Recent Advances in the Pseudospectral Method

John C. Bowman

University of Alberta

May 23, 2016

`www.math.ualberta.ca/~bowman/talks`

Acknowledgements: Malcolm Roberts (Université de Strasbourg)

Pseudospectral Method

- Pseudospectral simulations are a widely used numerical tool for the study of fluid turbulence:
    - Fast $N \log N$ scaling for N modes.
    - Spectral accuracy: more accurate than finite-difference or finite-element methods.
- Ideal choice for studying homogenous turbulence with periodic boundary conditions.
- Generalizations such as Chebyshev collocation and penalty methods allow them to handle more complicated boundary conditions and geometries.
- However, in many cases pseudospectral methods do not parallelize well on massively parallel distributed architectures due to the communication costs of the parallel transpose.

Dealiasing

- Patterson and Orszag pioneered the pseudospectral method over 40 years ago.

- They emphasized that the convolution theorem necessitates *dealiasing* unwanted harmonics arising from the periodicity of the discrete Fourier transform.

Discrete Cyclic Convolution

- The FFT provides an efficient tool for computing the *discrete cyclic convolution*

$$\sum_{p=0}^{N-1} F_p G_{k-p},$$

  where the vectors F and G have period N.

- Define the *Nth primitive root of unity:*

$$\zeta_N = \exp\left(\frac{2\pi i}{N}\right).$$

- The fast Fourier transform method exploits the properties that $\zeta_N^r = \zeta_{N/r}$ and $\zeta_N^N = 1$.

- However, the pseudospectral method requires a *linear convolution.*

- The unnormalized *backwards discrete Fourier transform* of $\{F_k : k = 0, \ldots, N\}$ is

$$f_j \doteq \sum_{k=0}^{N-1} \zeta_N^{jk} F_k \qquad j = 0, \ldots, N-1.$$

- The corresponding *forward transform* is

$$F_k \doteq \frac{1}{N} \sum_{j=0}^{N-1} \zeta_N^{-kj} f_j \qquad k = 0, \ldots, N-1.$$

- The orthogonality of this transform pair follows from

$$\sum_{j=0}^{N-1} \zeta_N^{\ell j} = \begin{cases} N & \text{if } \ell = sN \text{ for } s \in \mathbb{Z}, \\ \dfrac{1-\zeta_N^{\ell N}}{1-\zeta_N^{\ell}} = 0 & \text{otherwise.} \end{cases}$$

Convolution Theorem

$$
\begin{aligned}
\sum_{j=0}^{N-1} f_j g_j \zeta_N^{-jk} &= \sum_{j=0}^{N-1} \zeta_N^{-jk} \left(\sum_{p=0}^{N-1} \zeta_N^{jp} F_p \right) \left(\sum_{q=0}^{N-1} \zeta_N^{jq} G_q \right) \\
&= \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} F_p G_q \sum_{j=0}^{N-1} \zeta_N^{(-k+p+q)j} \\
&= N \sum_{s} \sum_{p=0}^{N-1} F_p G_{k-p+sN}.
\end{aligned}
$$

- The terms indexed by $s \neq 0$ are *aliases;* we need to remove them!

- Aliases can be avoided by *zero padding* input data vectors of length m to length $N \geq 2m - 1$.

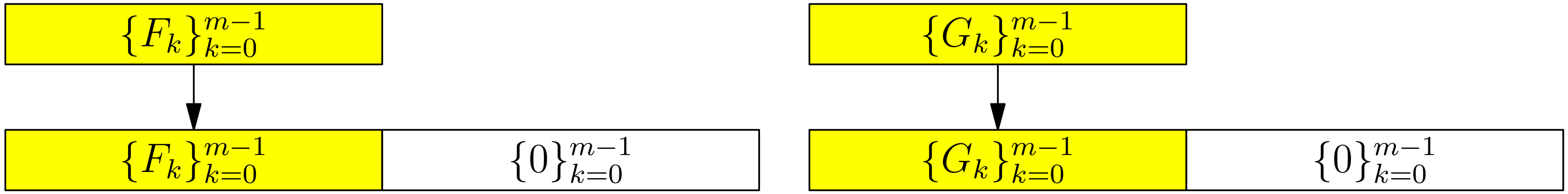
- *Explicit zero padding* prevents mode $m - 1$ from beating with itself and wrapping around to contaminate mode $N = 0 \bmod N$.

- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:

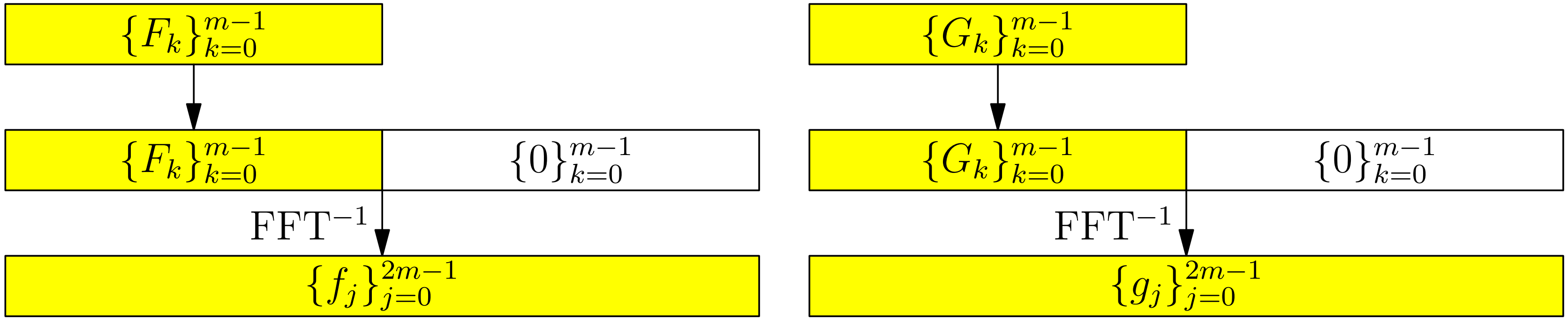
$\{F_k\}_{k=0}^{m-1}$

$\{G_k\}_{k=0}^{m-1}$

- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:



- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:



$\{F_k\}_{k=0}^{m-1}$

$\{F_k\}_{k=0}^{m-1}$ $\{0\}_{k=0}^{m-1}$

FFT^{-1}

$\{f_j\}_{j=0}^{2m-1}$

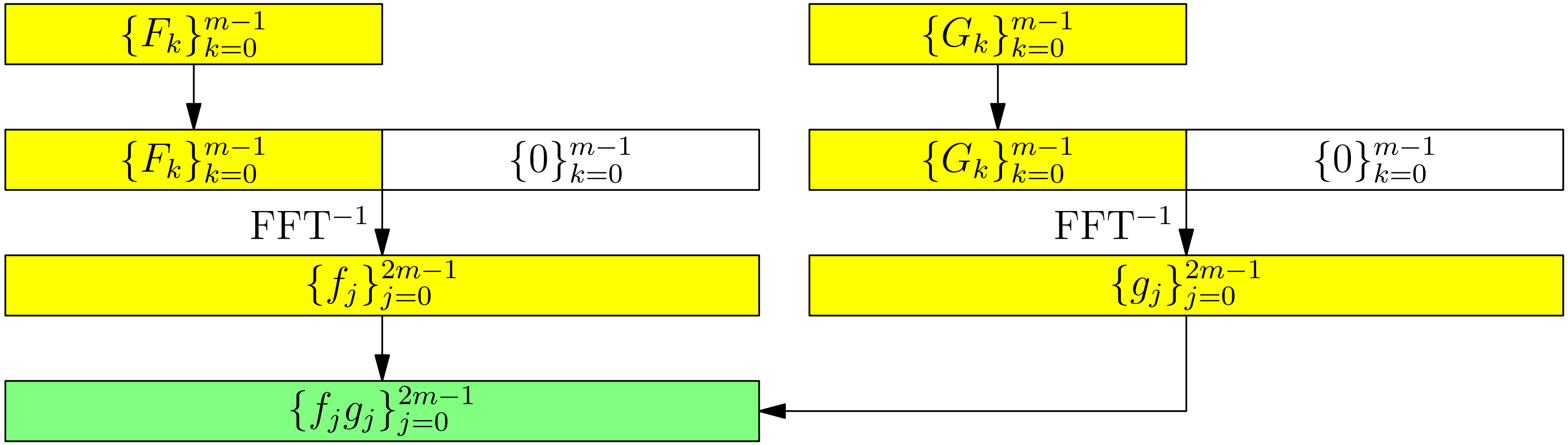
$\{G_k\}_{k=0}^{m-1}$

$\{G_k\}_{k=0}^{m-1}$ $\{0\}_{k=0}^{m-1}$

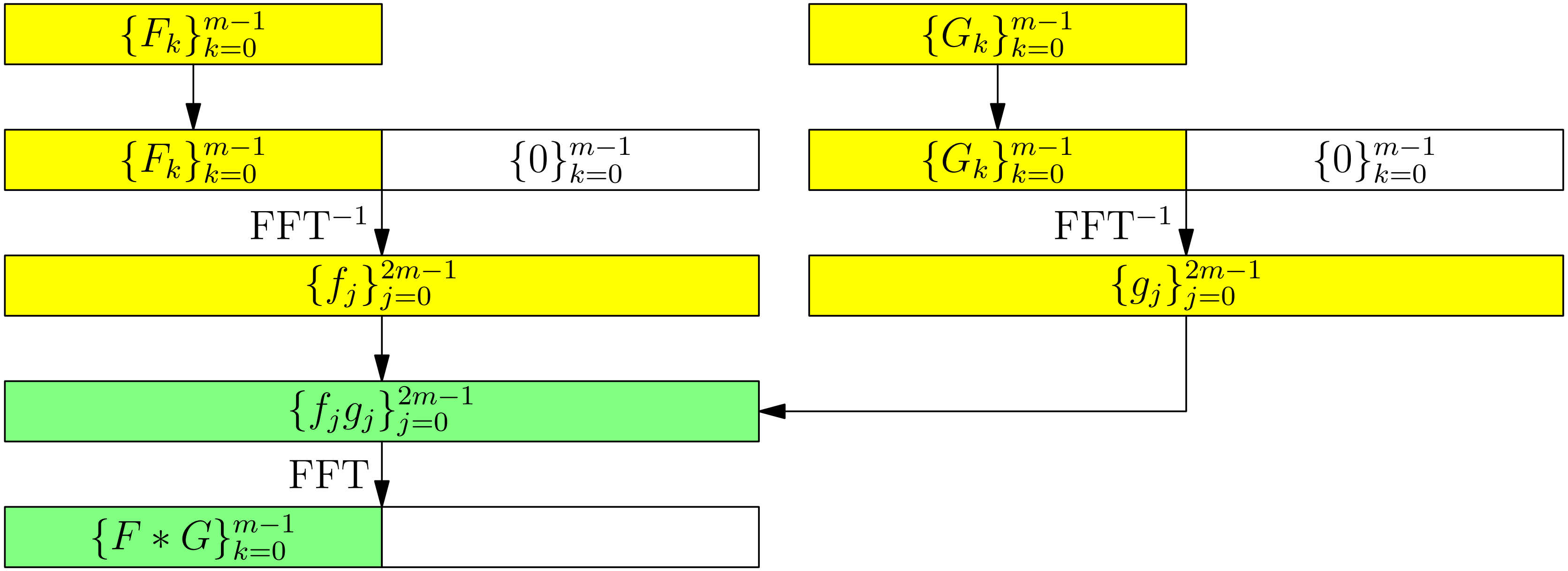
FFT^{-1}

$\{g_j\}_{j=0}^{2m-1}$

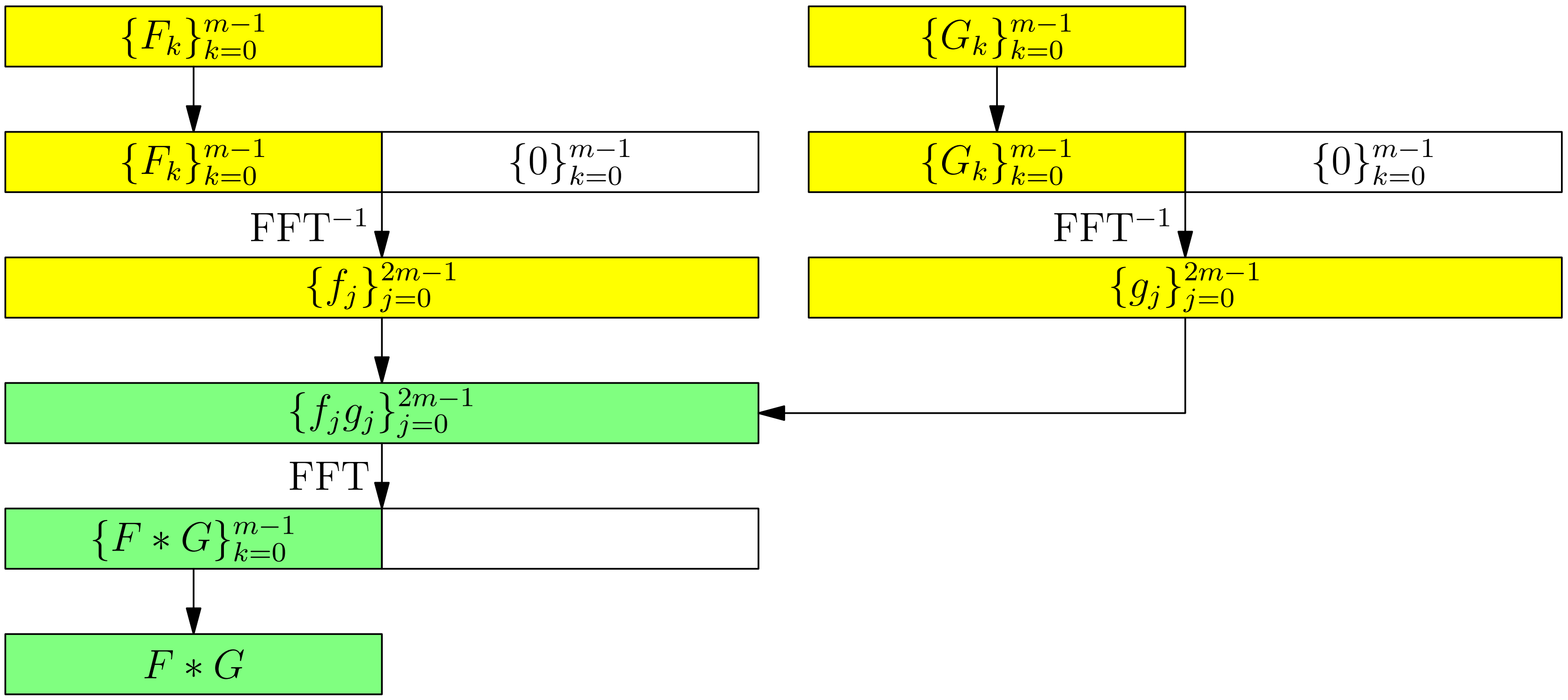
- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:



- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:



- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:



Implicit Padding

- Let $N = 2m$. For $j = 0, \ldots, 2m-1$ we want to compute

$$f_j = \sum_{k=0}^{2m-1} \zeta_{2m}^{jk} F_k.$$

- If $F_k = 0$ for $k \geq m$, one can easily avoid looping over the unwanted zero Fourier modes by decimating in wavenumber:

$$\begin{aligned} f_{2\ell} &= \sum_{k=0}^{m-1} \zeta_{2m}^{2\ell k} F_k = \sum_{k=0}^{m-1} \zeta_m^{\ell k} F_k, \\ f_{2\ell+1} &= \sum_{k=0}^{m-1} \zeta_{2m}^{(2\ell+1)k} F_k = \sum_{k=0}^{m-1} \zeta_m^{\ell k} \zeta_{2m}^{k} F_k, \qquad \ell = 0, 1, \ldots m-1. \end{aligned}$$

- This requires computing two subtransforms, each of size m, for an overall computational scaling of order $2m \log_2 m = N \log_2 m$.

- Odd and even terms of the convolution can then be computed separately, multiplied term-by-term, and transformed again to Fourier space:

$$
\begin{aligned}
2mF_k &= \sum_{j=0}^{2m-1} \zeta_{2m}^{-kj} f_j \\
&= \sum_{\ell=0}^{m-1} \zeta_{2m}^{-k2\ell} f_{2\ell} + \sum_{\ell=0}^{m-1} \zeta_{2m}^{-k(2\ell+1)} f_{2\ell+1} \\
&= \sum_{\ell=0}^{m-1} \zeta_{m}^{-k\ell} f_{2\ell} + \zeta_{2m}^{-k} \sum_{\ell=0}^{m-1} \zeta_{m}^{-k\ell} f_{2\ell+1} \qquad k = 0, \ldots, m-1.
\end{aligned}
$$

- No bit reversal is required at the highest level.

- A 1D implicitly padded convolution is implemented in our `FFTW++` library: `http://fftwpp.sourceforge.net/`.

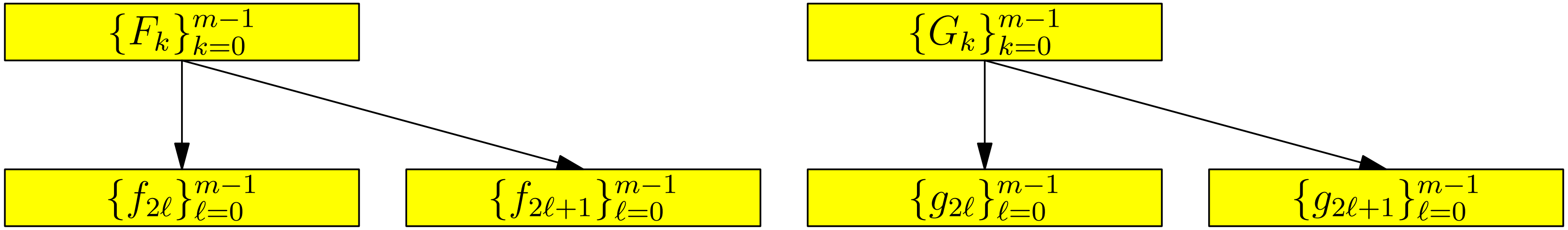
- This in-place convolution was written to use six out-of-place transforms, thereby avoiding bit reversal at all levels.

