How Important is Dealiasing for Turbulence Simulations?

John C. Bowman

University of Alberta

September 30, 2014

`www.math.ualberta.ca/∼bowman/talks`

Acknowledgements: Malcolm Roberts (Université de Strasbourg)

Dealiasing

- Over 40 years ago, Orszag pointed out the importance of dealiasing in the pseudospectral method.
- Dealiasing pseudospectral convolutions, either by padding or phase-shift dealiasing is expensive.
- Some researchers in the past have therefore neglected to dealias pseudospectral simulations of strongly damped flow.
- This shortcut is typically justified with the claim that high-wavenumber damping is sufficiently strong so that the dealiasing error contributes negligibly to the large energy-containing scales.
- On the other hand, Hou and Li proposed in 2007 that high-order Fourier smoothing captures nearly singular solutions of the 1D inviscid Burgers equations and the 3D Euler equations more accurately and efficiently than explicit dealiasing via 2/3 zero padding.

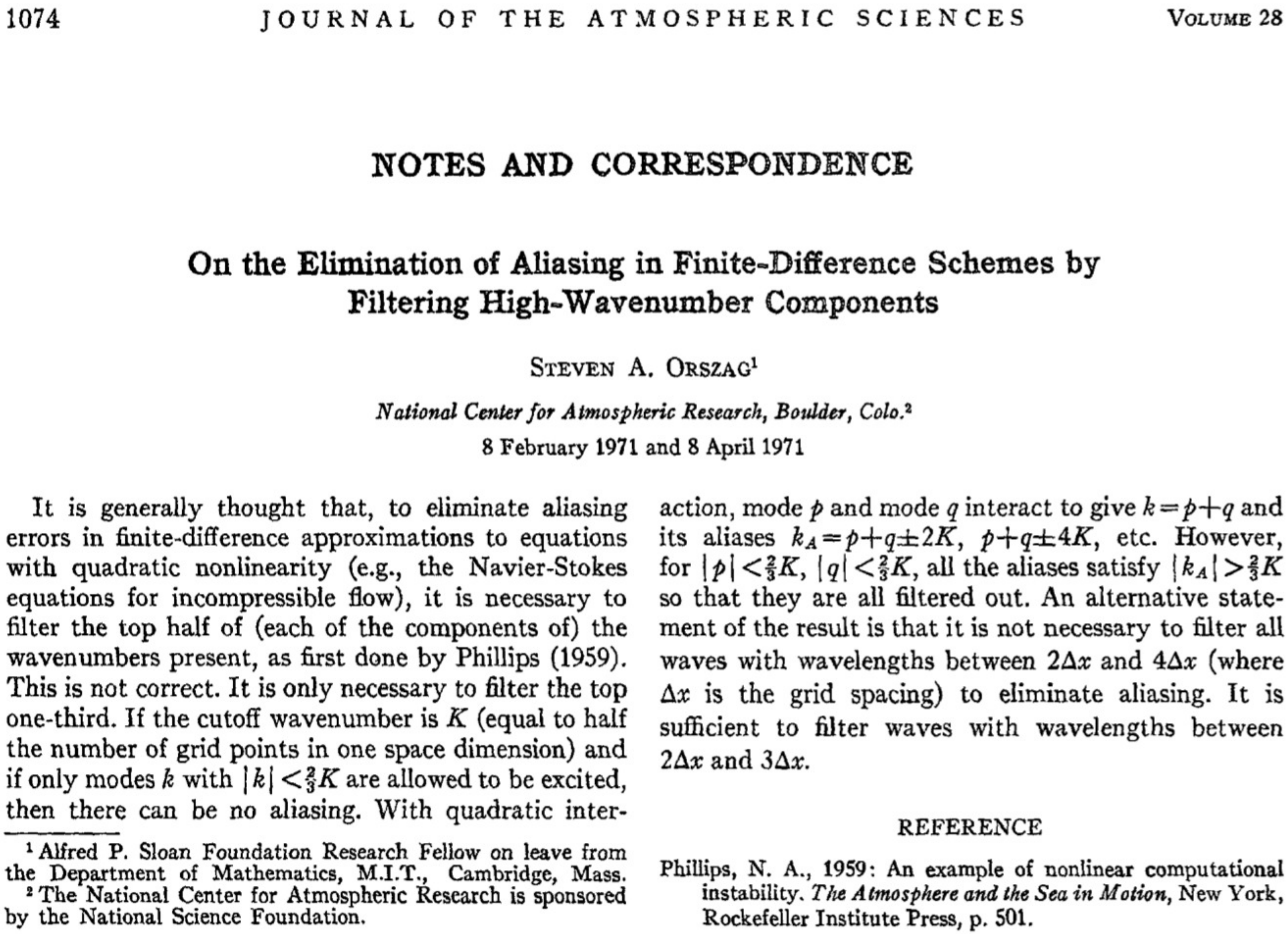
- Given that high-Reynolds number turbulence, with well-resolved inertial ranges, falls midway between these two limiting cases of large viscosity vs. vanishing viscosity, it seems prudent to reconfirm the importance of properly dealiasing turbulence simulations. . .

A Brief History of Dealiasing

- Gauss 1866: Nachlass: Theoria interpolationis methodo nova tractata
  - Earliest example of computing Fourier transforms via divide-and-conquer strategy (FFT)
- Phillips 1959: An example of non-linear computational instability
- Cooley & Tukey 1965: An Algorithm for the Machine Calculation of Complex Fourier Series
  - popularized general FFT algorithm

- Orszag 1971: Elimination of aliasing in finite-difference schemes by filtering high-wavenumber components
  - zero padding
- Patterson & Orszag 1971: Spectral calculations of isotropic turbulence: Efficient removal of aliasing interactions
  - phase-shift dealiasing
- Choi *et al.* 1995: Parallel matrix tranpose algorithms on distributed memory concurrent computers
- Hou & Li 2007: Computing nearly singular solutions using pseudo-spectral methods
- Bowman & Roberts 2011: Efficient dealiased convolutions without padding
  - implicit dealiasing

Origin of the 2/3 Rule



NOTES AND CORRESPONDENCE

On the Elimination of Aliasing in Finite-Difference Schemes by Filtering High-Wavenumber Components

STEVEN A. ORSZAG[1]

National Center for Atmospheric Research, Boulder, Colo.[2]

8 February 1971 and 8 April 1971

It is generally thought that, to eliminate aliasing errors in finite-difference approximations to equations with quadratic nonlinearity (e.g., the Navier-Stokes equations for incompressible flow), it is necessary to filter the top half of (each of the components of) the wavenumbers present, as first done by Phillips (1959). This is not correct. It is only necessary to filter the top one-third. If the cutoff wavenumber is K (equal to half the number of grid points in one space dimension) and if only modes k with $|k|<\frac{2}{3}K$ are allowed to be excited, then there can be no aliasing. With quadratic interaction, mode p and mode q interact to give $k=p+q$ and its aliases $k_A=p+q\pm 2K$, $p+q\pm 4K$, etc. However, for $|p|<\frac{2}{3}K$, $|q|<\frac{2}{3}K$, all the aliases satisfy $|k_A|>\frac{2}{3}K$ so that they are all filtered out. An alternative statement of the result is that it is not necessary to filter all waves with wavelengths between $2\Delta x$ and $4\Delta x$ (where Δx is the grid spacing) to eliminate aliasing. It is sufficient to filter waves with wavelengths between $2\Delta x$ and $3\Delta x$.

[1] Alfred P. Sloan Foundation Research Fellow on leave from the Department of Mathematics, M.I.T., Cambridge, Mass.

[2] The National Center for Atmospheric Research is sponsored by the National Science Foundation.

REFERENCE

Phillips, N. A., 1959: An example of nonlinear computational instability. *The Atmosphere and the Sea in Motion*, New York, Rockefeller Institute Press, p. 501.

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 $\zeta_N^N = 1$.

- 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

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

- The pseudospectral method requires a *linear convolution*.

Convolution Theorem

$$\sum_{j=0}^{N-1} f_j g_j \zeta_N^{-jk} = \sum_{j=0}^{N-1} \zeta_N^{-jk} \left(\sum_{p=0}^{N-1} \zeta_N^{jp} F_p \right) \left(\sum_{q=0}^{N-1} \zeta_N^{jq} G_q \right)$$

$$= \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} F_p G_q \sum_{j=0}^{N-1} \zeta_N^{(-k+p+q)j}$$

$$= N \sum_{s} \sum_{p=0}^{N-1} F_p G_{k-p+sN}.$$

- The terms indexed by $s \neq 0$ are *aliases;* we need to remove them by ensuring that $G_{k-p+sN} = 0$ whenever $s \neq 0$.

- If F_p and G_{k-p+sN} are nonzero only for $0 \leq p \leq m-1$ and $0 \leq k-p+sN \leq m-1$, then we want $k+sN \leq 2m-2$ to have no solutions for positive s.

- This can be achieved by choosing $N \geq 2m-1$.

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

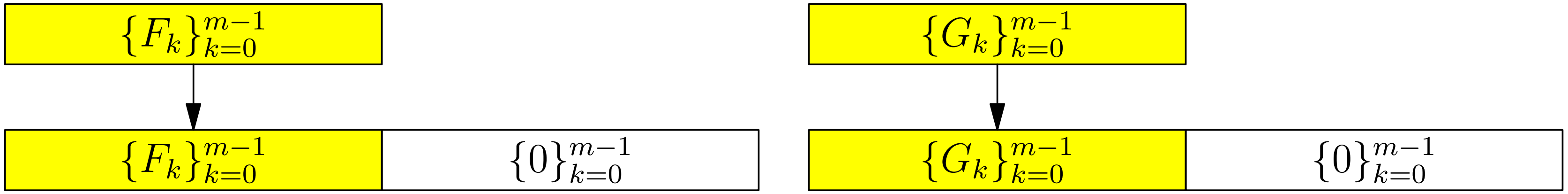
- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:

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

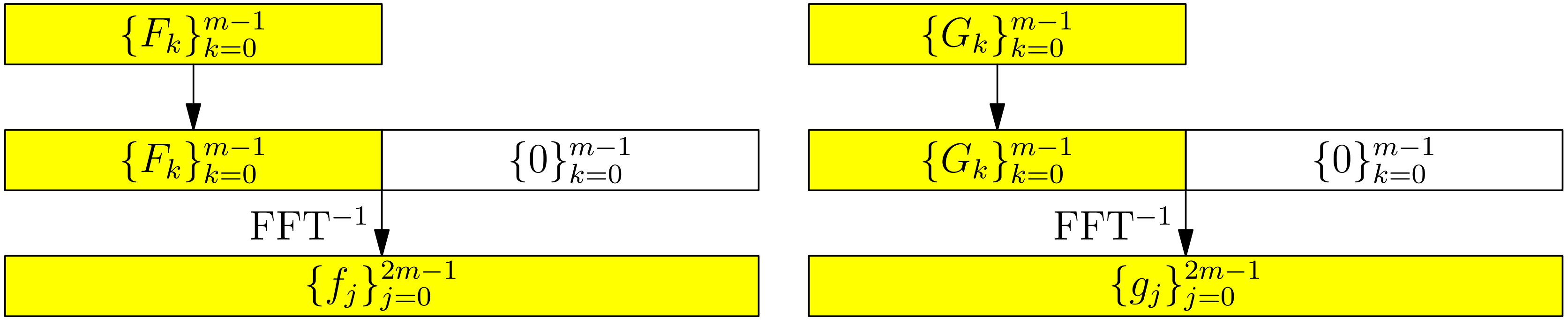
$\{G_k\}_{k=0}^{m-1}$

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

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

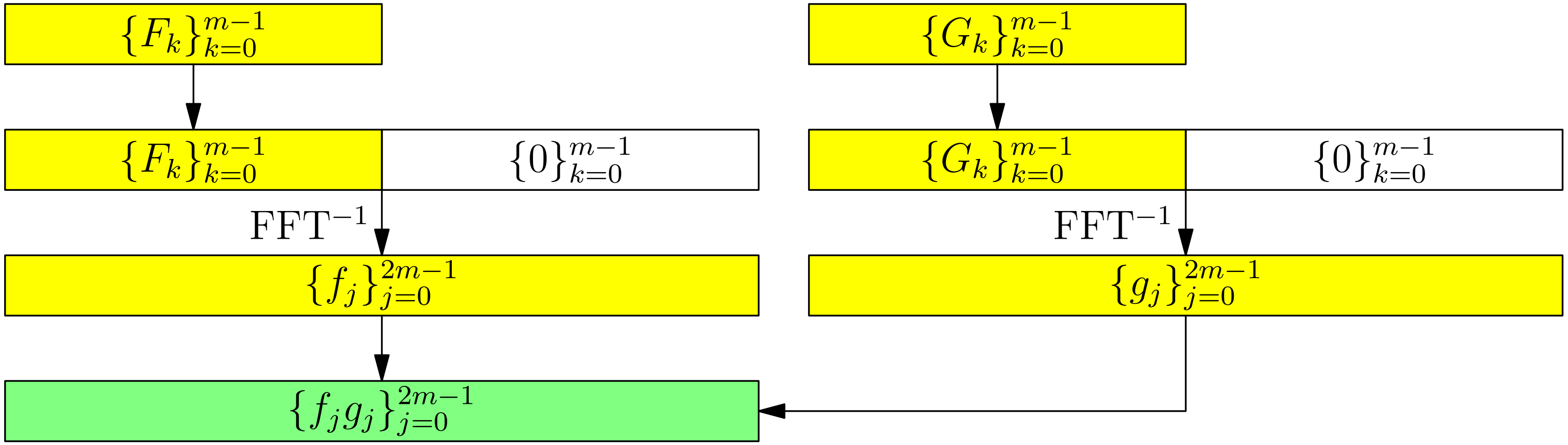


- One can dealias by *zero padding* input data vectors of length m to length $N \geq 2m-1$.
- Since FFT sizes with small prime factors in practice yield the most efficient implementations, the padding is normally extended to $N = 2m$:



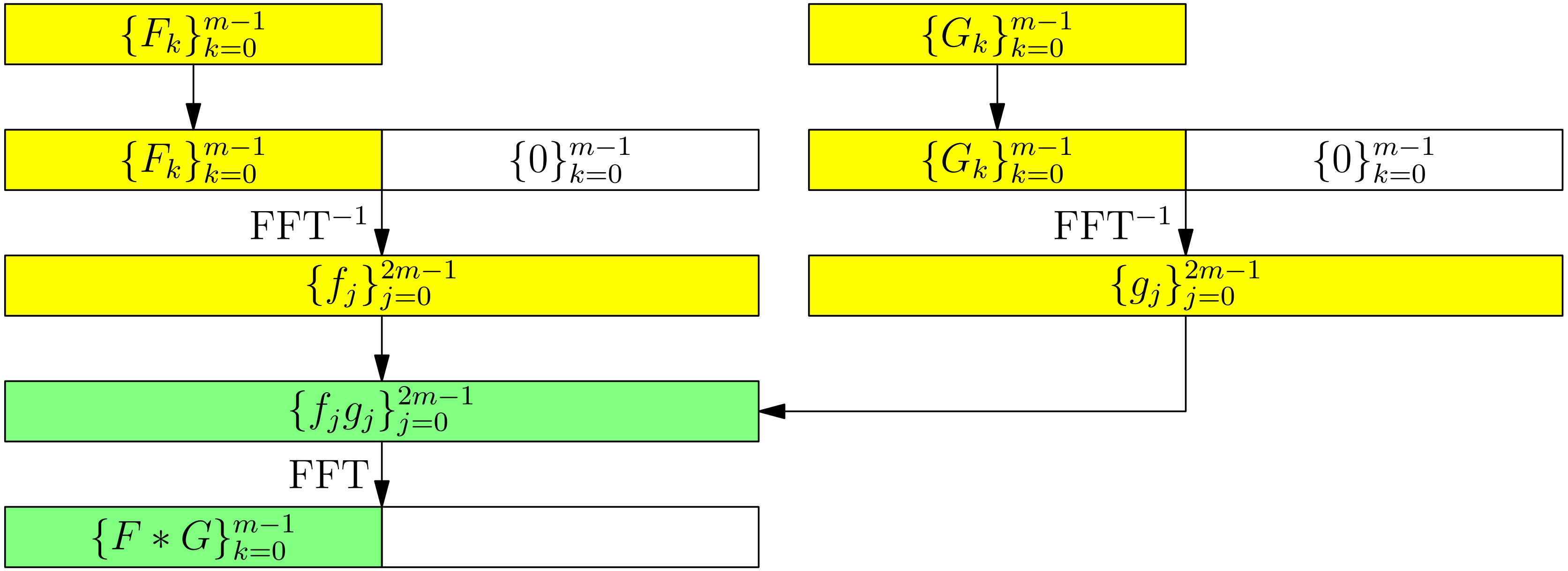
- One can dealias by *zero padding* input data vectors of length m to length $N \geq 2m-1$.

