spherical harmonic calculations on unstructured meshes is to perform an interpolation of the ζ field onto a Gaussian mesh, perform the SHT and SAL scaling on the Gaussian mesh, and then interpolate the ζ_{SAL} field back to the unstructured mesh. This allows for the use of highly optimized SHT libraries which employ fast Fourier transforms on the Gaussian mesh. However, many of these libraries do not support the distributed parallelism commonly used in ocean models. For this study, we have implemented the interpolation-based SHT using the following sequence:

1. The ζ values from each subdomain are gathered onto a single core
2. The ζ field is interpolated from the unstructured mesh onto a Gaussian mesh
3. The forward SHT is performed with a library call to `SHTns`
4. The SAL scaling is performed
5. An inverse SHT is done with a library call to `SHTns`
6. The resulting ζ_{SAL} field is interpolated from the Gaussian mesh back to the unstructured mesh
7. The ζ_{SAL} values are scattered back to the subdomains

The computations in steps 2–6 are done on a single processor. The bi-linear interpolation weights between the unstructured and Gaussian meshes are pre-computed using the Spherical Coordinate Remapping and Interpolation Package (SCRIP) (Jones, 1999). As will be shown in the results section, this approach is efficient for low core counts, but does not scale at the high processor counts needed for high-fidelity simulations.

2.6. Parallel unstructured mesh calculations

To perform the SHT calculations directly on an unstructured mesh, optimizations that rely on separating integrals in the θ and ϕ dimensions, equatorial symmetry, and grid nodes that correspond with quadrature points are not applicable. Therefore, the full integration from Eq. (12) is required over the domain. We approximate the forward transformation integrals by taking the values of ζ and Y to be constant over each mesh cell and summing together the integrals over each cell as follows:

$$\zeta^{(n,m)} = \sum_{k=1}^{K} \int_{\Omega_k} \zeta_k Y_k^{(n,m)} d\Omega_k, \tag{20}$$

$$= \sum_{k=1}^{K} \zeta_k Y_k^{(n,m)} A_k. \tag{21}$$

Here, K is the total number of mesh cells and Ω_k and A_k represent the domain and area of cell k, respectively. Since each computational core is assigned a subdomain of mesh cells, the integration can be separated into a local sum of integrals over the subdomain cells and a global sum of the subdomain integral contributions:

$$\zeta^{(n,m)} = \sum_{d=1}^{D} \left(\sum_{k=1}^{K_d} \zeta_k Y_k^{(n,m)} A_k \right). \tag{22}$$

The term in parentheses represents the local sum over the K_d cells in subdomain d and the outer sum represents the global sum of the D subdomain contributions. This global sum is implemented as a MPI_AllReduce operation. Following the MPI_AllReduce, all cores have the resulting global sum values for the $\zeta^{(n,m)}$ coefficients. Each core can then compute the SAL scaling in Eq. (15) and perform the inverse transformation in Eq. (17). Following Schaeffer (2013), the $P^{(n,m)}$ function values are not pre-computed and stored to prevent

memory bandwidth performance issues. Instead, they are computed on-the-fly using the recurrence relationship in Eqs. (6)–(8), taking advantage of the vectorization capabilities of modern CPUs. We also employ cache blocking among the forward and inverse computations in order to maintain good temporal cache locality.

2.7. Error assessment

There are four different types of error involved in the SHT calculations:

1. Truncation error
2. Integration error
3. Aliasing error
4. Interpolation error

For the parallel algorithm, only the truncation, integration, and aliasing errors are relevant, since no interpolation is necessary. Truncation error refers to the error in approximating

$$\zeta(\theta,\phi) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} \zeta^{(n,m)} Y^{(n,m)}(\theta,\phi), \tag{23}$$

$$\approx \sum_{n=0}^{N}\sum_{m=-n}^{n} \zeta^{(n,m)} Y^{(n,m)}(\theta,\phi). \tag{24}$$

This expansion is known to have spectral convergence $O(N^{-2q})$ for functions in $L_2(H^q(\mathbb{S}^2))$, where q is a non-negative integer and $H^q(\mathbb{S}^2))$ is the Sobolev space on the sphere (Frank et al., 2016). Integration error is introduced in calculating the $\zeta^{(n,m)}$ coefficients by the piecewise constant assumption used in Eq. (21). Because the integration is approximated by a mid-point rule, for a grid spacing of h, the error is of $O(h^2)$. For low orders of N, while the truncation error dominates the integration error, spectral convergence is achieved. Beyond a certain N order, the integration error will dominate and the convergence will be reduced to $O(h^2)$. Finally, as the N order increases beyond a certain point for a given grid resolution, aliasing errors occur as higher-frequency spherical harmonic modes are no longer well-resolved at a given grid resolution. This causes the approximation to diverge at high N.

For the serial algorithm, an $O(h^2)$ error is introduced due to the (bi)linear interpolation between the unstructured and Gaussian grids prior to the SHT. Therefore, these errors will dominate the truncation error and spectral convergence will not be achieved. In the case of the serial algorithm, the N order is set based on the resolution of the Gaussian grid, meaning that aliasing errors do not cause the approximation to diverge.

3. Results

In this section, the performance of the serial and parallel STH algorithms are compared using analytical functions on uniform and variable resolution meshes. In Sections 3.1 and 3.2, ζ is specified by an analytical function and errors are computed with respect to a truncated spherical harmonic series approximation to ζ. The error measure used is the root mean squared (RMS) error defined as:

$$E_{RMS} = \sqrt{\frac{\int_\Omega (\sum_{n=0}^{N}\sum_{m=-n}^{n} \zeta^{(n,m)} Y^{(n,m)} - \zeta)^2 d\Omega}{\int d\Omega}}. \tag{25}$$

Performance results are also shown for a global tidal simulation in Section 3.3. Following Pringle et al. (2021), the area-weighted RMS error for a tidal constituent, RMSE_t, is computed using the TPXO9-atlas v5 database (Egbert and Erofeeva, 2002) as the "ground-truth" solution:

$$D^2 = \frac{1}{2}(A_{TPXO}^2 - A_{MPAS}^2) - A_{TPXO}A_{MPAS}\cos(\varphi_{TPXO} - \varphi_{MPAS}), \tag{26}$$

Table 1
Uniform mesh size comparison.

Mesh	Number of cells
QU60	165,093
QU30	658,642
QU15	2,629,794

$$\mathrm{RMSE}_t = \sqrt{\frac{\int_\Omega D^2 d\Omega}{\int_\Omega d\Omega}}. \tag{27}$$

Here, A is the amplitude and φ is the Greenwich phase lag for a given tidal constituent.

All simulations were performed on a 460 node machine hosted at Pacific Northwest National Laboratory. Each node contains dual 20 core Intel Xeon Gold 6148 (Skylake) CPUs with 192 GB DRAM. The nodes are connected with the Intel OmniPath interconnect. The code was complied to use the AVX-512 instruction set. Wall clock times for all analytical function cases use the average of 100 forward/inverse transformations and are taken as the minimum of three separate runs. For the serial algorithm, the number of latitude nodes in the Gaussian mesh is related to the SH order by $N = N_\theta/2 - 1$ to avoid computing higher order spherical harmonics, which would be aliased at the Gaussian grid resolution. All meshes used in this study are produced with the `JIGSAW` mesh generation package (Engwirda, 2017).

3.1. Uniform resolution test case

The parallel and serial SHT implementations are compared by computing the errors between an order N SHT approximation and the analytical function

$$\zeta(\theta,\phi) = e^{-\frac{1}{2}\left(\frac{d(\theta,\phi)}{\sigma}\right)^2}. \tag{28}$$

In this equation, $\sigma = 800$ km and d is the minimum Haversine distance to the curve

$$x_s(t) = R_E\cos(t)\cos(c), \tag{29}$$

$$y_s(t) = R_E\sin(t)\cos(c), \tag{30}$$

$$z_s(t) = -R_E\sin(c), \tag{31}$$

with $c = \tan^{-1}(at)$, where $a = 0.15$ and $-L \geq t \geq L$ with $L = 5\pi$. The Earth's radius, $R_E = 6371.22$ km. The distance, d, is given by

$$d(\theta,\phi) = \min_{\forall t\in[-L,L]} 2R_E\sin^{-1} \times \left(\sqrt{\sin^2\left(\frac{\theta-\theta_s(t)}{2}\right) + \cos(\theta)\cos\left(\theta_s(t)\right)\sin^2\left(\frac{\phi-\phi_s(t)}{2}\right)}\right), \tag{32}$$

where

$$\theta_s(t) = \sin^{-1}\left(\frac{z_s(t)}{\sqrt{x_s(t)^2 + y_s(t)^2 + z_s(t)}}\right), \tag{33}$$

$$\phi_s(t) = \tan^{-1}\left(\frac{y_s(t)}{x_s(t)}\right). \tag{34}$$

This fuction is shown in Fig. 1. Three different quasi-uniform mesh resolutions are used: 60 km, 30 km, and 15 km. These are referred to as QU60, QU30, and QU15, respectively. The number of cells in each mesh is shown in Table 1.