- The computational complexity is $6Km\log_2 m$.
- The numerical error is similar to explicit padding and the memory usage is identical.

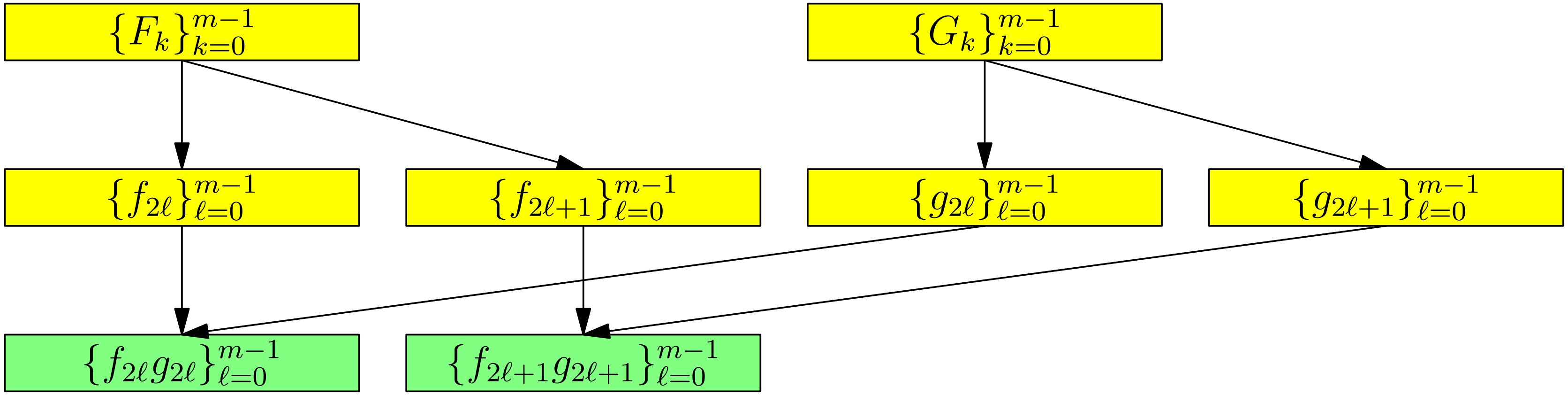
$\{F_k\}_{k=0}^{m-1}$

$\{G_k\}_{k=0}^{m-1}$

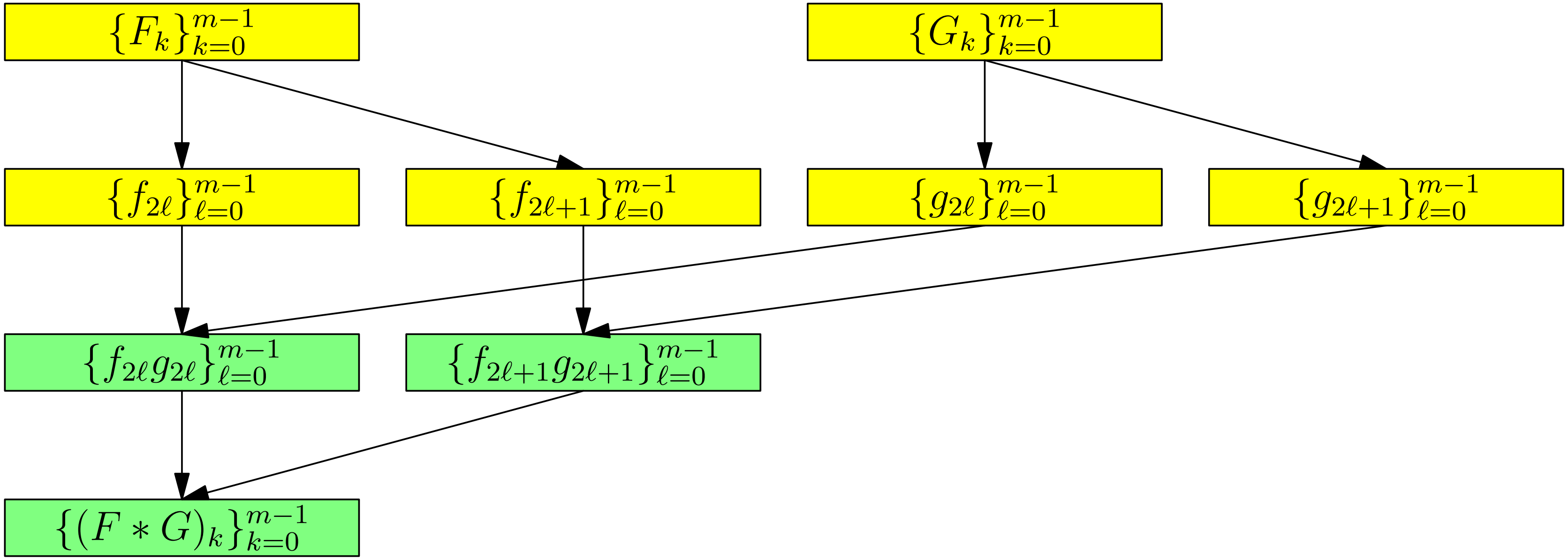
- The computational complexity is $6Km\log_2 m$.
- The numerical error is similar to explicit padding and the memory usage is identical.


$\{F_k\}_{k=0}^{m-1}$
$\{G_k\}_{k=0}^{m-1}$
$\{f_{2\ell}\}_{\ell=0}^{m-1}$
$\{f_{2\ell+1}\}_{\ell=0}^{m-1}$
$\{g_{2\ell}\}_{\ell=0}^{m-1}$
$\{g_{2\ell+1}\}_{\ell=0}^{m-1}$

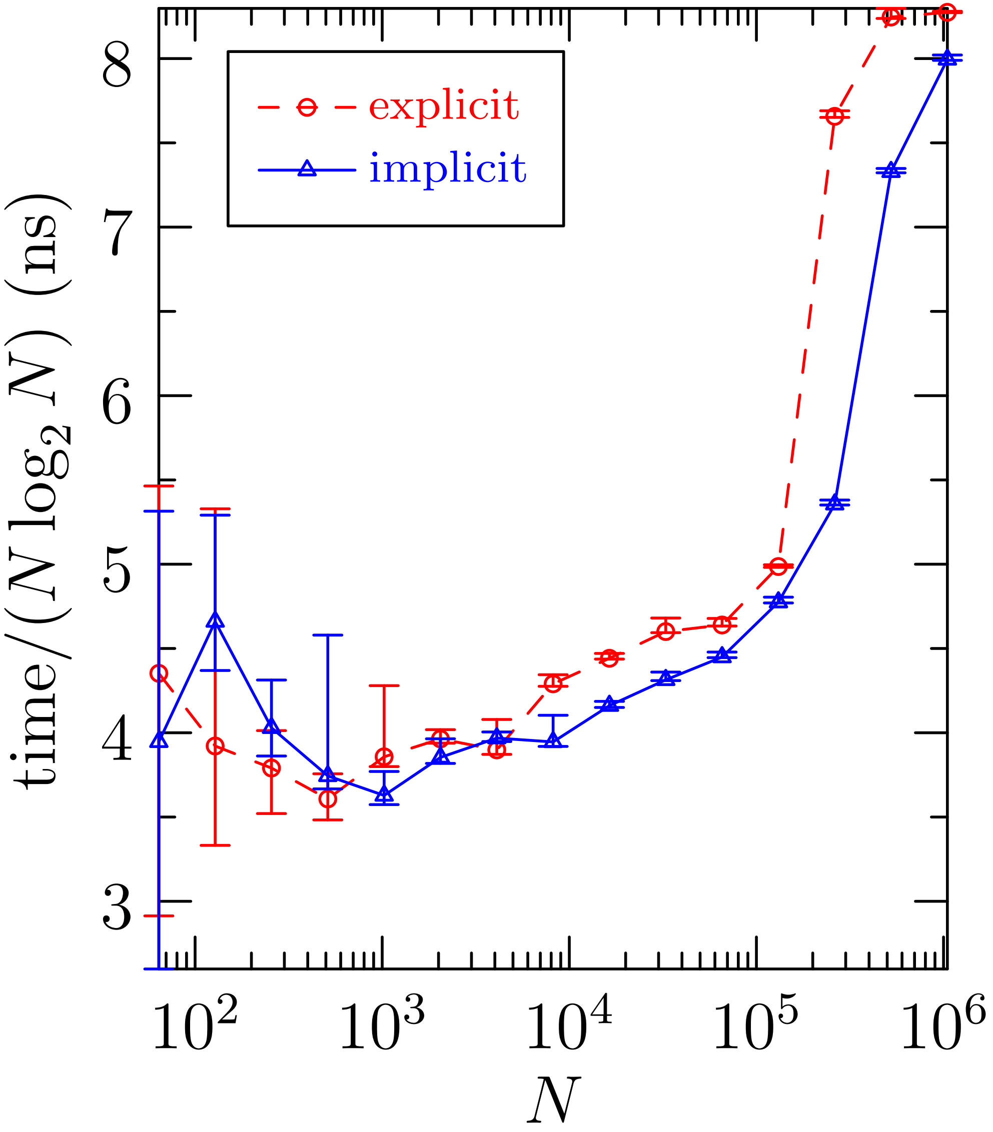
- The computational complexity is $6Km\log_2 m$.
- The numerical error is similar to explicit padding and the memory usage is identical.


$\{F_k\}_{k=0}^{m-1}$
$\{G_k\}_{k=0}^{m-1}$
$\{f_{2\ell}\}_{\ell=0}^{m-1}$
$\{f_{2\ell+1}\}_{\ell=0}^{m-1}$
$\{g_{2\ell}\}_{\ell=0}^{m-1}$
$\{g_{2\ell+1}\}_{\ell=0}^{m-1}$
$\{f_{2\ell}g_{2\ell}\}_{\ell=0}^{m-1}$
$\{f_{2\ell+1}g_{2\ell+1}\}_{\ell=0}^{m-1}$

- The computational complexity is $6Km\log_2 m$.
- The numerical error is similar to explicit padding and the memory usage is identical.


$\{F_k\}_{k=0}^{m-1}$
$\{G_k\}_{k=0}^{m-1}$
$\{f_{2\ell}\}_{\ell=0}^{m-1}$
$\{f_{2\ell+1}\}_{\ell=0}^{m-1}$
$\{g_{2\ell}\}_{\ell=0}^{m-1}$
$\{g_{2\ell+1}\}_{\ell=0}^{m-1}$
$\{f_{2\ell}g_{2\ell}\}_{\ell=0}^{m-1}$
$\{f_{2\ell+1}g_{2\ell+1}\}_{\ell=0}^{m-1}$
$\{(F*G)_k\}_{k=0}^{m-1}$

Implicit Padding in 1D

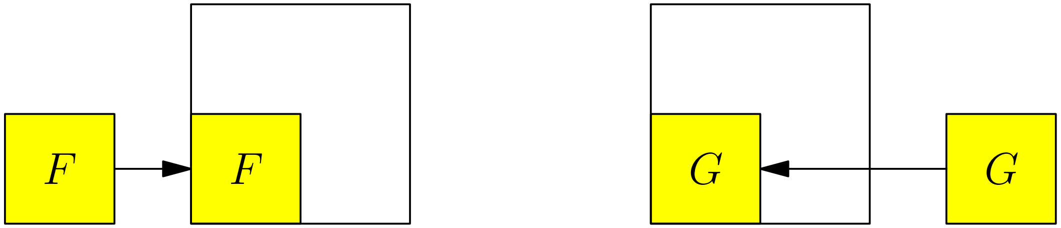


Convolutions in Higher Dimensions

- An explicitly padded convolution in 2 dimensions requires 12 padded FFTs, and 4 times the memory of a cyclic convolution.

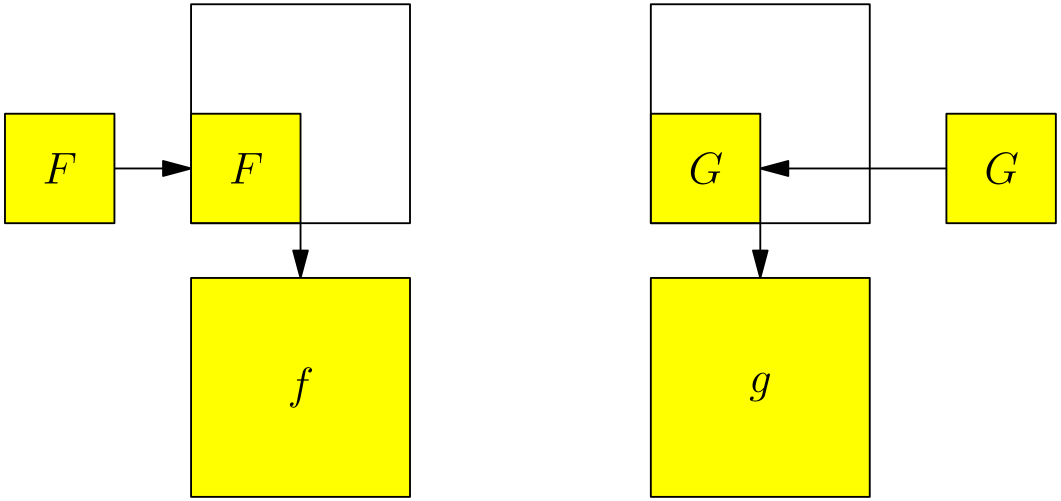
Convolutions in Higher Dimensions

- An explicitly padded convolution in 2 dimensions requires 12 padded FFTs, and 4 times the memory of a cyclic convolution.



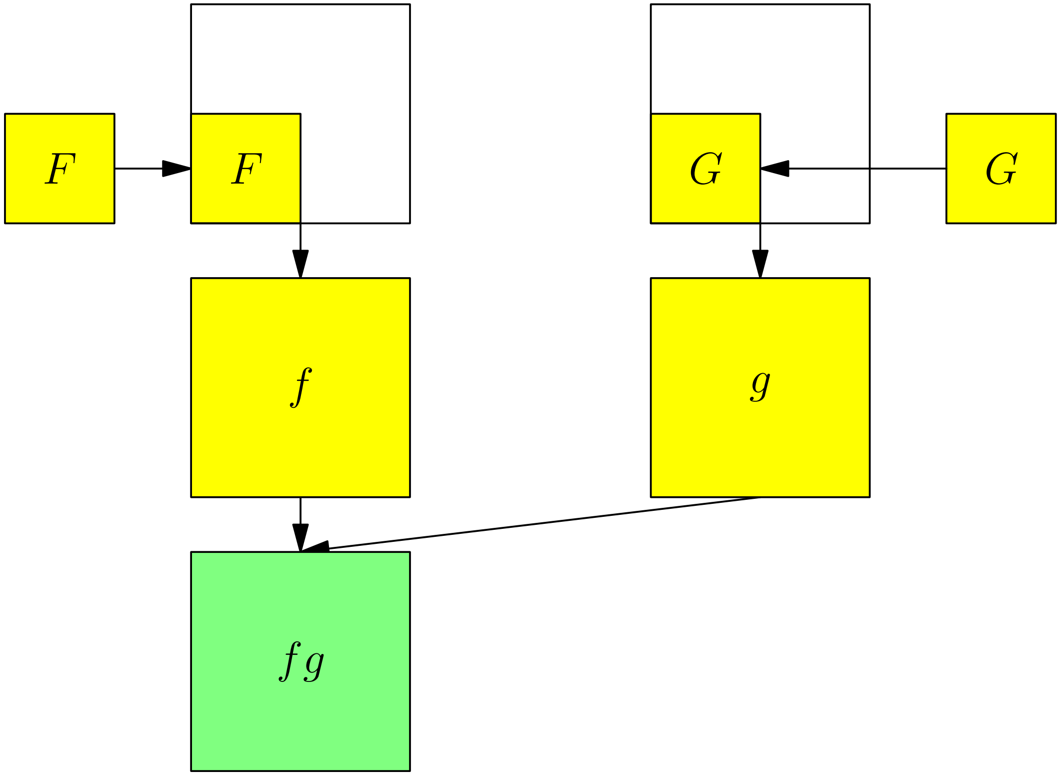
Convolutions in Higher Dimensions

- An explicitly padded convolution in 2 dimensions requires 12 padded FFTs, and 4 times the memory of a cyclic convolution.