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



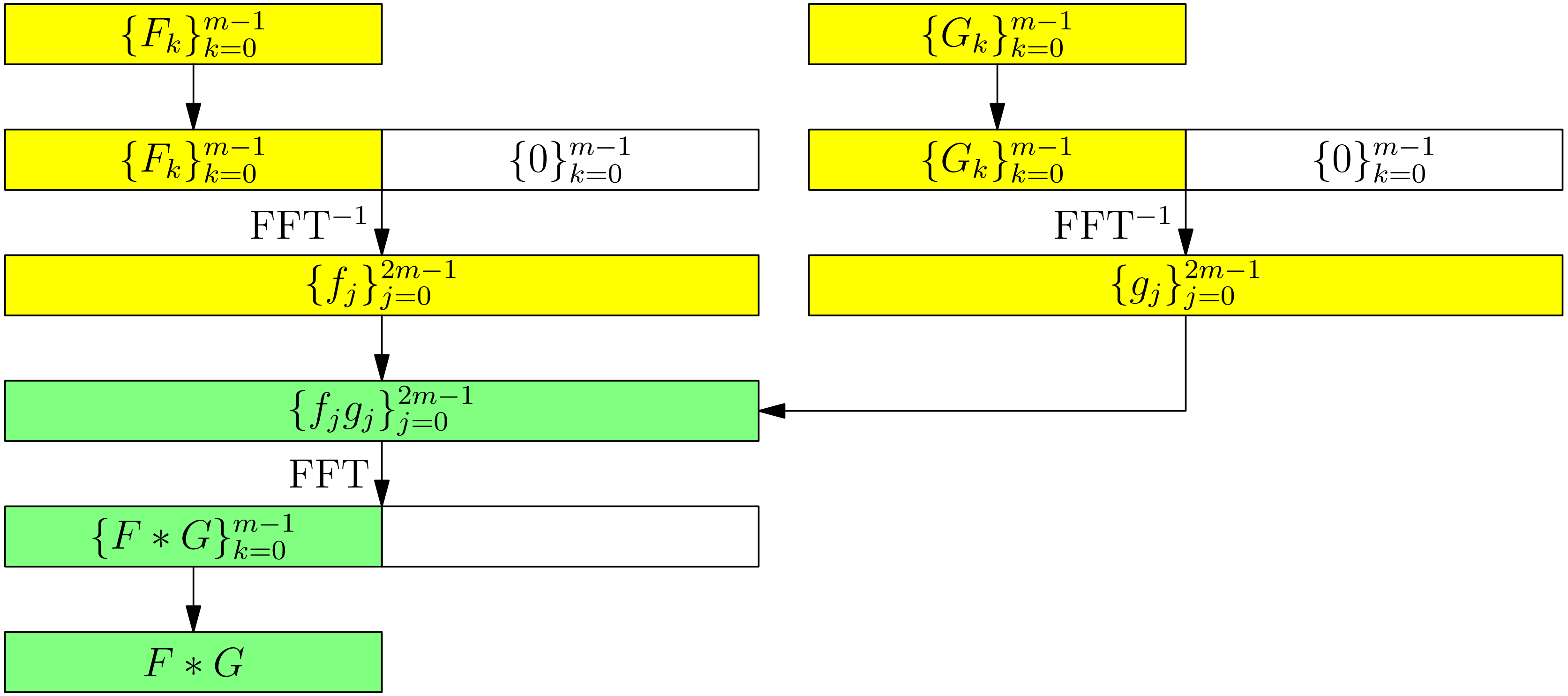
- One can dealias by *zero padding* input data vectors of length m to length $N \geq 2m-1$.

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



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

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



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

Hermitian Convolutions

- *Hermitian convolutions* arise when the input vectors are Fourier transforms of real data:

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

Centered Convolutions

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

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

- Here, one needs to pad to $N \geq 3m-2$ to prevent mode $m-1$ from beating with itself to contaminate the most negative (first) mode, corresponding to wavenumber $-m+1$. Since the ratio of the number of physical to total modes, $(2m-1)/(3m-2)$ is asymptotic to $2/3$ for large m, this padding scheme is often referred to as the *2/3 padding rule.*

- The Hermiticity condition then appears as

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

Aliasing Error

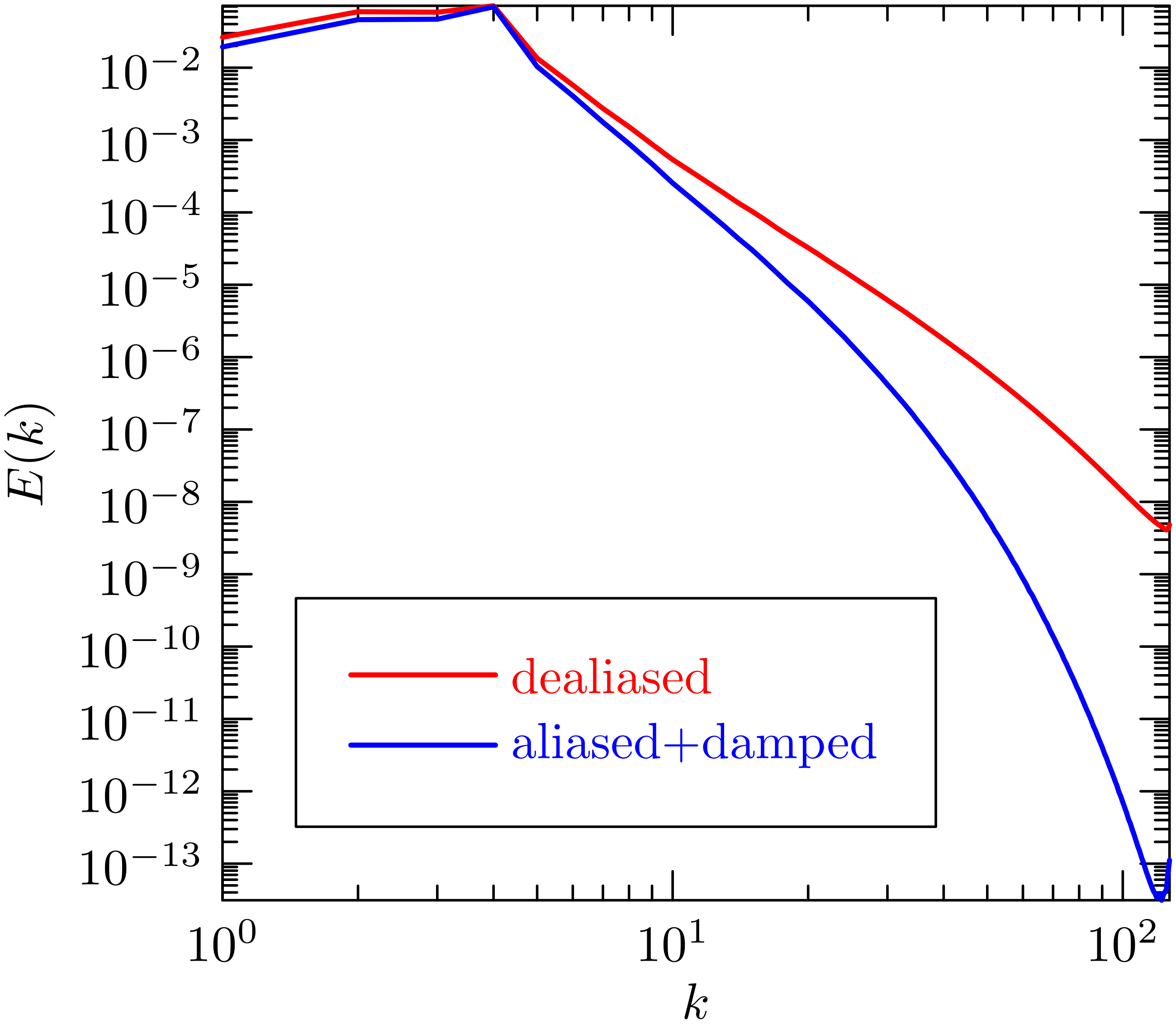
- The claim that aliasing error is negligible is problematic: in an undealiased pseudospectral simulation on $[-N/2, N/2]$, the inertial range mode at $N/3$ will beat with itself to generate a spurious harmonic at wavenumber $2N/3 = -N/3 \pmod{N}$.

- Moreover, as Phillips pointed out in 1959, aliasing errors not only contaminate the largest scales, they quickly lead to high-wavenumber explosive numerical instability.

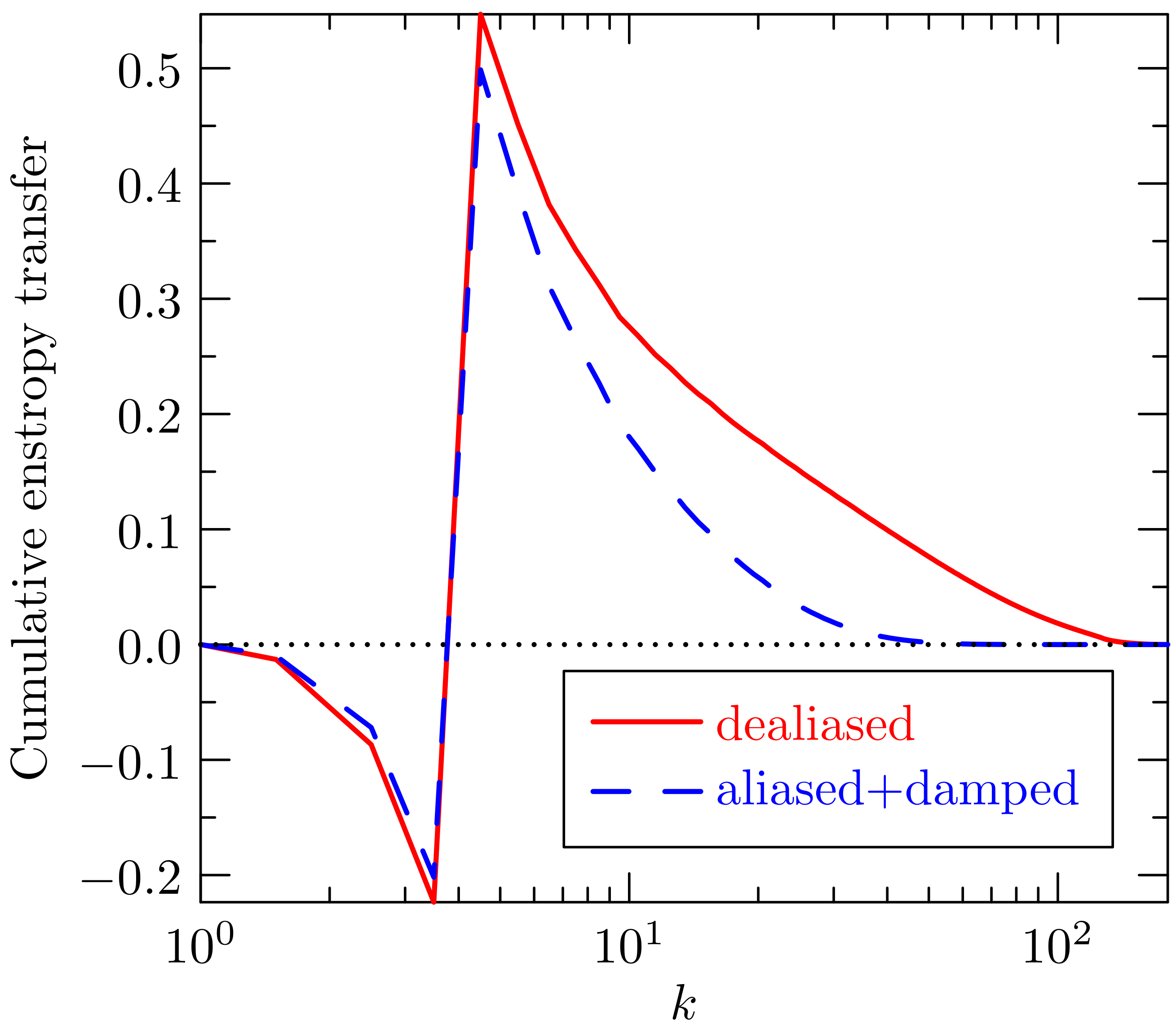
- In one experiment, we had to increase the viscosity by a factor of 15 to damp out the aliasing instability.

- This additional damping completely destroyed the inertial range and modified the energy transfers and large-scale energy spectrum.

