PSWM: a Periodic Shallow-Water Model

Scott R. Fulton

Department of Mathematics and Computer Science
Clarkson University, Potsdam, NY 13699–5815

Revised 19 July 2007

Abstract

These notes summarize several mathematical details underlying the periodic shallow water model `pswm`, including the equation forms used for semi-implicit time differencing and the details of the Fourier transforms used.

1 Equation Forms

The shallow-water equations may be formulated in various ways for semi-implicit time differencing, based on various choices of the form of equations and which terms to treat implicitly. A fairly comprehensive overview is given in [1]; here we consider only the details of the equations in the various forms used in the periodic shallow water model `pswm`. In each case we write the time-continuous equations in the split form

$$\psi'(t) - \mathcal{A}(\psi(t)) = \mathcal{B}(\psi(t)), \tag{1.1}$$

where $\psi(t)$ represents the solution variables, $\mathcal{A}(\psi(t))$ represents the terms to be treated implicitly, and $\mathcal{B}(\psi(t))$ represents the terms to be treated explicitly. When discretized by an m-step semi-implicit scheme using solution values $\psi^n \approx \psi(t_n)$ with $t_n = n\Delta t$, the corresponding implicit system to be solved for ψ^{n+1} takes the form

$$\psi^{n+1} - \tau\mathcal{A}(\psi^{n+1}) = \mathcal{C}^n, \tag{1.2}$$

where τ is a multiple of the time step and $\mathcal{C}^n$ is a linear combination of values of ψ, $\mathcal{A}(\psi)$, $\mathcal{B}(\psi)$ at times t_n, t_{n-1}, ..., t_{n-m+1}. In the model $\mathcal{C}^n$ is generated by the package `sitpack` based on the explicit and implicit schemes chosen; the details do not concern us here. In the sections that follow we express the model equations in a time-continuous split form corresponding to (1.1) and a time-discrete implicit system corresponding to (1.2), and show how to solve that system. We do this below for the momentum form, the vorticity/divergence form, and the potential vorticity/divergence form.

1.1 Momentum Form

We can write the momentum and mass continuity equations of the shallow-water system in the form

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + f\mathbf{k} \times \mathbf{v} + g\nabla h = \mathbf{F}, \tag{1.3}$$

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{v}) = F_h, \tag{1.4}$$

where $\mathbf{v}$ is the (horizontal) vector velocity, h is the free surface height, f is the Coriolis parameter, g is the gravitational constant, $\mathbf{k}$ is the vertical unit vector, and the operator ∇ is restricted to the horizontal. The source terms $\mathbf{F}$ for momentum and F_h for mass are regarded as known functions of t (and possibly of $\mathbf{v}$ and h).

In this model we identify the gravity-wave terms using a splitting based on a constant reference depth $\bar{h}$. It is convenient to work with the scaled deviation geopotential $p := g(h - \bar{h})/c$, where $c := (g\bar{h})^{1/2}$, in place of the depth.[1] With this change, (1.3) and (1.4) become

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + f\mathbf{k} \times \mathbf{v} + c\nabla p = \mathbf{F}, \tag{1.5}$$

$$\frac{\partial p}{\partial t} + c\nabla \cdot \mathbf{v} + \nabla \cdot (p\mathbf{v}) = F_p, \tag{1.6}$$

where $F_p := gF_h/c$. It should be noted that the advective form (1.5) is equivalent to the rotational form

$$\frac{\partial \mathbf{v}}{\partial t} + (f + \zeta)\mathbf{k} \times \mathbf{v} + \nabla(cp + K) = \mathbf{F}, \tag{1.7}$$

where $\zeta = \mathbf{k} \cdot \nabla \times \mathbf{v}$ is the relative vorticity and $K = \frac{1}{2}\mathbf{v} \cdot \mathbf{v}$ is the specific kinetic energy. The differences between these two forms affect only the nonlinear terms—which will be treated explicitly—and thus do not impact the implicit system to be solved.

To use a semi-implicit time discretization, we put the terms to be treated implicitly (height gradient and linear part of the divergence term) on the left and those to be treated explicitly (including the Coriolis terms for simplicity) on the right [cf. (1.1)]. This gives the split form

$$\frac{\partial \mathbf{v}}{\partial t} + c\nabla p = \mathbf{F} - \mathbf{v} \cdot \nabla \mathbf{v} - f\mathbf{k} \times \mathbf{v} \tag{1.8}$$

$$\frac{\partial p}{\partial t} + c\nabla \cdot \mathbf{v} = F_p - \nabla \cdot (p\mathbf{v}). \tag{1.9}$$

With any semi-implicit scheme, the implicit system [cf. (1.2)] generated from (1.8)–(1.9) is

$$\mathbf{v} + \tau c\nabla p = \mathbf{V}, \tag{1.10}$$

$$p + \tau c\nabla \cdot \mathbf{v} = P, \tag{1.11}$$

[1] This eliminates g from the problem and gives variables which have the same units.

where τ is a multiple of the time step Δt, $\mathbf{v}$ and p now represent the variables *at the new time level* (i.e., ψ^{n+1} with the superscript dropped for simplicity), and $\mathbf{V}$ and P are generated (by `sitpack`) from the right-hand side of (1.8)–(1.9) and values of $\mathbf{v}$ and p at the previous time level(s). To solve this system, we substitute for $\mathbf{v}$ from (1.10) in (1.11) to obtain the modified Helmholtz equation

$$p - \tau^2 c^2 \nabla^2 p = G := P - \tau c \nabla \cdot \mathbf{V}. \tag{1.12}$$

Once this is solved for p, the corresponding $\mathbf{v}$ can be obtained from (1.10). If the solution of (1.12) is only approximate, then p should be recomputed from $\mathbf{v}$ via (1.11) to ensure exact mass continuity in the discretized system.

1.2 Vorticity/divergence form

The momentum form (1.5)–(1.6) treated above has the disadvantage that the components of the velocity $\mathbf{v}$ are not true scalars (i.e., their values depend on the coordinate system chosen). An alternate approach uses vorticity and divergence instead. To do so, we take the dot product of $\mathbf{k}$ with the curl of the momentum equation in the form (1.7) to obtain the vorticity equation

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (\eta \mathbf{v}) = F_\zeta := \mathbf{k} \cdot \nabla \times \mathbf{F}, \tag{1.13}$$

where $\eta := f + \zeta$ is the absolute vorticity. Likewise, taking the divergence of (1.7) gives the divergence equation

$$\frac{\partial \delta}{\partial t} - \mathbf{k} \cdot \nabla \times (\eta \mathbf{v}) + \nabla^2 (cp + K) = F_\delta := \nabla \cdot \mathbf{F}, \tag{1.14}$$

where $\delta = \nabla \cdot \mathbf{v}$ is the divergence. Combining these equations with (1.6) gives a system for predicting ζ, δ, and h. To close this system we introduce a velocity potential χ and streamfunction ψ satisfying

$$\mathbf{v} = \nabla \chi + \mathbf{k} \times \nabla \psi, \tag{1.15}$$

which implies that

$$\nabla^2 \psi = \zeta, \qquad \nabla^2 \chi = \delta. \tag{1.16}$$

Coupling (1.13) and (1.14) with the continuity equation (1.6) and putting the terms to be treated implicitly on the left and those to be treated explicitly on the right gives the split form [cf. (1.1)]

$$\frac{\partial \zeta}{\partial t} = F_\zeta - \nabla \cdot (\eta \mathbf{v}), \tag{1.17}$$

$$\frac{\partial \delta}{\partial t} + c\nabla^2 p = F_\delta + \mathbf{k} \cdot \nabla \times (\eta \mathbf{v}) - \nabla^2 K, \tag{1.18}$$

$$\frac{\partial p}{\partial t} + c\delta = F_p - \nabla \cdot (p \mathbf{v}). \tag{1.19}$$