The error convergence for the two algorithms is depicted in Fig. 2. These results show that for the parallel algorithm, the convergence curve experiences a region of spectral convergence up to $\sim N = 40$. In this region, the truncation error of the SHT dominates the integration error. For $N > 40$, the $O(h^2)$ integration error dominates the SHT truncation error and the convergence curve becomes second order. As the SHT order increases, aliasing errors begin to limit the convergence of the SHT. These aliasing errors occur at higher N as the mesh resolution increases, as expected.

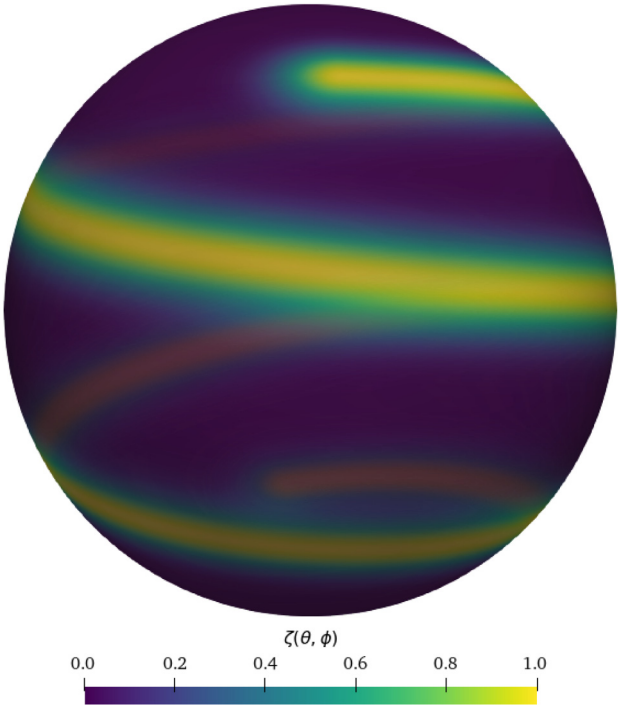

Fig. 1. Test function for uniform resolution case.

For the serial algorithm, the $O(h^2)$ error due to the interpolation from the unstructured to the Gaussian grid prevents the SHT error from achieving a region of spectral convergence. The convergence curves for all three mesh resolutions remain second order, resulting in a discrepancy between the N order required to achieve a given E_{RMS} level. This discrepancy widens as E_{RMS} decreases. The errors for the serial algorithm achieve the same error level as the parallel case at a higher N order due to the error introduced by interpolation onto the structured mesh. Since N is related to the structured grid resolution by $N = N_\theta/2 - 1$, the interpolation error decreases at a second-order rate as N increases. Although the serial errors do not diverge rapidly, as in the parallel case, they asymptotically approach a similar minimum error level for a given mesh resolution.

The effect of cache blocking in the parallel algorithm is shown in Fig. 3. As the calculation of the forward and inverse SHTs are done for smaller blocks of cells, greater temporal locality of memory access is achieved. This is particularly important for large meshes with low processor counts, since optimal cache blocking can lead to a 4× performance increase on the Skylake architecture. Cache blocking becomes less important when high core counts are used, since this naturally translates into smaller subdomains. However, it is critical for the block size to be under 2000 cells. A block size of 300–600 cells per block was found to achieve the best results.

The parallel performance for the QU60, QU30, and QU15 meshes is shown in Figs. 4, 5, and 6, respectively. These plots compare the parallel scalability and speedup for both the serial and parallel algorithms at several RMS error levels. These error levels are: 5×10^{-1}, 1×10^{-2}, 5×10^{-2}, 1×10^{-3}, 1×10^{-4}, and 2.5×10^{-5} for the QU15 mesh. These error levels are indicated by the dotted horizontal lines in plot (a) of Figs. 4–6 and each is represented by a different marker shape across plots (a)–(c). The coarser QU60 and QU30 meshes are compared at these same levels, up to the lowest error achieved at that resolution. As discussed above, the spherical harmonic order required to achieve a given error level is lower for the parallel algorithm than for the serial algorithm due to improved convergence. The scaling for the parallel algorithm is near ideal except for the coarsest mesh (QU60) beyond 2160 cores. This corresponds to around 80 cells per core, when the communication overhead begins to dominate the SHT computations. As expected, the serial algorithm does not scale with increasing core count. The gather and scatter communication overhead increases very little as the core count increases.

Table 2
Uniform mesh maximum speedup.

Error level	QU60	QU30	QU15
5×10^{-2}	10.49	26.10	62.83
1×10^{-2}	6.03	11.16	33.18
5×10^{-3}	5.02	8.60	28.11
1×10^{-3}	6.96	12.62	33.29
1×10^{-4}	–	23.57	21.97
2.5×10^{-5}	–	–	25.59

The speedup ratio between the parallel algorithm and serial algorithm is shown in Table 2. In general, the speedup decreases as the error level decreases. Since the expense of the gather/scatter operations for the serial algorithm is constant with regard to error level, at higher error levels the communication overhead takes a larger proportion of the overall compute time. As the error level decreases, the gather/scatter communication becomes increasingly negligible. This means that the parallel algorithm has an extra advantage for higher error levels, which translates to the largest speedups at the highest 5×10^{-1} error level for each mesh. The speedups tend to decrease for the next two error levels and then begin to increase again at the lowest errors, as the discrepancy between the N orders becomes larger. Speedups also generally increase as the mesh resolution increases for a given error level. For the QU60 mesh, the speedups range between 5 to 10.5 and increase to between 11 to 26 for QU30 and 22 to 63 for QU15. The break-even points for a speedup of 1 are usually around 200 cores across all the QU meshes used. Typically, realistic ocean simulations would be run with $O(1000)$ cores.

3.2. Variable resolution test case

We now evaluate the two methods on variable resolution meshes, using an oscillatory function with a period that varies with the mesh resolution. The mesh resolution varies with latitude and is refined in a band around the equatorial region with low resolution covering the rest of the sphere. The function used for the for this test case is

$$\zeta(\theta,\phi) = \omega_2\left(\omega_2(\Theta_{EQ}+\Phi_{EQ}) + \omega_1(\Theta_{LR}+\Phi_{LR})\right) + \omega_1\Theta_{LR}, \tag{35}$$

where

$$\Theta_{EQ} = \sin\left(\frac{2\pi R_E N_{WL}}{h_{EQ}}\theta\right), \tag{36}$$

$$\Theta_{LR} = \sin\left(\frac{2\pi R_E N_{WL}}{h_{LR}}\theta\right), \tag{37}$$

$$\Phi_{EQ} = \sin\left(\frac{2\pi R_E N_{WL}}{h_{EQ}}\phi\right), \tag{38}$$

$$\Phi_{LR} = \sin\left(\frac{2\pi R_E N_{WL}}{h_{LR}}\phi\right). \tag{39}$$

The value of N_{WL} controls the approximate number of grid cells per period. For this test case N_{WL} was taken to be 30. The weight functions, which blend the high and low resolution regions, are

$$\omega_1 = \frac{1}{2}\left(\tanh\left(\frac{|\theta| - t_s - \frac{1}{2}t_w}{\frac{1}{5}t_w}\right) + 1\right), \tag{40}$$

$$\omega_2 = 1 - \omega_1. \tag{41}$$

The transition start, $t_s = \pi/180$, and the transition width, $t_w = \pi/9$. The mesh resolution is specified as

$$h(\theta) = h_{EQ}\omega_2 + h_{LR}\omega_1. \tag{42}$$

The mesh spacing in the low resolution region is h_{LR} =120 km. Three different values are used for the high resolution equatorial region,

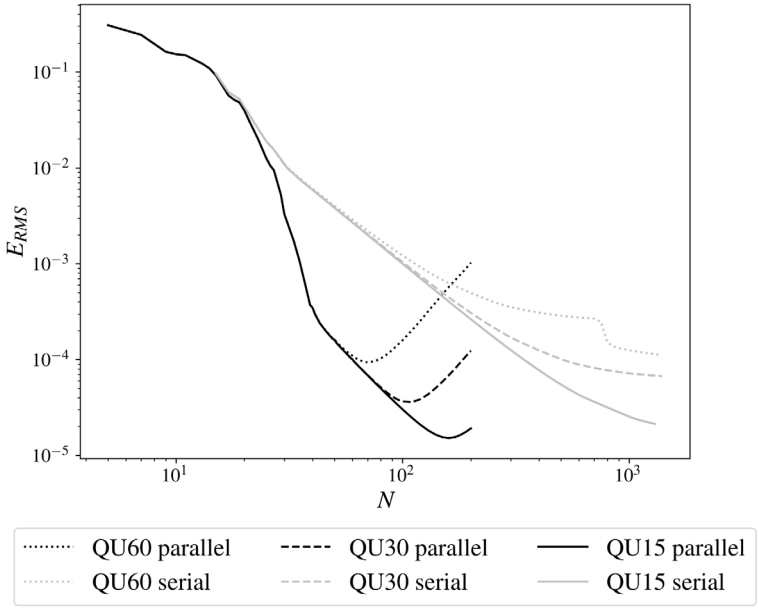

Fig. 2. Convergence for serial and parallel SHT algorithms on quasi-uniform resolution mesh.