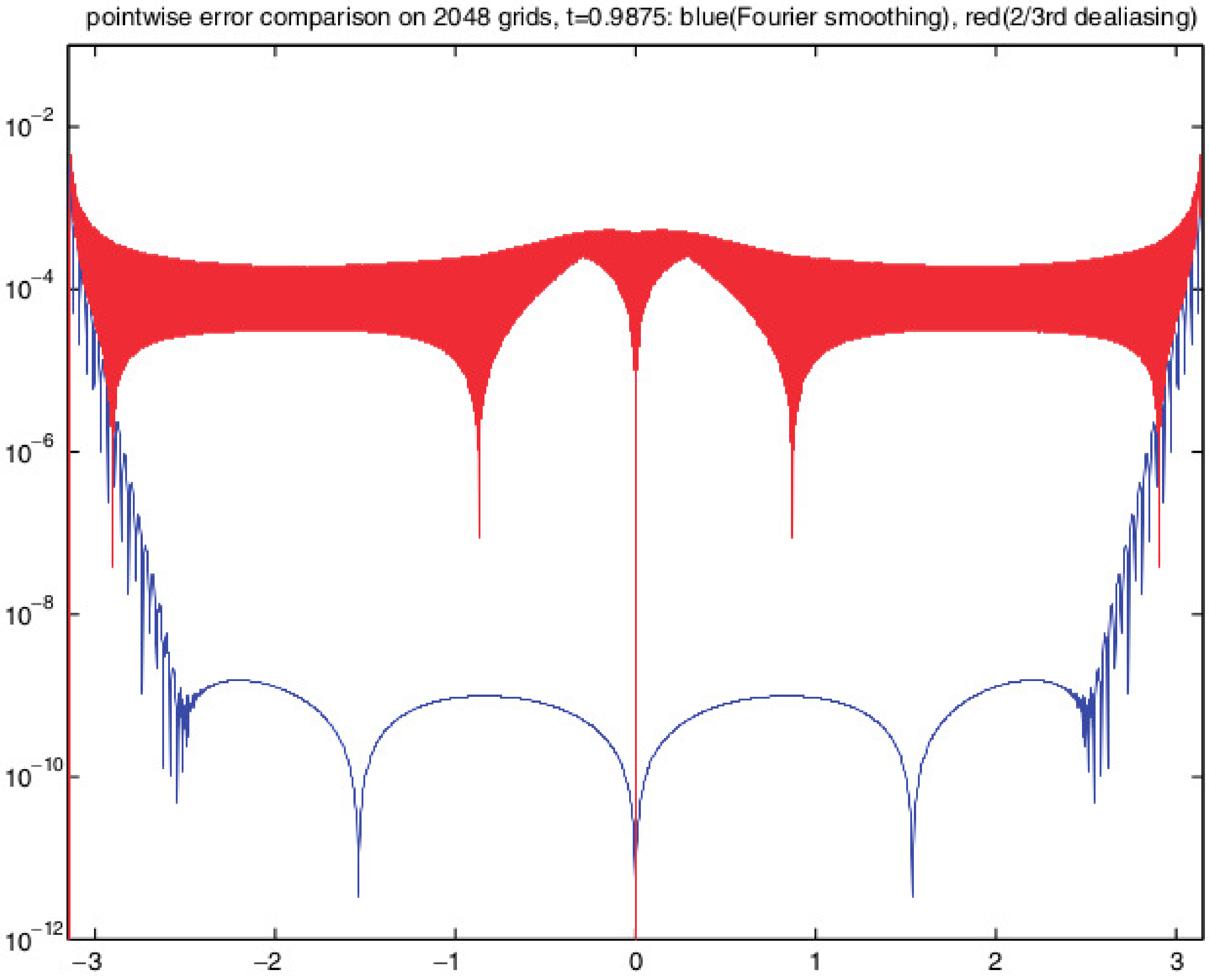
2D Enstrophy Cascade



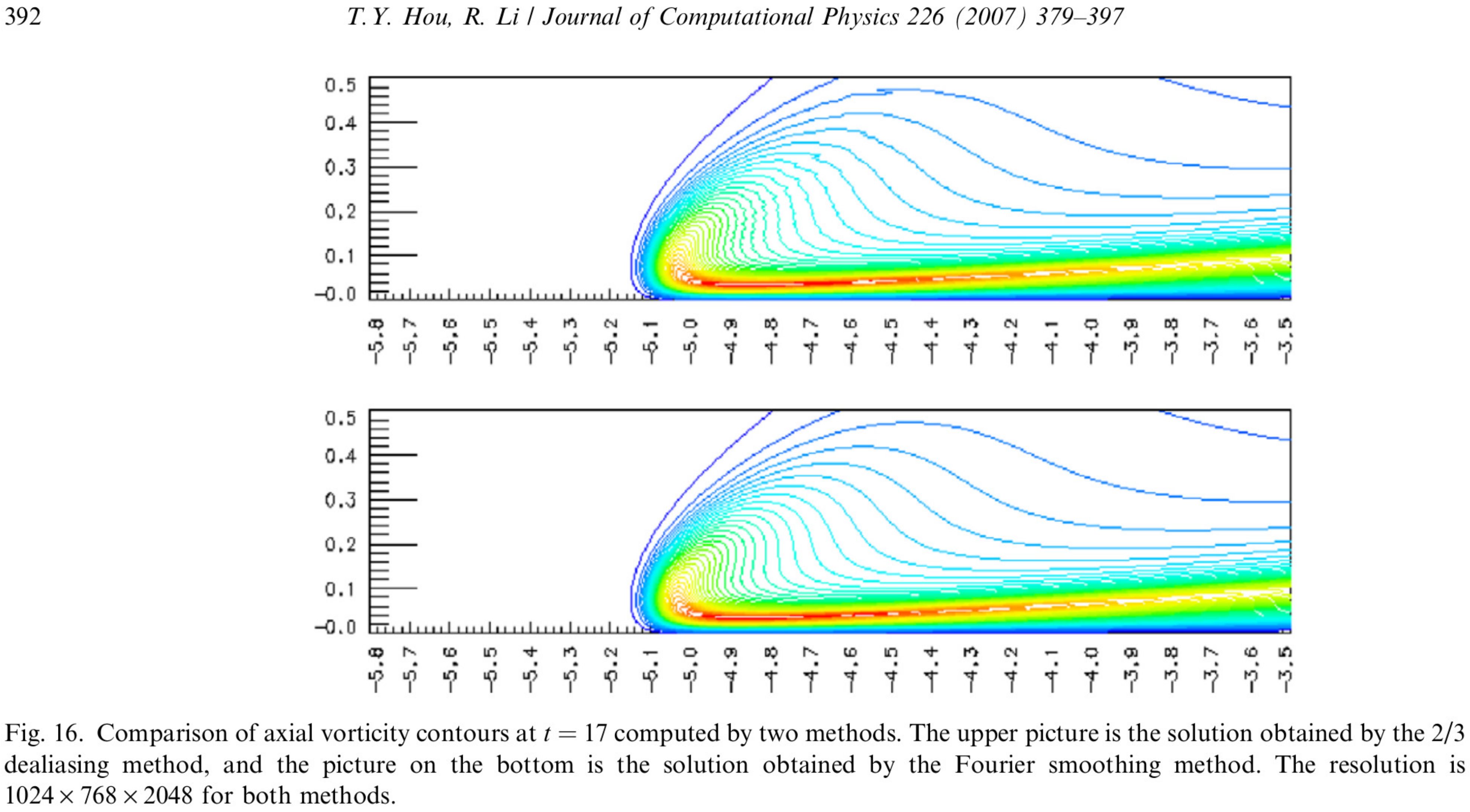
2D Enstrophy Transfer



Partial Dealiasing: Fourier Filter for 1D Burgers



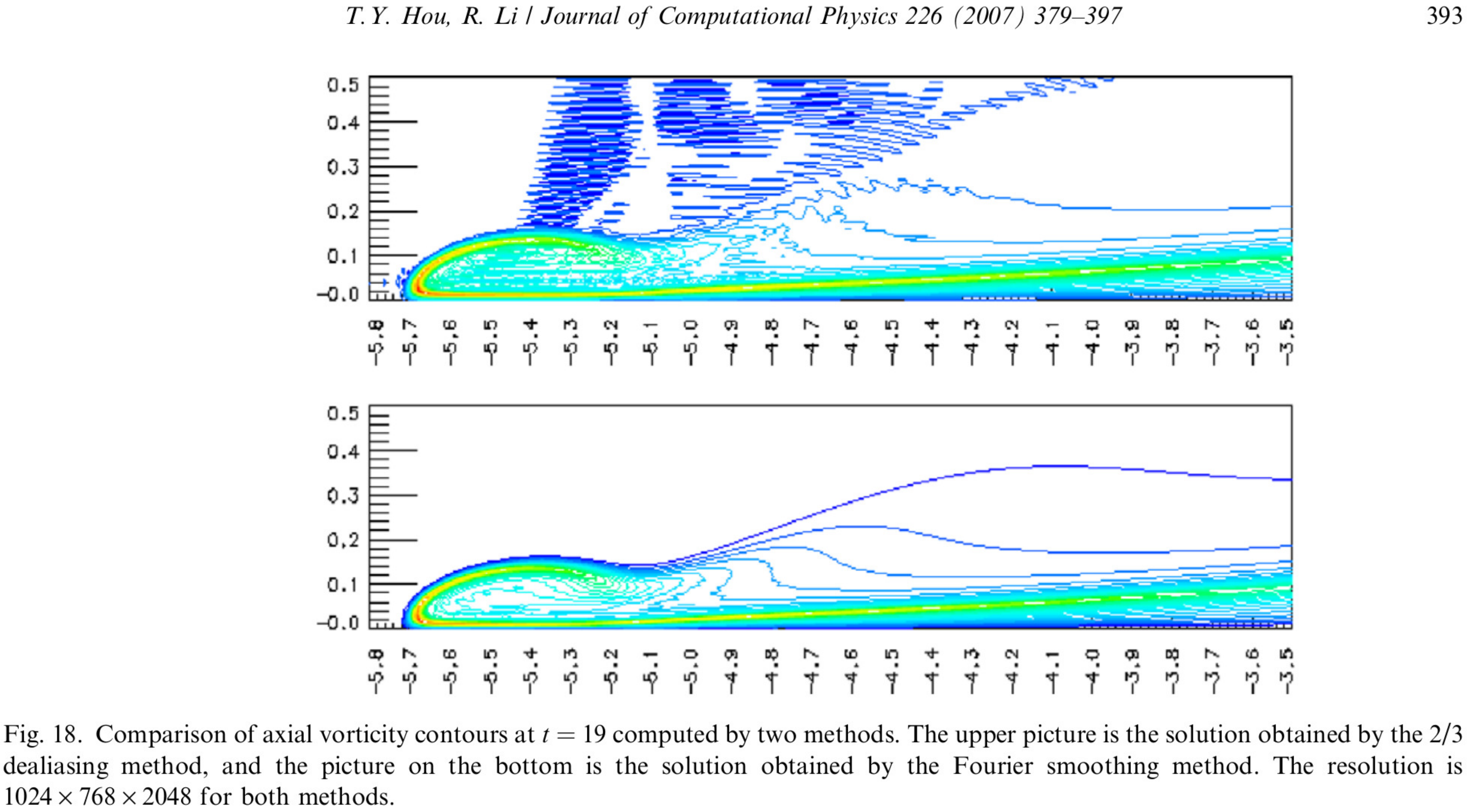
3D Euler At $t = 17$


392 *T.Y. Hou, R. Li / Journal of Computational Physics 226 (2007) 379–397*

0.5
0.4
0.3
0.2
0.1
−0.0
−5.8
−5.7
−5.6
−5.5
−5.4
−5.3
−5.2
−5.1
−5.0
−4.9
−4.8
−4.7
−4.6
−4.5
−4.4
−4.3
−4.2
−4.1
−4.0
−3.9
−3.8
−3.7
−3.6
−3.5

Fig. 16. Comparison of axial vorticity contours at $t = 17$ computed by two methods. The upper picture is the solution obtained by the 2/3 dealiasing method, and the picture on the bottom is the solution obtained by the Fourier smoothing method. The resolution is $1024 \times 768 \times 2048$ for both methods.

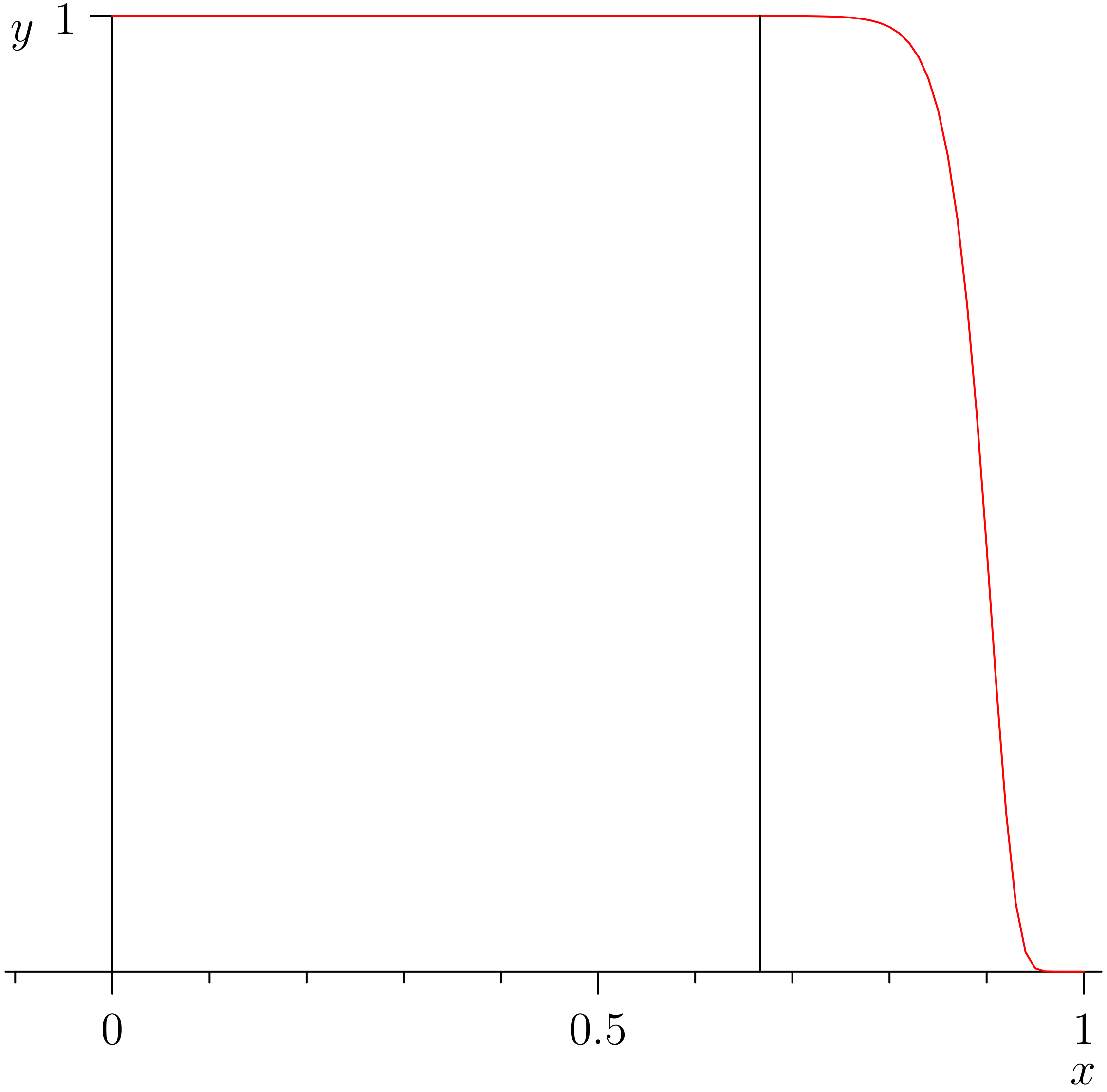
3D Euler At $t = 19$



T.Y. Hou, R. Li / Journal of Computational Physics 226 (2007) 379–397 393

Fig. 18. Comparison of axial vorticity contours at $t = 19$ computed by two methods. The upper picture is the solution obtained by the 2/3 dealiasing method, and the picture on the bottom is the solution obtained by the Fourier smoothing method. The resolution is $1024 \times 768 \times 2048$ for both methods.

Fourier Filter $y = e^{-36x^{36}}$



Critique of Fourier Smoothing Filter

- The fact that the Fourier filter encroaches beyond the 2/3 limit means that the computed solution is contaminated with unwanted aliases.

- Hou & Li state:

> *. . . the extra Fourier modes that are kept by the Fourier smoothing method give indeed an accurate approximation to the correct Fourier modes.*

- However, they did not compare to the pseudospectral error of an enlarged, properly dealiased, simulation that actually resolves these modes.

