The peril of the petascale: looming challenges in large-scale computational science

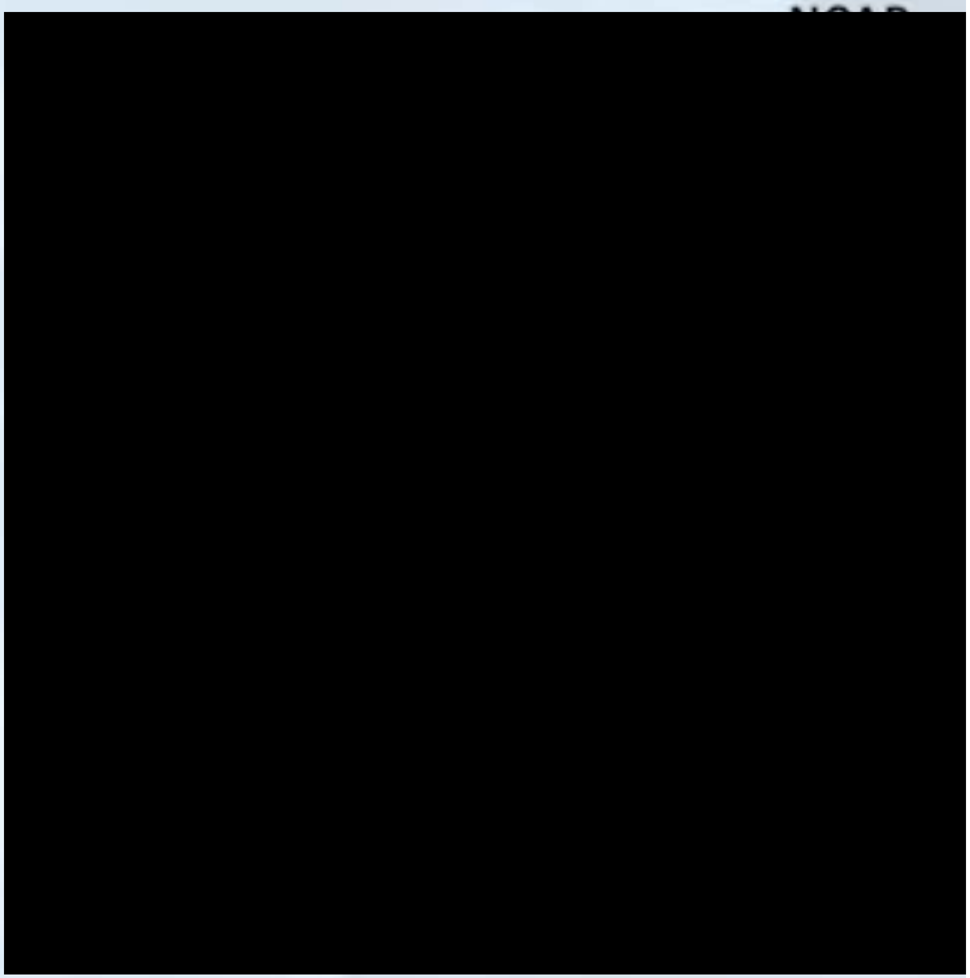
John Clyne, Alan Norton
National Center for Atmospheric Research

Acknowledgments: Mark Rast (CU), Bill Smyth (U. of Oregon), Pablo Mininni, (NCAR)

Pioneers at the dawn of *terascale* computing

Compressible thermal starting plume

- 2003 - Simulation
  - 6 months run time
  - 504x504x2048 grid
  - 5 variables (u,v,w,rho,temp)
  - ~500 time steps saved
  - 9 TBs storage (4GBs/var/timestep)
  - 112 IBM SP RS/6000 processors
- 2004 - Post-processing
  - 3 months
  - 3 derived variables (vorticity)
- 2004 - Analysis
  - **Abandoned!!!**
- 2006 - Analysis Resumed
- 2007 - Published
  - *New Journal of Physics*