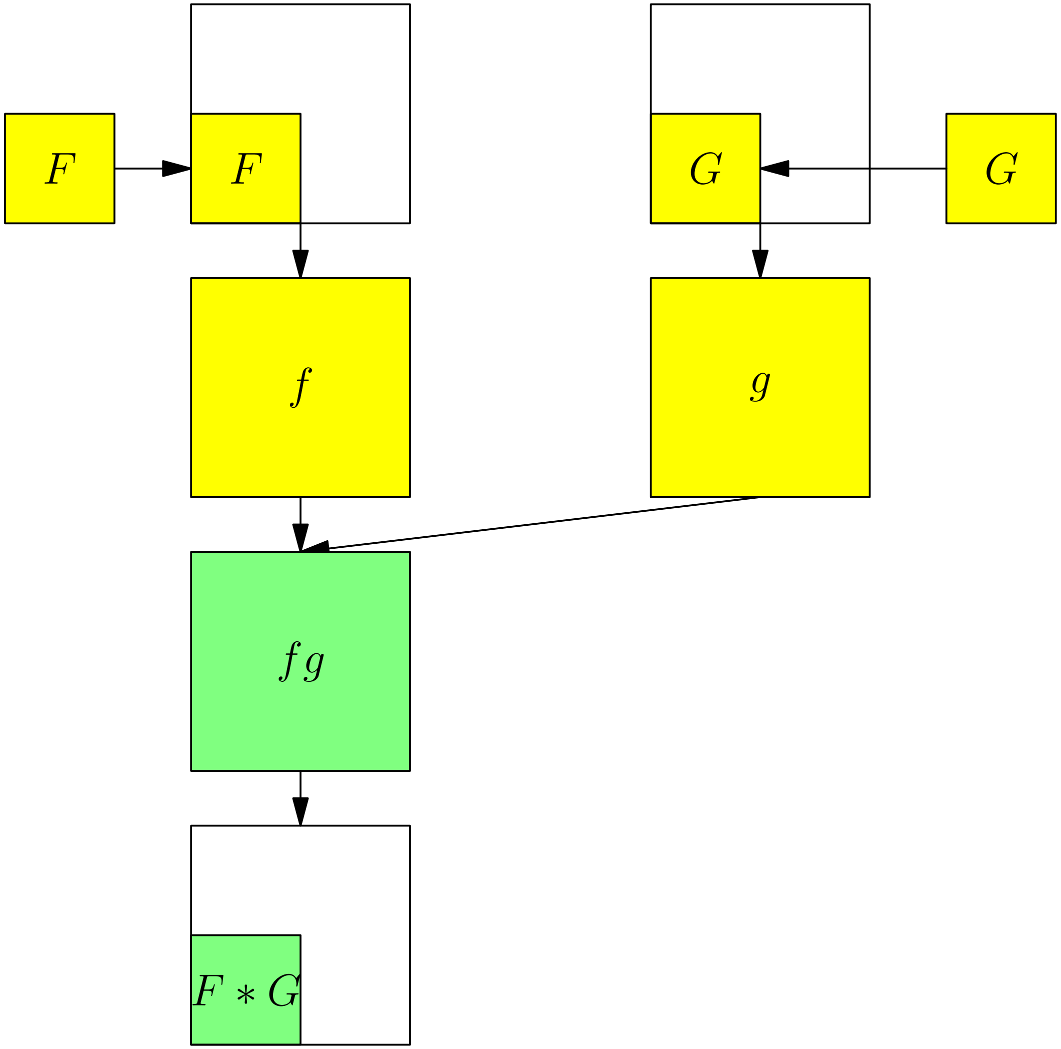
Convolutions in Higher Dimensions

- An explicitly padded convolution in 2 dimensions requires 12 padded FFTs, and 4 times the memory of a cyclic convolution.



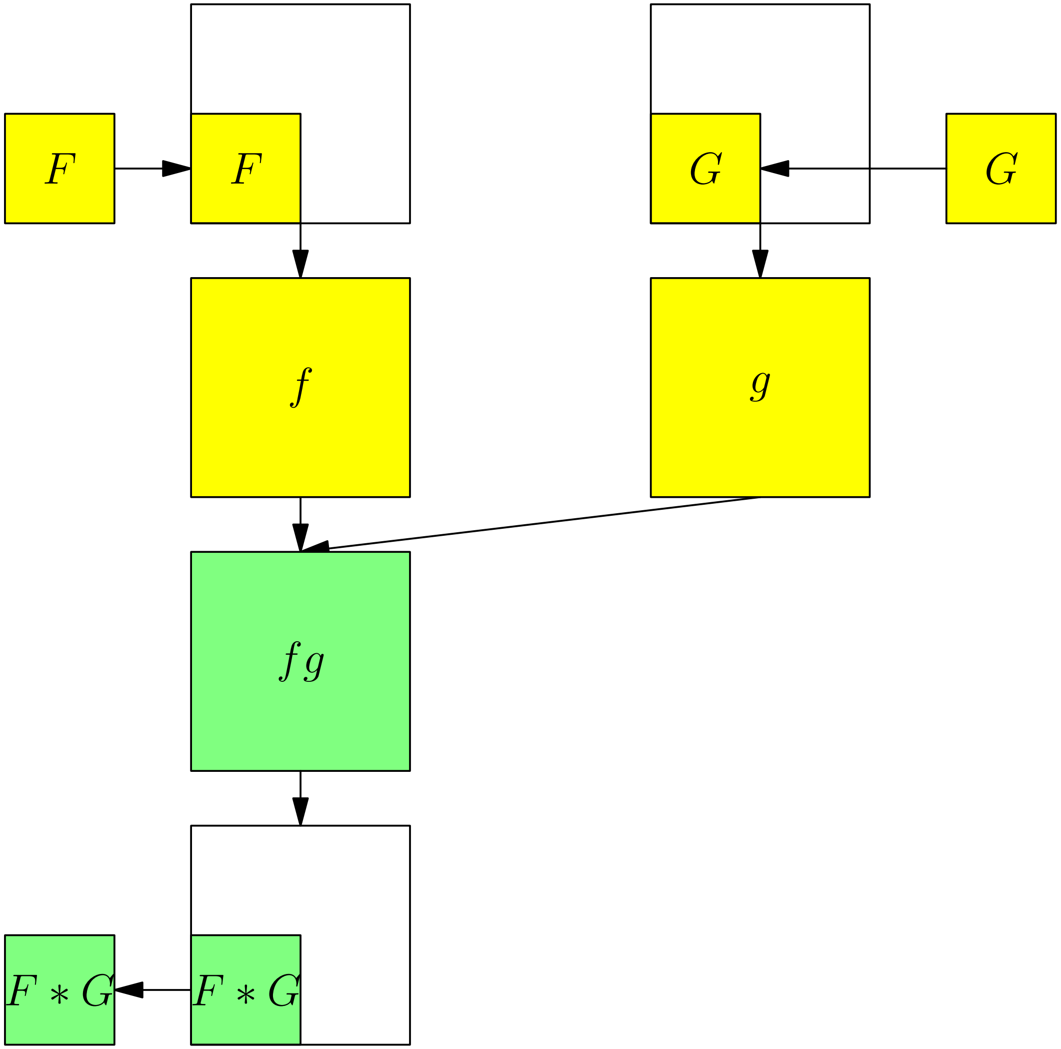
Convolutions in Higher Dimensions

- An explicitly padded convolution in 2 dimensions requires 12 padded FFTs, and 4 times the memory of a cyclic convolution.



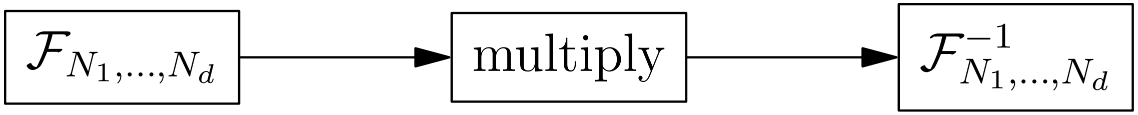
Convolutions in Higher Dimensions

- An explicitly padded convolution in 2 dimensions requires 12 padded FFTs, and 4 times the memory of a cyclic convolution.



Recursive Convolution

- Naive way to compute a multiple-dimensional convolution:

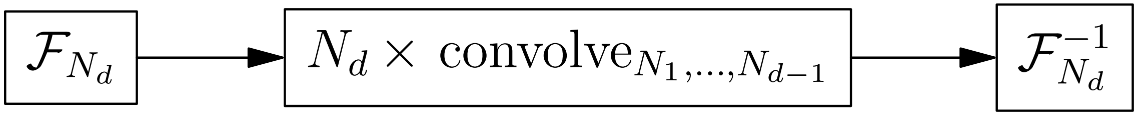


$\mathcal{F}_{N_1,\ldots,N_d}$

multiply

$\mathcal{F}^{-1}_{N_1,\ldots,N_d}$

- *Recursive convolution* allows one to avoid computing and storing the entire Fourier image of the data:



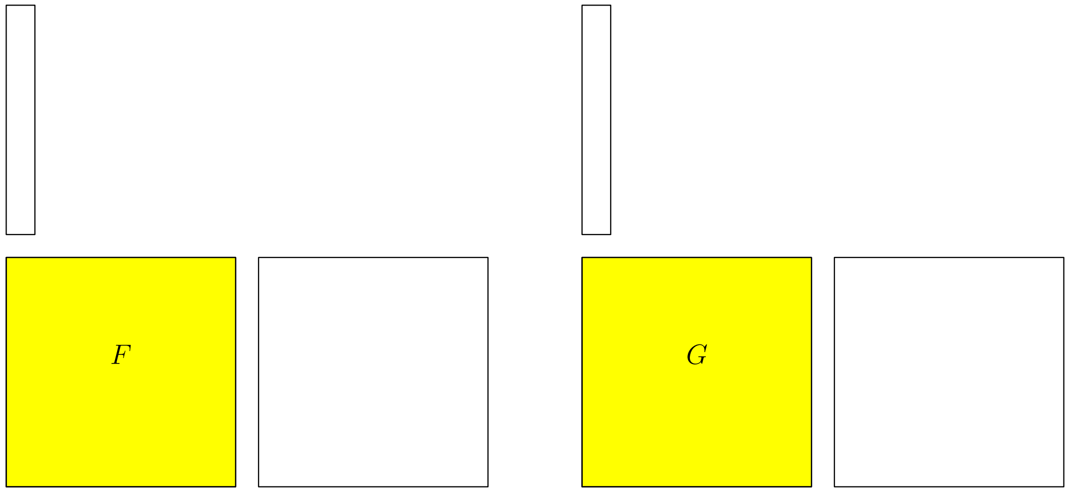
$\mathcal{F}_{N_d}$

$N_d \times$ convolve$_{N_1,\ldots,N_{d-1}}$

$\mathcal{F}^{-1}_{N_d}$

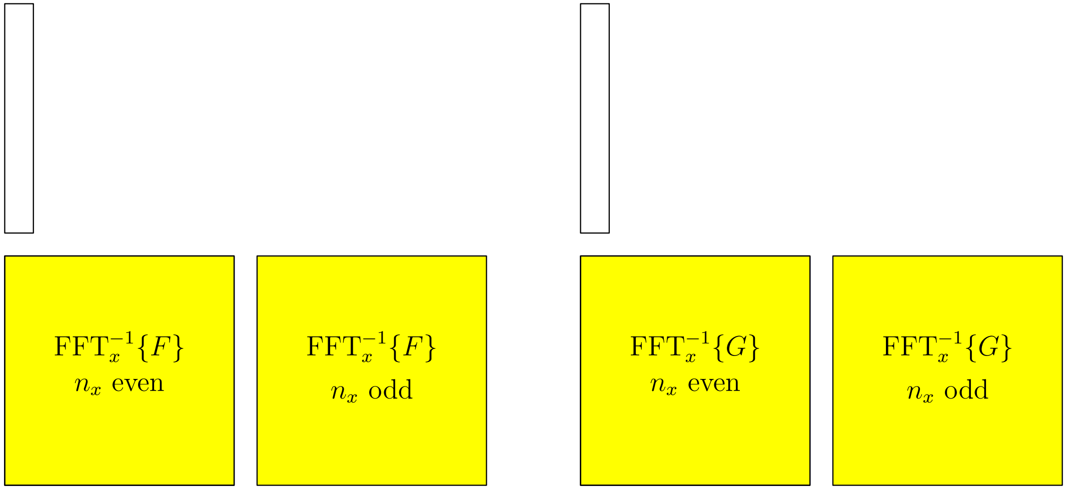
Implicit Padding in 2D

- Extra work memory need not be contiguous with the data.



Implicit Padding in 2D

- Extra work memory need not be contiguous with the data.



$\mathrm{FFT}_x^{-1}\{F\}$ n_x even

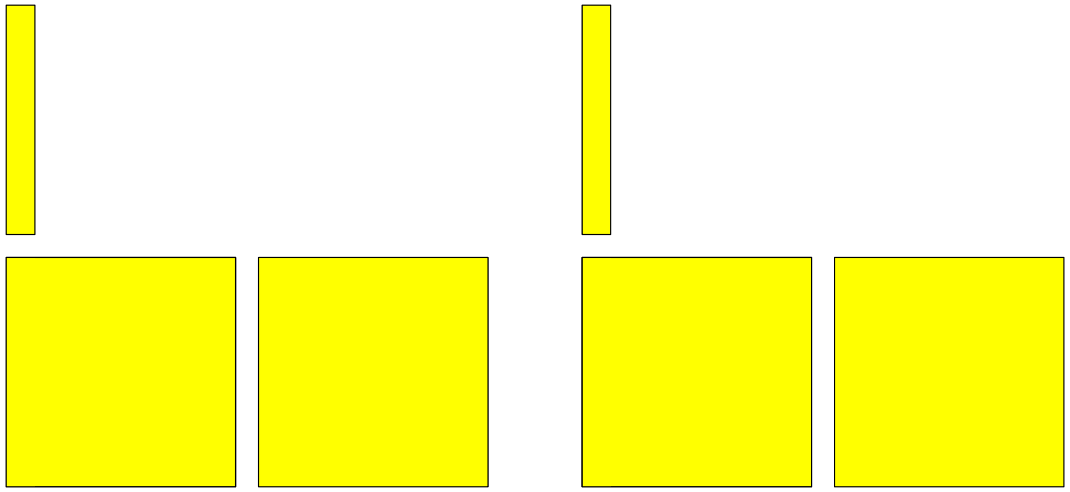
$\mathrm{FFT}_x^{-1}\{F\}$ n_x odd

$\mathrm{FFT}_x^{-1}\{G\}$ n_x even

$\mathrm{FFT}_x^{-1}\{G\}$ n_x odd

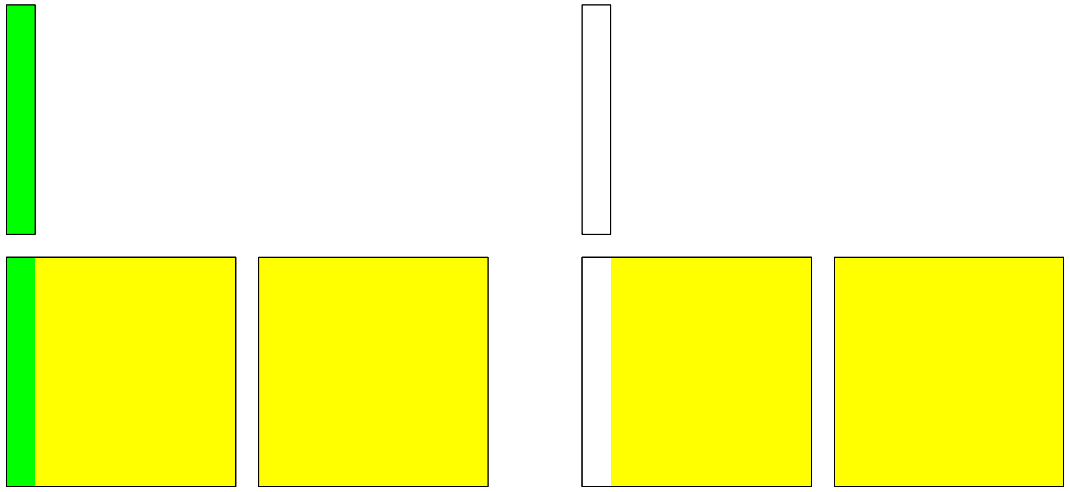
Implicit Padding in 2D

- Extra work memory need not be contiguous with the data.



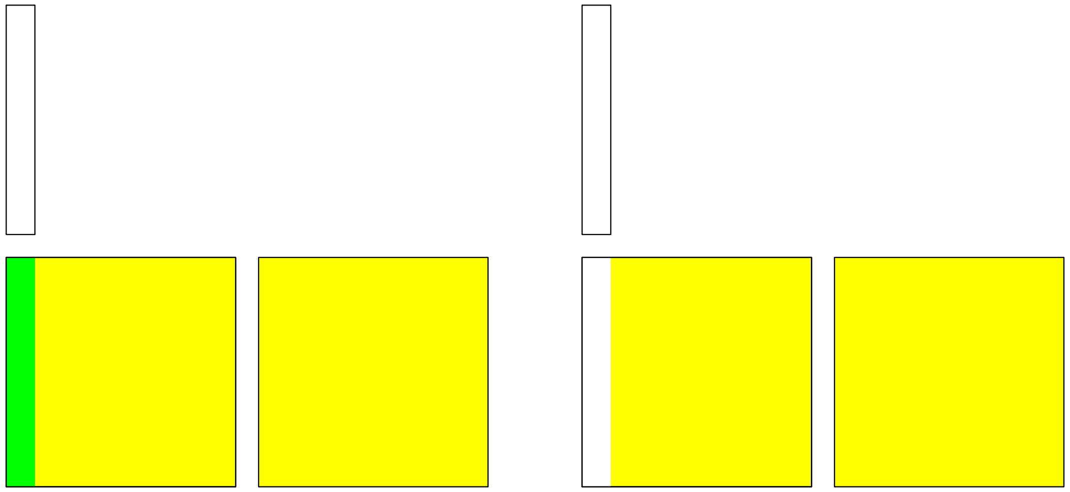
Implicit Padding in 2D

- Extra work memory need not be contiguous with the data.



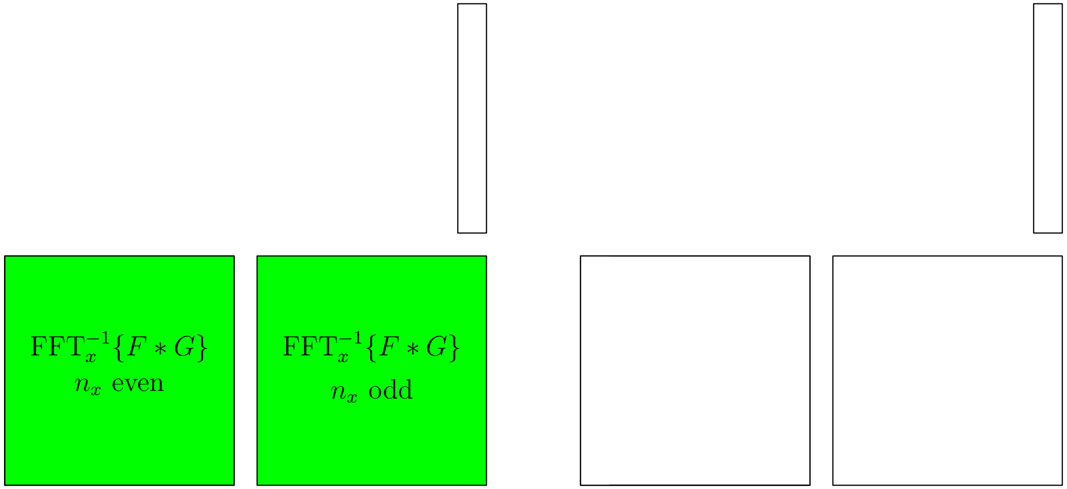
Implicit Padding in 2D

- Extra work memory need not be contiguous with the data.