If desired, the right-hand side of this system may be written in terms of χ and ψ instead of $\mathbf{v}$ using the identities

$$\nabla \cdot (\alpha \mathbf{v}) = \nabla \cdot (\alpha \nabla \chi) - \mathcal{J}(\alpha, \psi) \tag{1.20}$$

and

$$\mathbf{k} \cdot \nabla \times (\alpha \mathbf{v}) = \nabla \cdot (\alpha \nabla \psi) + \mathcal{J}(\alpha, \chi), \tag{1.21}$$

where $\mathcal{J}(\alpha, \beta) = \mathbf{k} \cdot (\nabla \alpha \times \nabla \beta)$ is the Jacobian operator and α and β represent any scalars. With these substitutions, (1.17)–(1.19) take the form

$$\frac{\partial \zeta}{\partial t} = F_\zeta - \nabla \cdot (\eta \nabla \chi) + \mathcal{J}(\eta, \psi), \tag{1.22}$$

$$\frac{\partial \delta}{\partial t} + c\nabla^2 p = F_\delta + \nabla \cdot (\eta \nabla \psi) + \mathcal{J}(\eta, \chi) - \nabla^2 K, \tag{1.23}$$

$$\frac{\partial p}{\partial t} + c\delta = F_p - \nabla \cdot (p \nabla \chi) + \mathcal{J}(p, \psi). \tag{1.24}$$

In this way, $\mathbf{v}$ is eliminated, and if we express the kinetic energy as

$$K = \frac{1}{2}\left[\nabla \cdot (\chi \nabla \chi) - \chi \nabla^2 \chi + \nabla \cdot (\psi \nabla \psi) - \psi \nabla^2 \psi\right] + \mathcal{J}(\psi, \chi). \tag{1.25}$$

then the only operators needed are the flux divergence, Jacobian, and Laplacian. However, this form is less efficient for the spectral model, since it requires more terms to be transformed than the form (1.17)–(1.19).

With a semi-implicit time discretization, the implicit system generated from (1.17)–(1.19) or (1.22)–(1.24) is

$$\zeta = Z, \tag{1.26}$$

$$\delta + \tau c \nabla^2 p = D, \tag{1.27}$$

$$p + \tau c \delta = P, \tag{1.28}$$

where ζ, δ, and p now represent the variables at the new time level and Z, D, and P are generated (via `sitpack`) from the right-hand side of (1.17)–(1.19) and values of ζ, δ, and p at the previous time level(s). To solve this system, we eliminate δ between (1.27) and (1.28) to obtain the modified Helmholtz equation

$$p - \tau^2 c^2 \nabla^2 p = G := P - \tau c D \tag{1.29}$$

[cf. (1.12)]. Once this is solved for p, the corresponding δ can be obtained from (1.28), ζ is given directly by (1.26), and ψ and χ are obtained by solving (1.16). If needed, the corresponding $\mathbf{v}$ can be computed from (1.15).

1.3 Potential vorticity/divergence form

By combining the vorticity equation (1.13) with the continuity equation (1.6) we can obtain the potential vorticity equation

$$\frac{\partial q}{\partial t} + \mathbf{v} \cdot \nabla q = F_q, \tag{1.30}$$

where

$$q := \frac{h}{\bar{h}}\eta = \frac{f + \zeta}{1 + p/c} \tag{1.31}$$

is the potential vorticity and

$$F_q := \frac{(1 + p/c)F_\zeta - \eta F_h/c}{(1 + p/c)^2}, \tag{1.32}$$

is the corresponding forcing. Coupling (1.30) with the divergence equation (1.14) and the continuity equation (1.6) and putting the terms to be treated implicitly on the left and those to be treated explicitly on the right gives the split form [cf. (1.1)]

$$\frac{\partial q}{\partial t} = F_q - \mathbf{v} \cdot \nabla q, \tag{1.33}$$

$$\frac{\partial \delta}{\partial t} + c\nabla^2 p = F_\delta + \mathbf{k} \cdot \nabla \times (\eta \mathbf{v}) - \nabla^2 K, \tag{1.34}$$

$$\frac{\partial p}{\partial t} + c\delta = F_p - \nabla \cdot (p\mathbf{v}). \tag{1.35}$$

As before, $\mathbf{v}$ may be replaced in terms of ψ and χ on the right-hand side of this system using (1.20) and (1.21).

With a semi-implicit time discretization, the implicit system generated from (1.33)–(1.35) is

$$q = Q, \tag{1.36}$$

$$\delta + \tau c\nabla^2 p = D, \tag{1.37}$$

$$p + \tau c\delta = P, \tag{1.38}$$

where q, δ, and p now represent the variables at the new time level and Q, D, and P are generated (via `sitpack`) from the right-hand side of (1.33)–(1.35) and values of q, δ, and p at the previous time level(s). To solve this system, we eliminate δ between (1.37) and (1.38) to obtain the modified Helmholtz equation

$$p - \tau^2 c^2 \nabla^2 p = G := P - \tau c D \tag{1.39}$$

[cf. (1.12)]. Once this is solved for p, the corresponding δ can be obtained from (1.38), q is given directly by (1.36), η is obtained from (1.31), and ψ and χ are obtained by solving (1.16). If needed, the corresponding $\mathbf{v}$ can be computed from (1.15).

1.4 Initialization

To initialize any of the above forms of the model we can specify the predicted variables, i.e., u, v, and p for the momentum form, ζ, δ, and p for the vorticity/divergence form, or q, δ, and p for the potential vorticity/divergence form. However, this will in general include a significant gravity-inertia wave component in the initial fields. A better approach is to use the nonlinear balance equation as follows.

Setting $\partial\delta/\partial t = 0$ and $F_\delta = 0$ in the divergence equation (1.14) and using Cartesian coordinates (x, y) we obtain

$$-(\eta v)_x + (\eta u)_y + \nabla^2 (cp + K) = 0, \tag{1.40}$$

where u and v are the velocity components in the x and y directions and the subscripts denote partial derivatives. Assuming the flow is purely rotational (nondivergent), then (1.15) and (1.25) reduce to

$$u = -\psi_y, \qquad v = \psi_x, \qquad K = \tfrac{1}{2}\left(\psi_x^2 + \psi_y^2\right). \tag{1.41}$$

With these substitutions (and a lot of algebra), (1.40) reduces to the nonlinear balance equation

$$f\nabla^2\psi + 2\left(\psi_{xx}\psi_{yy} - \psi_{xy}^2\right) = c\nabla^2 p. \tag{1.42}$$

If we specify the relative vorticity ζ, the corresponding streamfunction ψ can be obtained by solving

$$\nabla^2\psi = \zeta, \tag{1.43}$$

and then the corresponding height field p (in nonlinear balance) can be obtained by solving (1.42). Note that with ψ known this equation is linear, and thus easy to solve.

2 Fourier Transforms and Spectral-Space Operations

The periodic shallow water model `pswm` is formulated using a Fourier spectral representation on a two-dimensional rectangular domain which is periodic in both directions. In principle this is easy: the Fourier modes are eigenfunctions of the derivative and Laplacian operators, so these operations can be computed (and inverted) easily in spectral space. In practice, there are some subtleties in the discrete version of this representation which must be taken into account. In this section we first explain discrete Fourier representations in detail in one dimension, and then document how the corresponding two-dimensional representations are used in the model.