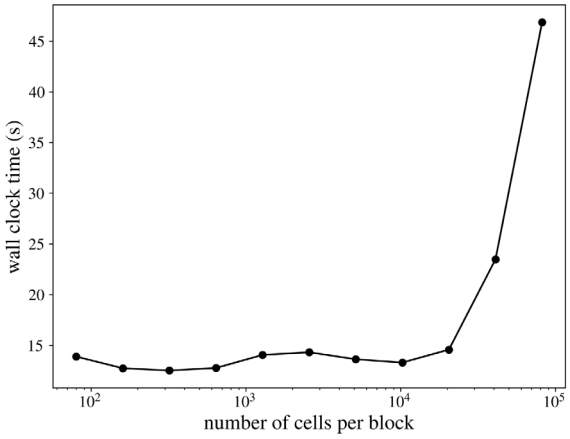

Fig. 3. Effect of cache blocking for parallel SHT algorithm.

h_{EQ} = 60, 30, 15 km. The high resolution region corresponds to ∼ 20% of the globe, which roughly corresponds to the area of the ocean with depths less than 2.5 km. This would represent the approximate area of a tide mesh that would be refined to resolve tides on the continental shelves. The number of cells in each of these meshes is shown in Table 3. The function and mesh resoluiton for the ER120-15 mesh are shown in Fig. 7.

The convergence for the serial and parallel algorithms across a range of truncation orders for the three meshes is shown in Fig. 8. The parallel algorithm experiences very steep convergence as it begins to resolve the fine scale features in the high resolution region. As the resolution of the refined region increases, the slope of the convergence curve steepens. The convergence of the serial algorithm is much slower due to the interpolation to the Gaussian mesh. The steep convergence of the parallel algorithm translates into large efficiency gains for higher levels of accuracy. Due to the structured nature of the Gaussian mesh, the mesh resolution required to resolve the high frequency wavelengths in the equatorial region must be used globally. This leads to a much larger mesh, which increases the number of calculations required to compute the SHTs compared to the variable resolution mesh.

The scalability and speedup of the parallel algorithm are shown in Figs. 9–11 for the ER120-60, ER120-30, and ER120-15 meshes. The serial and parallel algorithms are compared for four different levels of accuracy: 1×10^{-1}, 5×10^{-2}, 1×10^{-2}, and 5×10^{-3}. As in the previous section, these error levels are denoted by the dotted lines in the (a) panel of Figs. 9–11 and each is represented by a different marker shape across plots (a)–(c). Due to the smaller overall size of these meshes, the parallel algorithm does not scale as well for the ER120-60 and ER120-30 meshes as it does for the larger QU30 and QU15 meshes. The scalability limits for these meshes are consistent with the approximate 80 cells per core limit found for the QU60 meshes.

Table 3
Variable resolution mesh size comparison on ER120-15 mesh.

Mesh	Number of cells
ER120–60	60,755
ER120–30	117,935
ER120–15	286,193

Table 4
Variable resolution mesh maximum speedup.

Error level	ER120–60	ER120–30	ER120-15
1×10^{-1}	2.94	1.73	1.13
5×10^{-2}	2.57	1.79	1.10
1×10^{-2}	2.04	2.31	5.56
5×10^{-3}	3.94	9.21	29.24

The speedup ratios for the parallel algorithm for each error level are listed in Table 4. Compared to the QU meshes, the parallel speedup increases more drastically as the error tolerance decreases because of the steeper convergence rates experienced for the ER meshes. As for the QU meshes, the maximum speedups increase as the overall size for the mesh increases. The speedups for the ER120-60 meshes range between 2–4 and between 1.7 and 9 for the ER120-30. The ER120-15 mesh experiences speedups between 1.1 and 29. These maximum speedups are comparable with those of the QU meshes, despite the ER meshes having far fewer cells. For the 5×10^{-3} and 1×10^{-3} error levels, the break-even points for the speedup ratio are around 200 cores, as for the QU meshes.

3.3. Ocean tides

These approaches for computing SHTs have also been assessed in realistic simulations of tidal dynamics, where they are used for the calculation of SAL. Four different meshes have been used: three

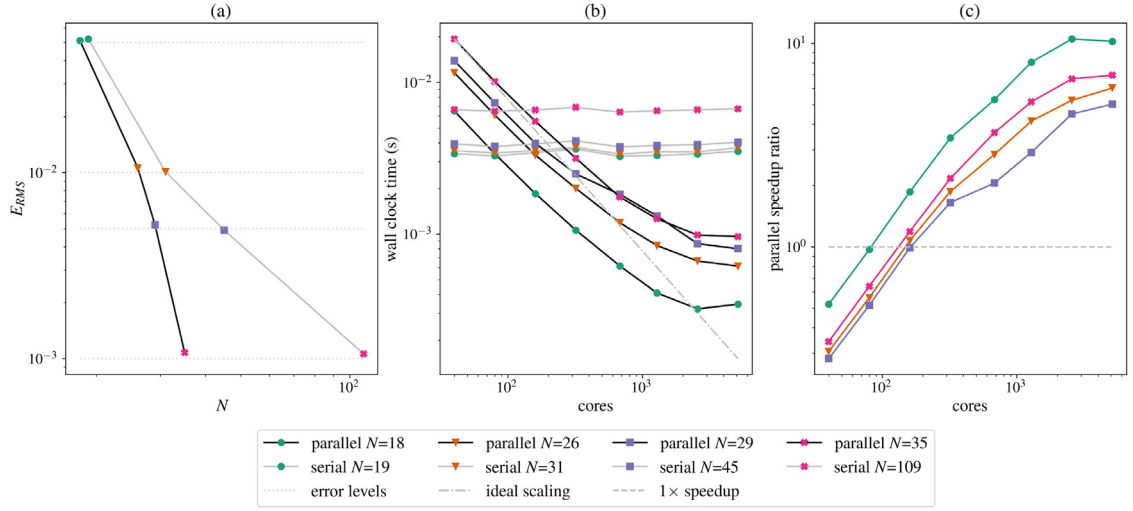

Fig. 4. Performance comparison between parallel and serial algorithms for QU60 mesh. The colored markers indicate the level of accuracy, while the black and grey solid lines represent the parallel and serial algorithms, respectively. (a) Error convergence for the N orders used to assess performance. Orders for the parallel and serial cases are taken at a consistent error level (represented by the dotted lines). (b) Strong scaling results for each error level. Ideal scaling is indicated by the dash–dot line. (c) Speedup ratio of wall clock times $\left(\frac{\text{serial}}{\text{parallel}}\right)$ for each error level. A speedup of one is indicated by the dashed line.