Mark Rast, NCAR/CU, 2003

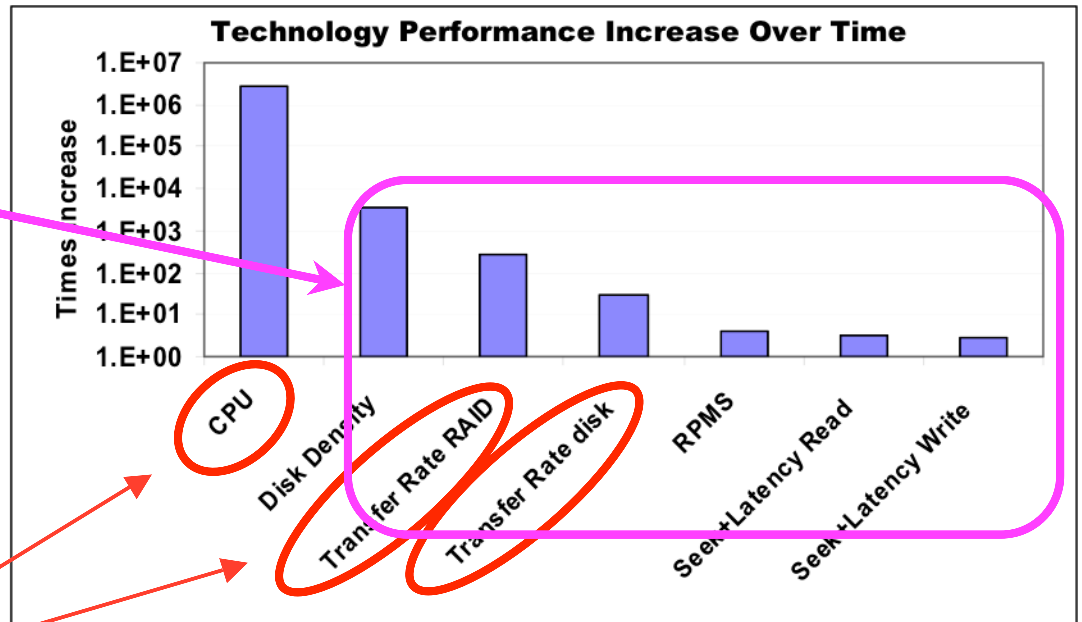
The path to petaflop computing: performance increases from 1977 to 2006

NCAR

Moore's Law does not apply to all computing technologies!!!

Orders of magnitude difference between improvements in CPU speed and IO bandwidth

Disparity between compute and IO is increasing rapidly



Increases in processor speed and disk density have both grown at alarming rates while disk transfer rates have only grown modestly and disk agility has hardly improved at all.

High End Computing Revitalization Task Force (HEC-RTF), Inter Agency Working Group (HEC-IWG) File Systems and I/O Research Workshop 5