2.1 Fourier representations in one dimension

Consider a function $f\colon \mathbb{R} \to \mathbb{C}$ which is L-periodic, i.e., $f(x+L) = f(x)$ for all $x \in \mathbb{R}$. Such a function has the Fourier series representation[2]

$$f(x) = \sum_{k=-\infty}^{\infty} \widetilde{f}_k e^{2\pi i k x/L}, \qquad x \in \mathbb{R}, \tag{2.1}$$

where

$$\widetilde{f}_k = \frac{1}{L}\int_0^L f(x) e^{-2\pi i k x/L}\,dx, \qquad k \in \mathbb{Z}. \tag{2.2}$$

Note from (2.2) that if f is real then $\widetilde{f}_{-k} = \widetilde{f}_k^*$ (where the asterisk represents the complex conjugate); in particular, $\widetilde{f}_0$ is real. We represent such a function on the computer by the discrete values $f_j := f(jL/N)$ for integer values of j, where N is some positive integer. Since $f_{j+N} = f_j$ by periodicity, only N values are needed; normally we will use $j = 0, \ldots, N-1$. Clearly this is not enough information to determine the Fourier coefficients $\widetilde{f}_k$ for all $k \in \mathbb{Z}$. Here we first derive the Discrete Fourier Transform pair, and then show how it can be used to represent functions which are band-limited, i.e., exactly represented by a *truncated* Fourier series. We then examine how such Fourier representations can be used to compute derivatives and nonlinear terms.

2.1.1 Discrete Fourier transforms

The key to discrete Fourier transforms is the following discrete orthogonality property:

Lemma 1 *If N is a positive integer and $j, k \in \mathbb{Z}$, then*

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi i j k/N} = \begin{cases} 1, & j \equiv 0 \pmod N, \\ 0, & \textit{otherwise.} \end{cases} \tag{2.3}$$

[2]The existence and convergence of this expansion depend on the smoothness of f. For example, if f is continuously differentiable, then the series exists and converges uniformly. Here we ignore such questions.

Proof: Setting $z = \exp(2\pi ij/N)$ we can sum the finite geometric series

$$\sum_{k=0}^{N-1} e^{2\pi ijk/N} = \sum_{k=0}^{N-1} z^k = \begin{cases} N, & z = 1, \\ \dfrac{1 - z^N}{1 - z}, & z \neq 1. \end{cases} \tag{2.4}$$

Since $z = 1$ if and only if $j \equiv 0 \pmod{N}$ and $z^N = \exp(2\pi ij) = 1$ for any $j \in \mathbb{Z}$, this reduces to (2.3). □

This orthogonality property leads directly to the Discrete Fourier Transform pair:

Lemma 2 *Two sequences $f_0, f_1, \ldots, f_{N-1}$ and $\widehat{f}_0, \widehat{f}_1, \ldots, \widehat{f}_{N-1}$ of $N > 0$ complex numbers are related by the Discrete Fourier Transform (DFT)*

$$\widehat{f}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-2\pi ijk/N}, \qquad k = 0, \ldots, N-1 \tag{2.5}$$

if and only if they are related by the Inverse Discrete Fourier Transform (IDFT)

$$f_j = \sum_{k=0}^{N-1} \widehat{f}_k e^{2\pi ijk/N}, \qquad j = 0, \ldots, N-1. \tag{2.6}$$

Proof: Direct calculation using (2.5) with j replaced by l shows that for $j = 0, \ldots, N-1$,

$$\sum_{k=0}^{N-1} \widehat{f}_k e^{2\pi ijk/N} = \sum_{k=0}^{N-1} \left[\frac{1}{N} \sum_{l=0}^{N-1} f_l e^{-2\pi ilk/N} \right] e^{2\pi ijk/N} = \sum_{l=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i(j-l)k/N} \right] f_l = f_j, \tag{2.7}$$

where the last step follows from Lemma 1. The converse follows by a similar argument. □

Both sequences f_j and $\widehat{f}_k$ may be extended by periodicity to be N-periodic. Also, if the sequence f_j is real then the sequence $\widehat{f}_k$ is "half-complex", i.e., $\widehat{f}_{N-k} = \widehat{f}_k^*$; in particular, $\widehat{f}_0$ is real, as is $\widehat{f}_{N/2}$ if N is even. The Fast Fourier Transform (FFT) is an algorithm for computing the DFT or IDFT in $O\big(N \log(N)\big)$ operations; it normally uses a transform length N which is either a power of 2 or an even number with many small prime factors.

2.1.2 Representing functions by the DFT

According to Lemma 2, the DFT pair (2.6)–(2.5) relates two sequences of numbers, namely, $f_0, \ldots, f_{N-1}$ and $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$: knowing either sequence is equivalent to knowing the other. To use this representation in solving differential equations, we must relate this data (i.e., either sequence of numbers) to a function $f(x)$ which we can differentiate. In particular, we want to relate the DFT coefficients $\widehat{f}_k$ to the Fourier coefficients $\widetilde{f}_k$ of a function f so we can

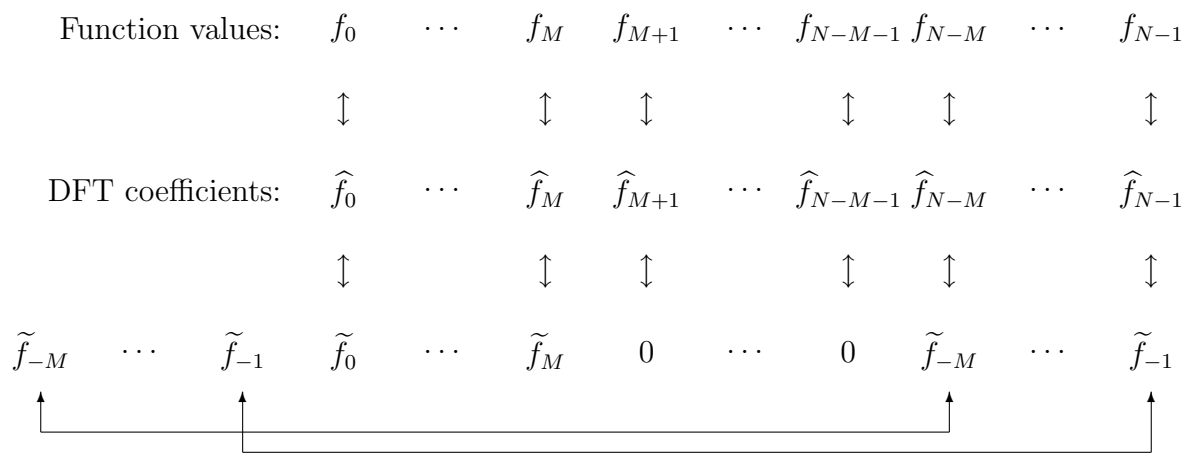

Figure 1: Relation between the coefficients of a DFT and a truncated Fourier series.

compute operations (such as derivatives) on f in spectral space (i.e., in terms of its Fourier coefficients). Since there are only finitely many DFT coefficients, we can at best expect them to determine a function with finitely many nonzero Fourier coefficients, i.e., a *band-limited* function of the form

$$f(x) = \sum_{k=-M}^{M} \widetilde{f}_k e^{2\pi i k x/L}, \qquad x \in \mathbb{R}. \tag{2.8}$$

We refer to the integer M here as the *spectral truncation*; it is the maximum index for which $\widetilde{f}_k$ or $\widetilde{f}_{-k}$ may be nonzero. When $N > 2M$ this relationship is as shown in Fig. 1. We can understand this relationship in two ways as follows.