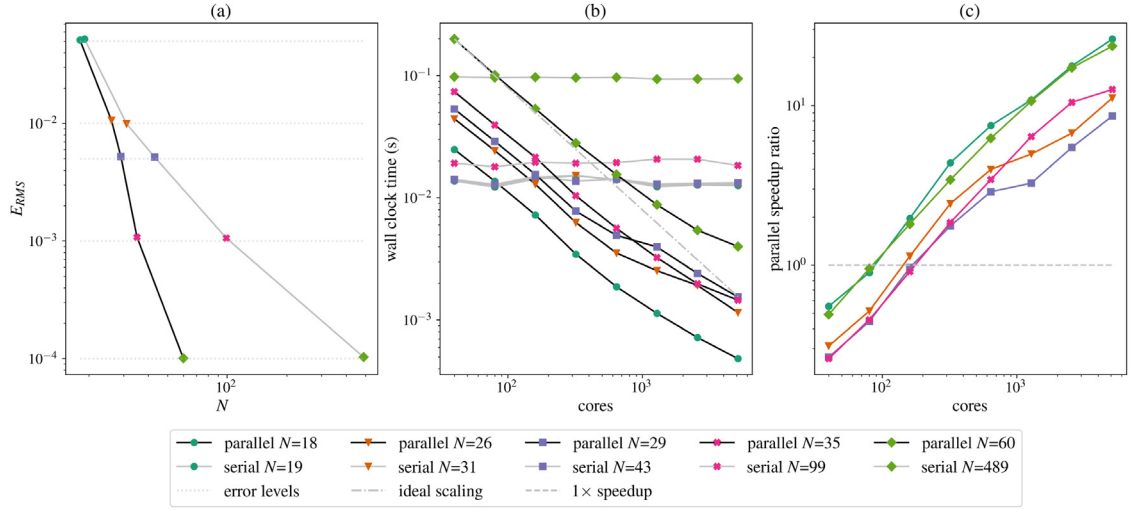

Fig. 5. Same as Fig. 4 for the QU30 mesh.

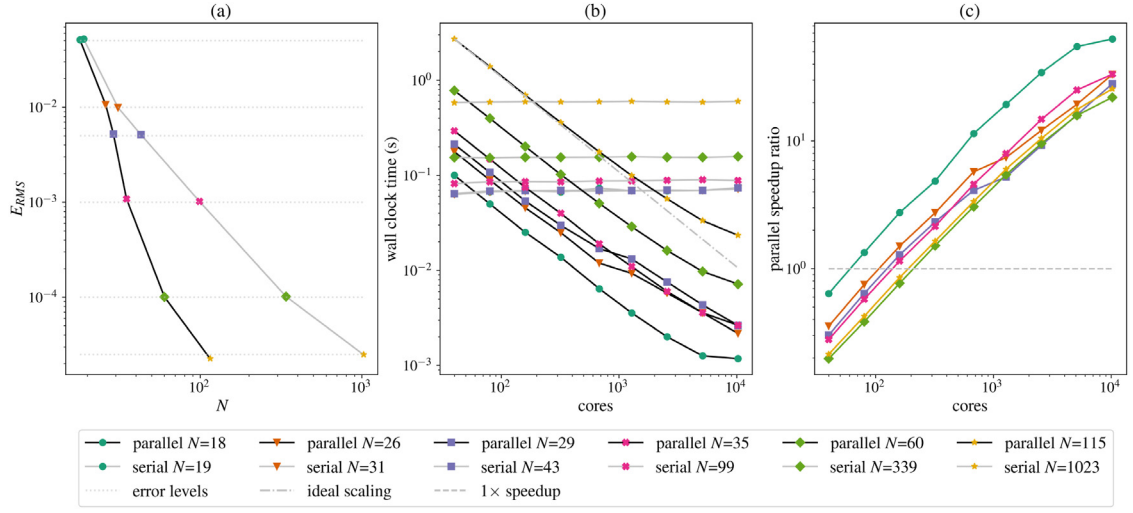

Fig. 6. Same as Fig. 4 for the QU15 mesh.

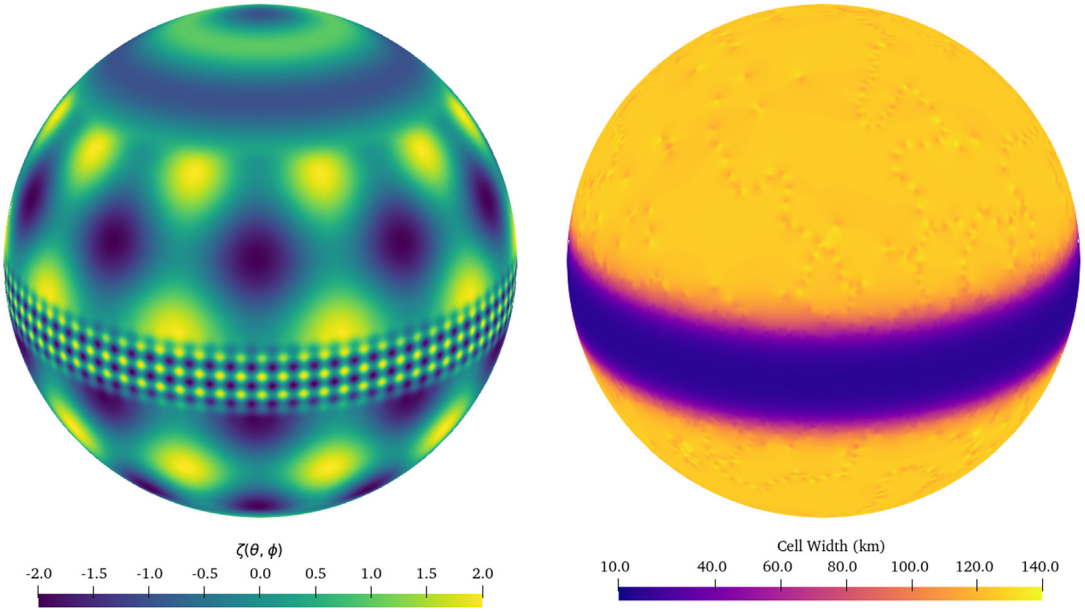

Fig. 7. Test function for the ER120-15 variable resolution case. Note that this function is symmetric about the equator.

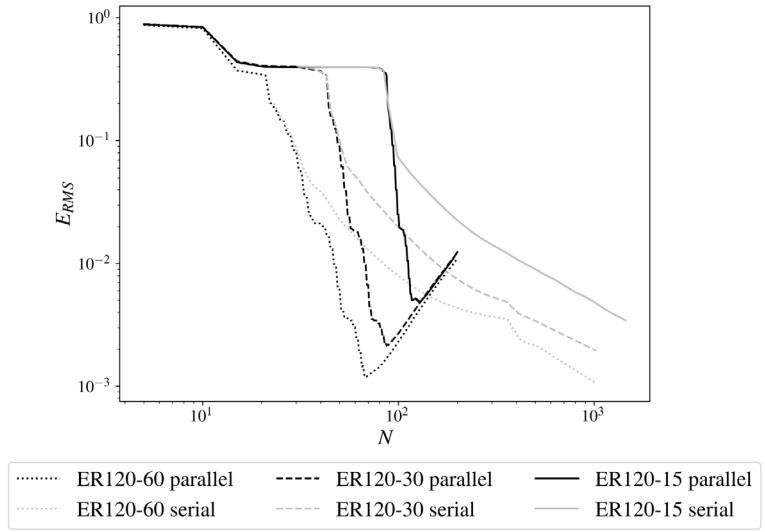

Fig. 8. Convergence for serial and parallel SHT algorithms on variable resolution mesh.

different refinements of icosahedral meshes and a variable resolution mesh, which provides increased resolution in shallow coastal regions and over steep bathymetric gradients. A comparison of these meshes can be found in Table 5.

Each simulation lasted 125 days with the first 15 days used as a spinup period for the tidal forcing. Equilibrium tidal forcing was specified for the 5 major constituents: M_2, S_2, N_2, K_1, and O_1. Harmonic analysis was performed for these 5 constituents over a 90 day period beginning 30 days after the start of the simulation, resulting in modeled amplitudes, A, and phase lags, φ, for each constituent. The model configurations, including topographic wave drag and bottom friction schemes, are the same as those used in Barton et al. (2022).

In order to compare the accuracy of the parallel and serial SAL schemes at each mesh resolution, several different SHT orders for the parallel ($N = 20, 40, 80, 100$) and serial ($N = 19, 39, 79, 159, 319$) SHT algorithms were used. The SAL terms were also calculated at different intervals, i.e. every 30, 10, and 1 min, with the resulting SAL field held fixed for timesteps between the SAL interval. These tests demonstrate how the frequency of computing the SAL terms affects the accuracy of the tidal simulation. Statistics for all tidal simulations are summarized in Table 6. This table shows the deep water area-weighted $RMSE_t$ from Eq. (27) for the M_2 constituent calculated between latitudes of 66°N and 66°S and for waters deeper than 1000 m (Arbic et al., 2004). The spherical harmonic orders, which resulted in an equivalent M_2 $RMSE_t$ value, are placed on the same row of Table 6 for comparison.