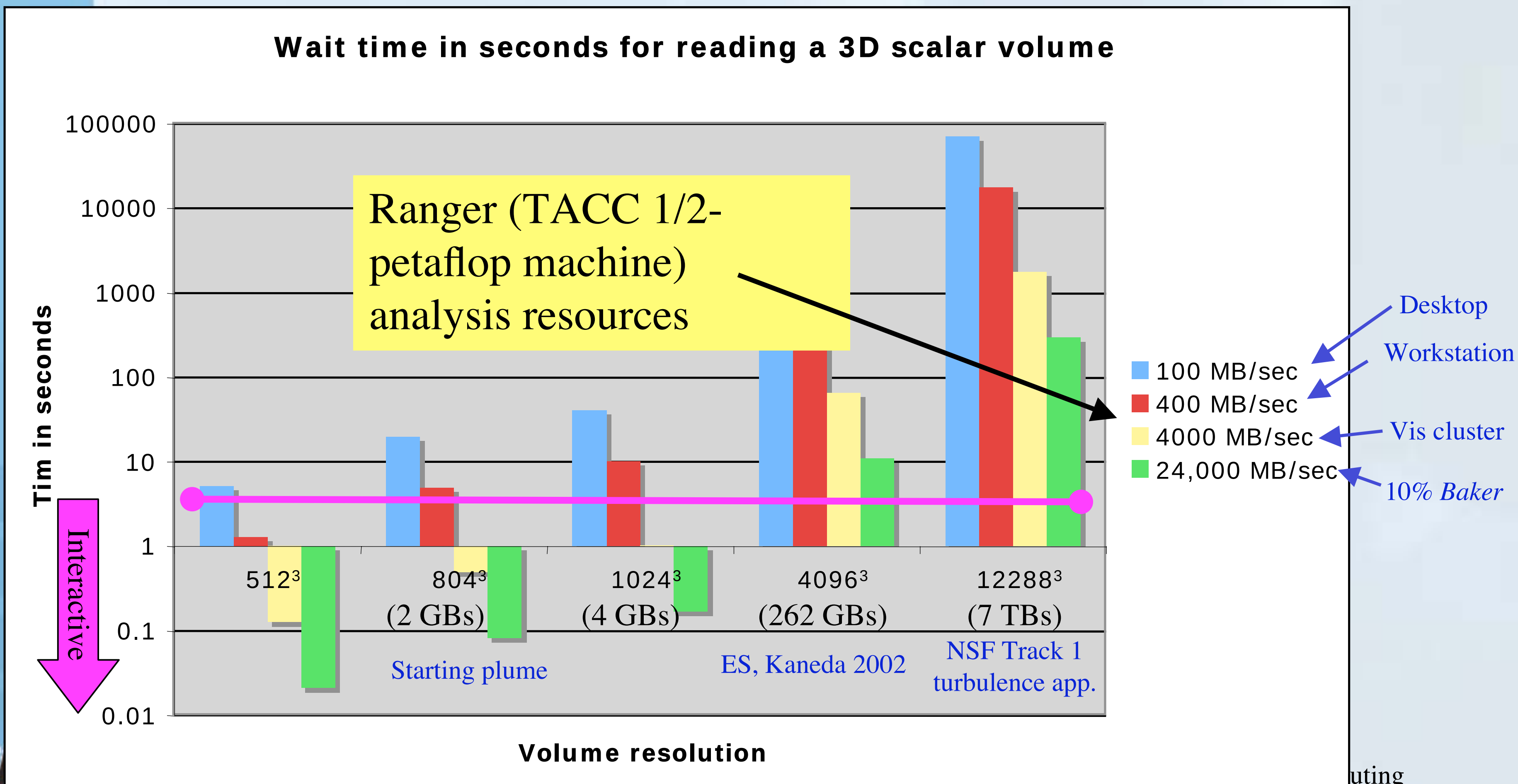
What is meant by *interactive* analysis?

Mark Rast, 2005

Definition: A system is *interactive* if the time between a user event and the response to that event is short enough maintain my full attention

If the response time is…

- 1-5 seconds : I'm engaged
- 5-60 seconds : I'm tapping my foot
- 1-3 minutes : I'm reading email
- > 3 minutes : I've forgotten why I asked the question!



Peril of the petascale…

We are in danger of computing more data than we can possibly examine in **depth**!

1. Data sets may be too large to store
2. IO bandwidth bottlenecks may prohibit **interactive** processing

Is the situation hopeless? Maybe not!

Many useful analysis operations can be performed without:

- Full data fidelity
  - (e.g. 64-bit precision, native solution sampling)
- Full data domain
  - Regions of interest typically are localized spatially and temporally

Data reduction needed

- Data model supporting:
  - Speed/quality tradeoffs (progressive data access)
  - Efficient region subsetting
- Tools that can effectively operate on data model

Discrete Wavelet Transforms

- Discrete Fourier transform

$$f(t) = \frac{1}{N}\sum_{n=0}^{N-1} a_t e^{j2\pi nt/N} \qquad (0 \le t \le N-1)$$

- Discrete Wavelet Transform

$$f(t) = \sum_k c(k)\phi_k(t) + \sum_k \sum_{j=0}^{\log_2 N} d_j(k)\,\psi_{j,k}(t)$$

Scaling term (coarse representation of signal)

Detail term (high frequency components of signal)

$$\phi(t) = \sum_k h_\phi(k)\sqrt{2}\phi(2t-k), \quad k \in Z \quad \text{scaling function}$$

$$\psi(t) = \sum_k h_\psi(k)\sqrt{2}\phi(2t-k), \quad k \in Z \quad \text{wavelet function}$$

- Properties
  - Multiresolution representation
  - Efficient: Linear time complexity
  - Adaptable: Can represent functions with discontinuities, bounded domains, and arbitrary topology
  - Time frequency localization: Many coefficients are zero or close to zero