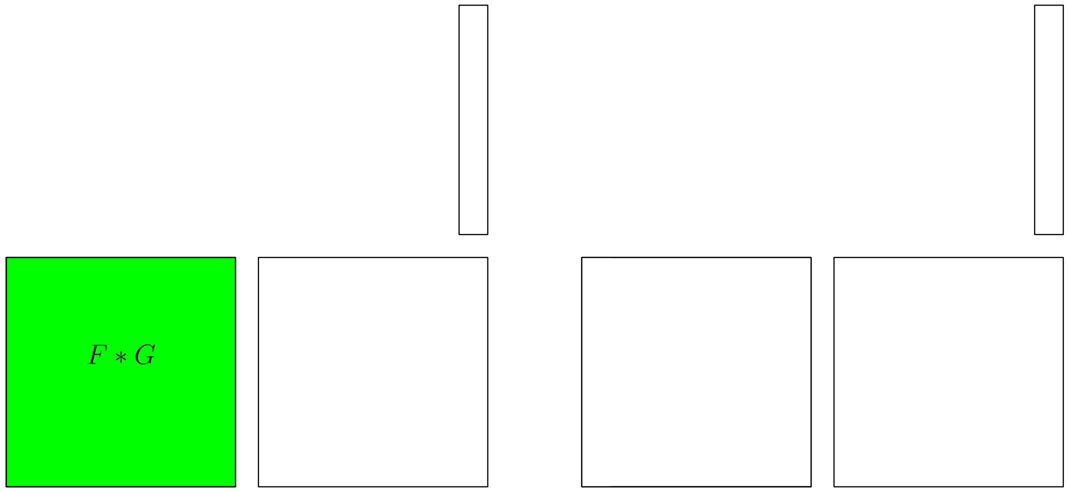
Implicit Padding in 2D

- Extra work memory need not be contiguous with the data.

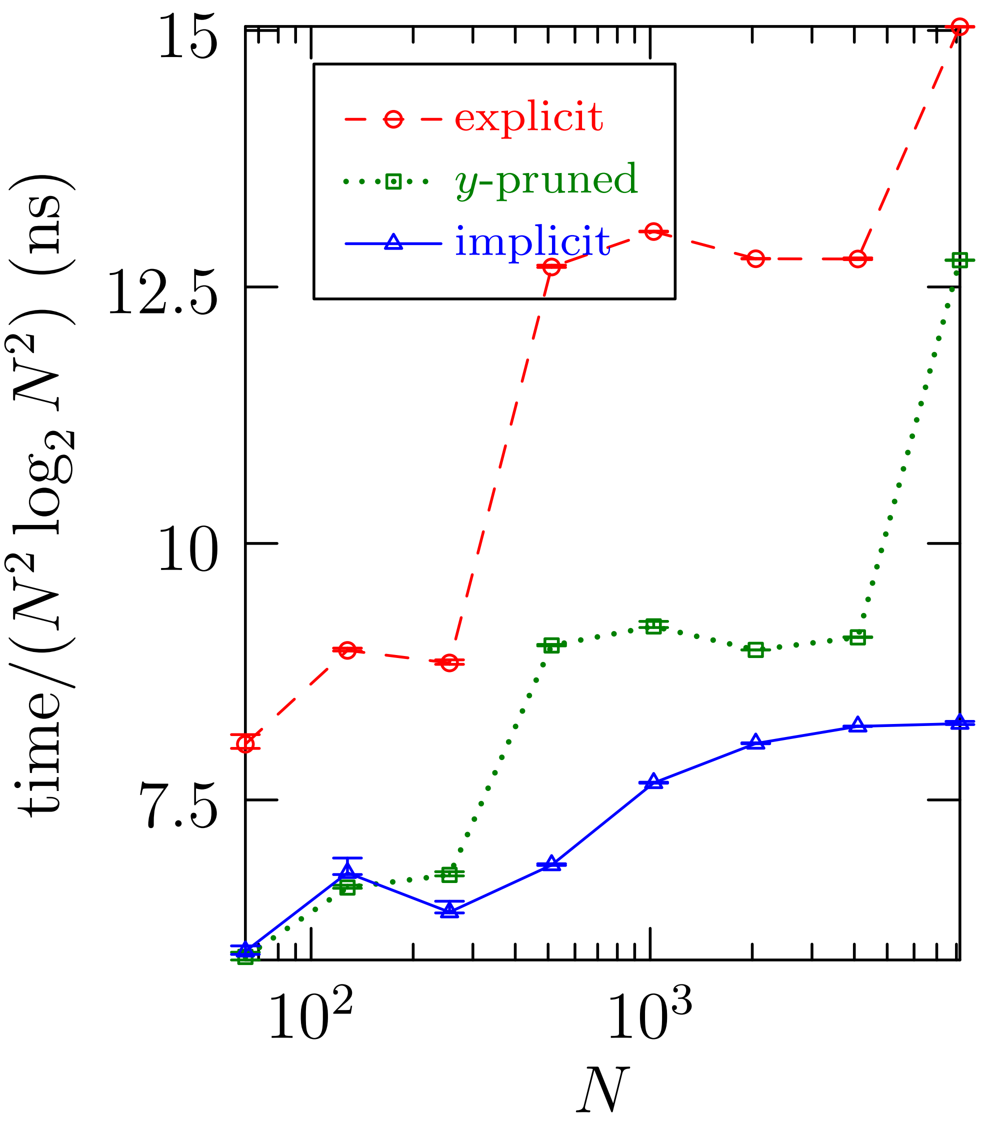

$\mathrm{FFT}_x^{-1}\{F * G\}$
n_x even
$\mathrm{FFT}_x^{-1}\{F * G\}$
n_x odd

Implicit Padding in 2D

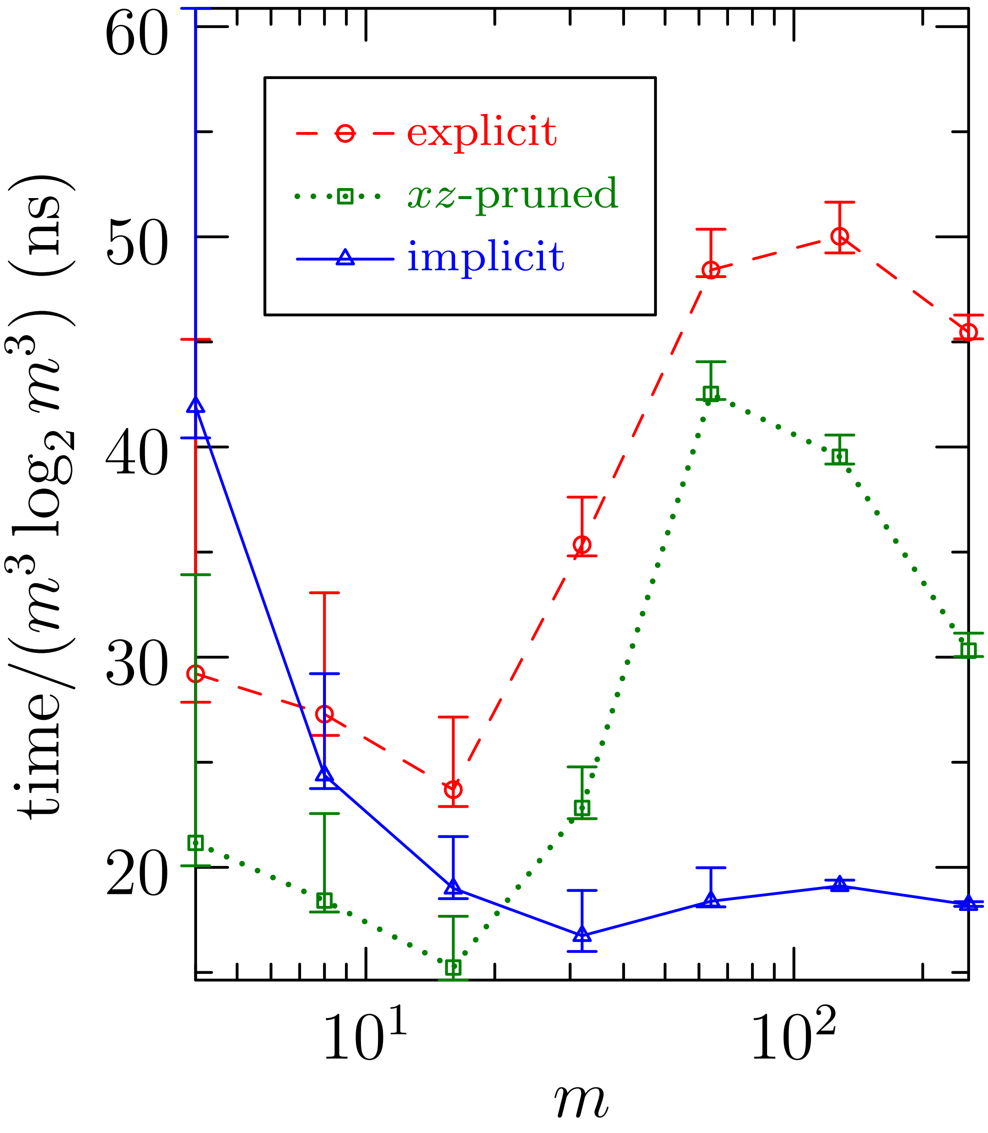
- Extra work memory need not be contiguous with the data.



Implicit Padding in 2D



Implicit Padding in 3D



Centered (Pseudospectral) Convolutions

- For a *centered convolution,* the Fourier origin ($k = 0$) is centered in the domain:

$$\sum_{p=k-m+1}^{m-1} f_p g_{k-p}$$

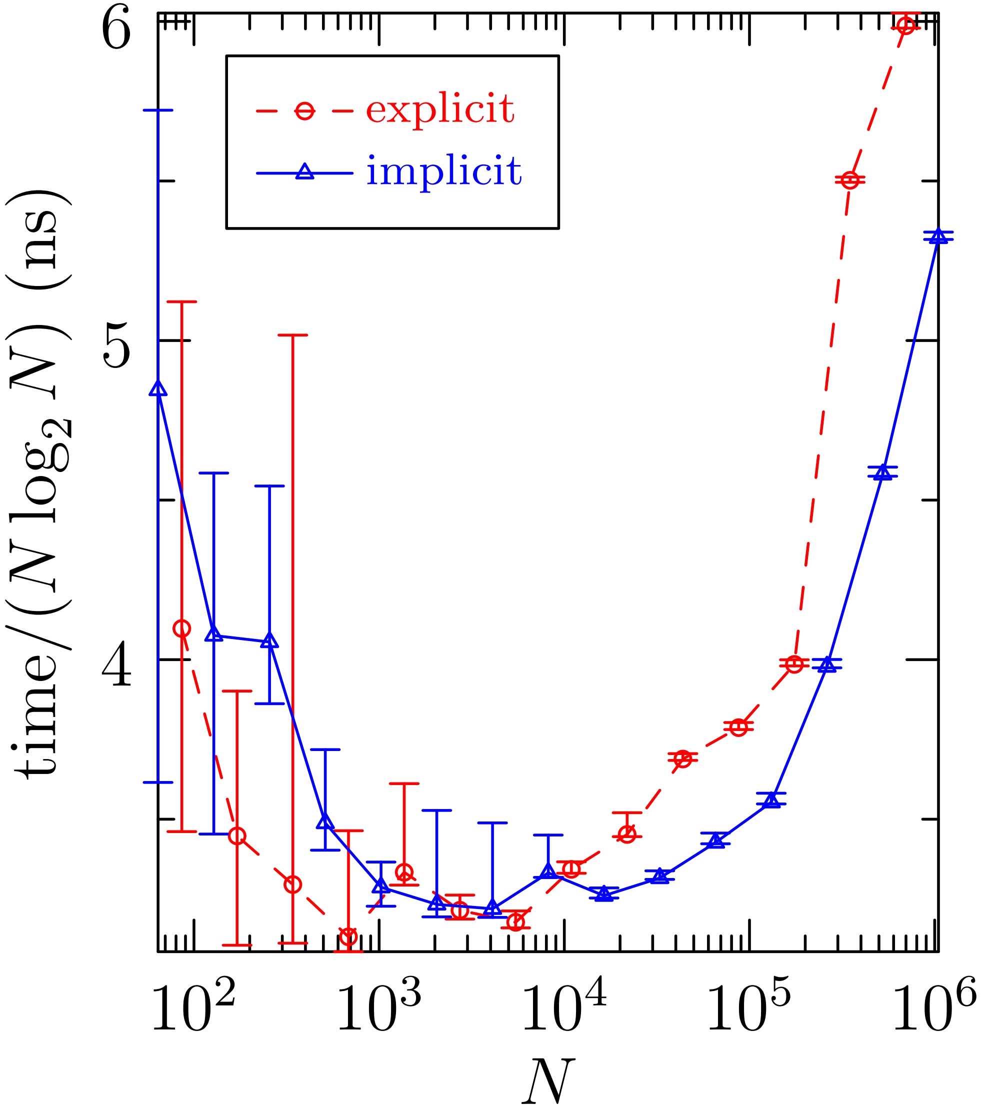
- Need to pad to $N \geq 3m - 2$ to remove aliases.

- The ratio $(2m-1)/(3m-2)$ of the number of physical to total modes is asymptotic to $2/3$ for large m.

- A *Hermitian convolution* arises since the input vectors are real:

$$f_{-k} = \overline{f_k}.$$

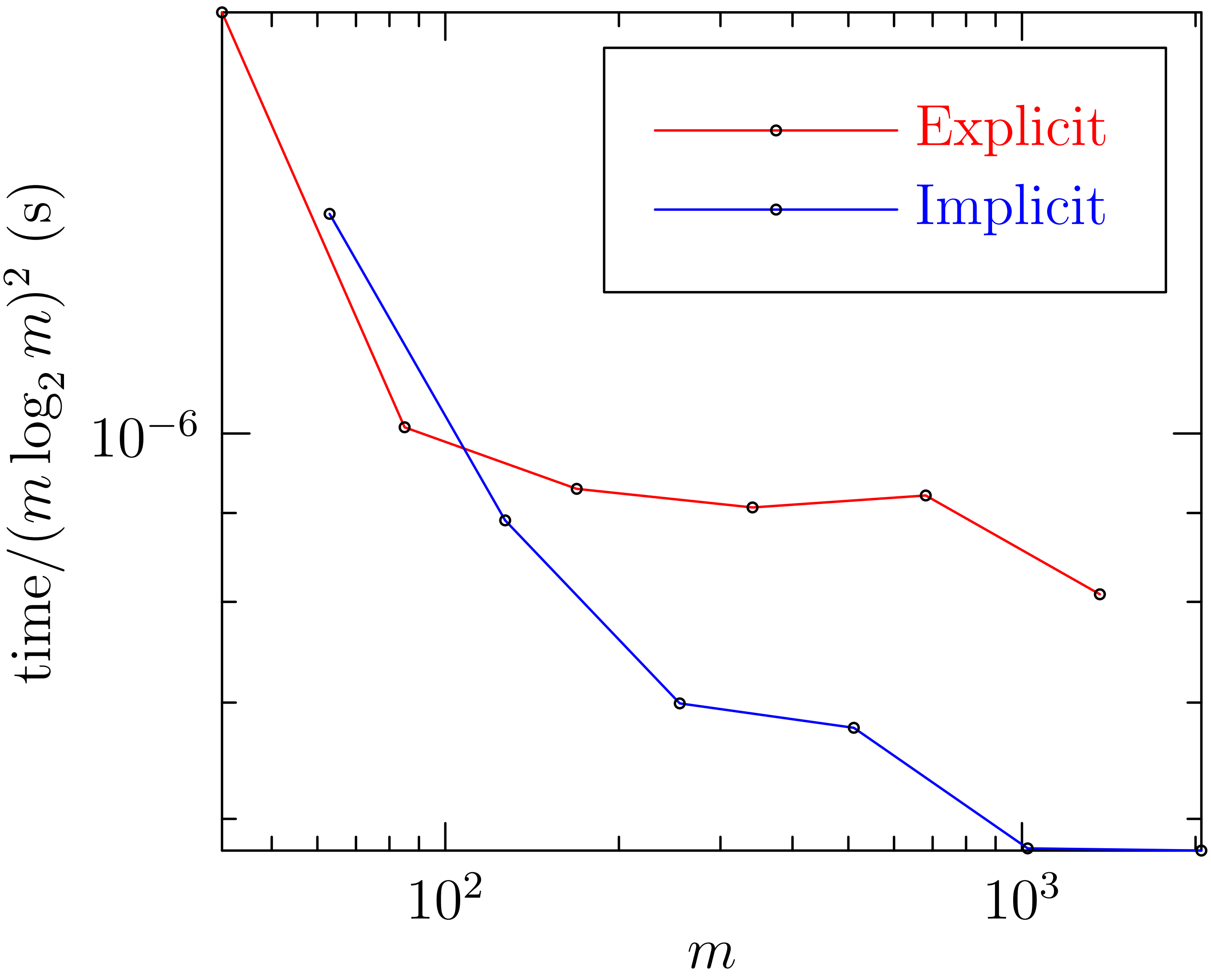
1D Implicit Hermitian Convolution



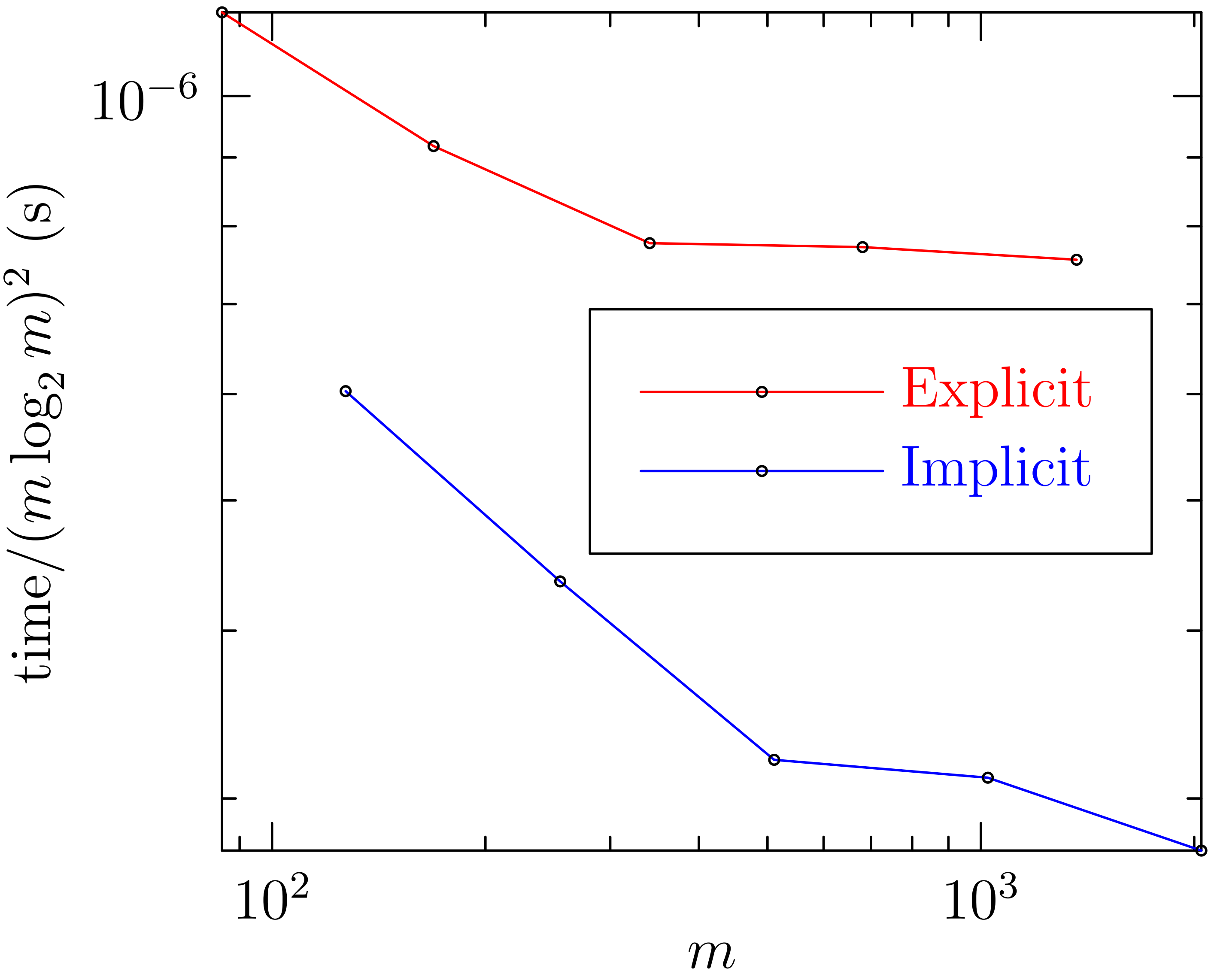
Shared-Memory Parallelization

- Our implicit and explicit convolution routines have been multithreaded for shared-memory architectures.

2D Navier–Stokes Pseudospectral [1 thread]



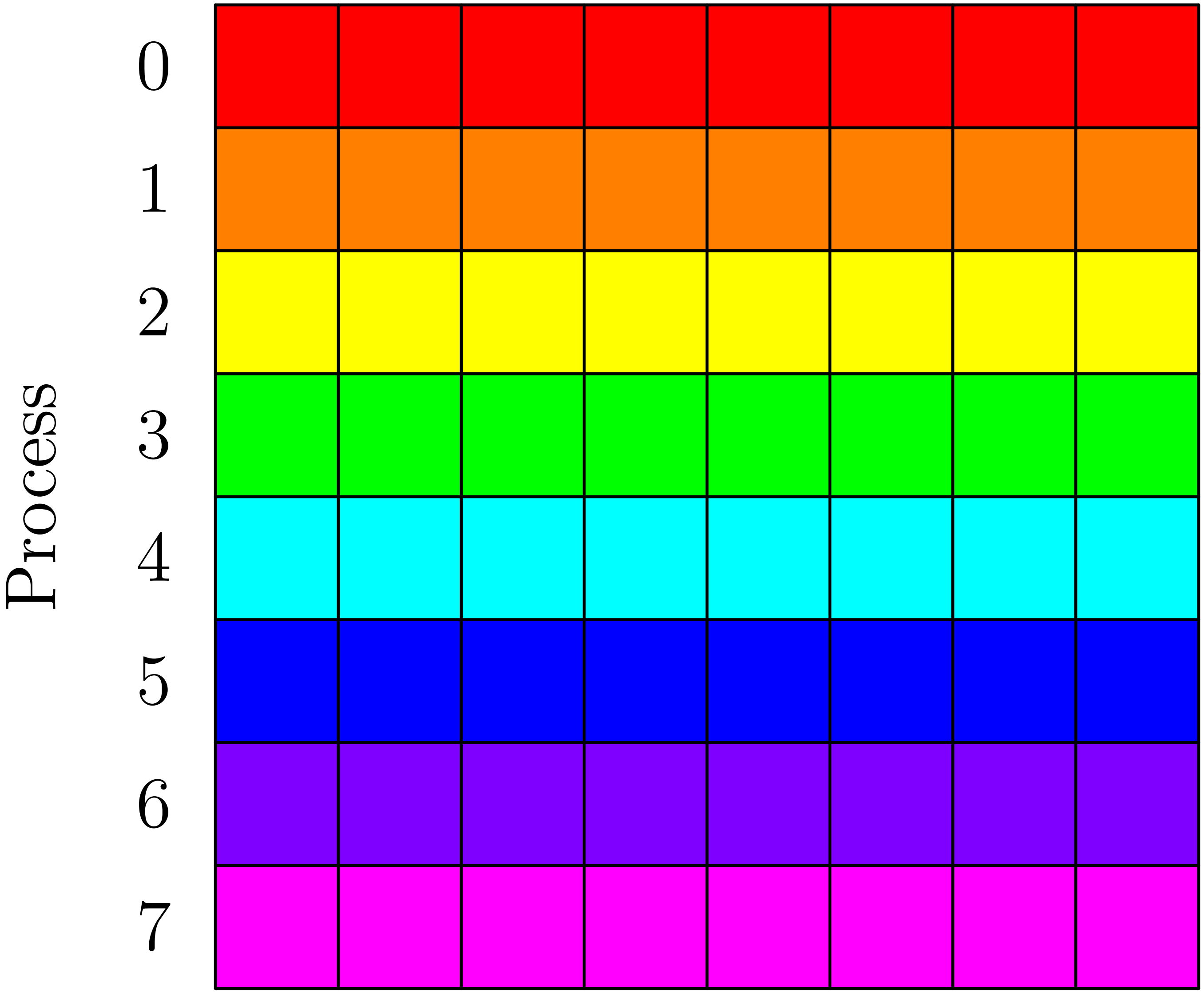
2D Navier–Stokes Pseudospectral [4 threads]



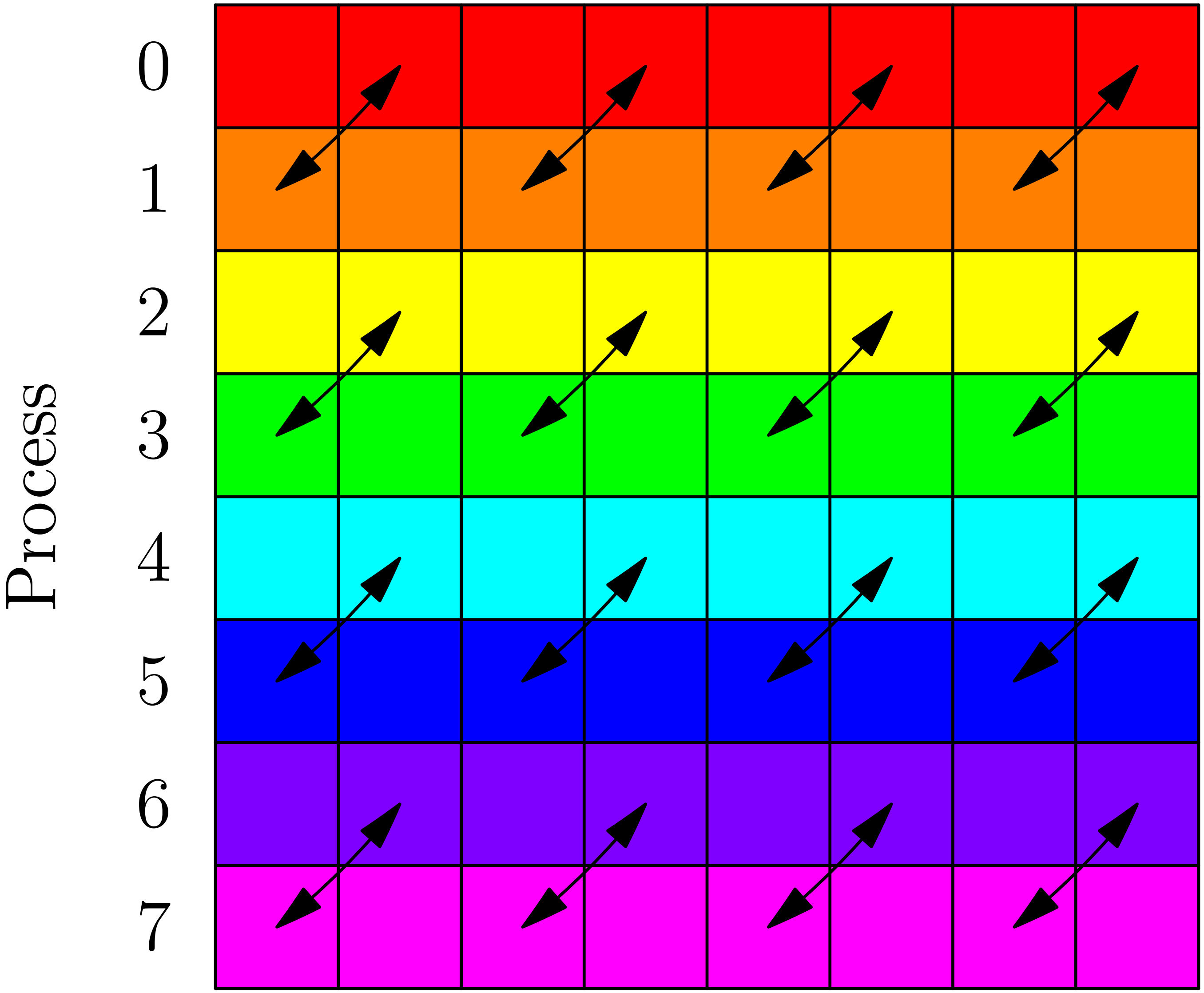
Distributed-Memory Parallelization

- The pseudospectral method uses a matrix transpose to localize the computation of the multi-dimensional FFTs onto individual processors.
- Parallel generalized slab/pencil decompositions have recently been developed for distributed-memory architectures.
- We compare several distributed matrix transpose algorithms, both blocking and nonblocking, under pure MPI and hybrid MPI/OpenMP architectures.
- Local transposition is not required within a single MPI node.
- We have developed an adaptive algorithm, dynamically tuned to choose the optimal block size.