All cases using serial and parallel SAL resulted in smaller errors than the scalar approximation. In general, the simulated M_2 error decreases with increasing spherical harmonic order. As shown in Table 6, accuracy does saturate at moderate values for these quantities even as mesh resolution increases. This is the result of the $(2n + 1)^{-1}$ factor on the SH coefficients, which means that any high-frequency variability resolved on finer meshes contributes less to the overall SAL effect. Across all meshes, $N = 79$ (160 Gaussian mesh latitudes) for serial SAL and $N = 40$ for parallel SAL performed equivalently and, in the worst case, resulted in errors within 0.1 cm of what was achieved at higher numbers of latitudes and spherical harmonic orders.

Increasing the frequency interval of the SAL calculation also modestly improves accuracy with 10 min being a reasonable choice, in

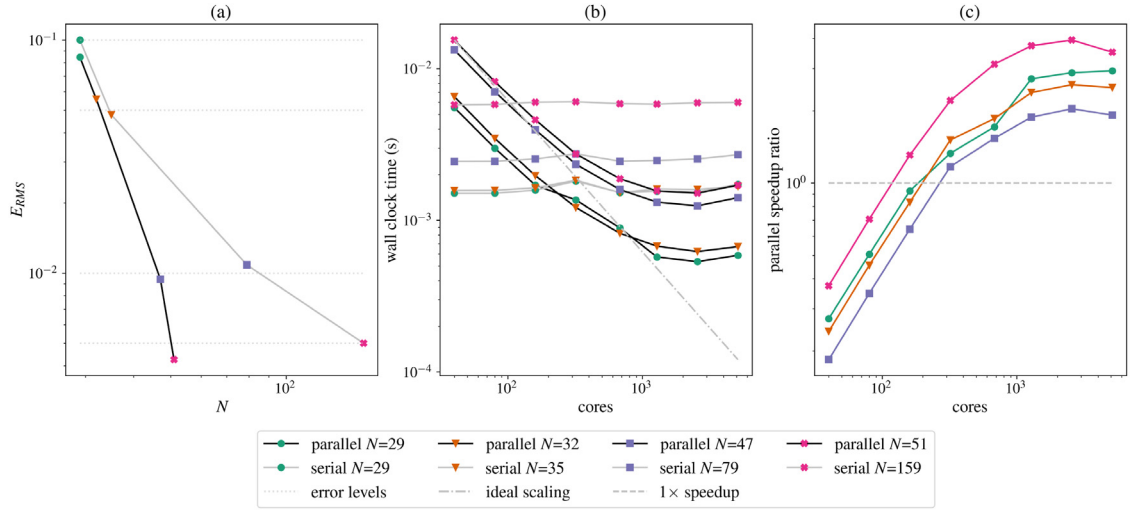

Fig. 9. Same as Fig. 4 for the ER120-60 mesh.

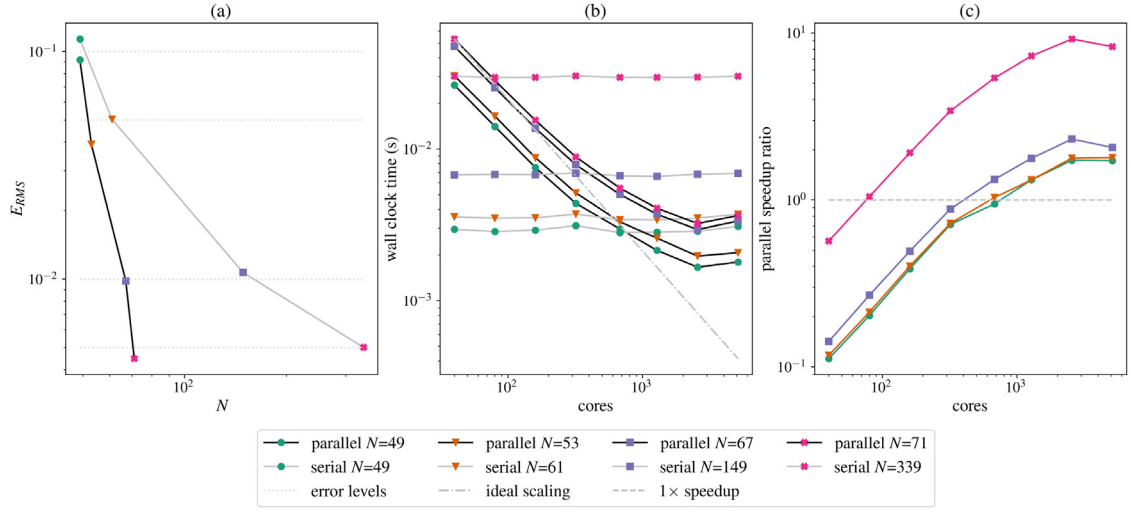

Fig. 10. Same as Fig. 4 for the ER120-30 mesh.

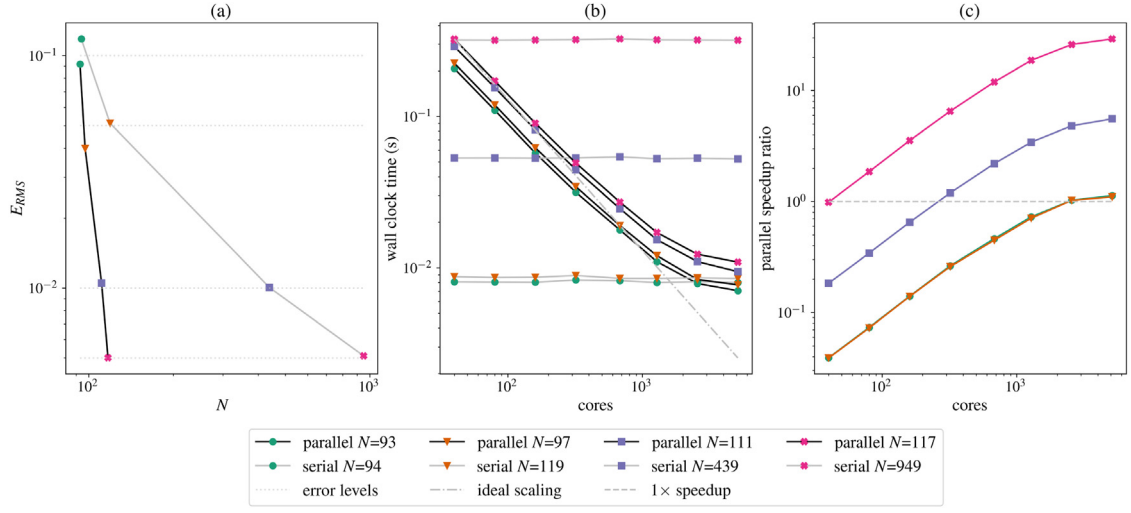

Fig. 11. Same as Fig. 4 for the ER120-15 mesh.

Table 5
Meshes used in tidal simulations.

Mesh	Resolution (km)	Number of cells	Timestep (s)
icos7	60	115,484	60
icos8	30	462,914	60
icos9	15	1,852,158	30
vr45to5	45 (open ocean) to 5 (coastal)	2,359,578	10

Table 6
M_2 constituent accuracy: Deep water M_2 RMSE$_t$ (cm).

Mesh	Order	Serial SAL			Order	Parallel SAL			Scalar SAL
		30 min	10 min	1 min		30 min	10 min	1 min	
icos7	–	–	–	–	–	–	–	–	13.0
	19	12.3	11.9	11.8	–	–	–	–	
	39	12.1	11.7	11.6	–	–	–	–	
	79	12.0	11.6	11.5	20	12.1	11.6	11.5	
	159	12.0	11.6	11.5	40	11.9	11.5	11.3	
	319	12.0	11.6	11.4	80	11.9	11.4	11.3	
icos8	–	–	–	–	–	–	–	–	11.3
	19	9.4	9.1	9.1	–	–	–	–	
	39	9.0	8.7	8.7	20	9.0	8.6	8.6	
	79	8.9	8.5	8.5	40	8.8	8.4	8.4	
	159	8.8	8.5	8.4	80	8.8	8.4	8.3	
	319	8.8	8.5	8.4	–	–	–	–	
icos9	–	–	–	–	–	–	–	–	10.5
	19	7.1	7.0	7.3	–	–	–	–	
	39	6.7	6.6	6.8	20	6.7	6.6	6.8	
	79	6.5	6.4	6.6	40	6.5	6.3	6.5	
	159	6.5	6.4	6.6	80	6.4	6.3	6.5	
	319	6.5	6.4	6.6	100	6.4	6.3	6.5	
vr45to5	–	–	–	–	–	–	–	–	8.9
	19	6.7	6.8	7.1	–	–	–	–	
	39	6.2	6.3	6.6	20	6.4	6.4	6.7	
	79	6.2	6.2	6.5	40	6.2	6.2	6.5	
	159	6.1	6.2	6.5	80	6.1	6.1	6.5	
	319	6.1	6.2	6.5	–	–	–	–	

general. For the icos9 mesh, decreasing the SAL interval to 1 min resulted in a slight increase in error. This was also true for the vr45to5 mesh as the errors increased for the 10 min and 1 min intervals over the 30 min interval.