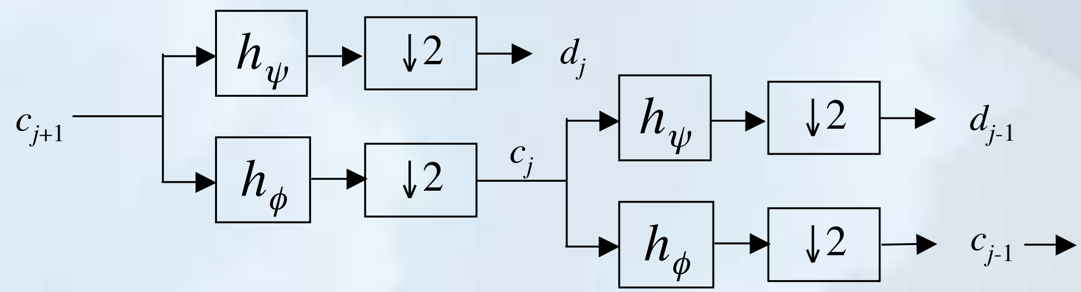
Computing wavelet transforms

1D Forward Transform

$$c_j = \sum_m h_\phi(m-2k)c_{j+1}(m)$$

$$d_j = \sum_m h_\psi(m-2k)c_{j+1}(m)$$

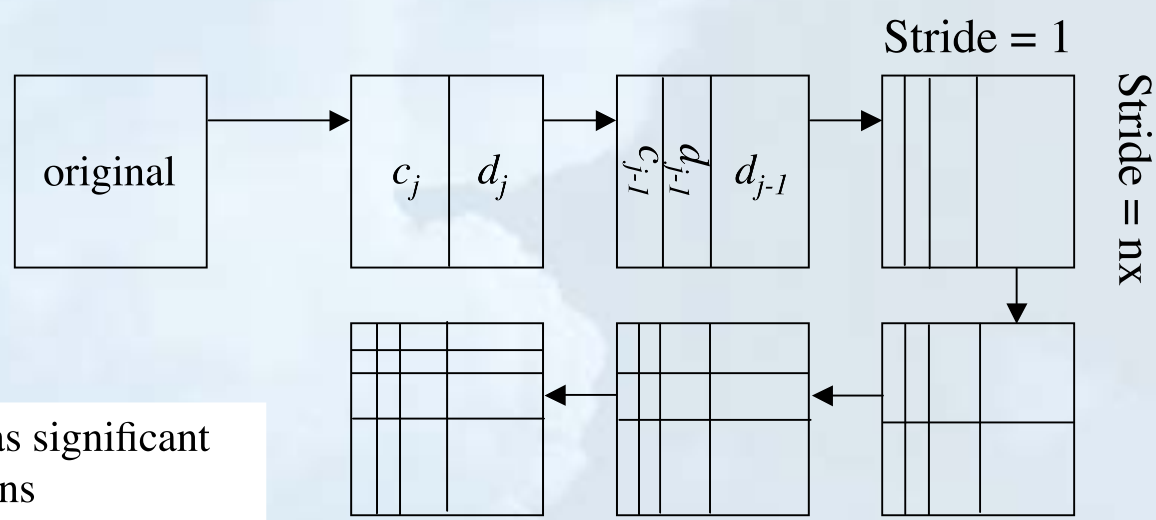
Forward transform filter bank



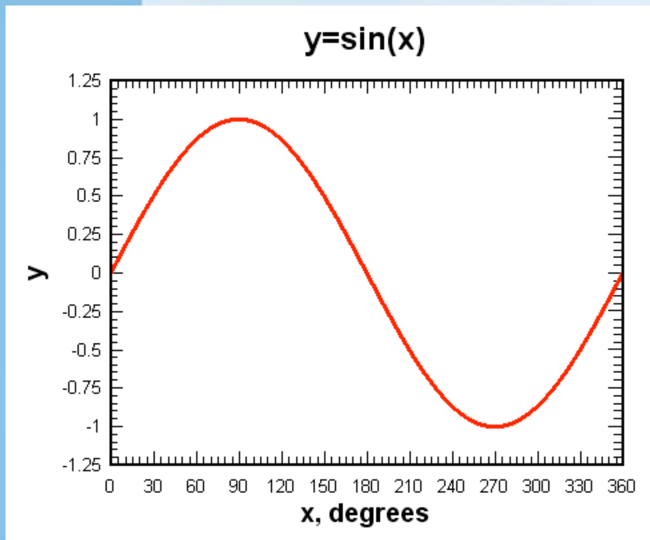
*n*D Forward Transform

- Extension to multiple dimensions is straight forward
- *Standard decomposition*: transform each dimension in sequence