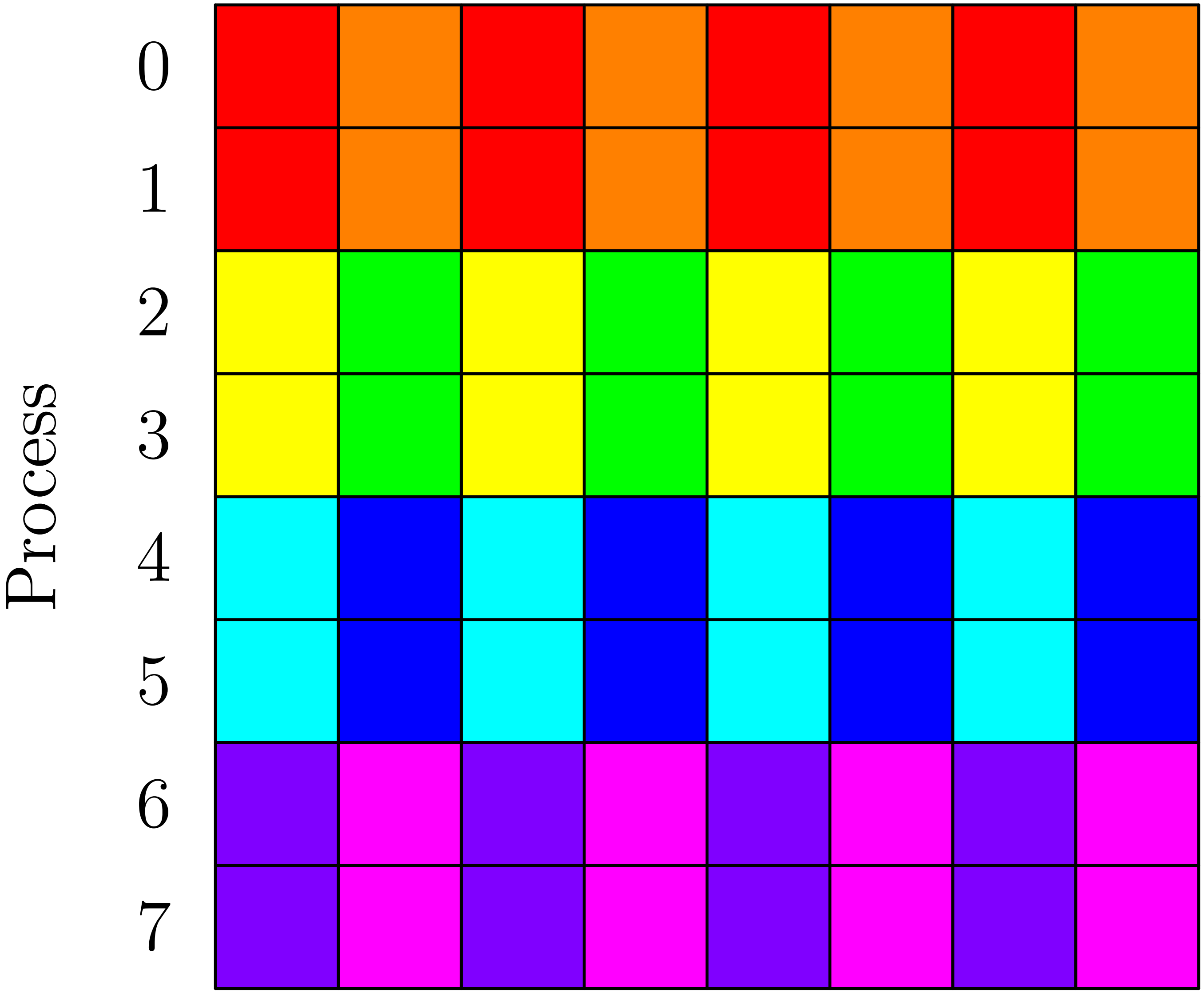
8×8 Block Transpose over 8 processors



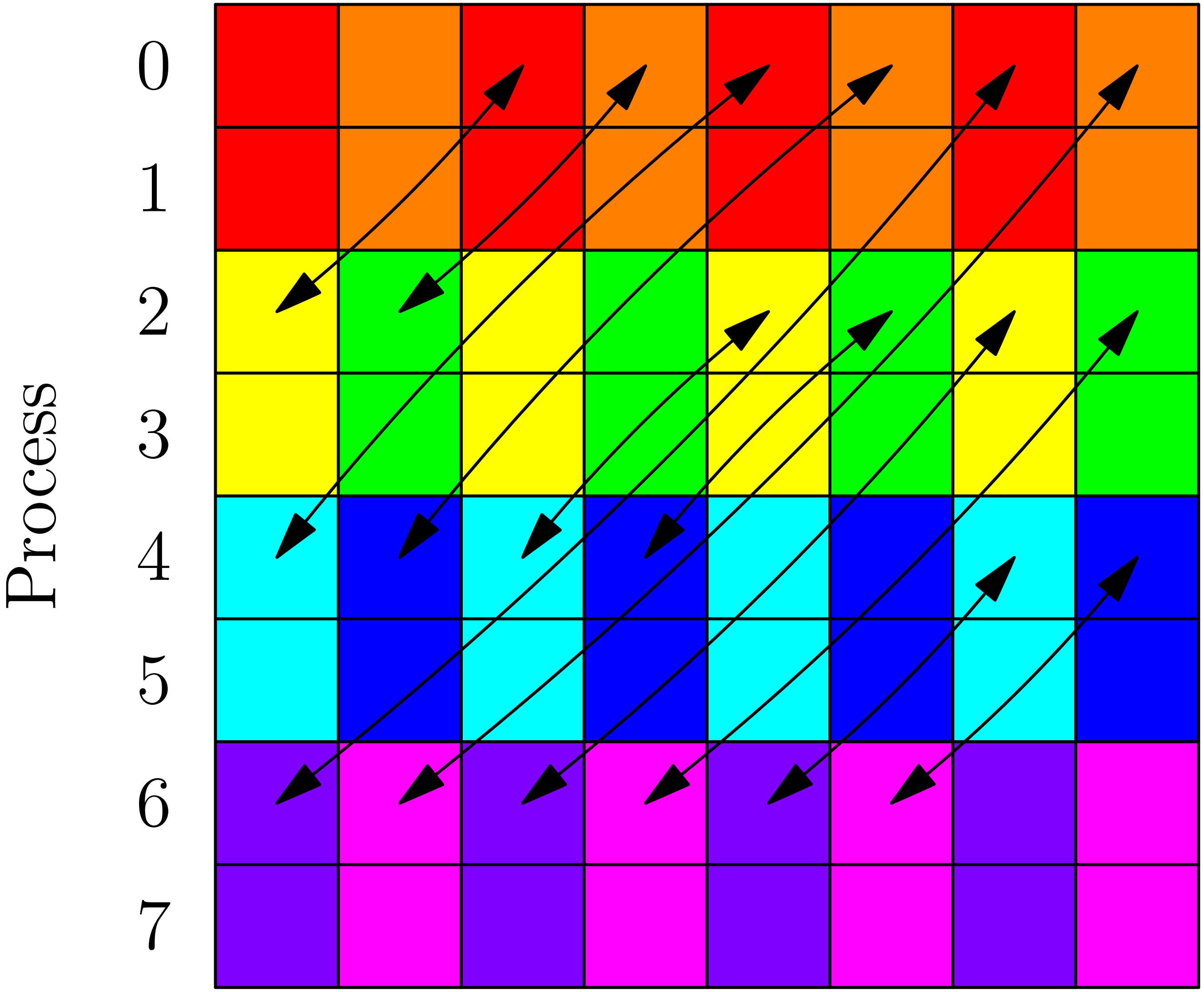
8×8 Block Transpose over 8 processors



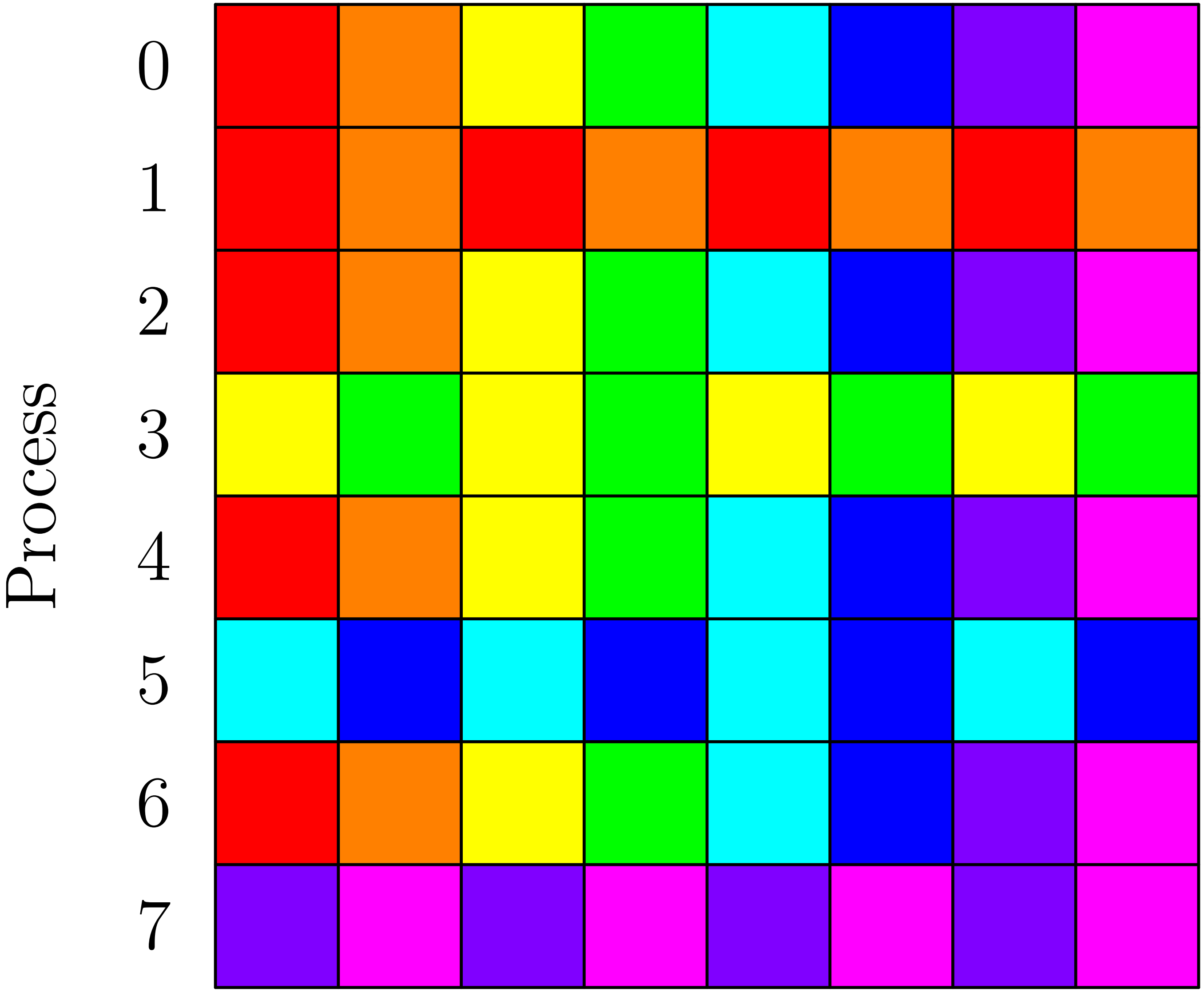
8×8 Block Transpose over 8 processors



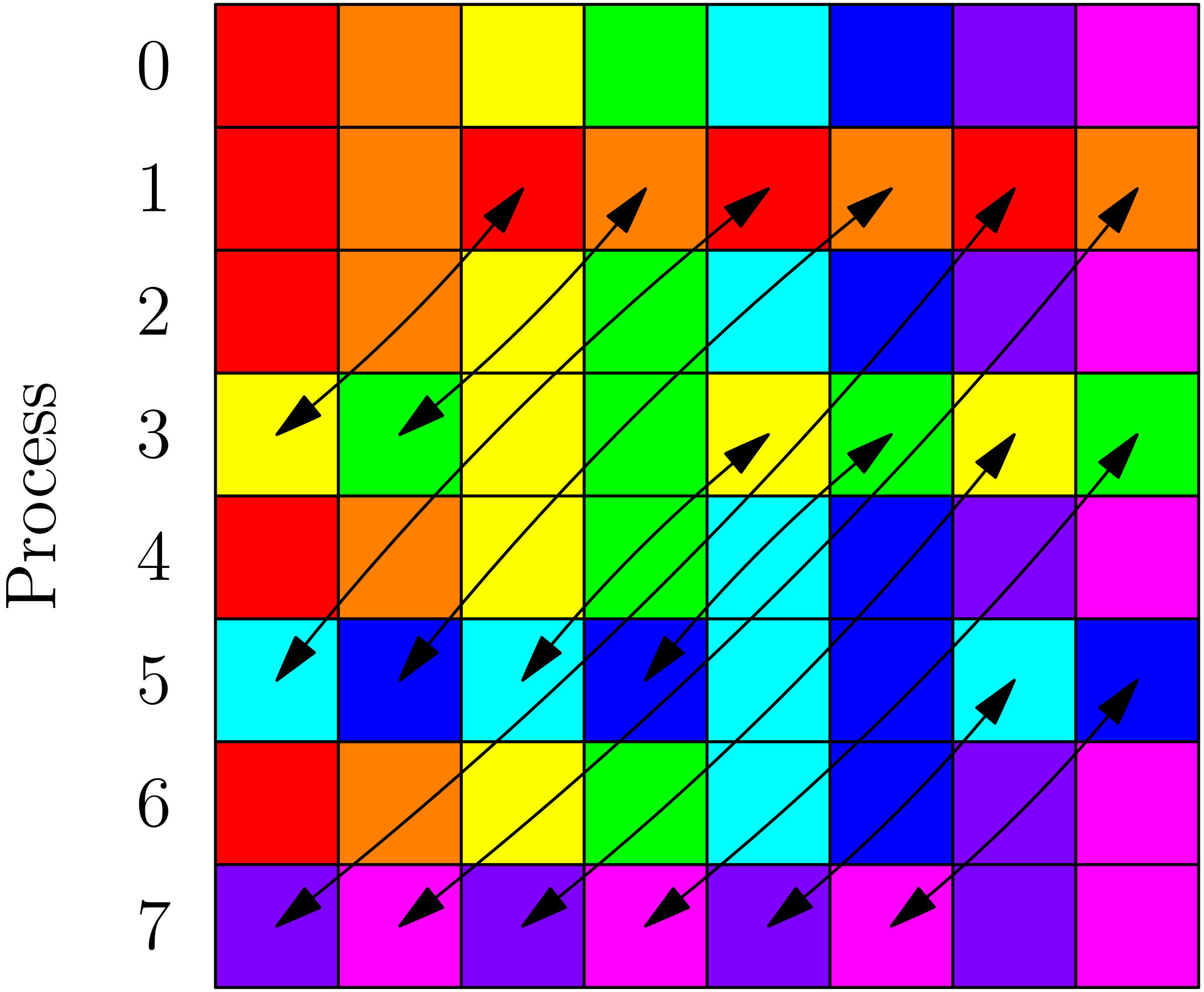
8 × 8 Block Transpose over 8 processors



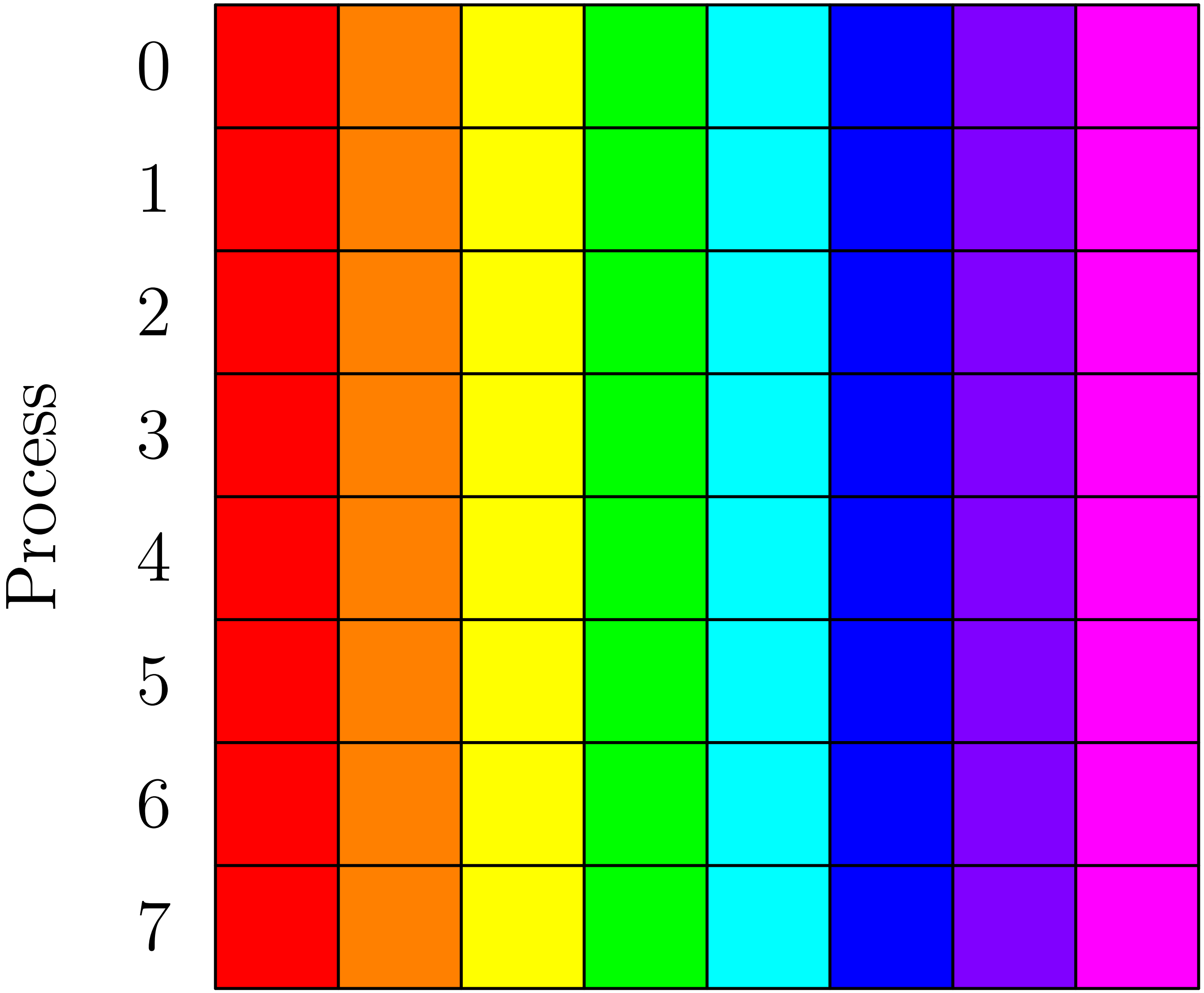
8 × 8 Block Transpose over 8 processors



8 × 8 Block Transpose over 8 processors



8 × 8 Block Transpose over 8 processors



Advantages of Hybrid MPI/OpenMP

- Use hybrid OpenMPI/MPI with the optimal number of threads:

  – yields larger communication block size;

  – local transposition is not required within a single MPI node;

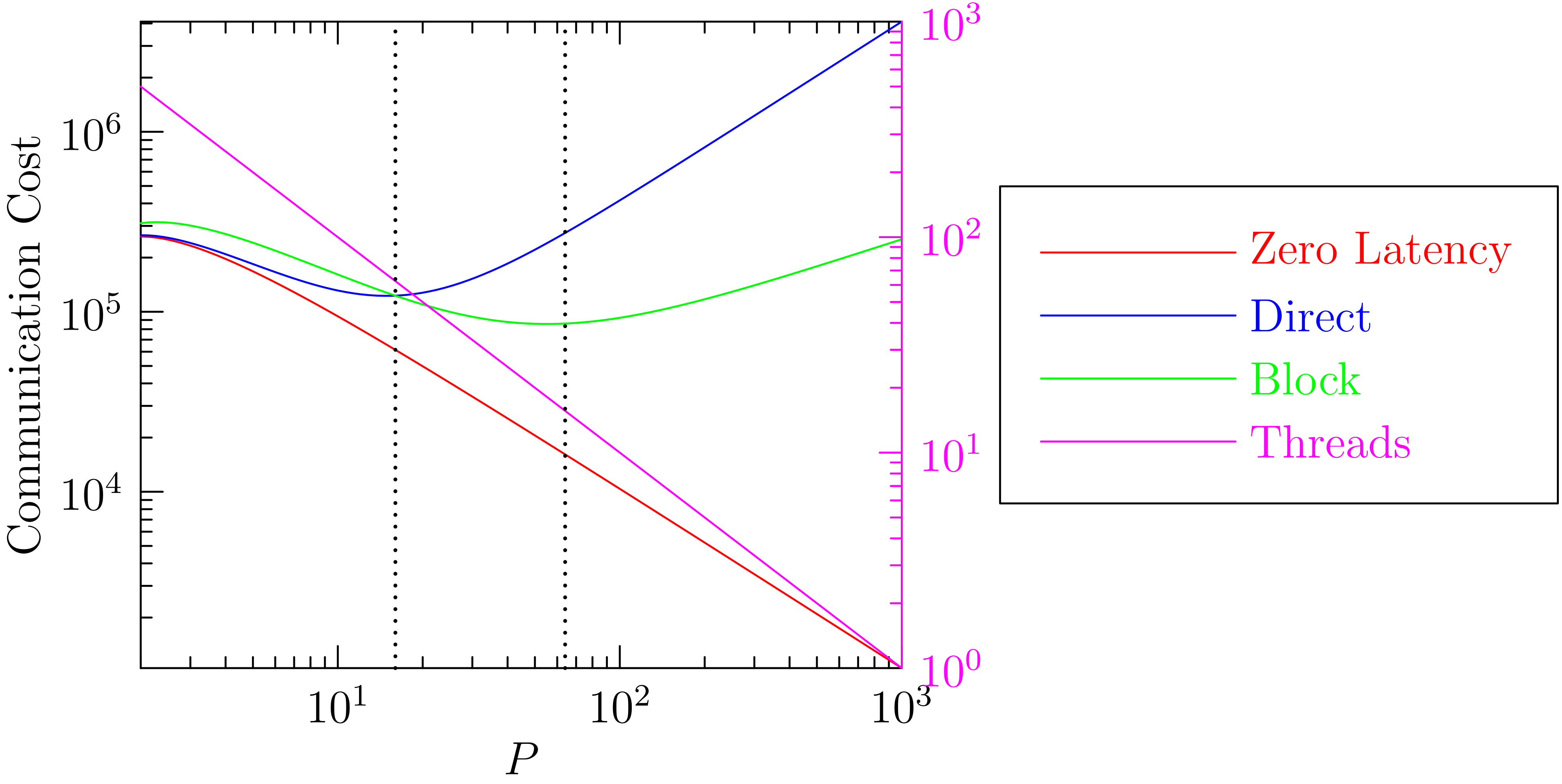
  – allows smaller problems to be distributed over a large number of processors;

  – for 3D FFTs, allows for more slab-like than pencil-like models, reducing the size of or even eliminating the need for a second transpose.

