Laurette TUCKERMAN
laurette@pmmh.espci.fr

Numerical Methods for Differential Equations in Physics

Fast Fourier Transform

$$\hat{u}_{k+N} = \sum_{\ell=0}^{N-1} u(x_\ell) e^{-i(k+N)2\pi\ell/N} = \sum u(x_\ell) e^{-ik2\pi\ell/N} e^{-iN2\pi\ell/N}$$

$$= \sum u(x_\ell) e^{-ik2\pi\ell/N} = \hat{u}_k \qquad \hat{u}_k \text{ is } N\text{-periodic}$$

$$\hat{u}_k = \sum_{\ell=0}^{N/2-1} u(x_{2\ell}) e^{-i2\pi(2\ell)k/N} + \sum_{\ell=0}^{N/2-1} u(x_{2\ell+1}) e^{-i2\pi(2\ell+1)k/N}$$

$$= \sum_{\ell=0}^{N/2-1} u(x_{2\ell}) e^{-i2\pi\ell k/(N/2)} + e^{-i2\pi k/N} \sum_{\ell=0}^{N/2-1} u(x_{2\ell+1}) e^{-i2\pi\ell k/(N/2)}$$

$$= \sum_{\ell=0}^{N/2-1} v(x_\ell) e^{-i2\pi\ell k/(N/2)} + e^{-i2\pi k/N} \sum_{\ell=0}^{N/2-1} w(x_\ell) e^{-i2\pi\ell k/(N/2)}$$

$$= \hat{v}_k + e^{-i2\pi k/N} \hat{w}_k$$

$$\hat{u}_{k+N/2} = \hat{v}_k - e^{-i2\pi k/N} \hat{w}_k$$

v, w are of length $N/2$

Two transforms each of size $N/2$, and N additions $\to 2(N/2)^2 + N$

Each trans of size $N/2$ becomes two trans of size $N/4$, and $N/2$ adds.

Each trans of size $N/4$ becomes two trans of size $N/8$, and $N/4$ adds.

$$\frac{N}{2} \longrightarrow 2\left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N$$

$$\frac{N}{4} \longrightarrow 2\left(2\left(\frac{N}{4}\right)^2 + \frac{N}{2}\right) + N = \frac{N^2}{4} + 2N$$

$$\frac{N}{8} \longrightarrow 2\left(2\left(2\left(\frac{N}{8}\right)^2 + \frac{N}{4}\right) + \frac{N}{2}\right) + N = \frac{N^2}{8} + 3N$$

$N = 2^p$:

$$\frac{N}{2^p} \longrightarrow \frac{N^2}{2^p} + pN = N + (\log_2 N)N = O(N \log_2 N)$$

Fourier transform in one dimension:

$$\widehat{u}_k = \sum_{\ell=0}^{N-1} u(x_\ell) e^{-i2\pi k\ell/N}$$

One multiplication for each k, ℓ, so $O(N_x^2)$ operations.

Fourier transform in two dimensions:

$$\widehat{\widehat{u}}_{k,m} = \sum_{n=0}^{N_y-1} \underbrace{\left(\sum_{\ell=0}^{N_x-1} u(x_\ell, y_n) e^{-i2\pi k\ell/N_x} \right)}_{\widehat{u}_k(y_n)} e^{-i2\pi mn/N_y}$$

$$\{u(x_\ell, y_n)\} \quad \xrightarrow{N_x^2 N_y} \quad \{\widehat{u}_k(y_n)\} \quad \xrightarrow{N_x N_y^2} \quad \{\widehat{\widehat{u}}_{k,m}\}$$

Total: $N_x N_y (N_x + N_y) \ll N_x^2 N_y^2$

With FFT: $N_x N_y (\log N_x + \log N_y) = N_x N_y \log(N_x N_y)$

Even without FFT, multidimensional Fourier transform would be fast because the different dimensions are decoupled: $N_x N_y N_z (N_x + N_y + N_z)$

Fourier transform in x is action with matrix $F^x_{k,\ell}\delta_{m,n}\delta_{i,j}$
Fourier transform in y is action with matrix $F^y_{m,n}\delta_{k,\ell}\delta_{i,j}$
Fourier transform in z is action with matrix $F^z_{i,j}\delta_{m,n}\delta_{k,\ell}$

Contrast with convolution: not separable

Convolution in one dimension

$$\widehat{fg}(k) = \left(\widehat{f} * \widehat{g}\right)(k) = \sum_{\ell} \widehat{f}_{\ell}\, \widehat{g}_{k-\ell}$$

One multiplication for each k, ℓ, so $O(N_x^2)$ operations.