- They also argued that:

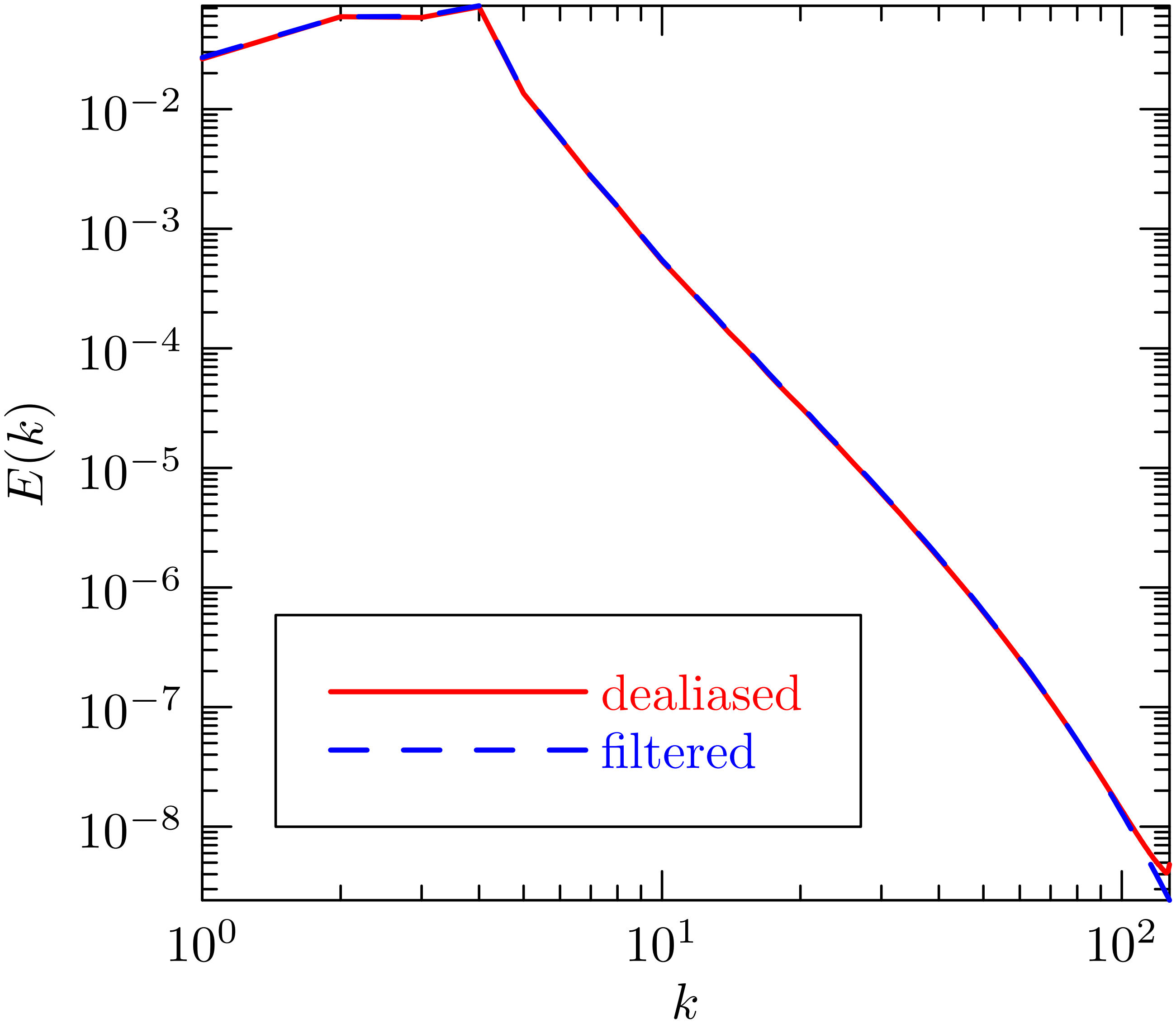
> This abrupt cut-off in the high frequency spectrum gives rise to the well-known Gibbs phenomenon and the loss of the L^2 energy, which is the main cause for the highly oscillatory and widespread pointwise error. . .

- The Gibbs phenomenon is not the issue: to machine precision, the dealiased FFT-based convolution produces results identical to a direct convolution and also to phase-shift dealiasing, where there is no 2/3 spectral cutoff of any sort.

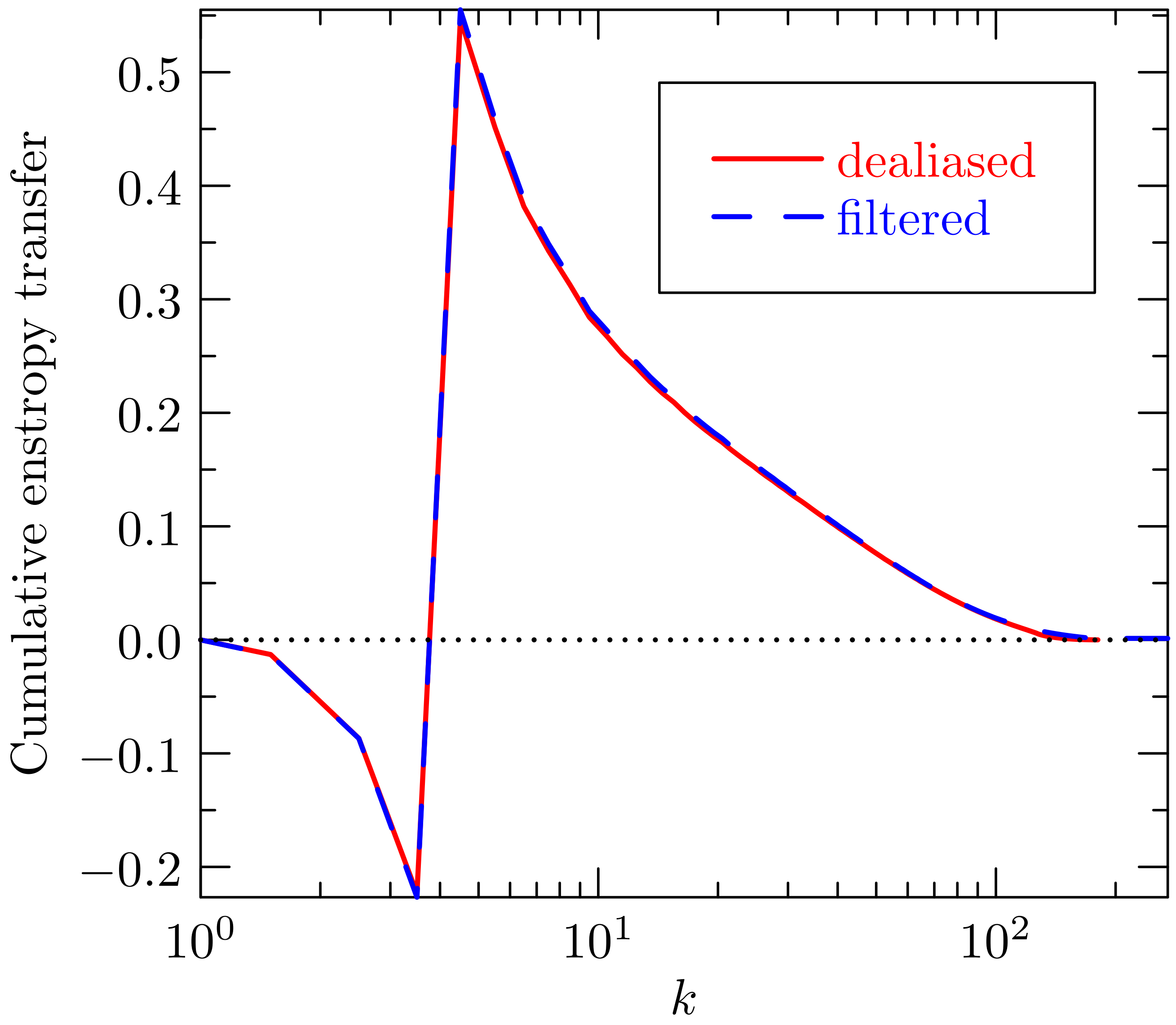
- Dealiasing by zero-padding does not contaminate the spectrum, but being underresolved does contaminate the spectrum (sometimes explosively). Indeed, Hou & Li themselves state:

> *We observe that when the solution is relatively smooth and can be resolved by the computational grid, the spectra of the two methods are almost indistinguishable.*

2D Enstrophy Cascade



2D Enstrophy Transfer



Efficiency

- Hou & Li argue that smoothing via a Fourier filter is 20% more efficient than dealiasing.

- However, the recently introduced pruning technique called implicit dealiasing [Bowman & Roberts 2011], which in two and three dimensions is roughly twice as fast as explicit dealiasing, more than offsets this 20% gain in speed.

Pruned FFTs

- Although explicit padding seems like an obvious waste of memory and computation, the conventional wisdom on avoiding this waste is well summed up by Steven G. Johnson, coauthor of the **FFTW** ("Fastest Fourier Transform in the West") library :

> *The most common case where people seem to want a pruned FFT is for zero-padded convolutions, where roughly 50% of your inputs are zero (to get a linear convolution from an FFT-based cyclic convolution). Here, a pruned FFT is hardly worth thinking about, at least in one dimension. In higher dimensions, matters change (e.g. for a 3d zero-padded array about 1/8 of your inputs are non-zero, and one can fairly easily save a factor of two or so simply by skipping 1d sub-transforms that are zero).*

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:

$$f_{2\ell} = \sum_{k=0}^{m-1} \zeta_{2m}^{2\ell k} F_k = \sum_{k=0}^{m-1} \zeta_m^{\ell k} F_k,$$

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

- 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.

- An implicitly padded convolution is implemented as in our `FFTW++` library (version 1.13) as `cconv(f,g,u,v)` computes an in-place implicitly dealiased convolution of two complex vectors `f` and `g` using two temporary vectors `u` and `v`, each of length m.

- This in-place convolution requires 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.

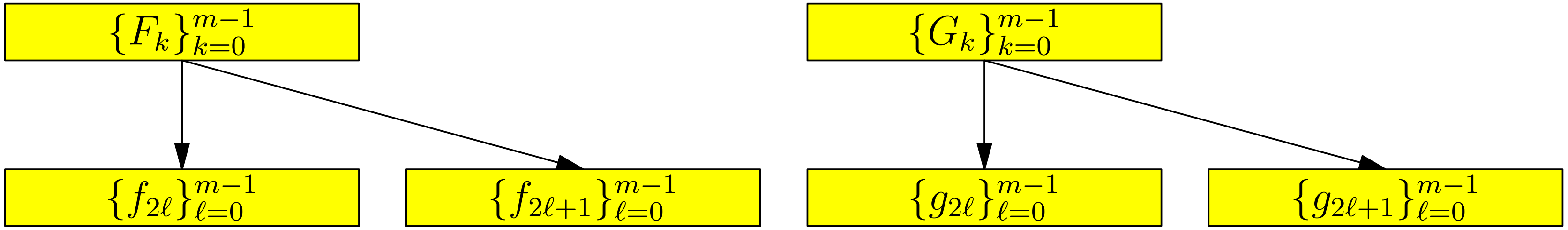
$\{F_k\}_{k=0}^{m-1}$

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

- This in-place convolution requires 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.



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

$\{f_{2\ell}\}_{\ell=0}^{m-1}$

$\{f_{2\ell+1}\}_{\ell=0}^{m-1}$

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

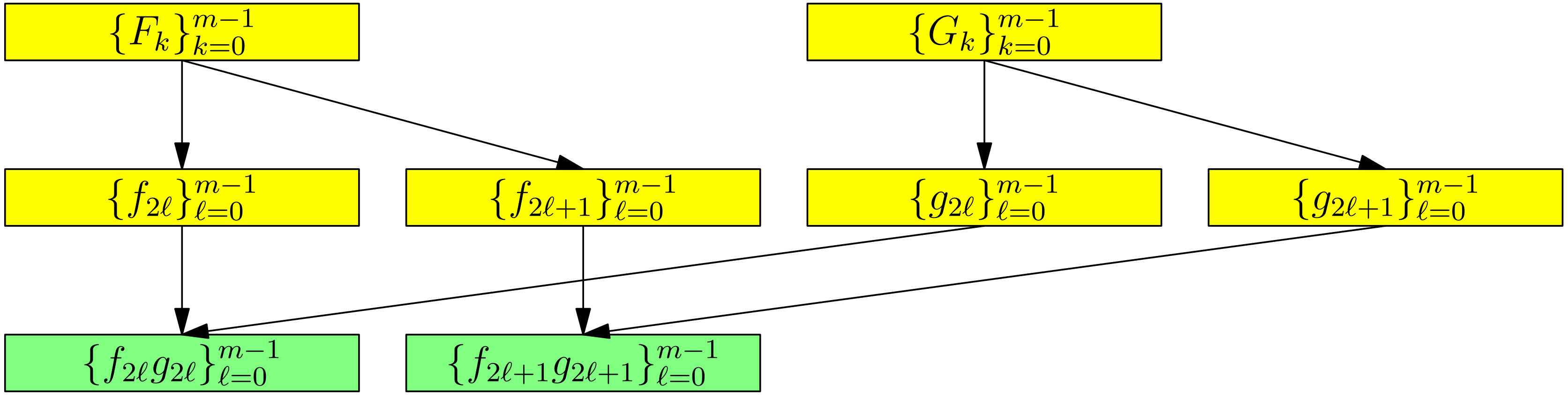
$\{g_{2\ell}\}_{\ell=0}^{m-1}$

$\{g_{2\ell+1}\}_{\ell=0}^{m-1}$

- This in-place convolution requires 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.



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

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

$\{f_{2\ell}\}_{\ell=0}^{m-1}$

$\{f_{2\ell+1}\}_{\ell=0}^{m-1}$

$\{g_{2\ell}\}_{\ell=0}^{m-1}$

$\{g_{2\ell+1}\}_{\ell=0}^{m-1}$

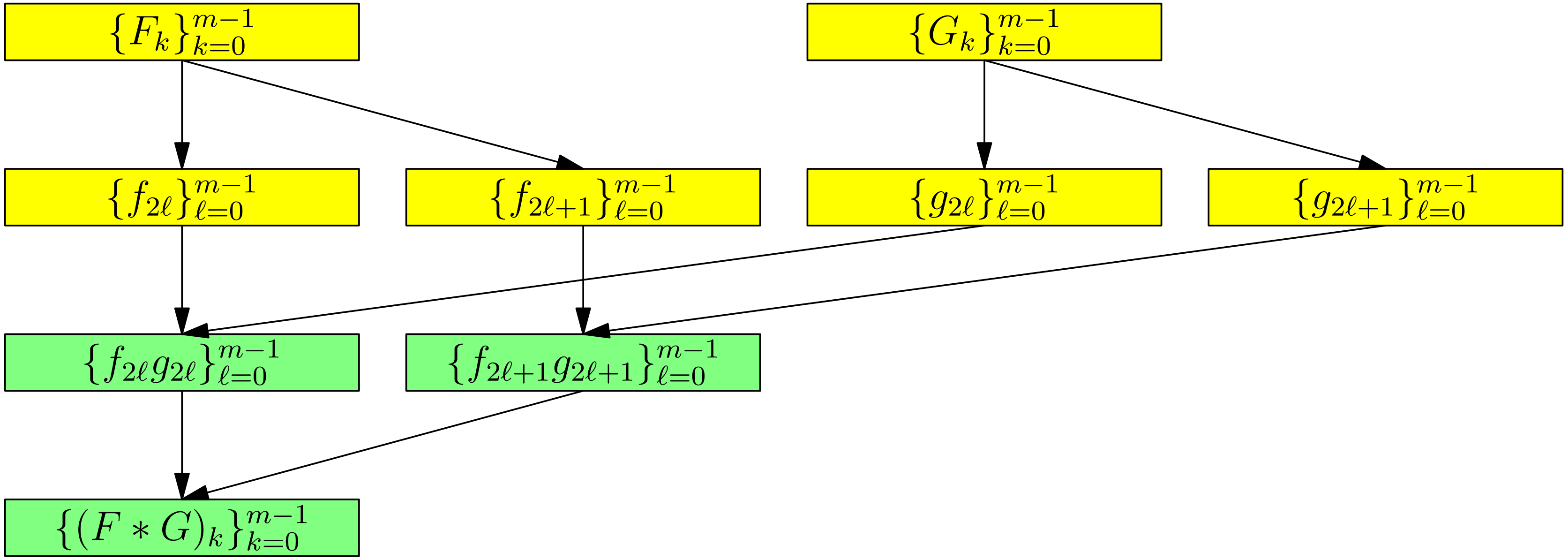
$\{f_{2\ell}g_{2\ell}\}_{\ell=0}^{m-1}$

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

- This in-place convolution requires six out-of-place transforms, thereby avoiding bit reversal at all levels.