First, suppose the function f is given as in (2.8). Evaluating this at the points $x_j = jL/N$ gives the discrete values

$$f_j = f(x_j) = \sum_{k=-M}^{M} \widetilde{f}_k e^{2\pi i j k/N}, \qquad j = 0, \ldots, N-1. \tag{2.9}$$

If $N > 2M$ we can define the sequence $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ by

$$\widehat{f}_k := \begin{cases} \widetilde{f}_k, & k = 0, \ldots, M, \\ 0, & k = M+1, \ldots, N-M-1, \\ \widetilde{f}_{k-N}, & k = N-M, \ldots, N-1 \end{cases} \tag{2.10}$$

(see Fig. 1). Then (2.9) reduces to the IDFT (2.6), so $\widehat{f}_k$ is given by the DFT (2.5). Thus, for the band-limited function (2.8) the DFT coefficients $\widehat{f}_k$ are identical to the Fourier series coefficients $\widetilde{f}_k$ (modulo N) provided that $N > 2M$. Mote that if we choose $N = 2M+1$ then both the DFT and Fourier series representations of f use $N = 2M+1$ degrees of freedom, and none of the coefficients (2.10) are automatically zero; if $N > 2M+1$ then there are "extra" terms in the DFT corresponding to Fourier coefficients which vanish.

Conversely, suppose we know either of the two N-periodic sequences $f_0, \ldots, f_{N-1}$ or $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ which are related by the DFT pair. If N is *odd*, this data uniquely determines a function f which satisfies $f(x_j) = f_j$ as follows: setting $M = (N-1)/2$, we invert (2.10) to yield $\widetilde{f}_k = \widehat{f}_k$ for $|k| \leq M$. The corresponding band-limited function f given by (2.8) then satisfies $f(x_j) = f_j$ for $j = 0, \ldots, N-1$. Thus, if N is odd we can interpret the DFT coefficients $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ directly as the Fourier series coefficients $\widetilde{f}_k$ (modulo N) of the band-limited function given by (2.8).

However, it is awkward to use N odd in discrete Fourier representations, since the FFT codes generally require N to be even. What happens if we try to relate the sequence $f_0, \ldots, f_{N-1}$ (or $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ via the DFT) to a band-limited function f of the form (2.8) when N is *even*? If we set $M = N/2$, it is clear that something must be lost: the N values $f_0, \ldots, f_{N-1}$ or corresponding coefficients $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ cannot contain the same information as the $2M+1 = N+1$ Fourier series coefficients $\widetilde{f}_{-M}, \ldots, \widetilde{f}_M$. If we insist that the function f match the specified values, i.e., require that $f(x_j) = f_j$ for $j = 0, \ldots, N-1$, then we find that (2.9) with $N = 2M$ reduces to

$$f_j = \sum_{k=-M+1}^{M-1} \widetilde{f}_k e^{2\pi i j k/N} + (-1)^j \left(\widetilde{f}_{-M} + \widetilde{f}_M \right), \qquad j = 0, \ldots, N-1. \tag{2.11}$$

If we set $\widetilde{f}_k = \widehat{f}_k$ for $|k| < M = N/2$ and choose $\widetilde{f}_{-M}$ and $\widetilde{f}_M$ satisfying $\widetilde{f}_{-M} + \widetilde{f}_M = \widehat{f}_M$, then (2.11) reduces to the IDFT (2.6), which implies that the function f defined by (2.8) with these Fourier series coefficients $\widetilde{f}_k$ matches the specified values f_j. It almost matches the DFT coefficients $\widehat{f}_k$ as well, doing so for $|k| < M$; however, only the sum $\widetilde{f}_{-M} + \widetilde{f}_M$ is determined by the data. Thus, *there is a one-parameter family of band-limited functions f* which match the values $f_0, \ldots, f_{N-1}$ when N is even, one function corresponding to each choice of the difference $d := \widetilde{f}_M - \widetilde{f}_{-M}$. Which one is "right" or "best"?

A simple approach is to set $d = \widehat{f}_M$, which gives $\widetilde{f}_M = \widehat{f}_M$ and $\widetilde{f}_{-M} = 0$ so that $\widetilde{f}_k = \widehat{f}_k$ for $k = -M+1, \ldots, M$. With this choice the numbers $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ correspond to the function

$$f(x) = f_1(x) := \sum_{k=-M+1}^{M} \widehat{f}_k e^{2\pi i k x/L}. \tag{2.12}$$

This function agrees with the data values $f_0, \ldots, N-1$ and its Fourier series coefficients $\widetilde{f}_k$ match the DFT coefficients $\widehat{f}_k$ (modulo N). However, this simple approach is problematic, at least in the case where f is real.[3] Specifically, if the data values f_j are real then we expect f to be a real function, but the condition $\widetilde{f}_{-M} = \widetilde{f}_M^*$ implies we must have $\widehat{f}_M = 0$—which is not necessarily what the data gives. Therefore, interpreting the (real) data $f_0, \ldots, f_{N-1}$ as corresponding to the function (2.12) is contradictory in general: unless $\widehat{f}_M$ happens to be zero, then the function is not strictly real (i.e., real for all x), and if it is forced to be real by setting $\widehat{f}_M = 0$, then it no longer matches the values f_j.

[3]Even if f is not necessarily real, the truncation is "unbalanced" in k (i.e., $\widetilde{f}_{-M}$ must vanish but $\widetilde{f}_M$ may not), which seems unreasonable. It is not clear whether this is really a problem.

A potentially better choice is to set $d = 0$, which gives $\widetilde{f}_M = \widetilde{f}_{-M} = \frac{1}{2}\widehat{f}_M$ and corresponds to the function

$$f(x) = f_2(x) := \sum_{k=-M+1}^{M-1} \widehat{f}_k e^{2\pi ikx/L} + \widehat{f}_M \cos\left(\frac{2\pi Mx}{L}\right). \tag{2.13}$$

Now in the case where the data $f_0, \ldots, f_{N-1}$ is real, the condition $\widetilde{f}_{-M} = \widetilde{f}_M^*$ implies only that $\widehat{f}_M$ must be real—which it is automatically, as a result of the DFT as noted above. Thus, interpreting the data as corresponding to the function (2.13) is at least consistent: f matches the function values and is real if they are real. However, with this interpretation the DFT coefficients $\widehat{f}_k$ do not (quite) match the Fourier series coefficients $\widetilde{f}_k$: the factor of two difference for the highest mode must be taken into account if the series is to be evaluated on a grid with a different number of points, as in a Fourier psuedospectral multigrid method [2].

2.1.3 Derivatives of Fourier representations

If f is any function given by a truncated Fourier series of the form (2.8), then its derivative $g = f'$ is given exactly by the truncated Fourier series

$$g(x) = \sum_{k=-M}^{M} \widetilde{g}_k e^{2\pi ikx/L}, \qquad x \in \mathbb{R}, \tag{2.14}$$

where

$$\widetilde{g}_k = \left(\frac{2\pi ik}{L}\right) \widetilde{f}_k, \qquad k = -M, \ldots, M. \tag{2.15}$$

Thus, the fact that the Fourier basis functions $\exp(2\pi ikx/L)$ are eigenfunctions of the derivative operator makes the representation of this operator "diagonal" in terms of the Fourier series coefficients $\widetilde{f}_k$. Is this same result true for the DFT coefficients $\widehat{f}_k$? That is, can we replace (2.15) by

$$\widehat{g}_k = \left(\frac{2\pi ik}{L}\right) \widehat{f}_k \tag{2.16}$$

for the appropriate values of k?