Convolution in two dimensions:

$$\widehat{fg}(k,m) = \left(\widehat{f} * \widehat{g}\right)(k,m) = \sum_{\ell}\sum_{n} \widehat{f}_{\ell,n}\, \widehat{g}_{k-\ell,m-n}$$

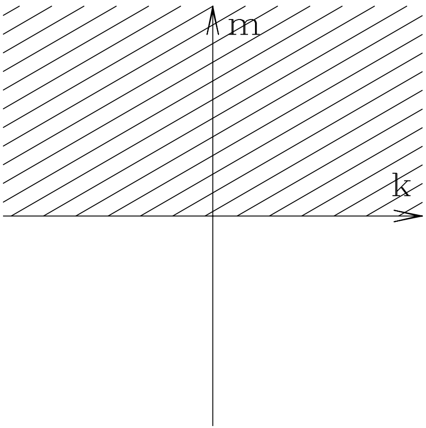
One multiplication for each k, ℓ, m, n, so $O(N_x^2 N_y^2)$ operations.

Multidimensional Fourier transform of real data

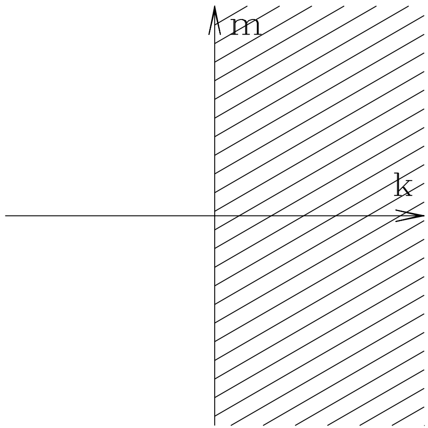
For $u(x)$ real, $\hat{u}_k$ is conjugate symmetric: $\hat{u}_{-k} = \hat{u}_k^*$ so that

$$u(x) \sim \hat{u}_k e^{ikx} + \hat{u}_{-k} e^{-ikx} = \hat{u}_k e^{ikx} + \left(\hat{u}_k e^{ikx}\right)^* = 2\mathcal{R}e\left(\hat{u}_k e^{ikx}\right)$$

For $u(x,y)$ real, $\hat{u}_{-k,-m} = \hat{u}_{k,m}^*$ so need half of (k,m) plane.



or



For $u(x,y,z)$ real, $\hat{u}_{-k,-m,-n} = \hat{u}_{k,m,n}^*$ so need half of (k,m,n) plane.

Discretization in Two or Three Dimensions

Periodic BCs in x and y: Fourier-Fourier

$$u(x,y) = \sum_{k,m} \hat{u}_{k,m} e^{ikx} e^{imy} \qquad g(x,y) = \sum_{k,m} \hat{g}_{k,m} e^{ikx} e^{imy}$$

$$\Delta u = \sum_{k,m} (-k^2 - m^2) \hat{u}_{k,m} e^{ikx} e^{imy} = \sum_{k,m} \hat{g}_{k,m} e^{ikx} e^{imy}$$

$$\hat{u}_{k,m} = \frac{-\hat{g}_{k,m}}{k^2 + m^2}$$

Have used interval $[0, 2\pi)$ for simplicity. More generally,

domain is $[0, L_x) \times [0, L_y)$ and basis functions are $e^{ikx2\pi/L_x} e^{imy2\pi/L_y}$

Periodic BCs in x, Dirichlet BCs in y:

Fourier-Finite Differences

$$u(x,y) = \sum_k \hat{u}_k(y) e^{ikx} \qquad g(x,y) = \sum_k \hat{g}_k(y) e^{ikx}$$

$$\begin{aligned}\Delta u &= \sum_k \left(-k^2 \hat{u}_k(y) + \frac{\hat{u}_k(y+\Delta y) - 2\hat{u}_k(y) + \hat{u}_k(y-\Delta y)}{(\Delta y)^2} \right) e^{ikx} \\ &= \sum_k \frac{\hat{u}_k(y+\Delta y) + \left(-2 - (k\Delta y)^2\right) \hat{u}_k(y) + \hat{u}_k(y-\Delta y)}{(\Delta y)^2} e^{ikx}\end{aligned}$$