Fig. 12 shows the scaling of the tidal simulations with serial, parallel, and scalar SAL. In all cases, the serial SAL was computed using a Gaussian grid with 160 latitudes ($N = 79$) and the parallel SAL used $N = 40$, since these result in equivalent M_2 RMSE$_t$ in Table 6. SAL calculation intervals of 10 min and 1 min were used for the parallel and serial SAL simulations to demonstrate the performance implications of computing SAL more frequently. The scalar SAL approach is used as the performance baseline, since it does not require the SHT calculations. The parallel SAL performance at 10 min is very close to that of using serial SAL. It outperforms and scales better than the serial approach, especially as the SAL interval is decreased to 1 min.

The slowdown incurred by calculating SAL via the serial and parallel methods vs. scalar SAL is shown in Fig. 13. This demonstrates that both the 10 min serial and parallel SAL computations add very little extra expense over the baseline scalar SAL run. However, the scalability of the parallel SAL allows it to maintain this very slight slowdown at higher processor counts. At the 1 min SAL interval, the parallel SAL becomes much more efficient than the serial SAL. The slowdown factor of the parallel approach reduces as the mesh size increases. Note that the serial slowdown factor for the vr45to5 mesh with a 1 min SAL interval is less than for the icos9 mesh. This difference is because computing SAL at a 1 min interval for the vr45to5 mesh adds less overhead due to the smaller timestep for that mesh, i.e., SAL is computed every 6 timesteps for vr45to5 as opposed to every other timestep for icos9. Overall, the serial SAL is between 2.25 and 4 times slower than the scalar SAL at high processor counts.

4. Discussion

A more accurate integration technique could be employed for the parallel algorithm, as opposed to the mid-point rule used in (21). For example, the integrals could be computed using a linear approximation over the triangular region defined by connecting the cell centers across each polygon edge (i.e. the dual mesh). This would decrease the integration error at the expense of memory locality and efficient vectorization.

There is a limit to how much finer the high resolution regions can be in idealized settings. As higher spherical harmonic orders are required to resolve smaller wavelengths in the high resolution regions, these spherical harmonic functions will incur aliasing errors in coarse regions of the mesh. It is likely that a specialized, mesh resolution-dependent set of basis functions would be required to address this shortcoming. However, for the purpose of SAL calculations, the high-frequency modes are less important because of the $(2n+1)^{-1}$ scaling factor for the spherical harmonic coefficients. Hence, in practice, as long as the most relevant low-frequency SH functions are resolved in both high and low resolution regions, aliasing errors due to differences in local mesh resolution should not effect the resulting SAL field. The wavelength of a SH mode of order n, $\lambda^{(n)}$, is given by the Jeans relation:

$$\lambda^{(n)} = \frac{2\pi R_E}{\sqrt{n(n+1)}}. \tag{43}$$

Our results show that for the parallel SAL, SH orders of $N = 40$ give good results for realistic tidal applications and that accuracy improvements beyond $N = 80$ are negligible. This corresponds to minimum SH wavelengths of 990 km for $N = 40$ and 488 km for $N = 80$. Both these wavelengths would be well-resolved globally at mesh resolutions required for accurate tides.

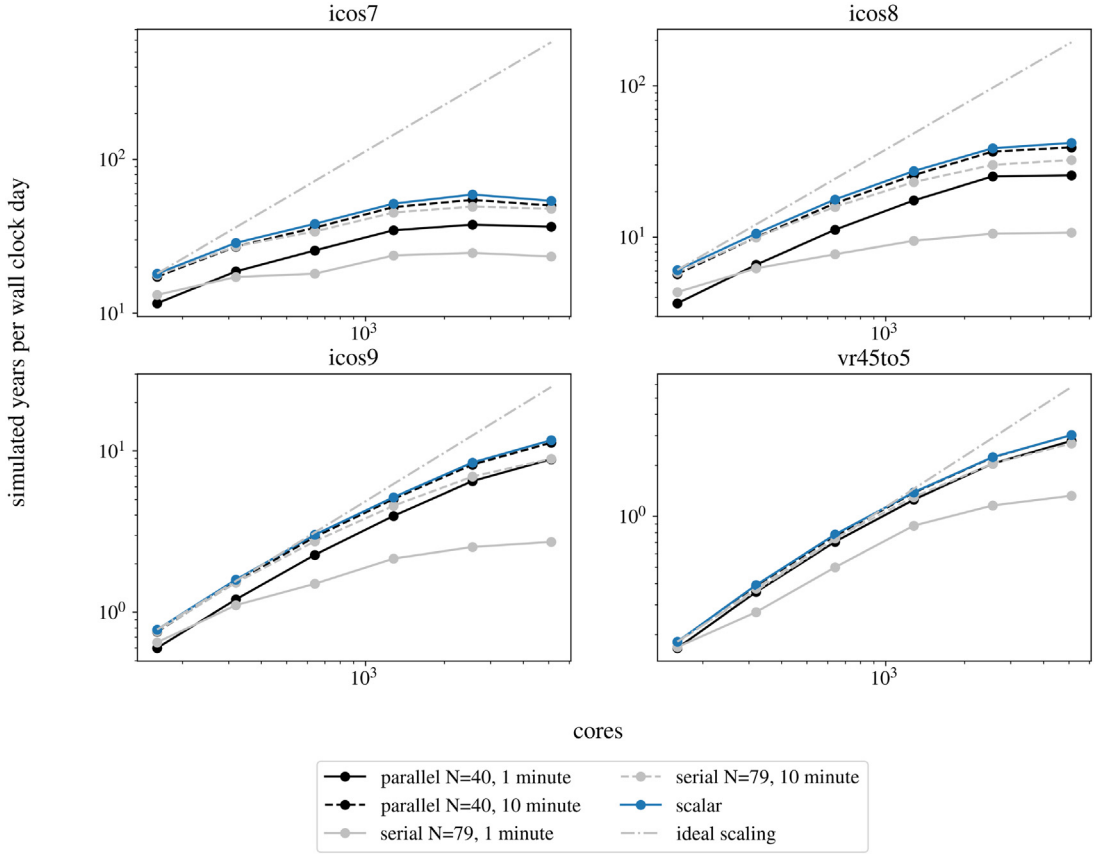

Fig. 12. Scaling curves for tidal simulations.

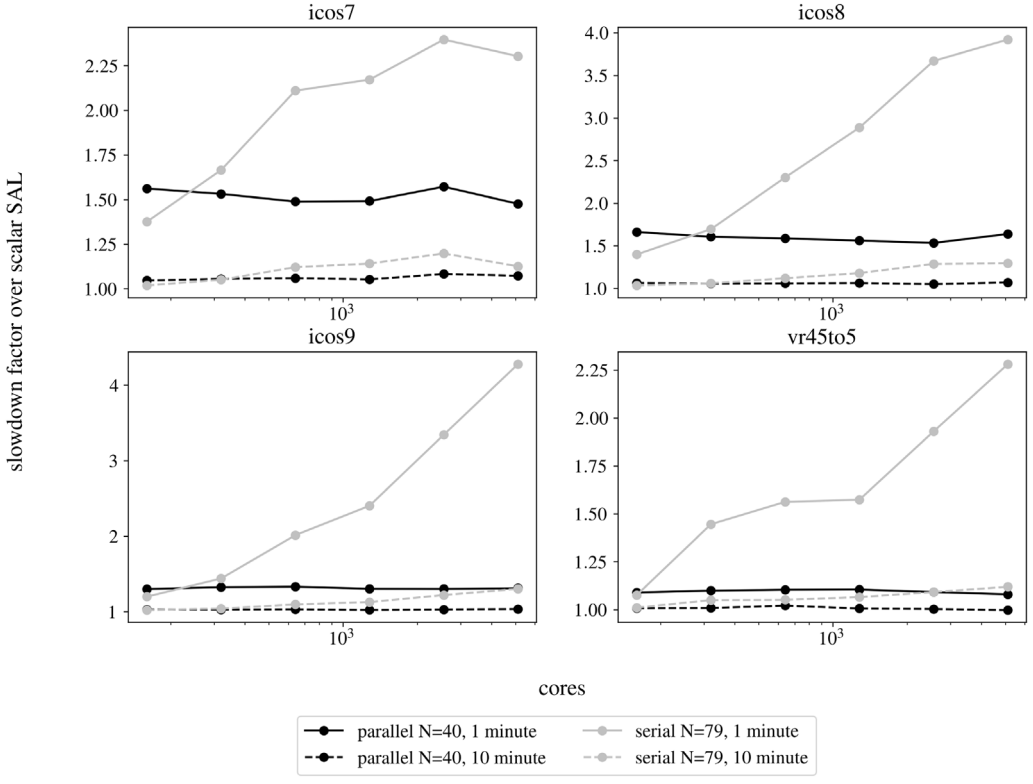

Fig. 13. Slowdown factor over scalar SAL for tidal simulations (lower is better).