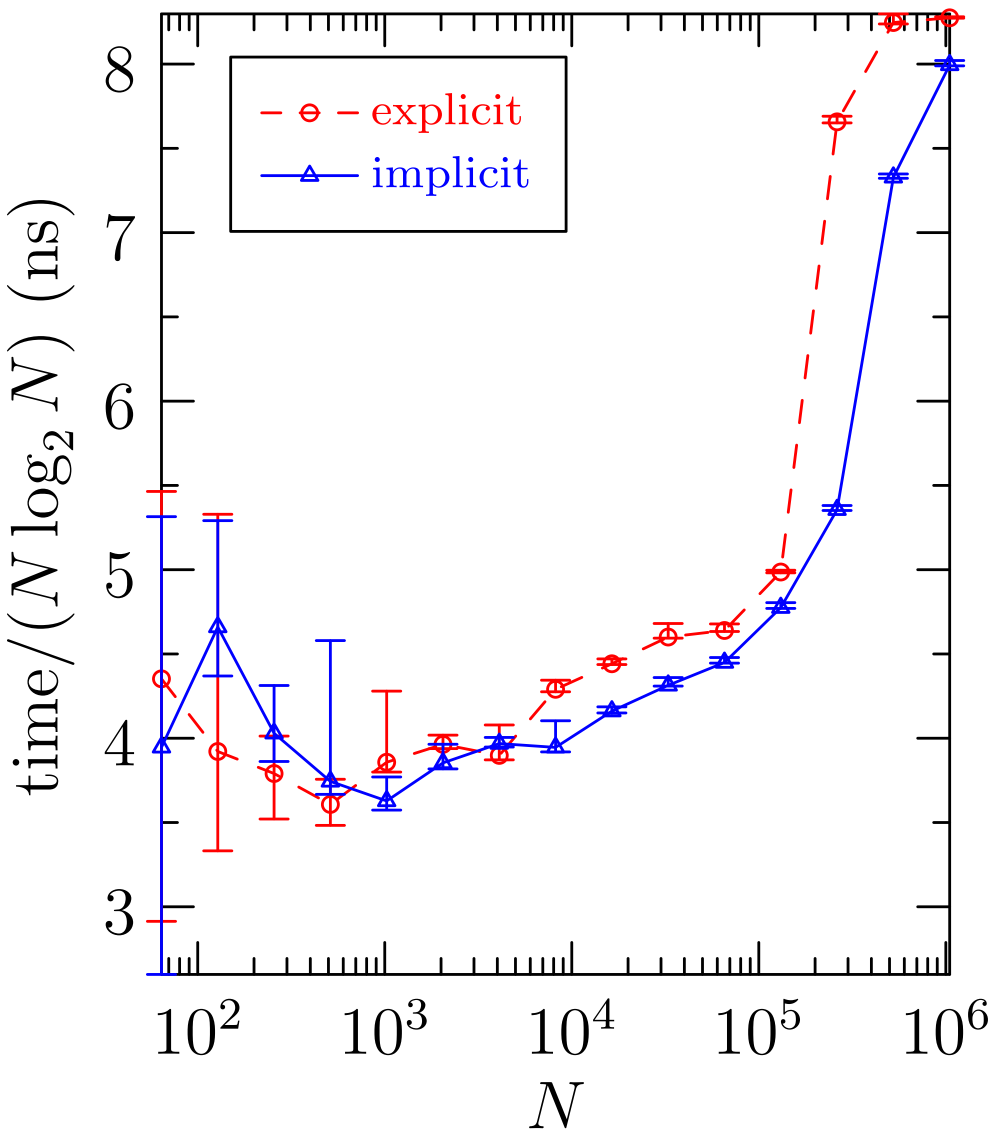
- The computational complexity is $6Km\log_2 m$.

- The numerical error is similar to explicit padding.


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

```
Input: vector f, vector g
Output: vector f
u ← $\texttt{fft}^{-1}(\mathsf{f})$;
v ← $\texttt{fft}^{-1}(\mathsf{g})$;
u ← $\mathsf{u} * \mathsf{v}$;
for $k = 0$ to $m - 1$ do
    $\mathsf{f}[k] \leftarrow \zeta_{2m}^{k}\mathsf{f}[k]$;
    $\mathsf{g}[k] \leftarrow \zeta_{2m}^{k}\mathsf{g}[k]$;
end
v ← $\texttt{fft}^{-1}(\mathsf{f})$;
f ← $\texttt{fft}^{-1}(\mathsf{g})$;
v ← $\mathsf{v} * \mathsf{f}$;
f ← $\texttt{fft}(\mathsf{u})$;
u ← $\texttt{fft}(\mathsf{v})$;
for $k = 0$ to $m - 1$ do
    $\mathsf{f}[k] \leftarrow \mathsf{f}[k] + \zeta_{2m}^{-k}\mathsf{u}[k]$;
end
return f/(2m);
```