If we represent the truncated series (2.8) using $N > 2M$ values f_j or coefficients $\widehat{f}_k$ related by the DFT pair, the answer is yes, since the DFT coefficients $\widehat{f}_k$ match the Fourier series coefficients $\widetilde{f}_k$ exactly. More precisely, just as evaluating the truncated series (2.8) at $x_j = jL/N$ gives the IDFT (2.6) with $\widehat{f}_k = \widetilde{f}_k$ for $k = 0, \ldots, N-1$ (with $\widetilde{f}_k = 0$ for $|k| > M$, and $\widehat{f}_k$ is interpreted modulo N), evaluating the truncated series (2.14) at the same points gives

$$g_j = g\left(\frac{jL}{N}\right) = \sum_{k=0}^{N-1} \widehat{g}_k e^{2\pi ijk/N}, \qquad j = 0, \ldots, N-1, \tag{2.17}$$

where $\widehat{g}_k = \widetilde{g}_k$ and (2.16) holds for $k = 0, \ldots, N-1$ (interpreted modulo N). Conversely, given either of the N-periodic sequences $f_0, \ldots, f_{N-1}$ or $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ related by the DFT pair, if N is *odd* there is a unique function f of the form (2.8) with $M = (N-1)/2$ satisfying $f(x_j) = f_j$, and its derivative (2.14) will have the values (2.17) as above.

However, if N is *even* and we attempt to use either of the sequences $f_0, \ldots, f_{N-1}$ or $\widehat{f}_0, \ldots, \widehat{f}_{N-1}$ (which are related by the DFT pair) to determine a band-limited function f and then find its derivative, the situation is more complicated. As discussed above, we will choose f of the form (2.8) with $M = N/2$ and coefficients $\widetilde{f}_k = \widehat{f}_k$ for $|k| < M$; however, the coefficients with $|k| = M$ need only satisfy $\widetilde{f}_{-M} + \widetilde{f}_M = \widehat{f}_M$. If we choose $\widetilde{f}_{-M} = 0$ and $\widetilde{f}_M = \widehat{f}_M$ so that f is given by (2.12), its derivative is

$$g(x) = f_1'(x) = \sum_{k=-M+1}^{M} \widehat{g}_k e^{2\pi i k x/L} \tag{2.18}$$

with $\widehat{g}_k$ given by (2.16). However, it must be recalled that for real data $f_0, \ldots, f_{N-1}$ the function $f_1(x)$ is real for all x only if the highest Fourier mode is dropped. In particular, evaluating (2.18) at $x_j = jL/N$ and using (2.16) gives

$$g_j = g(x_j) = \sum_{k=-M+1}^{M-1} \widehat{g}_k e^{2\pi i j k/N} + (-1)^j \left(\frac{2\pi i M}{L}\right) \widehat{f}_M, \qquad j = 0, \ldots, N-1, \tag{2.19}$$

and since $\widehat{f}_M$ is real, this results in *complex* values of the derivative.[4] Once again, we conclude that this interpretation of the data is problematic.

On the other hand, if we choose $\widetilde{f}_M = \widetilde{f}_{-M} = \frac{1}{2}\widehat{f}_M$ so that f is given by (2.13), its derivative is

$$g(x) = f_2'(x) = \sum_{k=-M+1}^{M-1} \widehat{g}_k e^{2\pi i k x/L} - \left(\frac{2\pi M}{L}\right) \widehat{f}_M \sin\left(\frac{2\pi M x}{L}\right), \tag{2.20}$$

with $\widehat{g}_k$ given by (2.16) for $|k| < M$. It should be noted that when evaluated at $x_j = jL/N$ the last term in (2.20) vanishes, so the contribution from the highest Fourier mode is lost at those points. This problem can be solved and exact values of the derivative obtained by evaluating the at the *midpoints* $x_{j+1/2} = (j + \frac{1}{2})L/N$ instead; we will not pursue this approach here (for details, see [2]).

2.1.4 Transforms of nonlinear terms

Suppose the model contains a quadratic nonlinear term $f(x) = g(x)h(x)$, where g and h are represented in the model by truncated Fourier series

$$g(x) = \sum_{k=-M}^{M} \widetilde{g}_k e^{2\pi i k x/L}, \qquad h(x) = \sum_{k=-M}^{M} \widetilde{h}_k e^{2\pi i k x/L}, \qquad x \in \mathbb{R}. \tag{2.21}$$

[4] If (2.19) is evaluated using an FFT code for real functions, the imaginary part will probably be dropped.

By direct multiplication we have

$$f(x) = \sum_{k=-2M}^{2M} \widetilde{f}_k e^{2\pi i k x/L}, \qquad x \in \mathbb{R}. \tag{2.22}$$

where the coefficients are given explicitly by

$$\widetilde{f}_k = \frac{1}{L}\int_0^L f(x) e^{-2\pi i k x/L}\, dx = \sum_{l=-M}^{M} \widetilde{g}_l \widetilde{h}_{k-l} \qquad k \in \mathbb{Z}, \tag{2.23}$$

where $\widetilde{g}_k$ and $\widetilde{h}_k$ vanish for $|k| > M$. While f contains contributions from Fourier modes $|k| \le 2M$, we assume that only the modes $|k| \le M$ will be retained in the model, so we want to compute the Fourier coefficients $\widetilde{f}_k$ for $k = -M, \ldots, M$ *without aliasing*, i.e., compute these coefficients exactly and simply drop the higher coefficients.

The formula on the right side of (2.23) shows how the Fourier modes in g and h interact to produce Fourier mode k in f. However, it is computationally too expensive to evaluate this sum directly, since it requires $O(M)$ operations for each k and thus $O(M^2)$ operations to compute $\widetilde{f}_k$ for $k = -M, \ldots, M$. Instead, we use the *transform method* (e.g., [3]) as follows:

1. From the Fourier coefficients $\widetilde{g}_k$ and $\widetilde{h}_k$ for $k = -M, \ldots, M$, evaluate g and h on the transform grid $x_j = jL/N$ for $j = 0, \ldots, N-1$ in physical space, i.e., compute $g_j = g(x_j)$ and $h_j = h(x_j)$.
2. Multiply these values together to evaluate f on the transform grid, i.e., $f_j = g_j h_j$ for $j = 0, \ldots, N-1$.
3. From the values f_j compute the Fourier coefficients $\widetilde{f}_k$ for $k = -M, \ldots, M$.

The key to the efficiency of the method is the fact that we can compute the transforms in steps 1 and 3 exactly by an IDFT and DFT, respectively, if the length N is large enough. Since step 2 requires N operations, if we use the FFT algorithm then the overall operation count is only $O\big(N \log(N)\big)$ operations.

To show that the transforms can be computed exactly—and to establish the transform length required—we use the following:

Lemma 3 *The left-sum approximation*

$$I(\phi) := \frac{1}{L}\int_0^L \phi(x)\, dx \approx \frac{1}{N}\sum_{j=0}^{N-1} \phi\left(\frac{jL}{N}\right) =: I_N(\phi) \tag{2.24}$$

is exact on the function $\phi_k(x) = \exp(2\pi i k x/L)$ *with* $k \in \mathbb{Z}$ *provided that* $|k| < N$.