Note: non-unit stride has significant performance implications



Standard 2D Wavelet Decomposition



Fourier transform basis function: sine, cosine

A very small sampling of wavelet transform basis functions

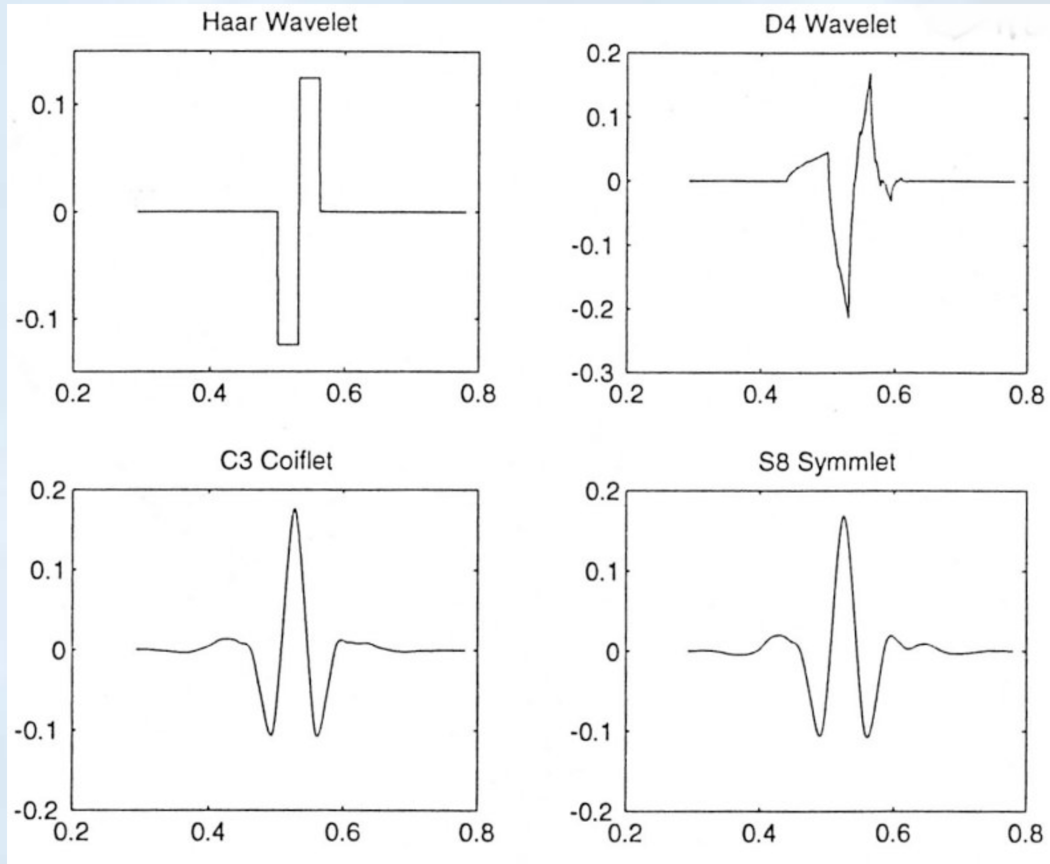


Image credit: K.H. Parker

Many wavelet families and parameterizations within each family to choose from. Best choice is often far from obvious.