For each Fourier component k

$$\frac{1}{(\Delta y)^2} \begin{pmatrix} -2-(k\Delta y)^2 & 1 & & & \\ 1 & -2-(k\Delta y)^2 & 1 & & \\ & 1 & -2-(k\Delta y)^2 & 1 & \\ & & & \ddots & \\ & & & 1 & -2-(k\Delta y)^2) \end{pmatrix} \begin{pmatrix} \hat{u}_k(y_1) \\ \hat{u}_k(y_2) \\ \hat{u}_k(y_3) \\ \vdots \\ \hat{u}_k(y_N) \end{pmatrix}$$

Boundary conditions needed for each Fourier component k, e.g.

$$\frac{1}{(\Delta y)^2}\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & -2-(k\Delta y)^2 & 1 & & \\ & 1 & -2-(k\Delta y)^2 & 1 & \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}\begin{pmatrix} \hat{u}_k(y_1) \\ \hat{u}_k(y_2) \\ \hat{u}_k(y_3) \\ \vdots \\ \hat{u}_k(y_N) \end{pmatrix} = \begin{pmatrix} \gamma_- \\ \hat{g}_k(y_2) \\ \hat{g}_k(y_3) \\ \vdots \\ \gamma_+ \end{pmatrix}$$

Elliptic equations need boundary conditions (Dirichlet, Neumann or periodic) along all boundaries of domain

Periodic BCs in x, Dirichlet BCs in y: Fourier-Chebyshev

$$u(x,y) = \sum_{k,n} u_{k,n} e^{ikx} T_n(y) \qquad g(x,y) = \sum_{k,n} g_{k,n} e^{ikx} T_n(y)$$

$$\begin{aligned}
\Delta u &= \sum_{k,n} -k^2 u_{k,n} e^{ikx} T_n(y) + \sum_{k,n} u_{k,n} e^{ikx} T_n''(y) \\
&= \sum_{k,n} -k^2 u_{k,n} e^{ikx} T_n(y) + \sum_{k,n} u_{k,n} e^{ikx} \sum_m R_{m,n} T_m(y) \\
&= \sum_{k,n} -k^2 u_{k,n} e^{ikx} T_n(y) + \sum_{k,m} \left(\sum_n R_{m,n} u_{k,n} \right) e^{ikx} T_m(y) \\
&= \sum_{k,n} -k^2 u_{k,n} e^{ikx} T_n(y) + \sum_{k,m} (Ru_k)_m e^{ikx} T_m(y) \\
&= \sum_{k,n} -k^2 u_{k,n} e^{ikx} T_n(y) + \sum_{k,n} (Ru_k)_n e^{ikx} T_n(y) \\
&= \sum_{k,n} \left(-k^2 u_{k,n} + (Ru_k)_n \right) e^{ikx} T_n(y)
\end{aligned}$$

where $(Ru_k)_m \equiv \sum_m R_{m,n} u_{k,n}$

$$\Delta u(x,y) = \sum_{k,n} \left(\sum_{k',n'} \Delta_{k,n,k',n'} u_{k',n'} \right) e^{ikx} T_n(y)$$

where $\quad \Delta_{k,n,k',n'} = -k^2 \delta_{k,k'} \delta_{n,n'} + R_{n,n'} \delta_{k,k'}$

Operators in x commute with operators in y.

Odd and even Chebyshev polynomials are decoupled.

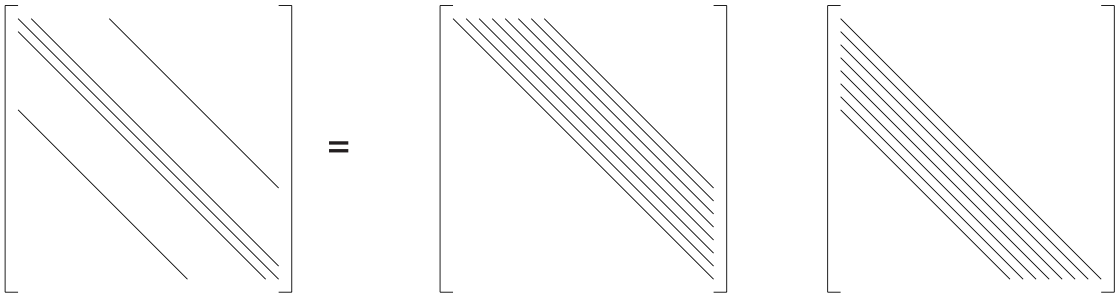
Recall: second-derivative Chebyshev matrix R is upper triangular
tridiagonal matrix B is such that BR is tridiagonal.