Currently the MPAS-Ocean tidal errors are dominated by processes other than SAL, as evidenced by the leveling off of errors with increasing SAL order and frequency in Table 6. However, as these other sources of error decrease due to model improvements, the ability to compute SAL more accurately, both in terms of increasing spherical harmonic order and SAL computation interval, may become more important. For other more accurate models that use unstructured meshes (e.g. ADCIRC Pringle et al., 2021), the parallel approach developed here provides an economical way to compute self-consistent SAL forcing, without relying on SAL data from other models. This is advantageous because it allows for non-periodic effects from atmospheric pressure loading and storm surge processes to be accounted for in the SAL forcing.

The `SHTns` library used in the serial approach could be replaced with a fast Gaussian grid SHT implementation that uses distributed memory parallelization (e.g. Foster and Worley, 1997). However, use of this approach would require a non-trivial communication mapping between the unstructured mesh and the Gaussian mesh. Better parallelization of the Gaussian grid SHT computations would still come with the interpolation errors that are incurred by the transfer of information from the unstructured mesh to the Gaussian grid. This would limit the opportunities for fast spectral convergence observed in the benchmark cases presented here. It would also require additional communication to perform the interpolation in parallel. As a result, a parallel Gaussian grid SHT may not entirely close the performance gap between these two approaches.

In many Earth system modeling contexts, bit-for-bit (BFB) reproducibility is an important consideration. For the parallel algorithm implemented in this paper, machine precision roundoff errors occur at different processor counts. This is due to the local computation of the forward transformation integrals paired with the use of a standard MPI-AllReduce implementation. This strategy does not guarantee BFB reproducible results when using different numbers of MPI ranks. BFB reproducible summations can be implemented by using quadruple precision (He and Ding, 2001), however, the additional overhead is significant in our experience.

Although this paper has considered only barotropic tides, the methodology presented here is also applicable to calculating SAL effects for baroclinic models under a variety of forcings. In a baroclinic setting, the mass loading which causes SAL is due to the bottom pressure anomaly, which may contain contributions from density variations, wind stress forcing, atmospheric pressure loading, as well as tidal motion (Vinogradova et al., 2015; Shihora et al., 2022). Thus, the parallel algorithm can be used to calculate SAL in a baroclinic model by performing the SHT and load Love numbers scaling on the bottom pressure field.

5. Conclusion

This paper has demonstrated an approach to computing SAL forcing based on SH transforms that is suitable for use in parallel unstructured ocean models. It avoids the interpolation error and communication cost of using fast SHT algorithms for Gaussian grids at the expense of an increase in operation count. However, this increase in operations can be overcome by the improved truncation error convergence properties and scalability of the new algorithm. For both idealized and realistic tidal simulations, these advantages make it many times faster than using fast computations on Gaussian grids. This paper has demonstrated speedups up to 30–60× over the serial Gaussian grid approach for idealized cases and up to 4× for tidal simulations at core counts typically used for parallel ocean simulation. Therefore, this parallel SHT algorithm can be used to improve the accuracy of computing in-line SAL forcing with only moderate performance sacrifice. The performance improvements of the parallel scheme are critical for the efficient inclusion of tidal processes within large-scale climate simulations. In addition, this parallel STH algorithm can be utilized in other ESM applications, which require scalable SHT computations, such as the glacial isostatic adjustment (GIA) process that is caused by ice sheet evolution.

CRediT authorship contribution statement

Steven R. Brus: Conceptualization, Methodology, Software, Investigation, Validation, Writing – original draft. **Kristin N. Barton:** Conceptualization, Methodology, Software, Validation, Writing – review & editing. **Nairita Pal:** Validation, Writing – review & editing. **Andrew F. Roberts:** Conceptualization, Writing – review & editing, Project administration. **Darren Engwirda:** Conceptualization, Methodology, Validation, Writing – review & editing, Project administration. **Mark R. Petersen:** Conceptualization, Methodology, Writing – review & editing. **Brian K. Arbic:** Writing – review & editing. **Damrongsak Wirasaet:** Methodology, Writing – review & editing. **Joannes J. Westerink:** Writing – review & editing. **Michael Schindelegger:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Earth System Model Development program area of the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research as part of the multi-program, collaborative Integrated Coastal Modeling (ICoM) project. Kristin N. Barton and Brian K. Arbic acknowledge support from PNNL contract DE-AC05-76RL01830. Joannes J. Westerink and Damrongsak Wirasaet received support from the Joseph and Nona Ahearn endowment at the University of Notre Dame and by the Department of Energy Grant DOE DE-SC0021105. Michael Schindelegger is grateful for funds provided by the DFG (German Research Foundation, project no. 451039647). This research was performed using BER Earth System Modeling program's Compy computing cluster located at Pacific Northwest National Laboratory. PNNL is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830.

References

Arbic, B.K., Garner, S.T., Hallberg, R.W., Simmons, H.L., 2004. The accuracy of surface elevations in forward global barotropic and baroclinic tide models. Deep Sea Res. II: Top. Stud. Oceanogr. 51 (25–26), 3069–3101. http://dx.doi.org/10.1016/j.dsr2.2004.09.014.

Barton, K.N., Pal, N., Brus, S.R., Petersen, M.R., Arbic, B.K., Engwirda, D., Roberts, A.F., Westerink, J.J., Wirasaet, D., Schindelegger, M., 2022. Global barotropic tide modeling using inline self-attraction and loading in MPAS-ocean. J. Adv. Modelling Earth Syst. http://dx.doi.org/10.1029/2022ms003207.

Cavanaugh, N.R., O'Brien, T.A., Collins, W.D., Skamarock, W.C., 2017. Spherical harmonic spectral estimation on arbitrary grids. Mon. Weather Rev. 145 (8), 3355–3363. http://dx.doi.org/10.1175/MWR-D-16-0259.1.

Conroy, C.J., Kubatko, E.J., West, D.W., 2012. ADMESH: an advanced, automatic unstructured mesh generator for shallow water models. Ocean Dyn. 62 (10–12), 1503–1517. http://dx.doi.org/10.1007/s10236-012-0574-0.

Driscoll, J., Healy, D., 1994. Computing Fourier transforms and convolutions on the 2-sphere. Adv. Appl. Math. 15 (2), 202–250. http://dx.doi.org/10.1006/aama.1994.1008.

Egbert, G.D., Erofeeva, S.Y., 2002. Efficient inverse modeling of barotropic ocean tides. J. Atmos. Ocean. Technol. 19 (2), 183–204. http://dx.doi.org/10.1175/1520-0426(2002)019<0183:eimobo>2.0.co;2.

Engwirda, D., 2017. JIGSAW-GEO (1.0): locally orthogonal staggered unstructured grid generation for general circulation modelling on the sphere. Geosci. Model Dev. 10 (6), 2117–2140. http://dx.doi.org/10.5194/gmd-10-2117-2017.

Farrell, W., Clark, J.A., 1976. On postglacial sea level. Geophys. J. Int. 46 (3), 647–667. http://dx.doi.org/10.1111/j.1365-246X.1976.tb01252.x.

Foster, I.T., Worley, P.H., 1997. Parallel algorithms for the spectral transform method. SIAM J. Sci. Comput. 18 (3), 806–837. http://dx.doi.org/10.1137/s1064827594266891.

Frank, M., Hauck, C., Kuepper, K., 2016. Convergence of filtered spherical harmonic equations for radiation transport. Commun. Math. Sci. 14 (5), 1443–1465. http://dx.doi.org/10.4310/CMS.2016.v14.n5.a10.