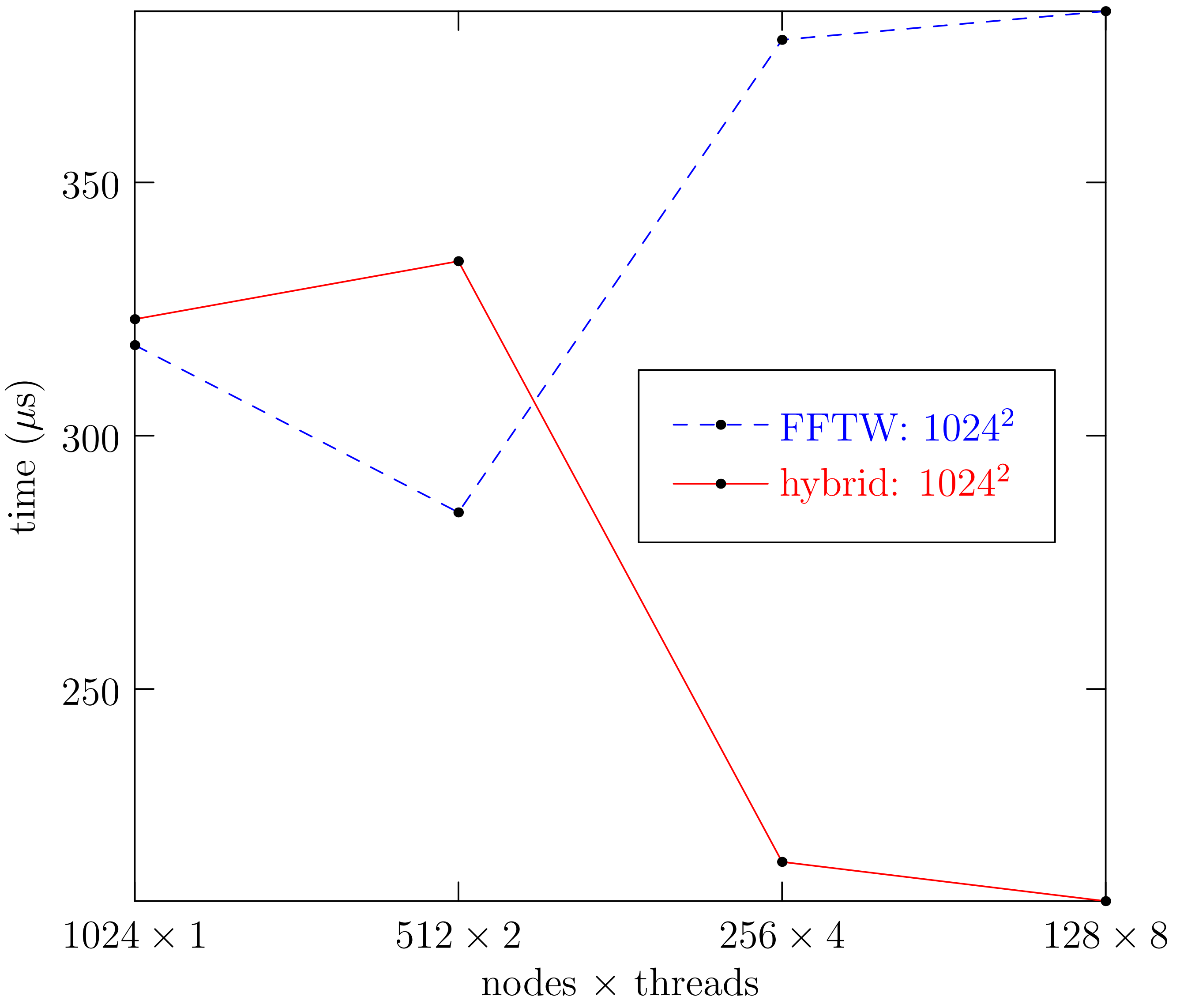
  – sometimes more efficient (by a factor of 2) than pure MPI.

- The use of nonblocking MPI communications allows us to overlap computation with communication: this can yield up to an additional 32% performance gain for implicitly dealiased convolutions, for which a natural parallelism exists between communication and computation.

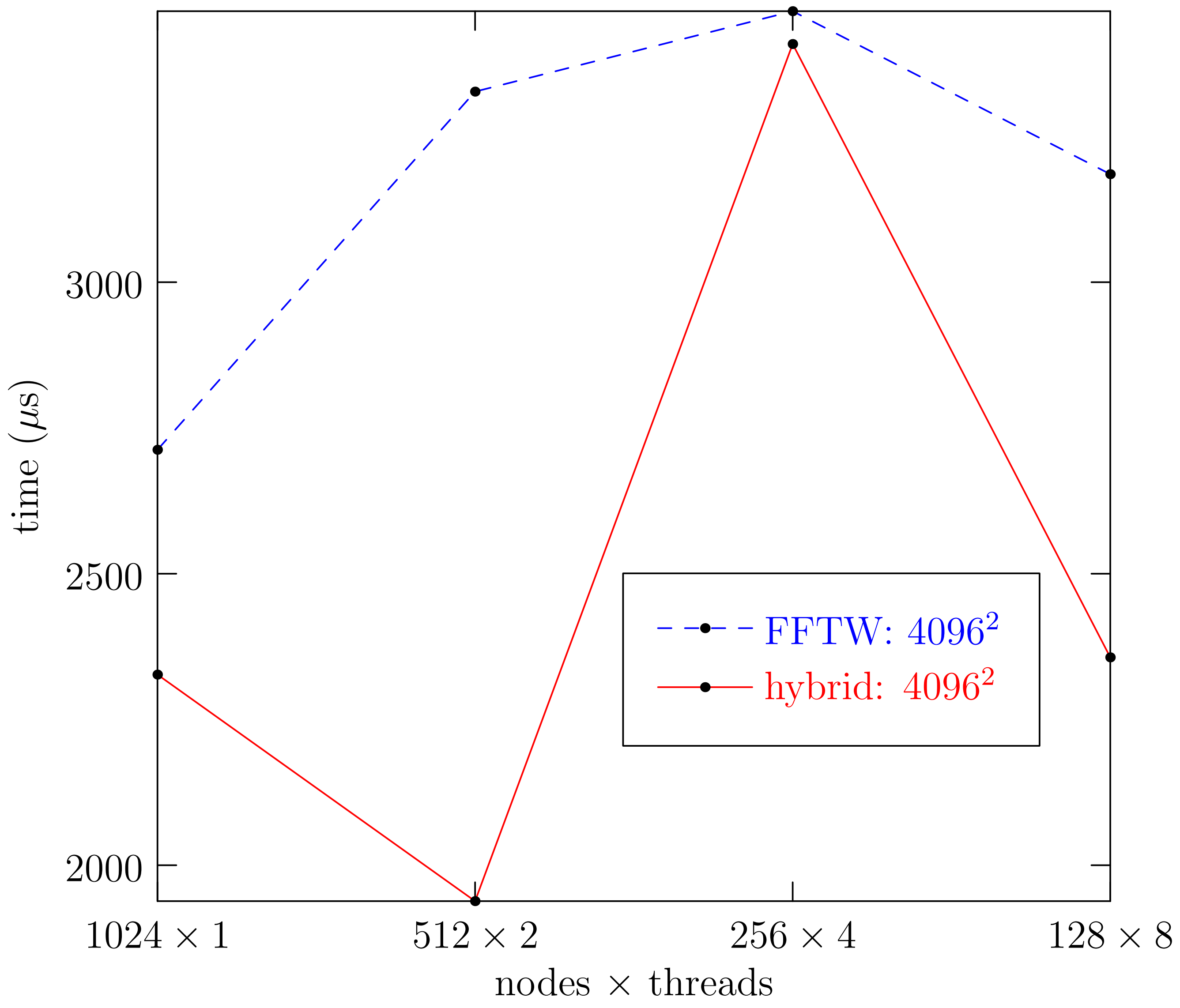
Transpose Communication Cost/Process



Optimal Number of Threads: 1024 × 1024



Optimal Number of Threads: 4096 × 4096



Windowed Time Averages

- Suppose we evolve a vorticity equation of the form

$$\frac{d\omega_{\boldsymbol{k}}}{dt} = S_{\boldsymbol{k}}.$$

- Once a statistically stationary state has been achieved, we may want to compute windowed time averages of moments like $|\omega_{\boldsymbol{k}}|^n$ and $S_{\boldsymbol{k}}\omega_{\boldsymbol{k}}^*$.

- But the saturation time is not normally known until after the simulation is completed!

Dynamic Moment Averaging

- Advantageous to precompute time-integrated moments like

$$M_n(t) = \int_0^t |\omega_{\boldsymbol{k}}(\tau)|^n \, d\tau.$$

- This can be accomplished done by evolving

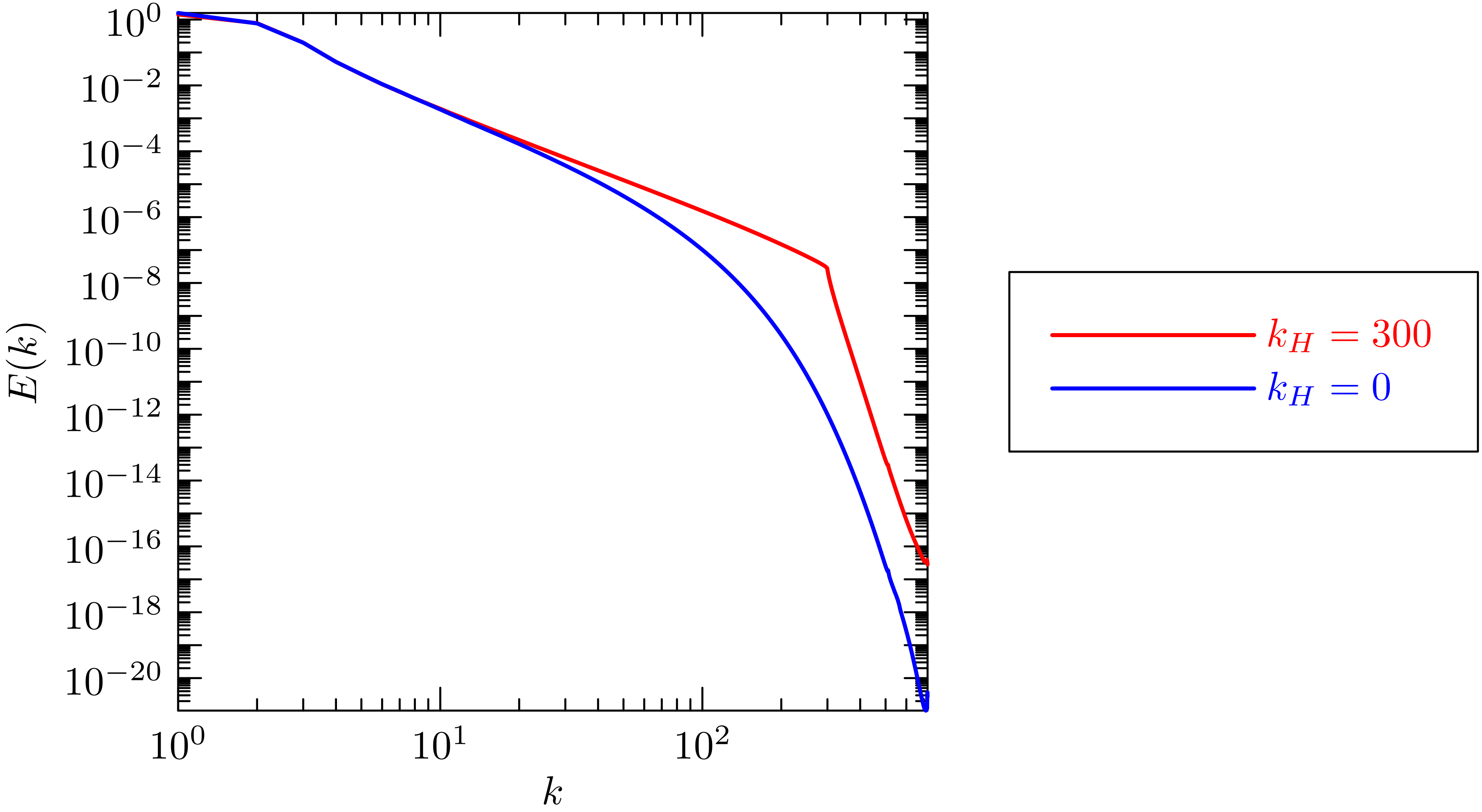
$$\frac{dM_n}{dt} = |\omega_{\boldsymbol{k}}|^n \,,$$

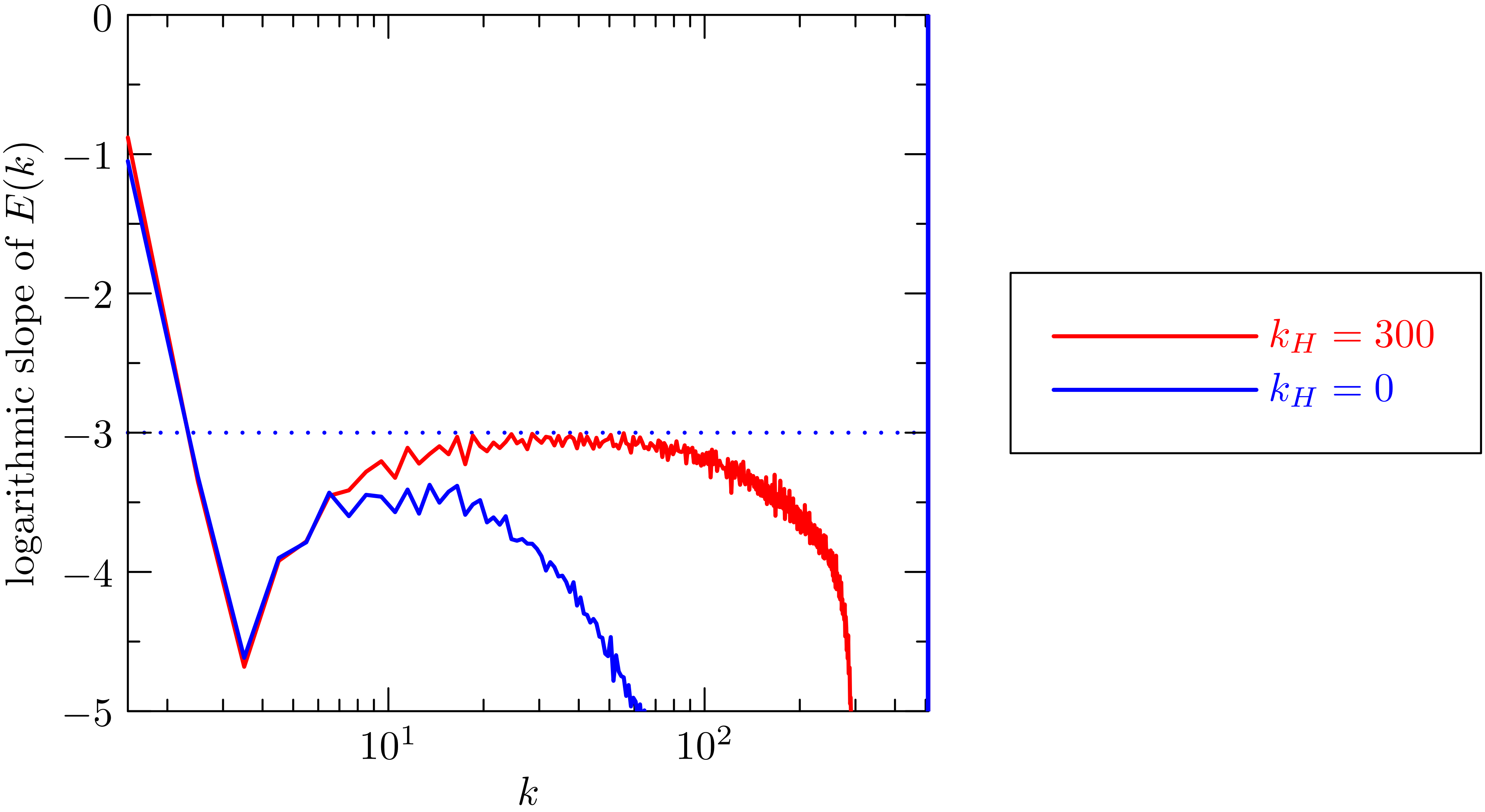
along with the vorticity $\omega_{\boldsymbol{k}}$ itself, using *the same* temporal discretization.

- These evolved quantities M_n can be used to extract accurate statistical averages during the post-processing phase, once the saturation time t_1 has been determined by the user:

$$\int_{t_1}^{t_2} |\omega_{\boldsymbol{k}}|^n (\tau) \, d\tau = M_n(t_2) - M_n(t_1).$$

Forcing at $k = 2$, friction for $k < 3$, viscosity for $k \geq k_H = 300$ (1023×1023 dealiased modes)




0
−1
−2
−3
−4
−5
logarithmic slope of $E(k)$
10^1
10^2
k
$k_H = 300$
$k_H = 0$

Conservative Integration

- Conservative integration [SIAM J. Appl. Math 59, 1112 (1999)] provides a useful diagnostic technique for ensuring that the underlying dynamical symmetries have been correctly implemented.

How to Reduce Degrees of Freedom?

- Spectral reduction reduces the number of degrees of freedom that must be explicitly evolved in turbulence simulations.

- Recognizing that spectral reduction yields inviscid equipartition spectra only with uniform binning, we developed an efficient FFT-based implementation, called pseudospectral reduction.