$$B\Delta u(x,y) = \sum_{k,n,n'} \left(-k^2 B_{n,n'} + (BR)_{n,n'} \right) u_{k',n'} e^{ikx} T_n(y)$$

Boundary conditions needed

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ (BR-k^2B)_{2,0} & (BR-k^2B)_{2,2} & (BR-k^2B)_{2,4} & & \\ & (BR-k^2B)_{4,2} & (BR-k^2B)_{4,4} & (BR-k^2B)_{4,6} & \\ & & (BR-k^2B)_{6,4} & (BR-k^2B)_{6,6} & (BR-k^2B)_{6,8} \end{pmatrix}$$

Dirichlet BCs in x and y: Finite differences/Finite differences



Fill-in to maximal bandwidth: bandedness not preserved

Bandwidth $J = N_x$, so operation count would be $O(J^2N) = O(N_x^2 N_x N_y)$ for LU decomposition and $O(JN) = O(N_x N_x N_y)$ for backsolve.

Alternative way of inverting: diagonalize in one or both directions.
Recall that operators in x commute with operators in y
so they are simultaneously diagonalizable

$$\Delta(n,m,n',m') = D_{xx}(n,n')\delta(m,m') + D_{yy}(m,m')\delta(n,n')$$
$$= \left(V_x\Lambda_x V_x^{-1}\right)(n,n')\delta(m,m') + \left(V_y\Lambda_y V_y^{-1}\right)(m,m')\delta(n,n')$$

Diagonalization (cost $N_x^3 + N_y^3$):

$$D_{xx} = V_x\Lambda_x V_x^{-1} \qquad D_{yy} = V_y\Lambda_y V_y^{-1}$$

V_x, V_y are the $N_x \times N_x$ and $N_y \times N_y$ matrices of eigenvectors
D_{xx}, D_{yy} are $N_x \times N_x$ and $N_y \times N_y$ matrices representing ∂_{xx}, ∂_{yy}
Λ_x and Λ_y are diagonal matrices containing the eigenvalues

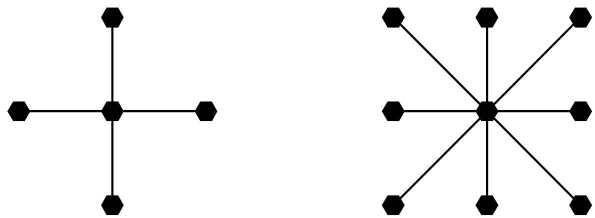
Transform to x and y eigenspace $O(N_x^2 N_y + N_y^2 N_x)$
Invert Laplacian in eigenspace: $O(N_x N_y)$
Inverse transform back from x and y eigenspace $O(N_x^2 N_y + N_y^2 N_x)$
Total: $O(N_x N_y (N_x + N_y))$ **Storage: $O(N_x^2 + N_y^2)$**
Can use diag in x and banded LU in y with cost $O(N_x N_y (N_x + 3))$
Can also do in 3D, with timing $O(N_x N_y N_z (N_x + N_y + N_z))$

Write $f(x,y)$ as $N_x \times N_y$ matrix instead of vector of length $N_x N_y$

$$
\begin{aligned}
\partial_{xx} f(x_i, y_j) &= \sum_k D_{xx}(i,k) f(x_k, y_j) = (D_{xx} f)(x_i, y_j) \\
\partial_{yy} f(x_i, y_j) &= \sum_k D_{yy}(j,k) f(x_i, y_k) = (f D_{yy}^T)(x_i, y_j) \\
\partial_{xx} f &= D_{xx} f = V_x \Lambda_x V_x^{-1} f \\
\partial_{yy} f &= f D_{yy}^T = f (V_y \Lambda_y V_y^{-1})^T = f (V_y^{-1})^T \Lambda_y V_y^T \\
\nabla^2 f &= g \\
V_x \Lambda_x V_x^{-1} f + f (V_y^{-1})^T \Lambda_y V_y^T &= g
\end{aligned}
$$

$$
\begin{aligned}
\Lambda_x \left[V_x^{-1} f (V_y^T)^{-1} \right] + \left[V_x^{-1} f (V_y^{-1})^T \right] \Lambda_y &= \left[V_x^{-1} g (V_y^T)^{-1} \right] \\
\left[V_x^{-1} f (V_y^T)^{-1} \right](i,j) &= \frac{\left[V_x^{-1} g (V_y^T)^{-1} \right](i,j)}{\Lambda_x(i) + \Lambda_y(j)} \\
f &= V_x \left[V_x^{-1} f (V_y^T)^{-1} \right] V_y^T
\end{aligned}
$$