Proof: The true integral is

$$I(\phi_k) = \frac{1}{L}\int_0^L e^{2\pi ikx/L}\,dx = \begin{cases} 1, & k = 0, \\ 0, & k \neq 0, \end{cases} \tag{2.25}$$

and by Lemma 1 the left-sum approximation is

$$I_N(\phi_k) = \frac{1}{N}\sum_{j=0}^{N-1} e^{2\pi ijk/N} = \begin{cases} 1, & k \equiv 0 \pmod N, \\ 0, & \text{otherwise.} \end{cases} \tag{2.26}$$

Comparing (2.25) and (2.26) we see that $I(\phi_k) = I_N(\phi_k)$ provided that $|k| < N$. □

To apply this Lemma we express the left side of (2.23) in the notation introduced above and substitute from (2.22) with k replaced by l to obtain

$$\widetilde{f}_k = I\left(f\phi_{-k}\right) = I\left(\sum_{l=-2M}^{2M} \widetilde{f}_l \phi_l \phi_{-k}\right) = \sum_{l=-2M}^{2M} \widetilde{f}_l I\left(\phi_{l-k}\right), \tag{2.27}$$

which we want to compute exactly for $|k| \leq M$. Likewise, the left-sum approximation using N points is

$$\frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-2\pi ijk/N} = I_N\left(f\phi_{-k}\right) = \sum_{l=-2M}^{2M} \widetilde{f}_l I_N\left(\phi_{l-k}\right). \tag{2.28}$$

Since $|l| \leq 2M$ and $|k| \leq n$, we have $|k-l| \leq 3M$, so by Lemma 3, $I_N(\phi_{k-l}) = I(\phi_{k-l})$ and thus

$$\widetilde{f}_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{2\pi ijk/N}, \qquad k = -M, \ldots, M \tag{2.29}$$

provided that $N > 3M$. Therefore, the transform method as described above computes the coefficients of the quadratic nonlinear term $f = gh$ exactly if $N > 3M$, where N is the transform length and M is the spectral truncation, i.e., the index of the highest Fourier coefficient retained in the spectral representations of f, g, and h. Since the transform length N is greater than the number $2M+1$ of Fourier coefficients retained, the coefficients of g and h must be "padded" with zeros before computing the IDFT (see Fig. 1 for an example of this padding). Likewise, the coefficients of f computed by the DFT must be truncated, i.e., the coeffients $\widetilde{f}_k$ for $|k| > M$ must be set to zero.[5]

[5]It is possible that by interpreting the DFT coefficients in terms of "unbalanced" discrete Fourier representations such as (2.12) the transform length N needed for exact results may be slightly smaller; we will not pursue this possiblity here.

2.2 Fourier representations in two dimensions

The model domain for `pswm` is doubly periodic with period L_x and L_y in x and y respectively. Here we give the details of two-dimensional Fourier representations on this domain.

2.2.1 Discrete Fourier transforms

Suppose that a given function f can be represented as a truncated double Fourier series with spectral truncation M_x and M_y in x and y, respectively. Repeating the one-dimensional representation developed above gives

$$f(x,y) = \sum_{l=-M_x}^{M_x} \widetilde{f}_l(y) e^{2\pi i l x/L_x} = \sum_{l=-M_x}^{M_x} \sum_{m=-M_y}^{M_y} \widetilde{f}_{l,m} e^{2\pi i l x/L_x} e^{2\pi i m y/L_y}, \qquad x,y \in \mathbb{R}, \tag{2.30}$$

where

$$\widetilde{f}_l(y) = \frac{1}{L_x}\int_0^{L_x} f(x,y) e^{-2\pi i l x/L_x}\, dx, \quad \widetilde{f}_{l,m} = \frac{1}{L_y}\int_0^{L_y} \widetilde{f}_l(y) e^{-2\pi i m y/L_y}\, dy, \qquad l,m \in \mathbb{Z}. \tag{2.31}$$

Evaluating the series in (2.30) at $x_j = jL_x/N_x$ and $y_k = kL_y/N_y$ gives

$$f_{j,k} = f(x_j,y_k) = \sum_{l=-M_x}^{M_x} \widetilde{f}_l(y_k) e^{2\pi i j l/N_x} = \sum_{l=-M_x}^{M_x} \sum_{m=-M_y}^{M_y} \widetilde{f}_{l,m} e^{2\pi i j l/N_x} e^{2\pi i k m/N_y} \tag{2.32}$$

for $j = 0,\ldots,N_x-1$ and $k = 0,\ldots,N_y-1$. As explained above, if $N_x > 2M_x$ and $N_y > 2M_y$, the Fourier series coefficients (2.31) may be evaluated exactly by the DFTs

$$\widehat{f}_l(y_k) = \frac{1}{N_x}\sum_{j=0}^{N_x-1} f_{j,k} e^{-2\pi i j l/N_x}, \quad \widehat{f}_{l,m} = \frac{1}{N_y}\sum_{k=0}^{N_y-1} \widehat{f}_l(y_k) e^{-2\pi i k m/N_y} \tag{2.33}$$

for $l = 0,\ldots,N_x-1$ and $m = 0,\ldots,N_y-1$; the DFT coefficients $\widehat{f}_l(y_k)$ and $\widehat{f}_{l,m}$ match the Fourier series coefficients $\widetilde{f}_l(y_k)$ and $\widetilde{f}_{l,m}$ for $|l| \le M_x$ and $|m| \le M_y$ and vanish for larger l and m. Since the DFT coeffcients are periodic, we can write the series (2.32) using IDFTs, i.e., in the form

$$f_{j,k} = \sum_{l=0}^{N_x-1} \widehat{f}_l(y_k) e^{2\pi i j l/N_x} = \sum_{l=0}^{N_x-1}\sum_{m=0}^{N_y-1} \widehat{f}_{l,m} e^{2\pi i j l/N_x} e^{2\pi i k m/N_y} \tag{2.34}$$

for $j = 0,\ldots,N_x-1$ and $k = 0,\ldots,N_y-1$. It should be noted that if $N_x = 2M_x$ and $N_y = 2M_y$ then the above results still hold, except that $\widetilde{f}_l(y_k) = \frac{1}{2}\widehat{f}_l(y_k)$ for $|l| = M_x$ and $\widetilde{f}_{l,m} = \frac{1}{2}\widehat{f}_{l,m}$ for $|m| = M_y$. Also, if f is real (the only case in which we are interested) then

$$\widetilde{f}_{-l}(y) = \widetilde{f}_l^*(y), \qquad \widetilde{f}_{-l,-m} = \widetilde{f}_{l,m}^*, \tag{2.35}$$

and

$$\widehat{f}_{-l}(y_k) = \widehat{f}_l^*(y_k). \qquad \widehat{f}_{-l,-m} = \widehat{f}_{l,m}^*. \tag{2.36}$$

2.2.2 Computation and storage

On the computer, we use the FFT routine `fft99` by Temperton [4, 5, 6] to compute the DFTs and IDFTs in (2.33) and (2.34), respectively. This routine computes real-to-"half-complex" transforms (or vice-versa), so applying it to the y-transforms—which are complex-to-complex transforms since $\widehat{f_l}(y_k)$ are complex numbers—does not produce the numbers $\widehat{f}_{k,l}$ directly. The procedure for using this routine is explained below and illustrated in Fig. 2.

We start with the numbers $f_{j,k}$ stored with explicit periodicity, i.e., stored for $j = -1, \ldots, N_x$ and $k = -1, \ldots, N_y$ as shown in Fig. 2 (top part). "Explicit periodicity" means that $f_{-1,k} = f_{N_x-1,k}$ and $f_{Nx,k} = f_{0,k}$ (with analogous periodicity in k); the two "extra" storage locations are needed for the calculations. The x-transform [first part of (2.33)] produces the half-complex sequence $\widehat{f_l}(y_k)$ for each $k = -1, \ldots, N_y$; what is actually returned by the routine `fft99` consists of the real and imaginary parts of this sequence, i.e., the real numbers $a_{l,k}$ and $b_{l,k}$ satisfying

$$\widehat{f_l}(y_k) = a_{l,k} + ib_{l,k} \tag{2.37}$$

for $l = 0, \ldots, M_x$, stored as shown in Fig. 2 (middle part). Here we use the notation $M_x = N_x/2$ for simplicity.[6] The remaining coefficients $\widehat{f_l}(y_k)$ for $l = Mx+1, \ldots, Nx-1$ are not computed explicitly but could be computed from $\widehat{f}_{Nx-l}(y_k) = \widehat{f_l}^*(y_k) = a_{l,k} - ib_{l,k}$. Also, $b_{0,k} = b_{M_x,k} = 0$ (since the values $f_{j,k}$ are real) corresponding to the two "extra" storage locations used for the input values.