2D Turbulence in Fourier Space

- Navier–Stokes equation for vorticity $\omega \doteq \hat{\boldsymbol{z}}\boldsymbol{\cdot}\boldsymbol{\nabla}\times\boldsymbol{u}$:

$$\frac{\partial\omega}{\partial t} + \boldsymbol{u}\boldsymbol{\cdot}\boldsymbol{\nabla}\omega = \nu\nabla^2\omega + f.$$

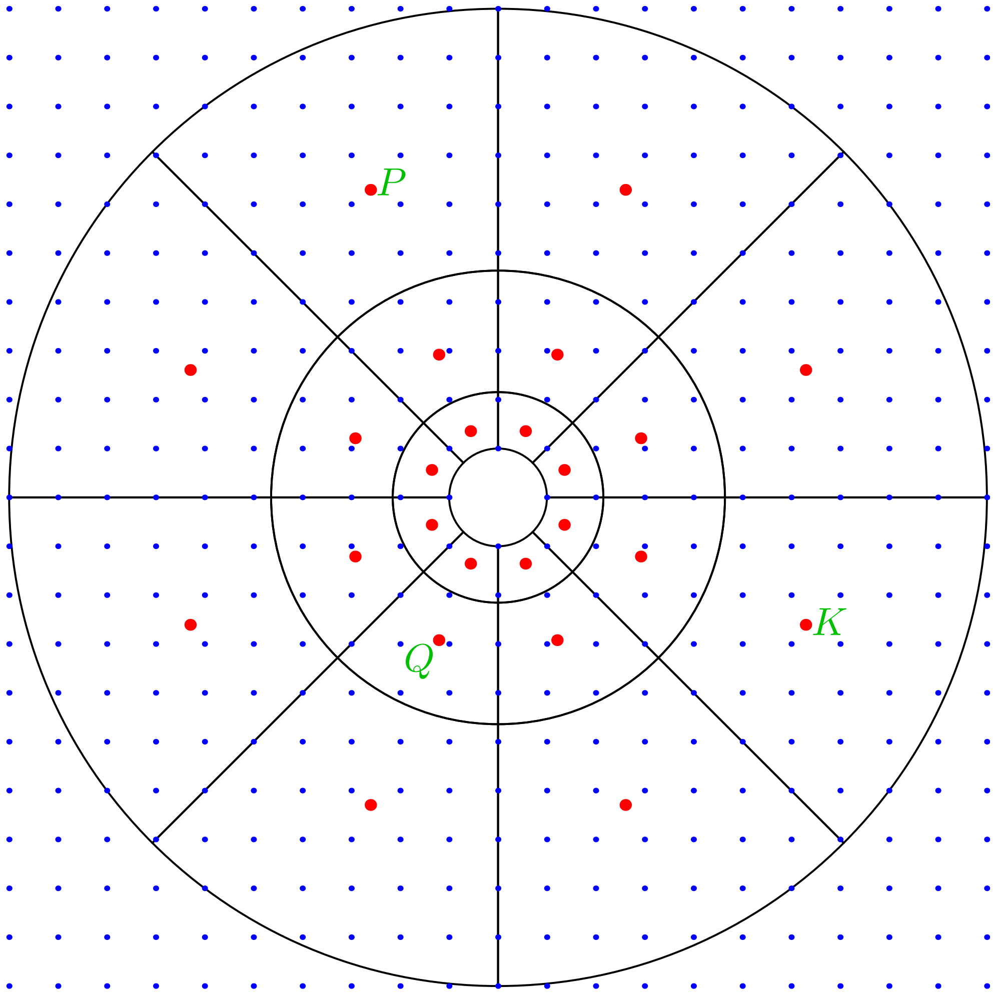
- In Fourier space:

$$\frac{\partial\omega_{\boldsymbol{k}}}{\partial t} + \nu_{\boldsymbol{k}}\omega_{\boldsymbol{k}} = \int d\boldsymbol{p}\int d\boldsymbol{q}\,\frac{\epsilon_{\boldsymbol{kpq}}}{q^2}\omega_{\boldsymbol{p}}^*\omega_{\boldsymbol{q}}^* + f_{\boldsymbol{k}},$$

where $\nu_{\boldsymbol{k}} \doteq \nu k^2$ and $\epsilon_{\boldsymbol{kpq}} \doteq (\hat{\boldsymbol{z}}\boldsymbol{\cdot}\boldsymbol{p}\times\boldsymbol{q})\,\delta(\boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q})$ is antisymmetric under permutation of any two indices.

Spectral Reduction

- Introduce a coarse-grained grid indexed by K:



Wavenumber Bin Geometry (8 × 3 bins)

- Define new variables

$$\Omega_{\boldsymbol{K}} = \langle \omega_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \doteq \frac{1}{\Delta_{\boldsymbol{K}}} \int_{\Delta_{\boldsymbol{K}}} \omega_{\boldsymbol{k}} \, d\boldsymbol{k},$$

where $\Delta_{\boldsymbol{K}}$ is the area of bin $\boldsymbol{K}$.

- Evolution of $\Omega_{\boldsymbol{K}}$:

$$\frac{\partial \Omega_{\boldsymbol{K}}}{\partial t} + \langle \nu_{\boldsymbol{k}} \omega_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} \Delta_{\boldsymbol{P}} \Delta_{\boldsymbol{Q}} \left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \omega^*_{\boldsymbol{p}} \omega^*_{\boldsymbol{q}} \right\rangle_{\boldsymbol{KPQ}},$$

where $\langle f \rangle_{\boldsymbol{KPQ}} = \dfrac{1}{\Delta_{\boldsymbol{K}} \Delta_{\boldsymbol{P}} \Delta_{\boldsymbol{Q}}} \displaystyle\int_{\Delta_{\boldsymbol{K}}} d\boldsymbol{k} \int_{\Delta_{\boldsymbol{P}}} d\boldsymbol{p} \int_{\Delta_{\boldsymbol{Q}}} d\boldsymbol{q} \, f.$

- Approximate $\omega_{\boldsymbol{p}}$ and $\omega_{\boldsymbol{q}}$ by bin-averaged values $\Omega_{\boldsymbol{P}}$ and $\Omega_{\boldsymbol{Q}}$:

$$\frac{\partial \Omega_{\boldsymbol{K}}}{\partial t} + \langle \nu_{\boldsymbol{k}} \rangle_{\boldsymbol{K}} \Omega_{\boldsymbol{K}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} \Delta_{\boldsymbol{P}} \Delta_{\boldsymbol{Q}} \left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \right\rangle_{\boldsymbol{KPQ}} \Omega^*_{\boldsymbol{P}} \Omega^*_{\boldsymbol{Q}}.$$

- Define the coarse-grained enstrophy Z and energy E:

$$Z \doteq \frac{1}{2}\sum_{\boldsymbol{K}} |\Omega_{\boldsymbol{K}}|^2 \Delta_{\boldsymbol{K}}, \qquad E \doteq \frac{1}{2}\sum_{\boldsymbol{K}} \frac{|\Omega_{\boldsymbol{K}}|^2}{K^2} \Delta_{\boldsymbol{K}}.$$

- Enstrophy is still conserved by the nonlinearity since

$$\left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \right\rangle_{\boldsymbol{KPQ}} \qquad \text{antisymmetric in} \qquad \boldsymbol{K} \leftrightarrow \boldsymbol{P}.$$

- But energy conservation has been lost!

$$\frac{1}{K^2}\left\langle \frac{\epsilon_{\boldsymbol{kpq}}}{q^2} \right\rangle_{\boldsymbol{KPQ}} \qquad \text{NOT antisymmetric in} \qquad \boldsymbol{K} \leftrightarrow \boldsymbol{Q}.$$

- Reinstate both desired symmetries with the modified coefficient

$$\frac{\langle \epsilon_{\boldsymbol{kpq}} \rangle_{\boldsymbol{KPQ}}}{Q^2}.$$

Properties

- We call the forced-dissipative version of this approximation *spectral reduction* (SR):

$$\boxed{\frac{\partial\Omega_{\boldsymbol{K}}}{\partial t}+\left\langle\nu_{\boldsymbol{k}}\right\rangle_{\boldsymbol{K}}\Omega_{\boldsymbol{K}}=\sum_{\boldsymbol{P},\boldsymbol{Q}}\Delta_{\boldsymbol{P}}\Delta_{\boldsymbol{Q}}\frac{\left\langle\epsilon_{\boldsymbol{kpq}}\right\rangle_{\boldsymbol{KPQ}}}{Q^2}\Omega^*_{\boldsymbol{P}}\Omega^*_{\boldsymbol{Q}}.}$$

- SR conserves both energy and enstrophy and reduces to the exact dynamics in the limit of small bin size.

- It has the same general structure and symmetries as the original equation and in this sense may be considered a *renormalization*.

- SR obeys a Liouville Theorem; in the inviscid limit, it yields statistical-mechanical (equipartition) solutions.

- However: since the $\delta_{\boldsymbol{k}+\boldsymbol{p}+\boldsymbol{q},0}$ factor in the nonlinear coefficient $\epsilon_{\boldsymbol{kpq}}$ has been smoothed over, spectral reduction is no longer a convolution: pseudospectral collocation does not apply.

Statistical Equipartition

- The Liouville Theorem and the coarse-grained invariants

$$E \doteq \frac{1}{2}\sum_{\boldsymbol{K}} \frac{|\Omega_{\boldsymbol{K}}|^2}{K^2}\Delta_{\boldsymbol{K}}, \qquad Z \doteq \frac{1}{2}\sum_{\boldsymbol{K}} |\Omega_{\boldsymbol{K}}|^2 \Delta_{\boldsymbol{K}},$$

  lead to statistical equipartition of $(\alpha/K^2+\beta)\,|\Omega_{\boldsymbol{K}}|^2\,\Delta_{\boldsymbol{K}}$. But this is correct only for uniform bins!

- However, for nonuniform bins, a rescaling of time by $\Delta_{\boldsymbol{K}}$,

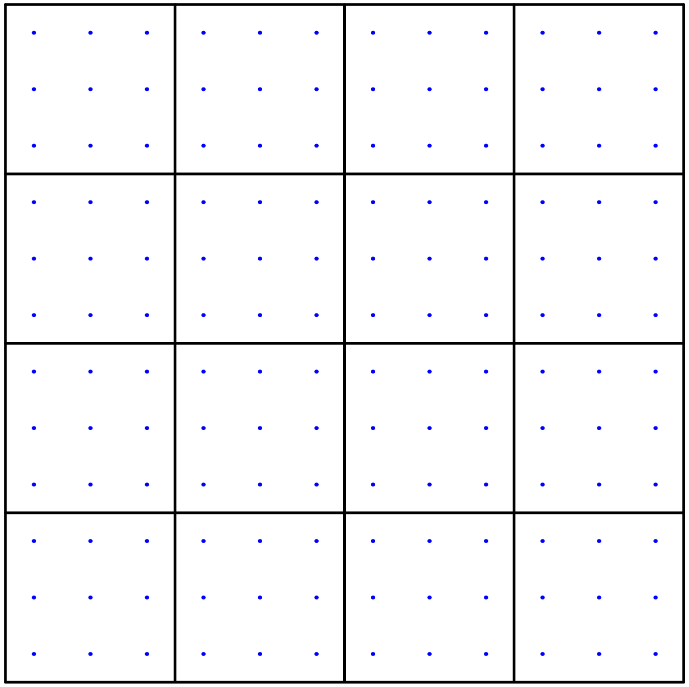
$$\frac{1}{\Delta_{\boldsymbol{K}}}\frac{\partial \Omega_{\boldsymbol{K}}}{\partial t} + \langle \nu_{\boldsymbol{k}} \rangle_{\boldsymbol{K}}\,\Omega_{\boldsymbol{K}} = \sum_{\boldsymbol{P},\boldsymbol{Q}} \Delta_{\boldsymbol{P}}\Delta_{\boldsymbol{Q}}\,\frac{\langle \epsilon_{\boldsymbol{kpq}} \rangle_{\boldsymbol{KPQ}}}{Q^2}\,\Omega^*_{\boldsymbol{P}}\Omega^*_{\boldsymbol{Q}},$$

  yields the correct inviscid equipartition: $\left\langle |\Omega_{\boldsymbol{K}}|^2 \right\rangle = \left(\frac{\alpha}{K^2}+\beta\right)^{-1}$.

- Unfortunately, the rescaled equations are hopelessly stiff.

Spectral Reduction on a Lattice

- Reluctantly, we accept the fact that each bin must contain the same number of modes.
- Imposing uniform bins has an important advantage: it affords a pseudospectral implementation of spectral reduction!
- Consider spectral reduction on a coarse-grained lattice, with $r \times r$ modes per rectangular bin (here $r = 3$):



1D Coarse-Grained Convolution

- The coarse-grained convolution $\langle f * g \rangle_K$ of f and g can then be computed as

$$\langle f * g \rangle_K \doteq \frac{1}{r} \sum_{\ell=0}^{r-1} (f * g)_{rK+\ell} = \frac{1}{r^2 M} \sum_{J=0}^{M-1} \zeta_M^{-KJ} \sum_{s=0}^{r-1} S_{J,s}^{*} \hat{f}_{sM+J} \hat{g}_{sM+J}$$

$$= \frac{1}{r^2 M} \sum_{J=0}^{M-1} \zeta_M^{-KJ} W_J \hat{F}_J \hat{G}_J,$$

in terms of the spatial weight factors $W_J \doteq \sum_{s=0}^{r-1} \left| S_{J,s} \right|^2 S_{J,s}$.

- Similarly, the bin-averaged Fourier transform of F_K weighted by ℓ is given by

$$\hat{f}_{sM+J} = \sum_{K=0}^{M-1} \sum_{\ell=0}^{r-1} \zeta_N^{(sM+J)(rK+\ell)} \ell F_K = T_{J,s} \hat{F}_J,$$

where

$$T_{J,s} \doteq \sum_{\ell=0}^{r-1} \ell \zeta_N^{J\ell} \zeta_r^{s\ell}.$$

- Let $W'_J \doteq \sum_{s=0}^{r-1} \left|S_{J,s}\right|^2 T_{J,s}$.

Pseudospectral reduction

- In terms of $F^0 \doteq K_x\Omega_{\boldsymbol{K}}$, $F^1 \doteq K_y\Omega_{\boldsymbol{K}}$, $F^2 \doteq \Omega_{\boldsymbol{K}}$, $G^0 \doteq K_xK^{-2}\Omega_{\boldsymbol{K}}$, $G^1 \doteq K_yK^{-2}\Omega_{\boldsymbol{K}}$, and $G^2 \doteq K^{-2}\Omega_{\boldsymbol{K}}$:

$$
\begin{aligned}
&\sum_{\boldsymbol{P},\boldsymbol{Q}} \frac{1}{Q^2} \left\langle \delta_{\boldsymbol{p}+\boldsymbol{q},\boldsymbol{k}}(p_xq_y - p_yq_x)\right\rangle_{\boldsymbol{KPQ}} \Omega_{\boldsymbol{P}}\Omega_{\boldsymbol{Q}} \\
&= \frac{1}{r^2}\sum_{\boldsymbol{\ell}} \Big(\big[(rK_x+\ell_x)\Omega_{\boldsymbol{K}}\big] * \big[(rK_y+\ell_y)K^{-2}\Omega_{\boldsymbol{K}}\big]\Big)_{r\boldsymbol{K}+\boldsymbol{\ell}} \\
&\qquad -\frac{1}{r^2}\sum_{\boldsymbol{\ell}} \Big(\big[(rK_y+\ell_y)\Omega_{\boldsymbol{K}}\big] * \big[(rK_x+\ell_x)K^{-2}\Omega_{\boldsymbol{K}}\big]\Big)_{r\boldsymbol{K}+\boldsymbol{\ell}} \\
&= \frac{1}{r^4M^2}\sum_{\boldsymbol{J}} \zeta_M^{-\boldsymbol{K}\cdot\boldsymbol{J}} \Big[r^2 W_{J_x}W_{J_y}(\hat{F}^0_{\boldsymbol{J}}\hat{G}^1_{\boldsymbol{J}} - \hat{F}^1_{\boldsymbol{J}}\hat{G}^0_{\boldsymbol{J}}) \\
&\qquad + rW'_{J_x}W_{J_y}(\hat{F}^2_{\boldsymbol{J}}\hat{G}^1_{\boldsymbol{J}} - \hat{F}^1_{\boldsymbol{J}}\hat{G}^2_{\boldsymbol{J}}) + rW_{J_x}W'_{J_y}(\hat{F}^0_{\boldsymbol{J}}\hat{G}^2_{\boldsymbol{J}} - \hat{F}^2_{\boldsymbol{J}}\hat{G}^0_{\boldsymbol{J}})\Big].
\end{aligned}
$$

- Computational complexity is $\mathcal{O}(N\log N)$, with a coefficient $7/5 = 1.4$ times greater that for pseudospectral collocation.

Conclusions

- For centered convolutions in d dimensions implicit padding asymptotically uses $(2/3)^{d-1}$ of the conventional storage.

- The factor of 2 speedup is largely due to increased data locality.

- Highly optimized and parallelized implicit dealiasing routines have been implemented as a software layer `FFTW++` (v 2.00) on top of the `FFTW` library and released under the Lesser GNU Public License: `http://fftwpp.sourceforge.net/`

- The advent of implicit dealiasing of convolutions makes overlapping transposition with FFT computation feasible.

- Dynamic moment averaging allows the integration time window to be specified by the user *a posteriori*. The cumulative enstrophy transfer function is an excellent diagnostic for determining the saturation time.

- Writing of a high-performance dealiased pseudospectral code is now a relatively straightforward exercise!

Asymptote: 2D & 3D Vector Graphics Language

Andy Hammerlindl, John C. Bowman, Tom Prince

`http://asymptote.sf.net`

(freely available under the Lesser GNU Public License)

References

[Bowman & Roberts 2011] J. C. Bowman & M. Roberts, SIAM J. Sci. Comput., **33**:386, 2011.

[Bowman *et al.* 1996] J. C. Bowman, B. A. Shadwick, & P. J. Morrison, "Spectral reduction for two-dimensional turbulence," in *Transport, Chaos, and Plasma Physics 2*, edited by S. Benkadda, F. Doveil, & Y. Elskens, pp. 58–73, New York, 1996, Institute Méditerranéen de Technologie (Marseille, 1995), World Scientific.

[Bowman *et al.* 2001] J. C. Bowman, B. A. Shadwick, & P. J. Morrison, "Numerical challenges for turbulence computation: Statistical equipartition and the method of spectral reduction," in *Scientific Computing and Applications*, edited by P. Minev, Y. S. Wong, & Y. Lin, volume 7 of *Advances in Computation: Theory and Practice*, pp. 171–178, Huntington, New York, 2001, Nova Science Publishers.

[Orszag 1971] S. A. Orszag, Journal of the Atmospheric Sciences, **28**:1074, 1971.

[Roberts 2011] M. Roberts, *Multispectral Reduction of Two-Dimensional Turbulence*, Ph.D. thesis, University of Alberta, Edmonton, AB, Canada, 2011, `http://www.math.ualberta.ca/~bowman/group/roberts_phd.pdf`.

[Shadwick *et al.* 1999] B. A. Shadwick, J. C. Bowman, & P. J. Morrison, SIAM J. Appl. Math., **59**:1112, 1999.