Golaz, J.-C., Caldwell, P.M., Roekel, L.P.V., Petersen, M.R., Tang, Q., Wolfe, J.D., Abeshu, G., Anantharaj, V., Asay-Davis, X.S., Bader, D.C., Baldwin, S.A., Bisht, G., Bogenschutz, P.A., Branstetter, M., Brunke, M.A., Brus, S.R., Burrows, S.M., Cameron-Smith, P.J., Donahue, A.S., Deakin, M., Easter, R.C., Evans, K.J., Feng, Y., Flanner, M., Foucar, J.G., Fyke, J.G., Griffin, B.M., Hannay, C., Harrop, B.E., Hoffman, M.J., Hunke, E.C., Jacob, R.L., Jacobsen, D.W., Jeffery, N., Jones, P.W., Keen, N.D., Klein, S.A., Larson, V.E., Leung, L.R., Li, H.-Y., Lin, W., Lipscomb, W.H., Ma, P.-L., Mahajan, S., Maltrud, M.E., Mametjanov, A., McClean, J.L., McCoy, R.B., Neale, R.B., Price, S.F., Qian, Y., Rasch, P.J., Eyre, J.E.J.R., Riley, W.J., Ringler, T.D., Roberts, A.F., Roesler, E.L., Salinger, A.G., Shaheen, Z., Shi, X., Singh, B., Tang, J., Taylor, M.A., Thornton, P.E., Turner, A.K., Veneziani, M., Wan, H., Wang, H., Wang, S., Williams, D.N., Wolfram, P.J., Worley, P.H., Xie, S., Yang, Y., Yoon, J.-H., Zelinka, M.D., Zender, C.S., Zeng, X., Zhang, C., Zhang, K., Zhang, Y., Zheng, X., Zhou, T., Zhu, Q., 2019. The DOE E3SM coupled model version 1: Overview and evaluation at standard resolution. J. Adv. Modelling Earth Syst. 11 (7), 2089–2129. http://dx.doi.org/10.1029/2018ms001603.

Gordeev, R.G., Kagan, B.A., Polyakov, E.V., 1977. The effects of loading and self-attraction on global ocean tides: The model and the results of a numerical experiment. J. Phys. Oceanogr. 7 (2), 161–170. http://dx.doi.org/10.1175/1520-0485(1977)007<0161:teolas>2.0.co;2.

He, Y., Ding, C.H.Q., 2001. Using accurate arithmetics to improve numerical reproducibility and stability in parallel applications. J. Supercomput. 18 (3), 259–277. http://dx.doi.org/10.1023/a:1008153532043.

Healy, D., Rockmore, D., Kostelec, P., Moore, S., 2003. FFTs for the 2-sphere-improvements and variations. J. Fourier Anal. Appl. 9 (4), 341–385. http://dx.doi.org/10.1007/s00041-003-0018-9.

Hendershott, M.C., 1972. The effects of solid earth deformation on global ocean tides. Geophys. J. Int. 29 (4), 389–402. http://dx.doi.org/10.1111/j.1365-246x.1972.tb06167.x.

Hoch, K.E., Petersen, M.R., Brus, S.R., Engwirda, D., Roberts, A.F., Rosa, K.L., Wolfram, P.J., 2020. MPAS-Ocean simulation quality for variable-resolution North American coastal meshes. J. Adv. Modelling Earth Syst. 12 (3), e2019MS001848. http://dx.doi.org/10.1029/2019MS001848.

Holmes, S.A., Featherstone, W.E., 2002. A unified approach to the clenshaw summation and the recursive computation of very high degree and order normalised associated Legendre functions. J. Geod. 76 (5), 279–299. http://dx.doi.org/10.1007/s00190-002-0216-2.

Jones, P.W., 1999. First-and second-order conservative remapping schemes for grids in spherical coordinates. Mon. Weather Rev. 127 (9), 2204–2210. http://dx.doi.org/10.1175/1520-0493(1999)127%3C2204:FASOCR%3E2.0.CO;2.

Karypis, G., Kumar, V., 1998. Multilevelk-way partitioning scheme for irregular graphs. J. Parallel Distrib. Comput. 48 (1), 96–129. http://dx.doi.org/10.1006/jpdc.1997.1404.

Keiner, J., Kunis, S., Potts, D., 2009. Using NFFT 3—A software library for various nonequispaced fast Fourier transforms. ACM Trans. Math. Softw. 36 (4), 1–30. http://dx.doi.org/10.1145/1555386.1555388.

Kunis, S., Potts, D., 2003. Fast spherical Fourier algorithms. J. Comput. Appl. Math. 161 (1), 75–98. http://dx.doi.org/10.1016/s0377-0427(03)00546-6.

Mohlenkamp, M.J., 1999. A fast transform for spherical harmonics. J. Fourier Anal. Appl. 5 (2–3), 159–184. http://dx.doi.org/10.1007/bf01261607.

Munk, W.H., MacDonald, G.J., 1960. The Rotation of the Earth; A Geophysical Discussion. Cambridge [Eng.] University Press, http://dx.doi.org/10.1119/1.10629.

Petersen, M.R., Jacobsen, D.W., Ringler, T.D., Hecht, M.W., Maltrud, M.E., 2015. Evaluation of the arbitrary Lagrangian–Eulerian vertical coordinate method in the MPAS-Ocean model. Ocean Model. 86, 93–113. http://dx.doi.org/10.1016/j.ocemod.2014.12.004.

Pringle, W.J., Wirasaet, D., Roberts, K.J., Westerink, J.J., 2021. Global storm tide modeling with ADCIRC v55: unstructured mesh design and performance. Geosci. Model Dev. 14 (2), 1125–1145. http://dx.doi.org/10.5194/gmd-14-1125-2021.

Ray, R., 1998. Ocean self-attraction and loading in numerical tidal models. Mar. Geod. 21 (3), 181–192. http://dx.doi.org/10.1080/01490419809388134.

Ringler, T., Petersen, M., Higdon, R.L., Jacobsen, D., Jones, P.W., Maltrud, M., 2013. A multi-resolution approach to global ocean modeling. Ocean Model. 69, 211–232. http://dx.doi.org/10.1016/j.ocemod.2013.04.010.

Roberts, K.J., Pringle, W.J., Westerink, J.J., 2019. OceanMesh2D 1.0: MATLAB-based software for two-dimensional unstructured mesh generation in coastal ocean modeling. Geosci. Model Dev. 12 (5), 1847–1868. http://dx.doi.org/10.5194/gmd-12-1847-2019.

Schaeffer, N., 2013. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. Geochem. Geophys. Geosyst. 14 (3), 751–758. http://dx.doi.org/10.1002/ggge.20071.

Schindelegger, M., Green, J., Wilmes, S.-B., Haigh, I.D., 2018. Can we model the effect of observed sea level rise on tides? J. Geophys. Res.: Oceans 123 (7), 4593–4609. http://dx.doi.org/10.1029/2018JC013959.

Shihora, L., Sulzbach, R., Dobslaw, H., Thomas, M., 2022. Self-attraction and loading feedback on ocean dynamics in both shallow water equations and primitive equations. Ocean Model. 169, 101914. http://dx.doi.org/10.1016/j.ocemod.2021.101914.

Stammer, D., Cazenave, A., Ponte, R.M., Tamisiea, M.E., 2013. Causes for contemporary regional sea level changes. Annu. Rev. Mar. Sci. 5 (1), 21–46. http://dx.doi.org/10.1146/annurev-marine-121211-172406.

Stepanov, V.N., Hughes, C.W., 2004. Parameterization of ocean self-attraction and loading in numerical models of the ocean circulation. J. Geophys. Res.: Oceans 109 (C3), http://dx.doi.org/10.1029/2003jc002034.

Suda, R., Takami, M., 2001. A fast spherical harmonics transform algorithm. Math. Comp. 71 (238), 703–716. http://dx.doi.org/10.1090/s0025-5718-01-01386-2.

Tamisiea, M.E., Hill, E.M., Ponte, R.M., Davis, J.L., Velicogna, I., Vinogradova, N.T., 2010. Impact of self-attraction and loading on the annual cycle in sea level. J. Geophys. Res. 115 (C7), http://dx.doi.org/10.1029/2009jc005687.

Vinogradova, N.T., Ponte, R.M., Quinn, K.J., Tamisiea, M.E., Campin, J.-M., Davis, J.L., 2015. Dynamic adjustment of the ocean circulation to self-attraction and loading effects. J. Phys. Oceanogr. 45 (3), 678–689. http://dx.doi.org/10.1175/jpo-d-14-0150.1.