Likewise, for the y transform [second part of (2.33)], since the DFT is linear we have

$$\widehat{f}_{l,m} = \widehat{a}_{l,m} + i\widehat{b}_{l,m} \tag{2.38}$$

where

$$\widehat{a}_{l,m} = \frac{1}{N_y}\sum_{k=0}^{N_y-1} a_{l,k}e^{-2\pi ikm/N_y}, \qquad \widehat{b}_{l,m} = \frac{1}{N_y}\sum_{k=0}^{N_y-1} b_{l,k}e^{-2\pi ikm/N_y} \tag{2.39}$$

are the DFTs of the (real) sequences $a_{l,k}$ and $b_{l,k}$, respectively. The sequences $\widehat{a}_{l,m}$ and $\widehat{b}_{l,m}$ are half-complex, i.e.,

$$\widehat{a}_{l,N_y-m} = \widehat{a}^*_{l,m}, \qquad \widehat{b}_{l,N_y-m} = \widehat{b}^*_{l,m}, \tag{2.40}$$

so `fft99` returns their real and imaginary parts, which we denote here by

$$\widehat{a}_{l,m} = A_{l,m} + iC_{l,m}, \qquad \widehat{b}_{l,m} = B_{l,m} + iD_{l,m} \tag{2.41}$$

for $m = 0, \ldots, M_y = N_y/2$ as shown in Fig. 2 (bottom part). Once again, since the sequences $a_{l,k}$ and $b_{l,k}$ are real, we have $C_{l,0} = C_{l,M_y} = 0$ and $D_{l,0} = D_{l,M_y} = 0$.

[6] This is just notation here, *not* the spectral truncation. The model actually uses the spectral trunctation $M_x < N_x/3$ in order to compute quadratic nonlinearities without aliasing as discussed previously.

$$
\begin{array}{cccccccccc}
k{=}N_y & f_{-1,N_y} & f_{0,N_y} & f_{1,N_y} & f_{2,N_y} & \cdots & f_{j,N_y} & \cdots & f_{N_x-1,N_y} & f_{N_x,N_y} \\
k{=}N_y{-}1 & f_{-1,N_y-1} & f_{0,N_y-1} & f_{1,N_y-1} & f_{2,N_y-1} & \cdots & f_{j,N_y-1} & \cdots & f_{N_x-1,N_y-1} & f_{N_x,N_y-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots \\
k & f_{-1,k} & f_{0,k} & f_{1,k} & f_{2,k} & \cdots & f_{j,k} & \cdots & f_{N_x-1,k} & f_{N_x,k} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots \\
k{=}0 & f_{-1,0} & f_{0,0} & f_{1,0} & f_{2,0} & \cdots & f_{j,0} & \cdots & f_{N_x-1,0} & f_{N_x,0} \\
k{=}{-}1 & f_{-1,-1} & f_{0,-1} & f_{1,-1} & f_{2,-1} & \cdots & f_{j,-1} & \cdots & f_{N_x-1,-1} & f_{N_x,-1} \\
 & j{=}{-}1 & j{=}0 & j{=}1 & j{=}2 & \cdots & j & \cdots & j{=}N_x{-}1 & j{=}N_x
\end{array}
$$

$\Downarrow$ forward x-transform $\Downarrow$ $\qquad$ $\Uparrow$ inverse x-transform $\Uparrow$

$$
\begin{array}{cccccccccc}
k{=}N_y & a_{0,N_y} & b_{0,N_y} & a_{1,N_y} & b_{1,N_y} & \cdots & a_{l,N_y} & b_{l,N_y} & \cdots & a_{M_x,N_y} & b_{M_x,N_y} \\
k{=}N_y{-}1 & a_{0,N_y-1} & b_{0,N_y-1} & a_{1,N_y-1} & b_{1,N_y-1} & \cdots & a_{l,N_y-1} & b_{l,N_y-1} & \cdots & a_{M_x,N_y-1} & b_{M_x,N_y-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & \\
k & a_{0,k} & b_{0,k} & a_{1,k} & b_{1,k} & \cdots & a_{l,k} & b_{l,k} & \cdots & a_{M_x,k} & b_{M_x,k} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & \\
k{=}0 & a_{0,0} & b_{0,0} & a_{1,0} & b_{1,0} & \cdots & a_{l,0} & b_{l,0} & \cdots & a_{M_x,0} & b_{M_x,0} \\
k{=}{-}1 & a_{0,-1} & b_{0,-1} & a_{1,-1} & b_{1,-1} & \cdots & a_{l,-1} & b_{l,-1} & \cdots & a_{M_x,-1} & b_{M_x,-1} \\
 & j{=}{-}1 & j{=}0 & j{=}1 & j{=}2 & \cdots & j{=}2l{-}1 & j{=}2l & \cdots & j{=}N_x{-}1 & j{=}N_x
\end{array}
$$

$\Downarrow$ forward y-transform $\Downarrow$ $\qquad$ $\Uparrow$ inverse y-transform $\Uparrow$

$$
\begin{array}{cccccccccc}
k{=}N_y & C_{0,M_y} & D_{0,M_y} & C_{1,M_y} & D_{1,M_y} & \cdots & C_{l,M_y} & D_{l,M_y} & \cdots & C_{M_x,M_y} & D_{M_x,M_y} \\
k{=}N_y & A_{0,M_y} & B_{0,M_y} & A_{1,M_y} & B_{1,M_y} & \cdots & A_{l,M_y} & B_{l,M_y} & \cdots & A_{M_x,M_y} & B_{M_x,M_y} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & \\
k{=}2m & C_{0,m} & D_{0,m} & C_{1,m} & D_{1,m} & \cdots & C_{l,m} & D_{l,m} & \cdots & C_{M_x,m} & D_{M_x,m} \\
k{=}2m{-}1 & A_{0,m} & B_{0,m} & A_{1,m} & B_{1,m} & \cdots & A_{l,m} & B_{l,m} & \cdots & A_{M_x,m} & B_{M_x,m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & \\
k{=}2 & C_{0,1} & D_{0,1} & C_{1,1} & D_{1,1} & \cdots & C_{l,1} & D_{l,1} & \cdots & C_{M_x,1} & D_{M_x,1} \\
k{=}1 & A_{0,1} & B_{0,1} & A_{1,1} & B_{1,1} & \cdots & A_{l,1} & B_{l,1} & \cdots & A_{M_x,1} & B_{M_x,1} \\
k{=}0 & C_{0,0} & D_{0,0} & C_{1,0} & D_{1,0} & \cdots & C_{l,0} & D_{l,0} & \cdots & C_{M_x,0} & D_{M_x,0} \\
k{=}{-}1 & A_{0,0} & B_{0,0} & A_{1,0} & B_{1,0} & \cdots & A_{l,0} & B_{l,0} & \cdots & A_{M_x,0} & B_{M_x,0} \\
 & j{=}{-}1 & j{=}0 & j{=}1 & j{=}2 & \cdots & j{=}2l{-}1 & j{=}2l & \cdots & j{=}N_x{-}1 & j{=}N_x
\end{array}
$$

Figure 2: Storage for values of a function f and corresponding Fourier coefficients (here $M_x = N_x/2$ and $M_y = N_y/2$).