Implicit Padding in 1D

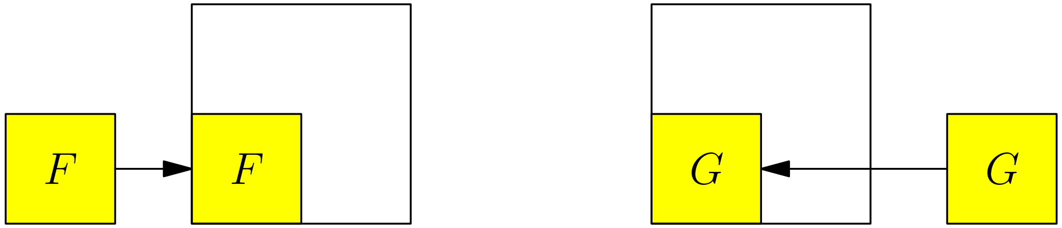


Convolutions in Higher Dimensions

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

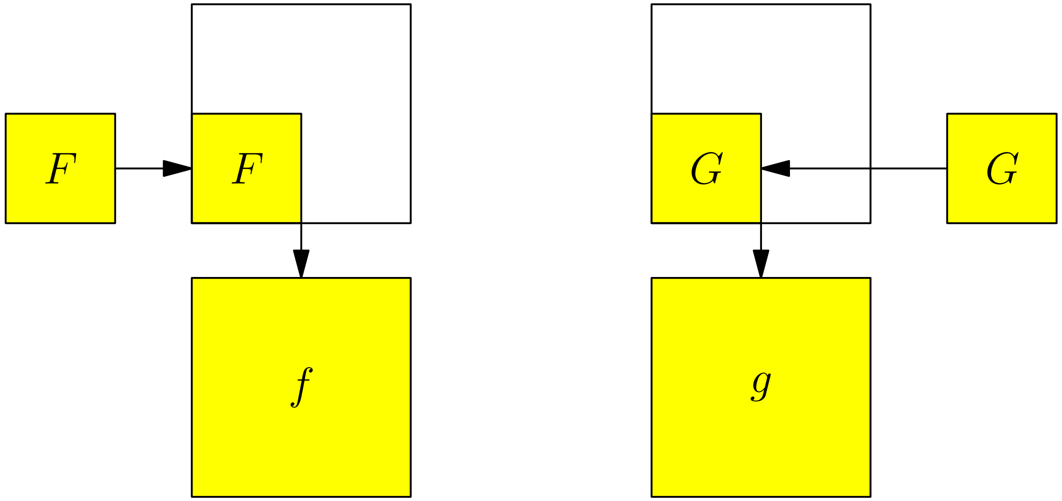
Convolutions in Higher Dimensions

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



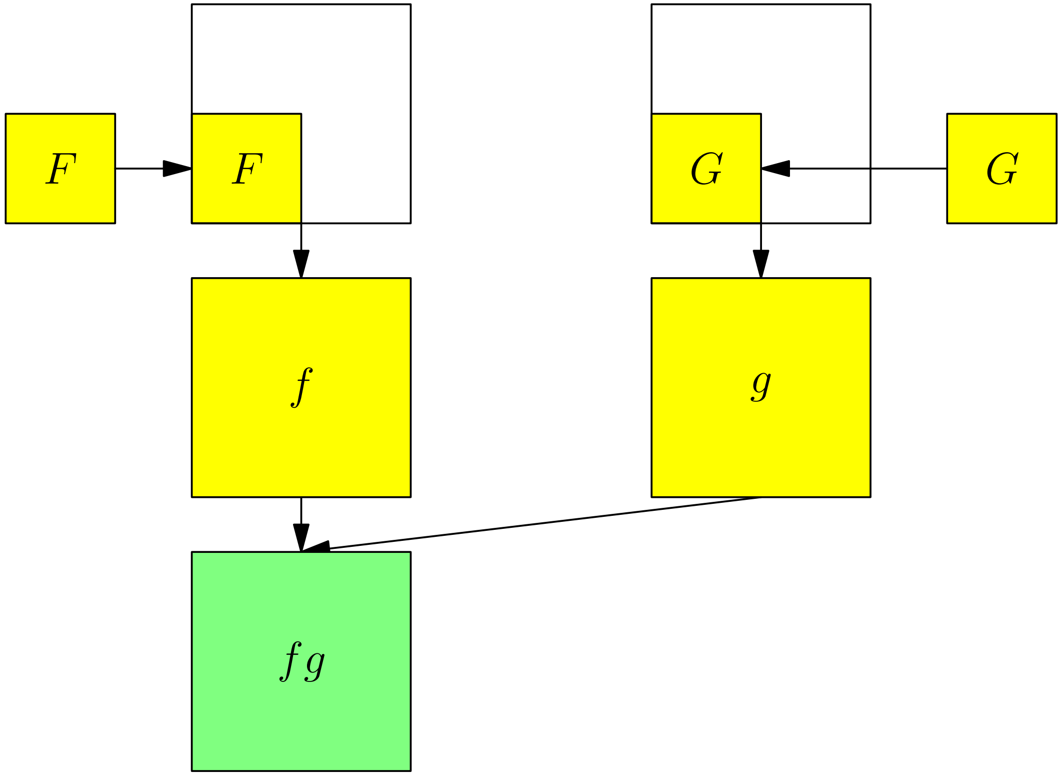
Convolutions in Higher Dimensions

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



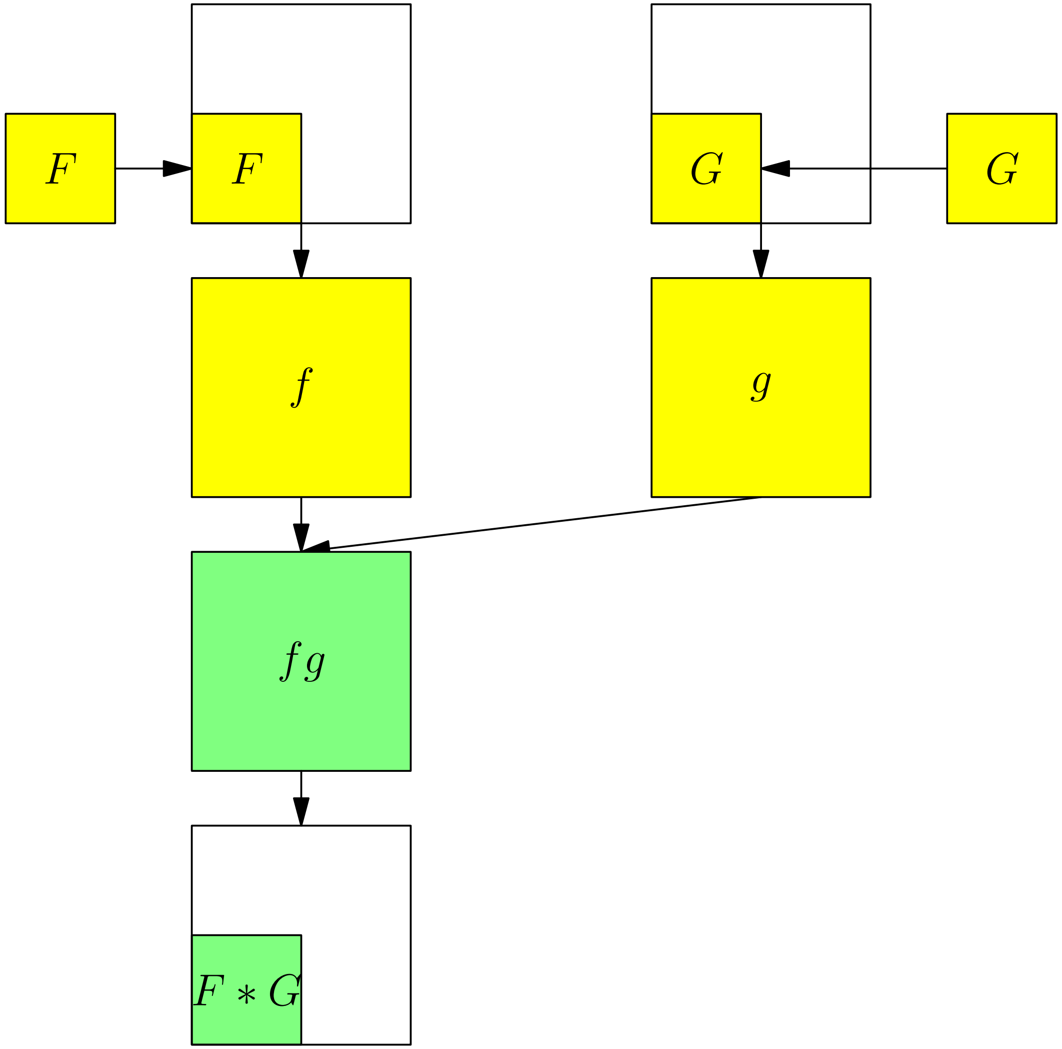
Convolutions in Higher Dimensions

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



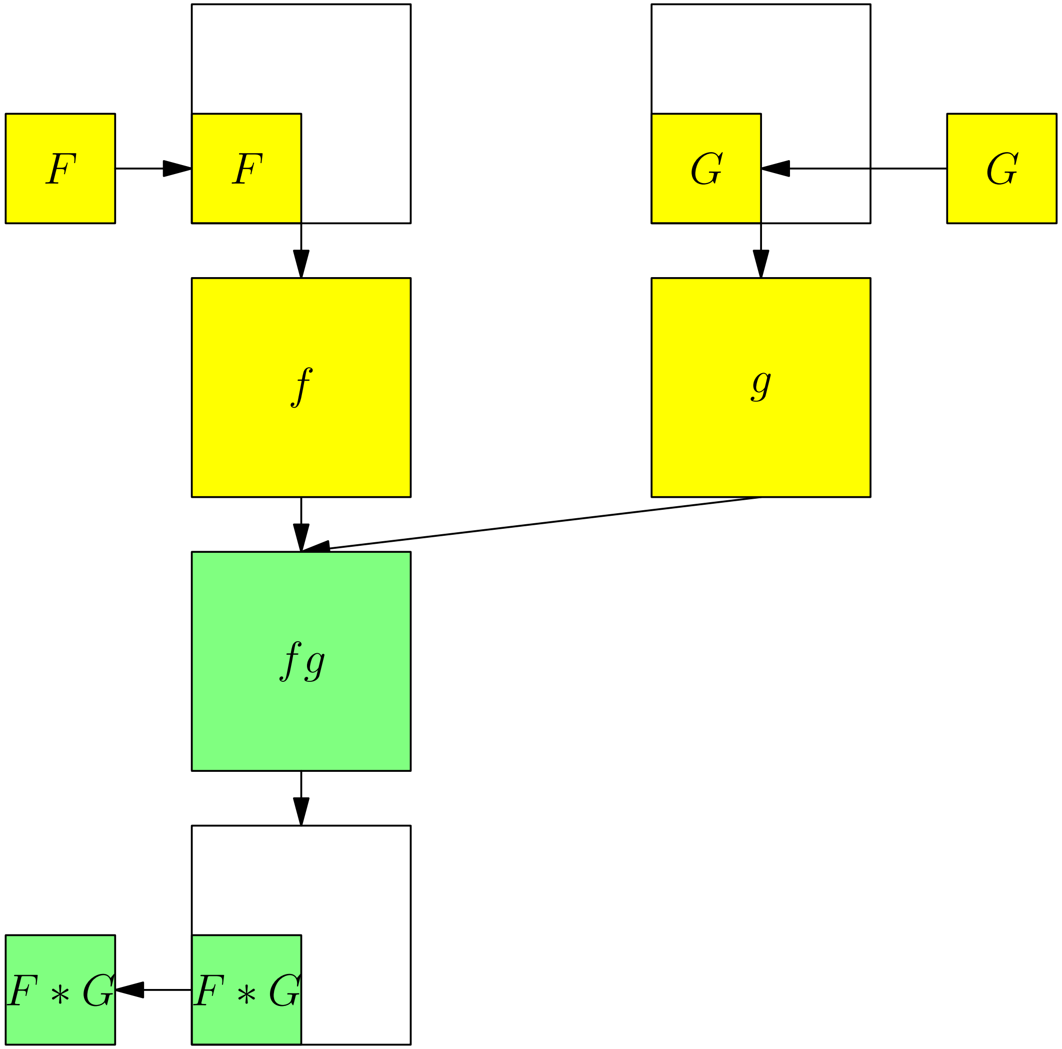
Convolutions in Higher Dimensions

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



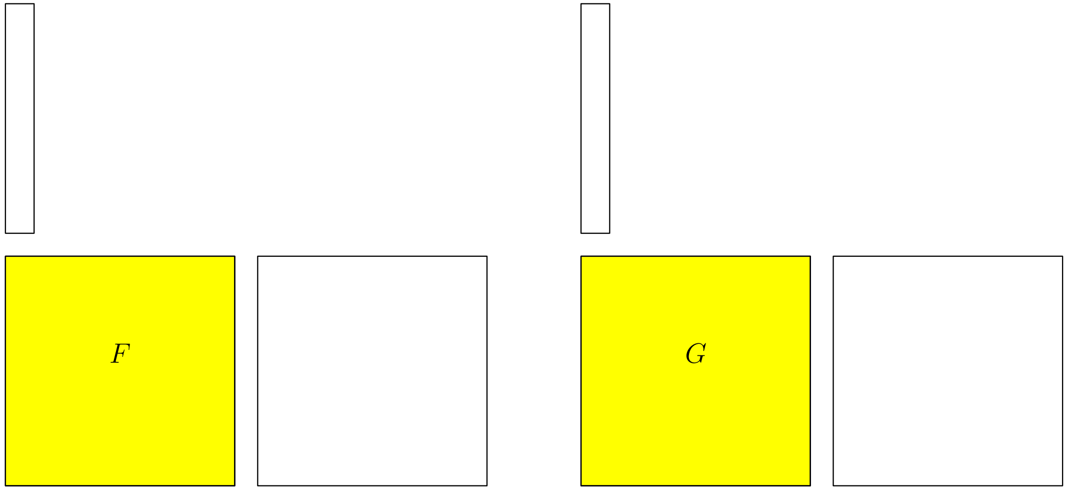
Convolutions in Higher Dimensions

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



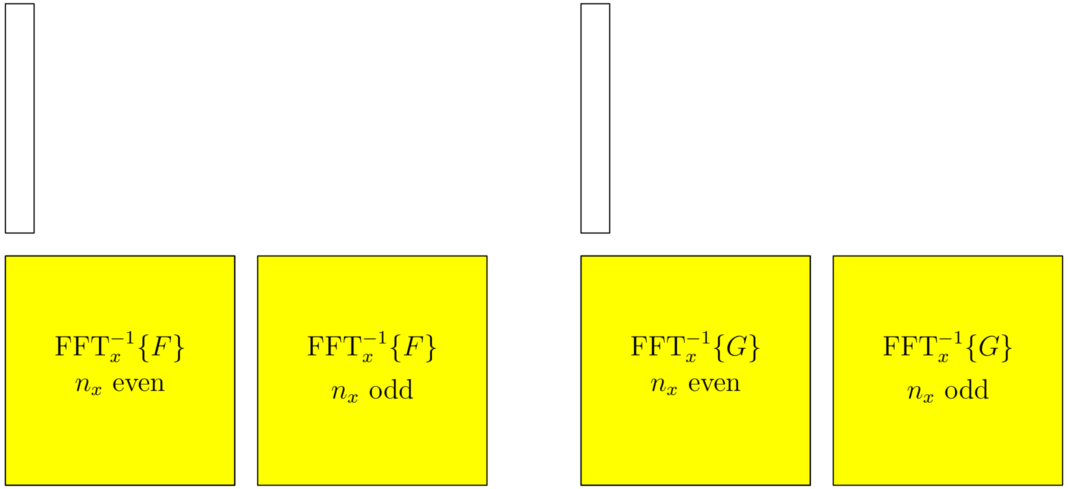
Implicit Padding in 2D

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



Implicit Padding in 2D

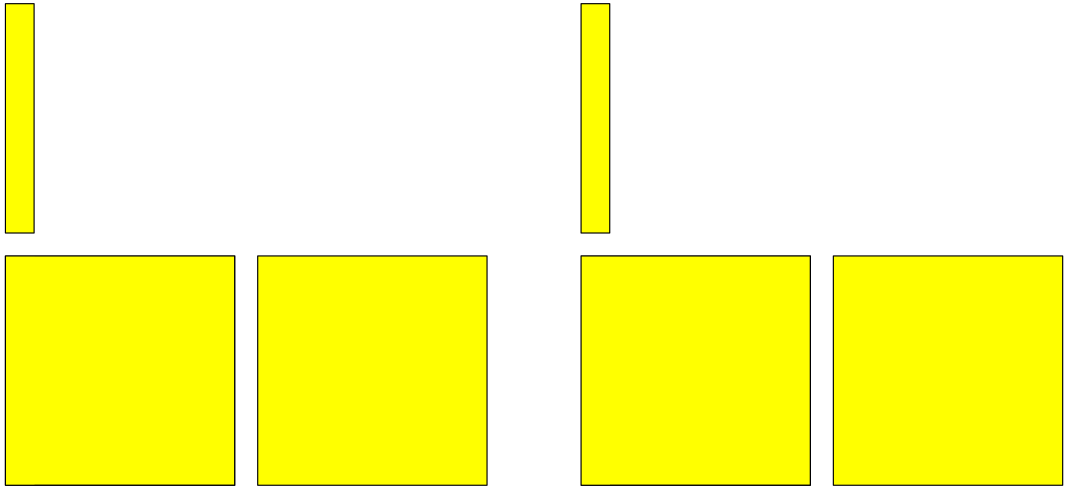
- Extra work memory need not be contiguous with the data.