Stencils for two-dimensional finite-difference Laplacian



Five-point stencil: $\frac{d^2u}{dx^2}(x_j, y_k) + \frac{d^2u}{dy^2}(x_j, y_k)$

$$\approx \frac{1}{h^2}\left(u(x_{j+1}, y_k) - 2u(x_j, y_k) + u(x_{j-1}, y_k)\right)$$

$$+ \frac{1}{h^2}\left(u(x_j, y_{k+1}) - 2u(x_j, y_k) + u(x_j, y_{k-1})\right)$$

Error is $\frac{h^2}{24}(\partial_x^4 + \partial_y^4)u + \ldots$

This error is not isotropic, unlike the Laplacian itself.

We show that the continuous Laplacian IS isotropic:

Rotate $(x, y) \to (x', y')$: $\quad x' = \alpha x + \beta y, \quad y' = -\beta x + \alpha y, \quad \alpha^2 + \beta^2 = 1$

$$\frac{\partial}{\partial x} = \frac{\partial x'}{\partial x}\frac{\partial}{\partial x'} + \frac{\partial y'}{\partial x}\frac{\partial}{\partial y'} = \alpha\frac{\partial}{\partial x'} - \beta\frac{\partial}{\partial y'}$$

$$\frac{\partial^2}{\partial x^2} = \left(\alpha\frac{\partial}{\partial x'} - \beta\frac{\partial}{\partial y'}\right)^2$$

$$= \alpha^2\frac{\partial^2}{\partial x'^2} - 2\alpha\beta\frac{\partial^2}{\partial x'\partial y'} + \beta^2\frac{\partial^2}{\partial y'^2}$$

$$\frac{\partial}{\partial y} = \frac{\partial x'}{\partial y}\frac{\partial}{\partial x'} + \frac{\partial y'}{\partial y}\frac{\partial}{\partial y'} = \beta\frac{\partial}{\partial x'} + \alpha\frac{\partial}{\partial y'}$$

$$\frac{\partial^2}{\partial y^2} = \left(\beta\frac{\partial}{\partial x'} + \alpha\frac{\partial}{\partial y'}\right)^2$$

$$= \beta^2\frac{\partial^2}{\partial x'^2} + 2\alpha\beta\frac{\partial^2}{\partial x'\partial y'} + \alpha^2\frac{\partial^2}{\partial y'^2}$$

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \frac{\partial^2}{\partial x'^2} + \frac{\partial^2}{\partial y'^2}$$

Error of nine-point stencil is isotropic (to lowest order)

$$\frac{d^2u}{dx^2}(x_j, y_k) + \frac{d^2u}{dy^2}(x_j, y_k) \approx$$

$$\frac{1}{6h^2}\left[4\left(u(x_{j+1}, y_k) + u(x_{j-1}, y_k) + u(x_j, y_{k+1}) + u(x_j, y_{k-1})\right)\right.$$
$$+ u(x_{j+1}, y_{k+1}) + u(x_{j-1}, y_{k-1}) + u(x_{j+1}, y_{k-1}) + u(x_{j-1}, y_{k+1})$$
$$\left.-20u(x_j, y_k)\right]$$

Error: $\frac{h^2}{12}\left(\partial_x^2 + \partial_y^2\right)^2 = \frac{h^2}{12}\Delta^2$

An important iterative method for solving the Poisson equation:

Multigrid

Like the FFT, the multigrid algorithm relies on coarsening the grid recursively, solving on coarse grids, then returning to fine grids.

An additional reason to emphasize solutions to the Poisson equation:

Helmholtz problems

Parabolic problems (like heat equation, Navier-Stokes equations, ...)

Implicit schemes lead to Helmholtz-type problems such as

$$(I - \Delta t \nabla^2) f = g$$

which can be solved by the same techniques as the Poisson problem.