From (2.38) and (2.41) we see that for each $l = 0, \ldots, N_x$ and $m = 0, \ldots, N_y$ the four numbers $A_{l,m}$, $B_{l,m}$, $C_{l,m}$, and $D_{l,m}$ yield the Fourier coefficients

$$\widehat{f}_{l,m} = (A_{l,m} - D_{l,m}) + i(B_{l,m} + C_{l,m}), \tag{2.42}$$

and using (2.36) and (2.40) we find that

$$\widehat{f}_{Nx-l,m} = (A_{l,m} + D_{l,m}) - i(B_{l,m} - C_{l,m}), \tag{2.43}$$

$$\widehat{f}_{l,N_y-m} = (A_{l,m} + D_{l,m}) + i(B_{l,m} - C_{l,m}), \tag{2.44}$$

$$\widehat{f}_{N_x-l,N_y-m} = (A_{l,m} - D_{l,m}) - i(B_{l,m} + C_{l,m}). \tag{2.45}$$

We can denote these formulas by the relation

$$\widehat{f}_{l,m} \leftrightarrow \begin{bmatrix} C_{l,m} & D_{l,m} \\ A_{l,m} & B_{l,m} \end{bmatrix} \tag{2.46}$$

where the array on the right-hand side is neither a matrix nor a stencil, but simply a schematic representation of the locations of the four numbers $A_{l,m}$, $B_{l,m}$, $C_{l,m}$, and $D_{l,m}$ related to the coefficient $\widehat{f}_{l,m}$ in the array produced by two applications of the FFT routine `fft99`. However, it is probably never necessary to actually compute the Fourier coefficients $\widehat{f}_{l,m}$ explicitly; all necessary calculations can be done with these four numbers directly. For testing the transforms, we can invert (2.42)–(2.45) to obtain

$$A_{l,m} = \frac{1}{4}\left(\widehat{f}_{l,m} + \widehat{f}_{-l,m} + \widehat{f}_{l,-m} + \widehat{f}_{-l,-m}\right), \tag{2.47}$$

$$B_{l,m} = \frac{1}{4i}\left(\widehat{f}_{l,m} - \widehat{f}_{-l,m} + \widehat{f}_{l,-m} - \widehat{f}_{-l,-m}\right), \tag{2.48}$$

$$C_{l,m} = \frac{1}{4i}\left(\widehat{f}_{l,m} + \widehat{f}_{-l,m} - \widehat{f}_{l,-m} - \widehat{f}_{-l,-m}\right), \tag{2.49}$$

$$D_{l,m} = \frac{1}{4}\left(-\widehat{f}_{l,m} + \widehat{f}_{-l,m} + \widehat{f}_{l,-m} - \widehat{f}_{-l,-m}\right). \tag{2.50}$$

2.2.3 Derivatives

The most important operations computed in spectral space are the derivatives. Letting g and h denote the x and y derivatives of f, respectively, then by (2.16) their spectral coefficients are given by

$$\widehat{g}_{l,m} = \left(\frac{2\pi i l}{L_x}\right)\widehat{f}_{l,m}, \qquad \widehat{h}_{l,m} = \left(\frac{2\pi i m}{L_y}\right)\widehat{f}_{l,m}. \tag{2.51}$$

Using (2.45) in these formulas gives

$$\widehat{g}_{l,m} = \left(\frac{2\pi l}{L_x}\right)\Big[-(B_{l,m}+C_{l,m})+i(A_{l,m}-D_{l,m})\Big], \tag{2.52}$$

$$\widehat{g}_{Nx-l,m} = \left(\frac{2\pi l}{L_x}\right)\Big[-(B_{l,m}-C_{l,m})-i(A_{l,m}+D_{l,m})\Big], \tag{2.53}$$

$$\widehat{g}_{l,N_y-m} = \left(\frac{2\pi l}{L_x}\right)\Big[-(B_{l,m}-C_{l,m})+i(A_{l,m}+D_{l,m})\Big], \tag{2.54}$$

$$\widehat{g}_{N_x-l,N_y-m} = \left(\frac{2\pi l}{L_x}\right)\Big[-(B_{l,m}+C_{l,m})-i(A_{l,m}-D_{l,m})\Big] \tag{2.55}$$

and

$$\widehat{h}_{l,m} = \left(\frac{2\pi m}{L_y}\right)\Big[-(B_{l,m}+C_{l,m})+i(A_{l,m}-D_{l,m})\Big], \tag{2.56}$$

$$\widehat{h}_{Nx-l,m} = \left(\frac{2\pi m}{L_y}\right)\Big[(B_{l,m}-C_{l,m})+i(A_{l,m}+D_{l,m})\Big], \tag{2.57}$$

$$\widehat{h}_{l,N_y-m} = \left(\frac{2\pi m}{L_y}\right)\Big[(B_{l,m}-C_{l,m})-i(A_{l,m}+D_{l,m})\Big], \tag{2.58}$$

$$\widehat{h}_{N_x-l,N_y-m} = \left(\frac{2\pi m}{L_y}\right)\Big[-(B_{l,m}+C_{l,m})-i(A_{l,m}-D_{l,m})\Big]. \tag{2.59}$$

These may be summarized in the schematic notation of (2.46) in the form

$$(\widehat{f_x})_{l,m} \leftrightarrow \left(\frac{2\pi l}{L_x}\right)\begin{bmatrix}-D_{l,m} & C_{l,m}\\ -B_{l,m} & A_{l,m}\end{bmatrix}, \qquad (\widehat{f_y})_{l,m} \leftrightarrow \left(\frac{2\pi m}{L_y}\right)\begin{bmatrix}A_{l,m} & B_{l,m}\\ -C_{l,m} & -D_{l,m}\end{bmatrix}. \tag{2.60}$$

Also, second derivatives have the simpler representation

$$(\widehat{f_{xx}})_{l,m} \leftrightarrow -\left(\frac{2\pi l}{L_x}\right)^2\begin{bmatrix}A_{l,m} & B_{l,m}\\ C_{l,m} & D_{l,m}\end{bmatrix}, \qquad (\widehat{f_{yy}})_{l,m} \leftrightarrow -\left(\frac{2\pi m}{L_y}\right)^2\begin{bmatrix}A_{l,m} & B_{l,m}\\ C_{l,m} & D_{l,m}\end{bmatrix}. \tag{2.61}$$

The important point here is that the operations of taking derivatives may be represented in spectral space by scaling the four numbers corresponding to a given Fourier (l,m) and rearranging them and changing their signs as needed.

Also note the point made in the previous section about discrete Fourier representations when the transform length N is twice the spectral truncation M. Suppose, for example, that we wish to take the x-derivative of a real-valued function f represented in Fourier spectral space with $N_x = 2M_x$. Since $\widehat{f}_l(y_k)$ is real for $l = M_x$ in (2.37), we have $b_{M_x,k} = 0$ and thus $B_{M_x,m} = D_{M_x,m} = 0$ from (2.41). From the first part of (2.60) we see that the information in Fourier mode $l = M_x$ (i.e., $A_{M_x,m}$ and $C_{M_x,m}$) will be lost in taking the derivative, since `fft99` ignores the imaginary part of this mode (it must be zero to produce real values). This problem does not affect the second derivative as in (2.61), nor will it affect derivatives when $M_x < 2N_x$.

References

[1] Fulton, S. R., 2007: Semi-Implicit Discretization of the Shallow-Water Equations. Unpublished manuscript.

[2] Brandt, A., S. R. Fulton, and G. D. Taylor, 1984: Improved spectral multigrid methods for periodic elliptic problems. *J. Comp. Phys.*, **58**, 96–112.

[3] Orszag, S. A., 1970: Transform method for the calculation of vector-coupled sums: application to the spectral form of the vorticity equation. *J. Atmos. Sci.*, **27**, 890–895.

[4] Temperton, C., 1983a: Self-sorting mixed-radix fast Fourier transforms. *J. Comp. Phys.*, **52**, 1–23.

[5] Temperton, C., 1983b: A note on prime factor FFT algorithms. *J. Comp. Phys.*, **52**, 198–204.

[6] Temperton, C., 1983c: Fast mixed-radix real Fourier transforms. *J. Comp. Phys.*, **52**, 340–350.