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

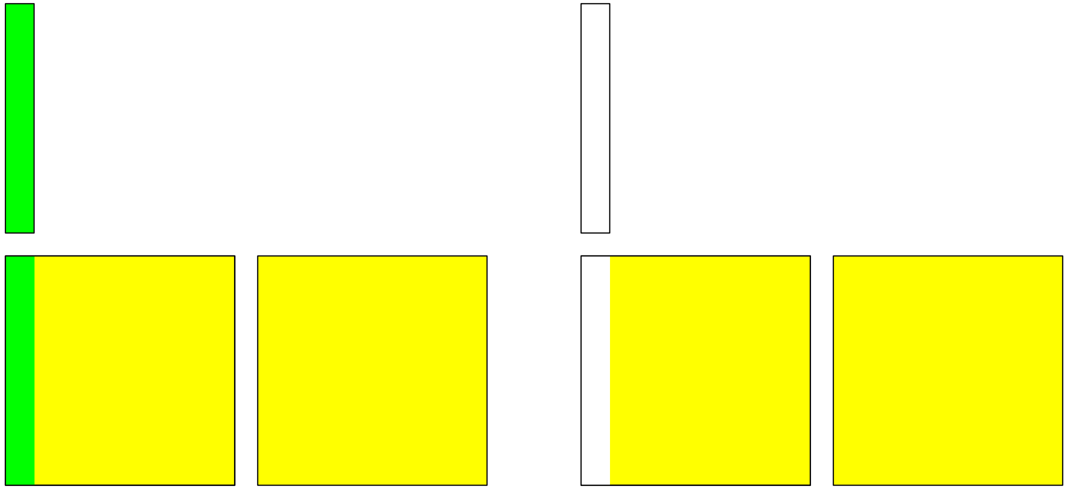
Implicit Padding in 2D

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



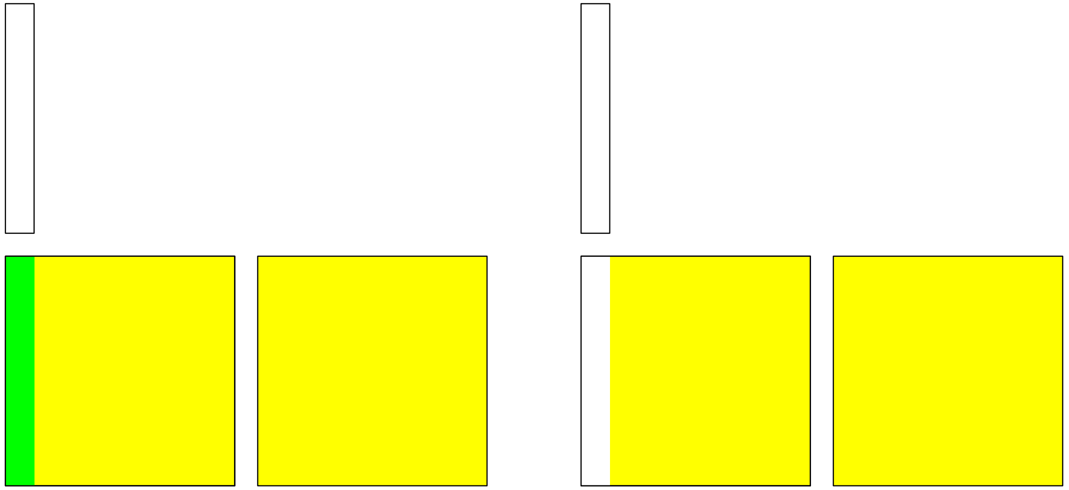
Implicit Padding in 2D

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



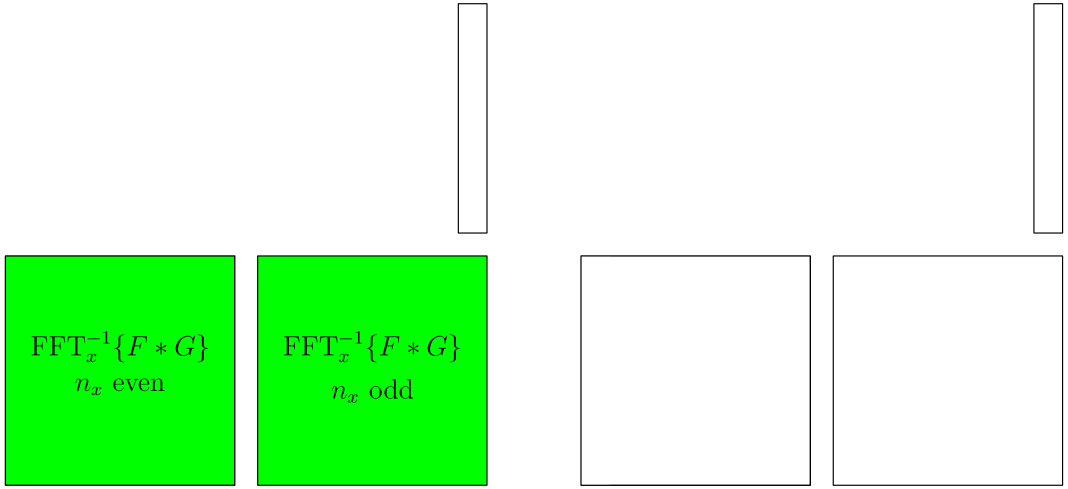
Implicit Padding in 2D

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



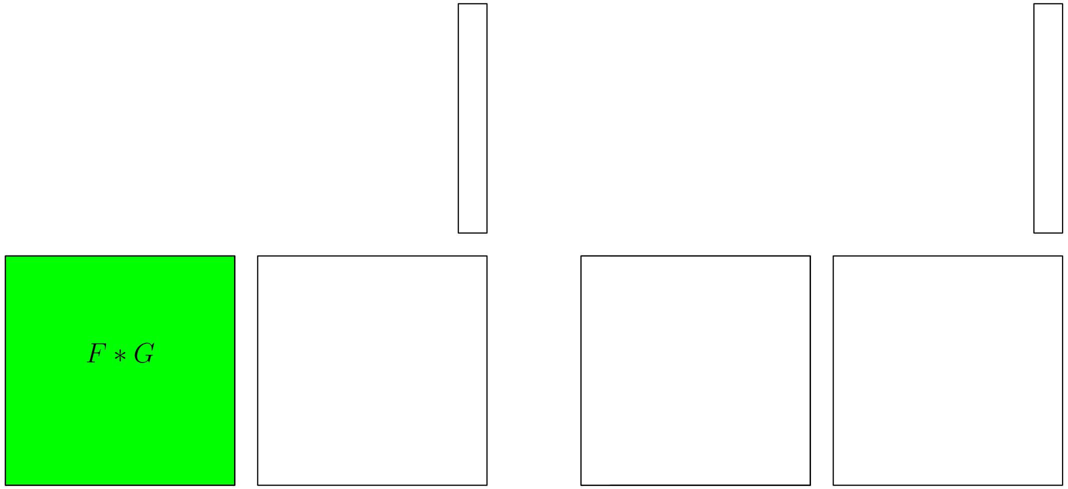
Implicit Padding in 2D

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

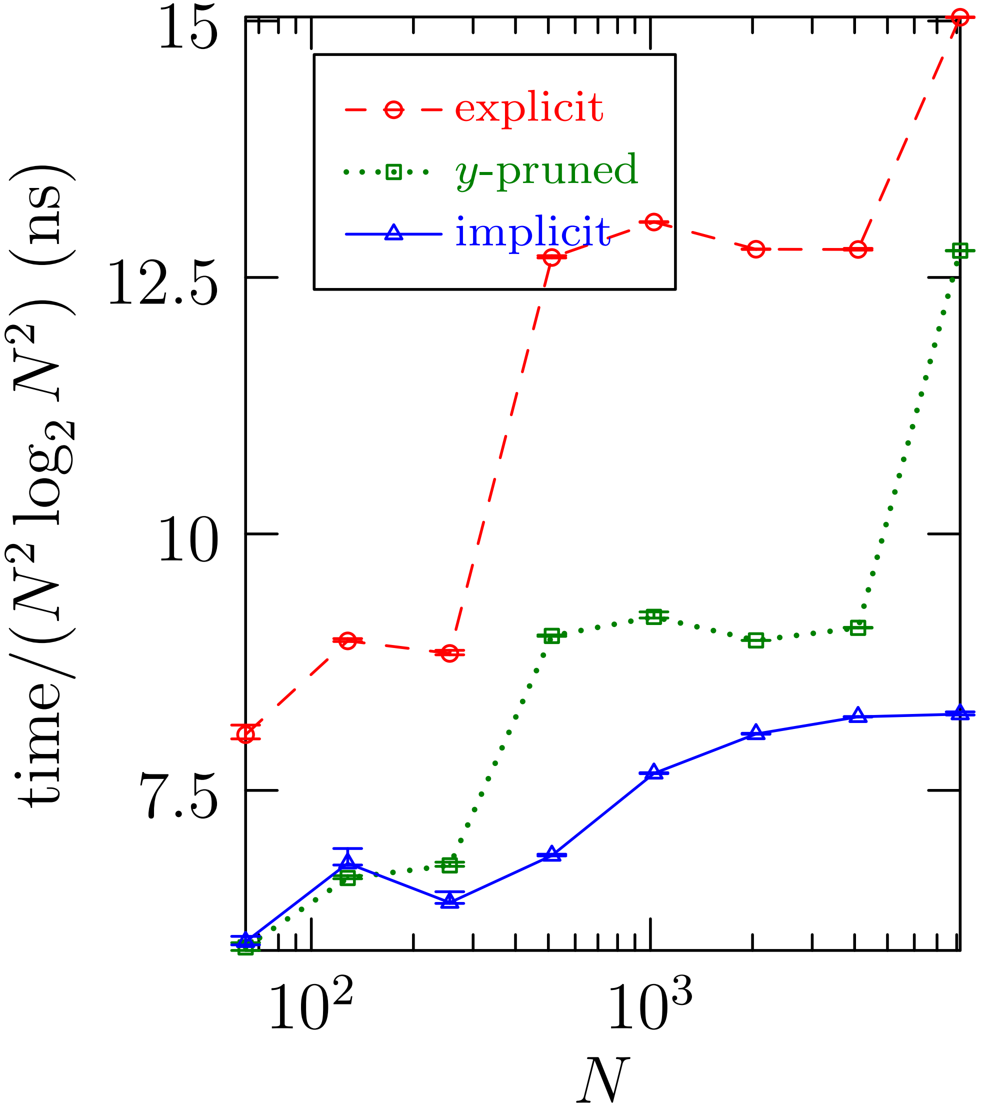

$\mathrm{FFT}_x^{-1}\{F * G\}$
n_x even
$\mathrm{FFT}_x^{-1}\{F * G\}$
n_x odd

Implicit Padding in 2D

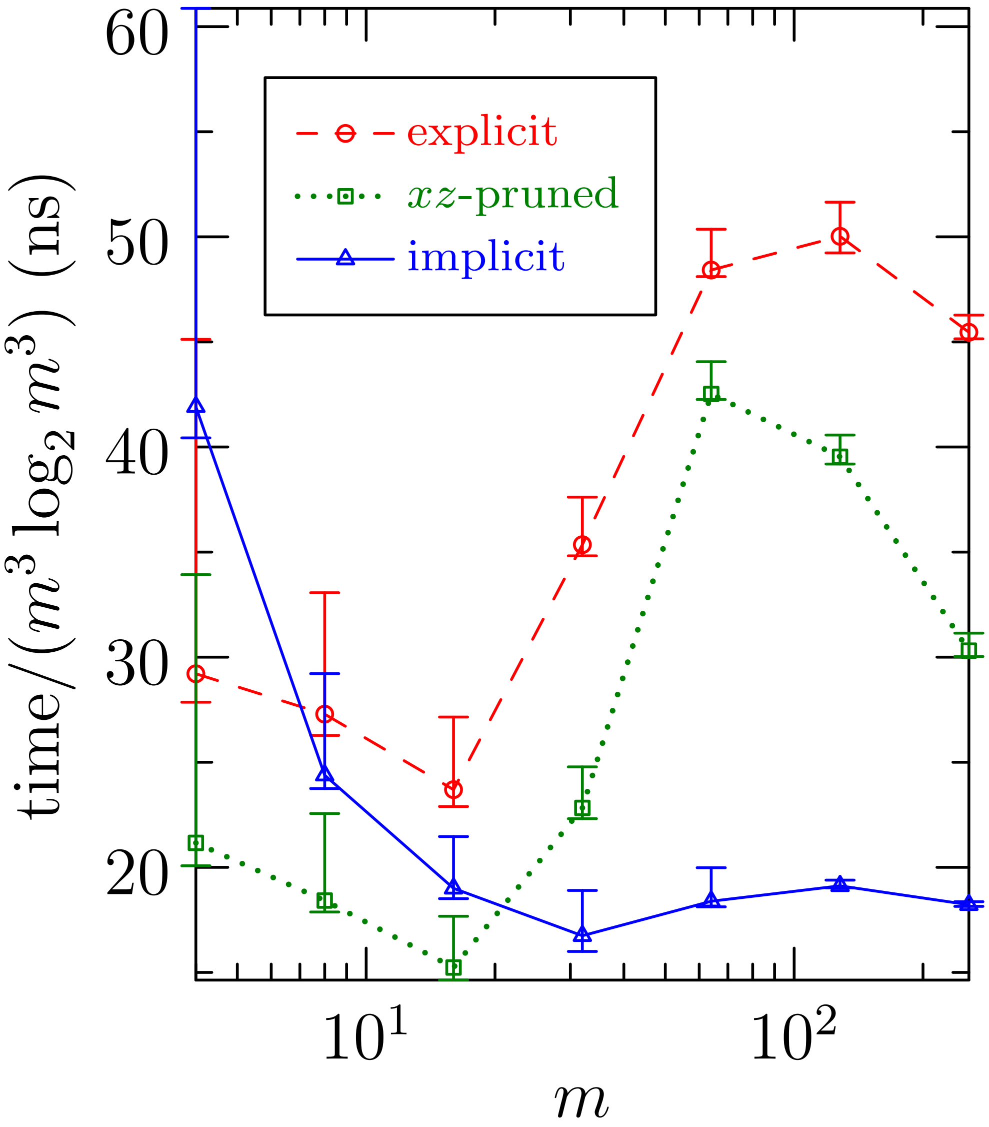
- Extra work memory need not be contiguous with the data.



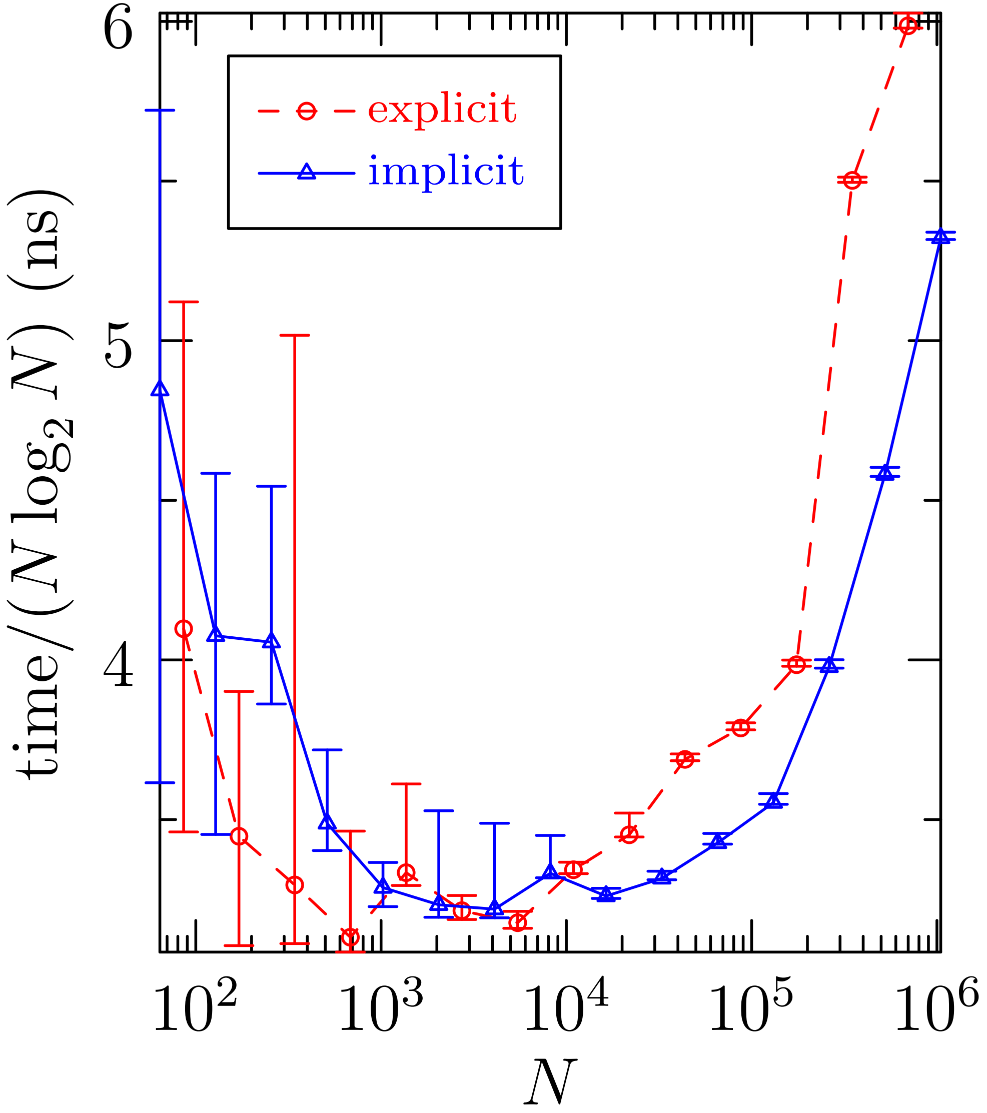
Implicit Padding in 2D



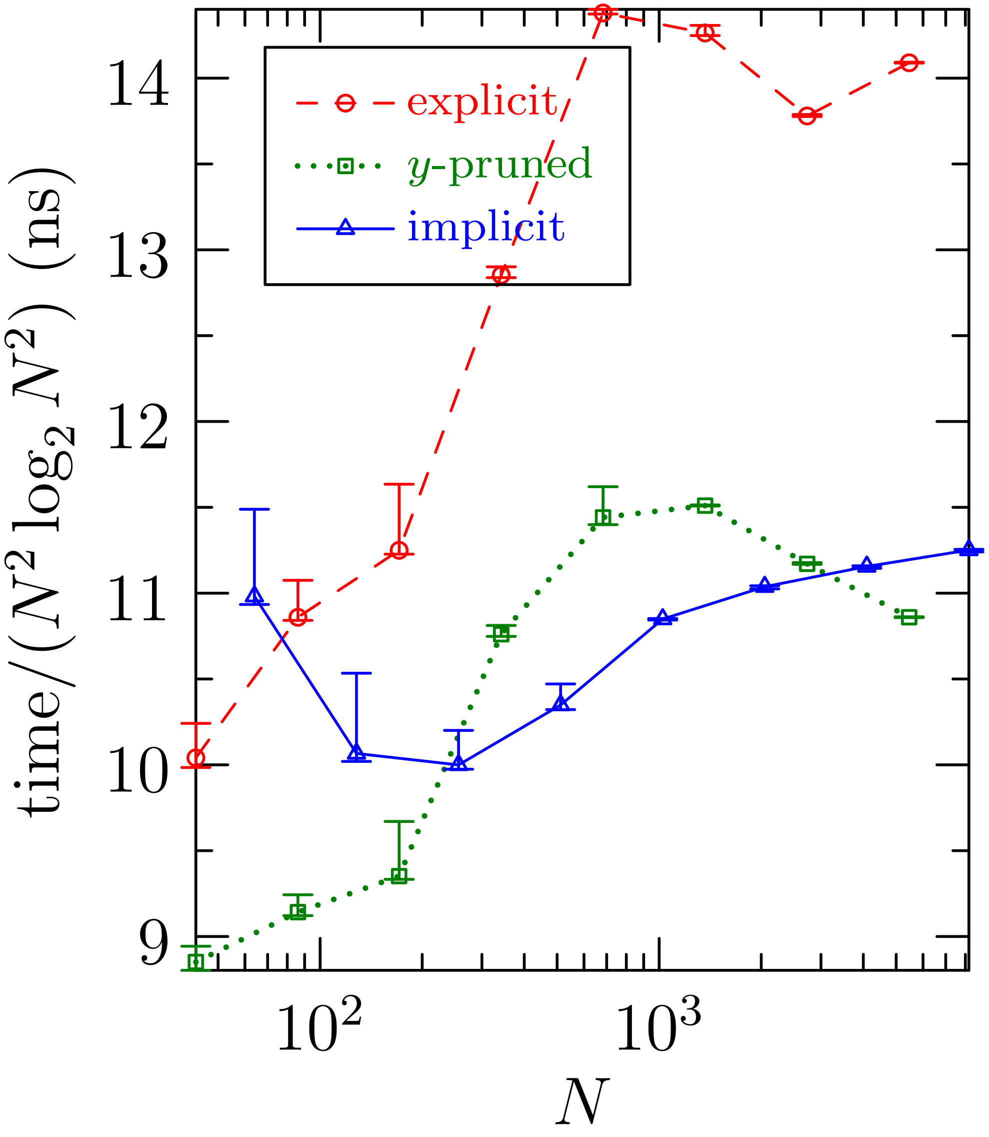
Implicit Padding in 3D



Implicit Hermitian Centered Padding in 1D



Implicit Hermitian Centered Padding in 2D

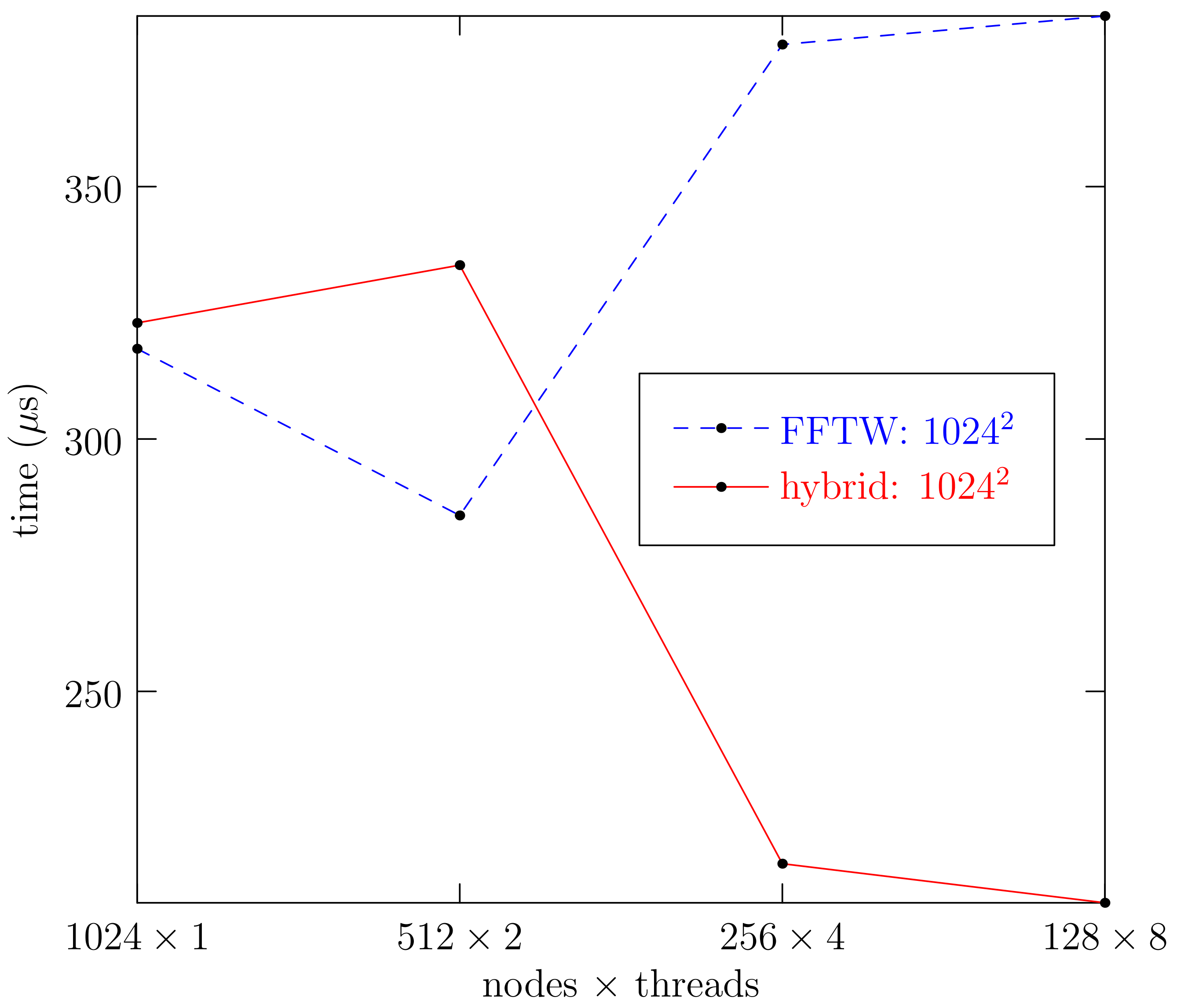


Parallelization

- Our implicit and explicit convolution routines have been multithreaded for shared-memory architectures.
- Parallel generalized slab/pencil model implementations have recently been developed for distributed-memory architectures (available in svn repository and upcoming 1.14 release).
- The key bottleneck is the distributed matrix transpose.
- We have compared several distributed matrix transpose algorithms, both blocking and nonblocking, under both pure MPI and hybrid MPI/OpenMP architectures.
- Local transposition is not required within a single MPI node.
- Hybrid MPI/OpenMP offers a larger communication block size than pure MPI for matrix transposition.

- Hybrid MPI/OpenMP is sometimes more efficient (by a factor of 2) than pure MPI for computing distributed matrix transposes [Bowman & Roberts 2013].

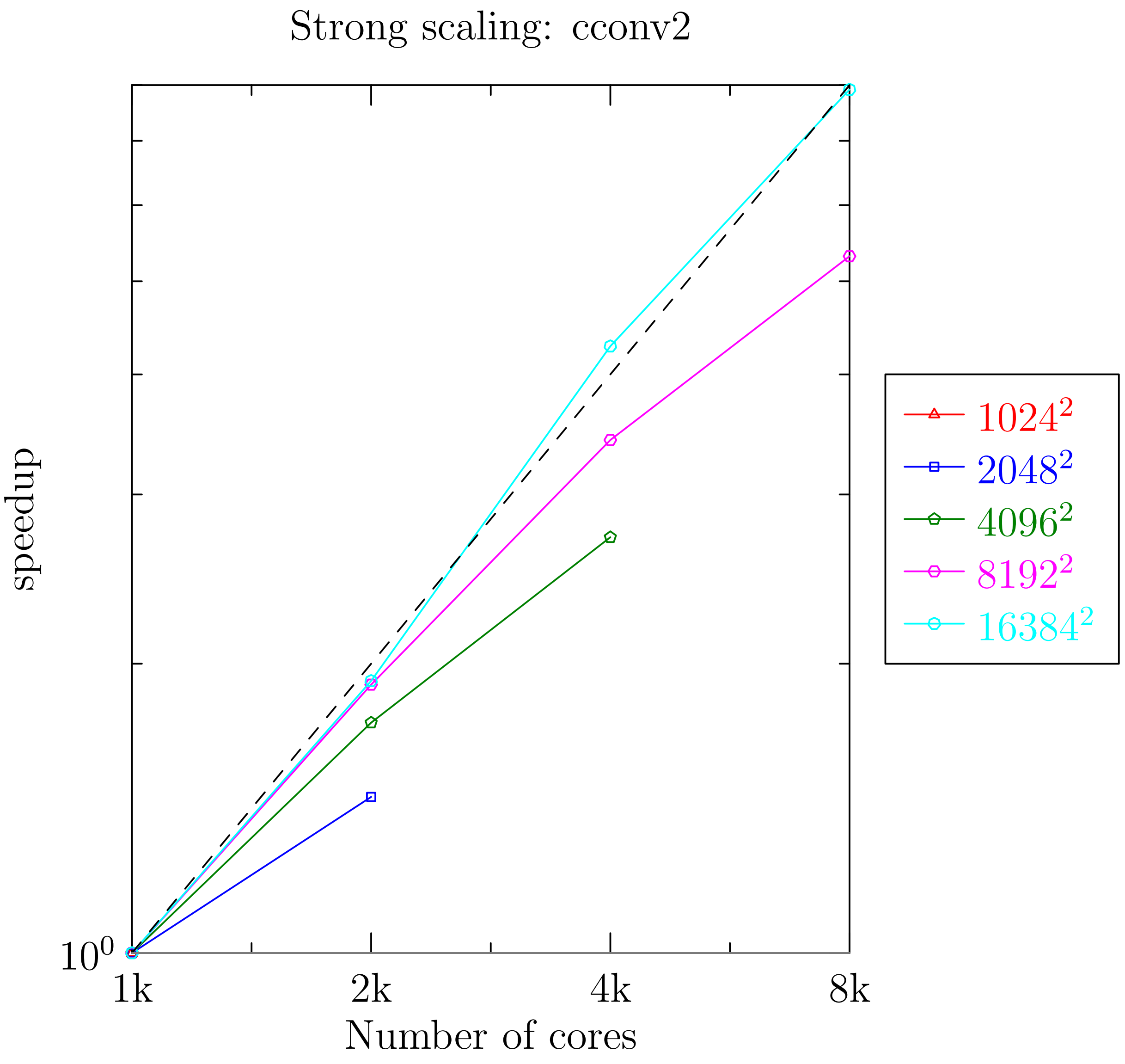
Matrix Transpose: Optimal Number of Threads



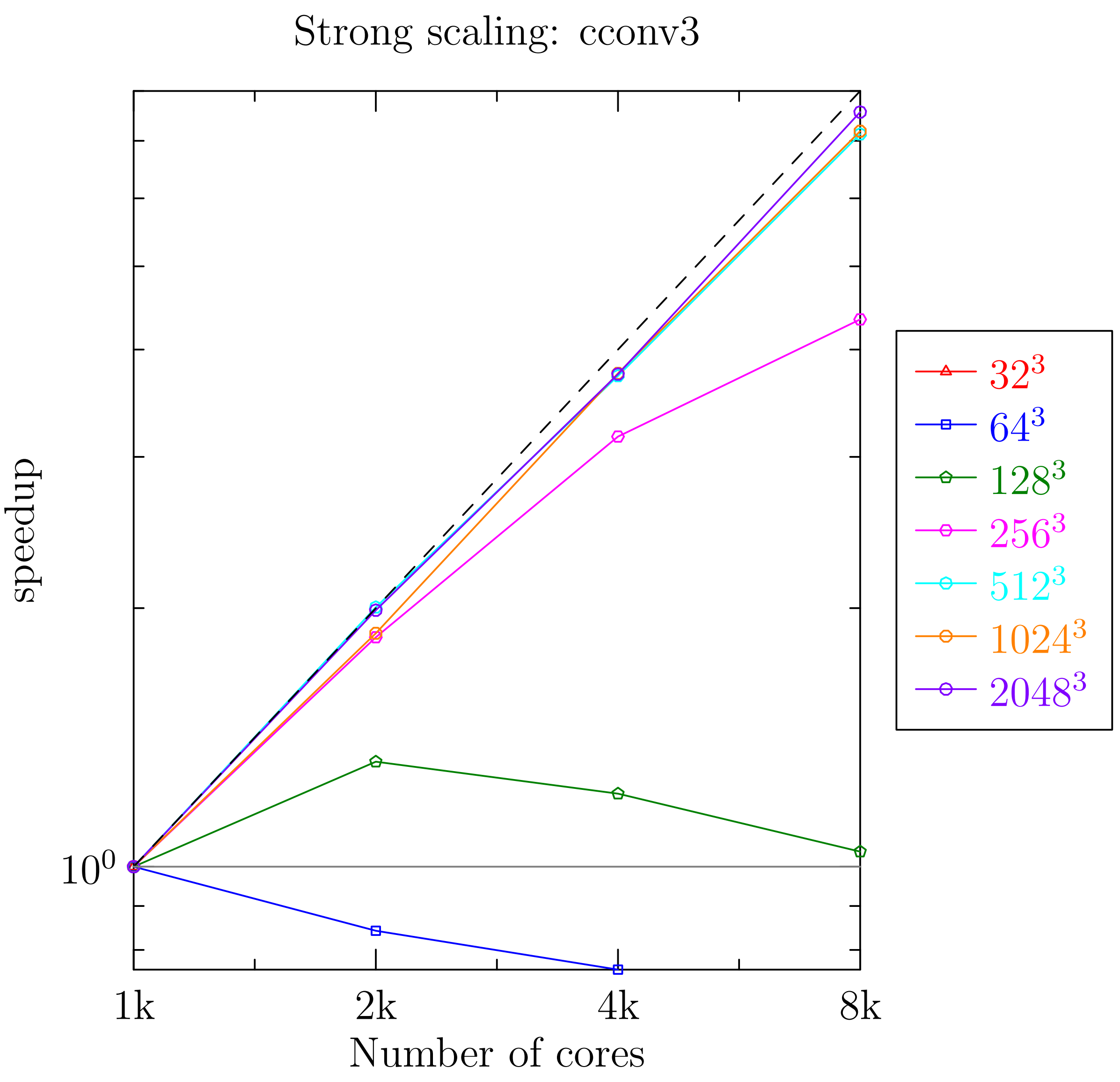
Advantages of Hybrid MPI/OpenMP

- Smaller problems sizes to be distributed over a large number of processors;
- More slab-like than pencil-like model; this reduces the size of or even eliminates the need for the second transpose.
- Overlapping computation with communication can yield a 10% speedup for 3D implicitly dealiased convolutions, where a natural parallelism exists between communication and computation.

Pure MPI Scaling of 2D Implicit Convolutions



Pure MPI Scaling of 3D Implicit Convolution



Conclusions

- Dealiasing of pseudospectral simulations is essential for stability and accuracy.
- Partial dealiasing via a Fourier filter is neither necessary nor desirable.
- With the advent of implicit dealiasing, partial Fourier filtering is no longer more efficient than full dealiasing.
- Memory savings: in d dimensions implicit padding asymptotically uses $1/2^{d-1}$ of the memory required by conventional explicit padding.
- The factor of 2 speedup with implicit dealiasing is due to increased data locality.
- Highly optimized and parallelized implicit dealiasing routines have been implemented as a software layer `FFTW++` on top of the `FFTW` library and released under the Lesser GNU Public License: `http://fftwpp.sourceforge.net/`

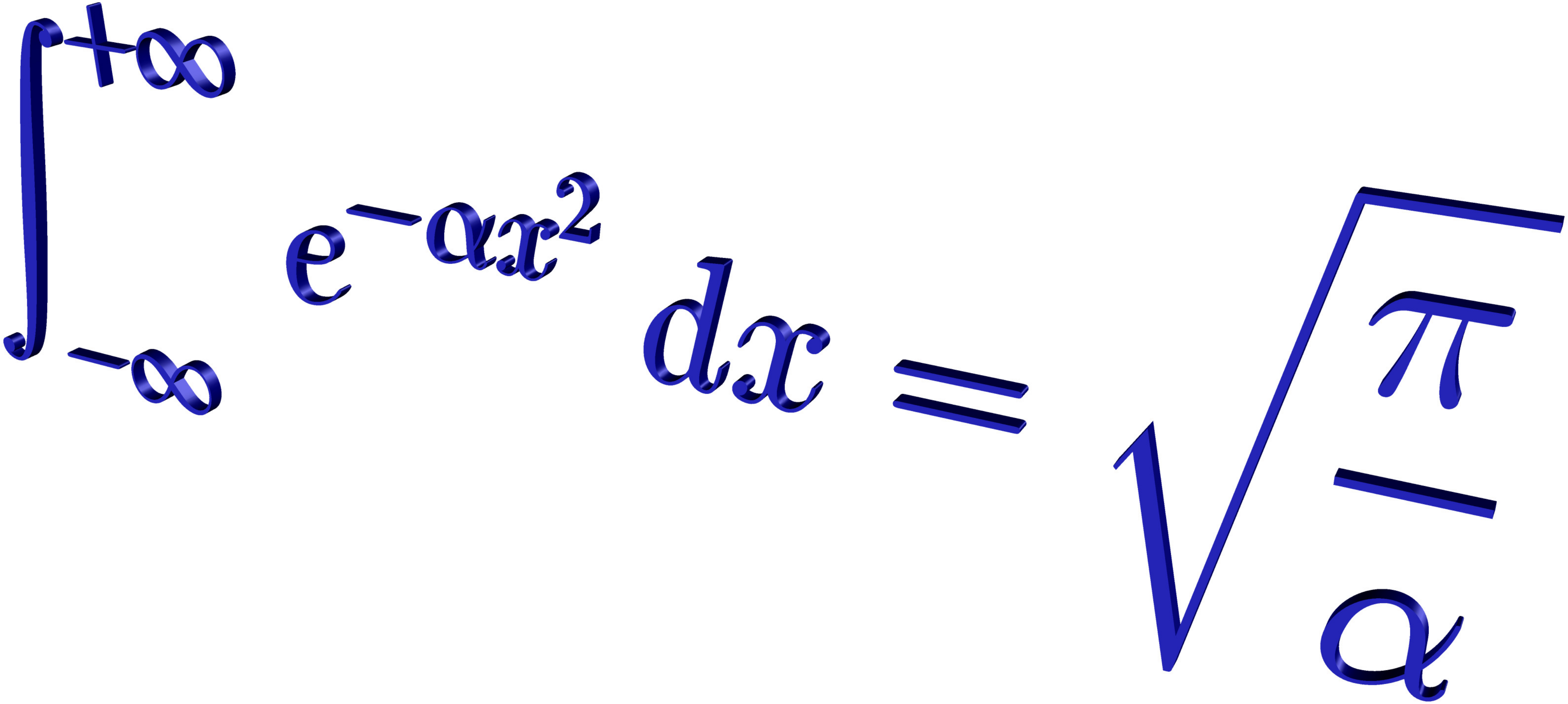
- Writing a high-performance dealiased pseudospectral code is now a relatively straightforward exercise!

Asymptote: 2D & 3D Vector Graphics Language

Andy Hammerlindl, John C. Bowman, Tom Prince

`http://asymptote.sf.net`

Asymptote Lifts TEX to 3D



$$\int_{-\infty}^{+\infty} e^{-\alpha x^2}\,dx = \sqrt{\frac{\pi}{\alpha}}$$

`http://asymptote.sf.net`

Acknowledgements: Orest Shardt (U. Alberta)

References

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

[Bowman & Roberts 2013] J. C. Bowman & M. Roberts, "Adaptive matrix transpose algorithms for distributed multicore processors," in *Proceedings of the International Conference on Applied Mathematics, Modeling and Computational Science*, edited by M. Cojocaru, Springer, 2013.

[Cooley & Tukey 1965] J. W. Cooley & J. W. Tukey, Mathematics of Computation, **19**:297, 1965.

[Franchetti & Puschel 2009] F. Franchetti & M. Puschel, "Generating high performance pruned FFT implementations," in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 549–552, IEEE Computer Society, 2009.

[Frigo & Johnson] M. Frigo & S. G. Johnson, Pruned FFTs and `FFTW`, `http://www.fftw.org/pruned.html`.

[Gauss 1866] C. F. Gauss, "Nachlass: Theoria interpolationis methodo nova tractata," in *Carl Friedrich Gauss Werke*, volume 3, pp. 265–327, Königliche Gesellschaft der Wissenschaften, Göttingen, 1866.

[Hou & Li 2007] T. Y. Hou & R. Li, Journal of Computational Physics, **226**:379, 2007.

[Markel 1971] J. Markel, IEEE Transactions on Audio and Electroacoustics, **19**:305, 1971.

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

[Patterson Jr. & Orszag 1971] G. S. Patterson Jr. & S. A. Orszag, Physics of Fluids, **14**:2538, 1971.

[Phillips 1959] N. A. Phillips, The Atmosphere and the Sea in Motion, p. 501, 1959.

[Sorensen & Burrus 1993] H. Sorensen & C. Burrus, IEEE Transactions on Signal Processing, **41**:1184, 1993.