Wavelet based progressive data access (1)
Frequency truncation method

NCAR

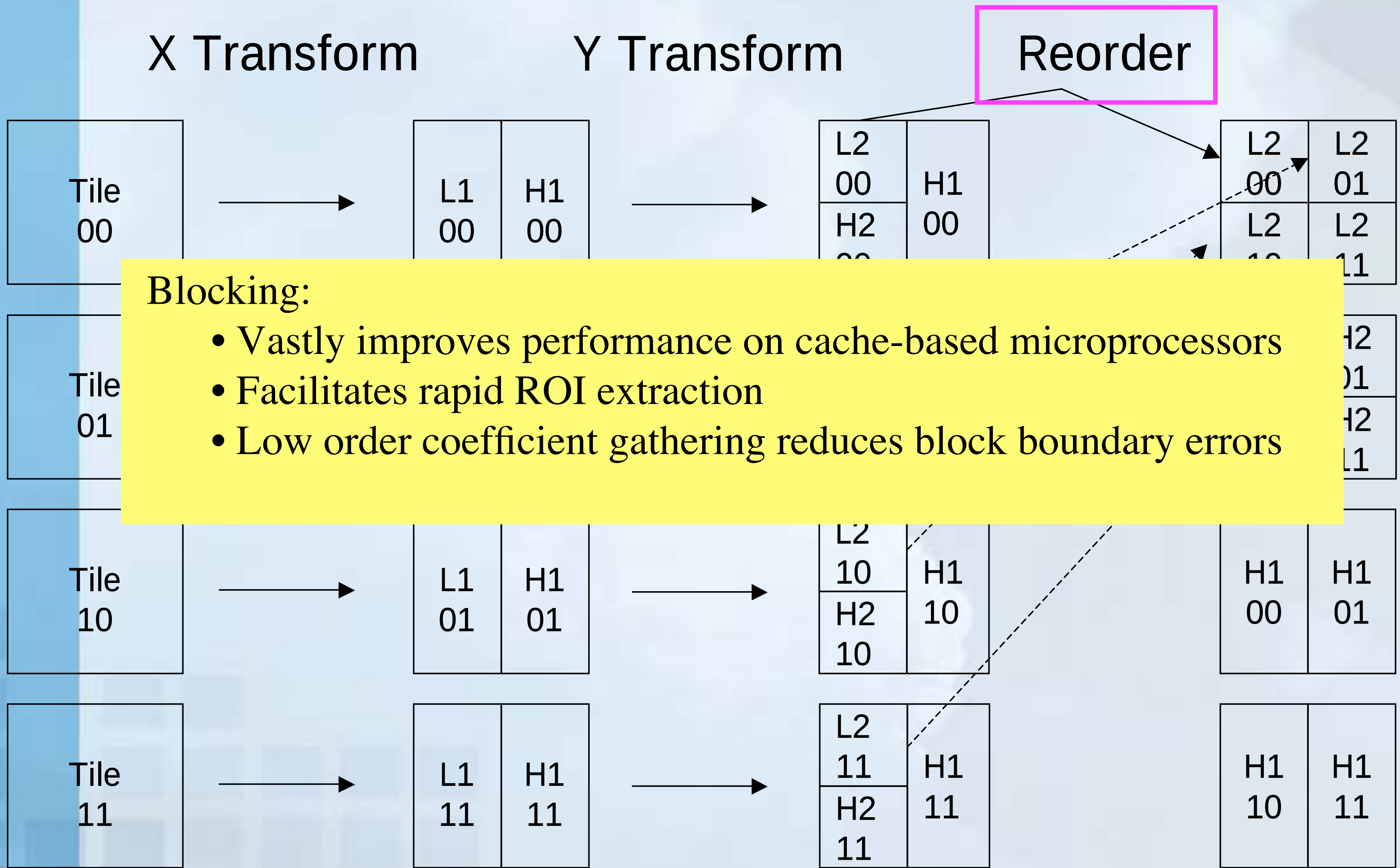
- Truncate "*j*" parameter of expansion:

$$f(t) = \sum_k c(k)\phi_k(t) + \sum_k \sum_{j=0}^{\log_2 N} d_j(k)\,\psi_{j,k}(t)$$

- Provides coarsened approximations at power-of-two increments
- Good:
  - Simple
  - Fast
  - Implicit surviving coefficient coordinates
  - **Preserves topology of original grid**
- Not so good:
  - Limited to power-of-two reductions
  - Compression quality

Strategies for large, multidimensional data:
Block (tile) based decomposition with low order coefficient gathering

NCAR



Blocking:

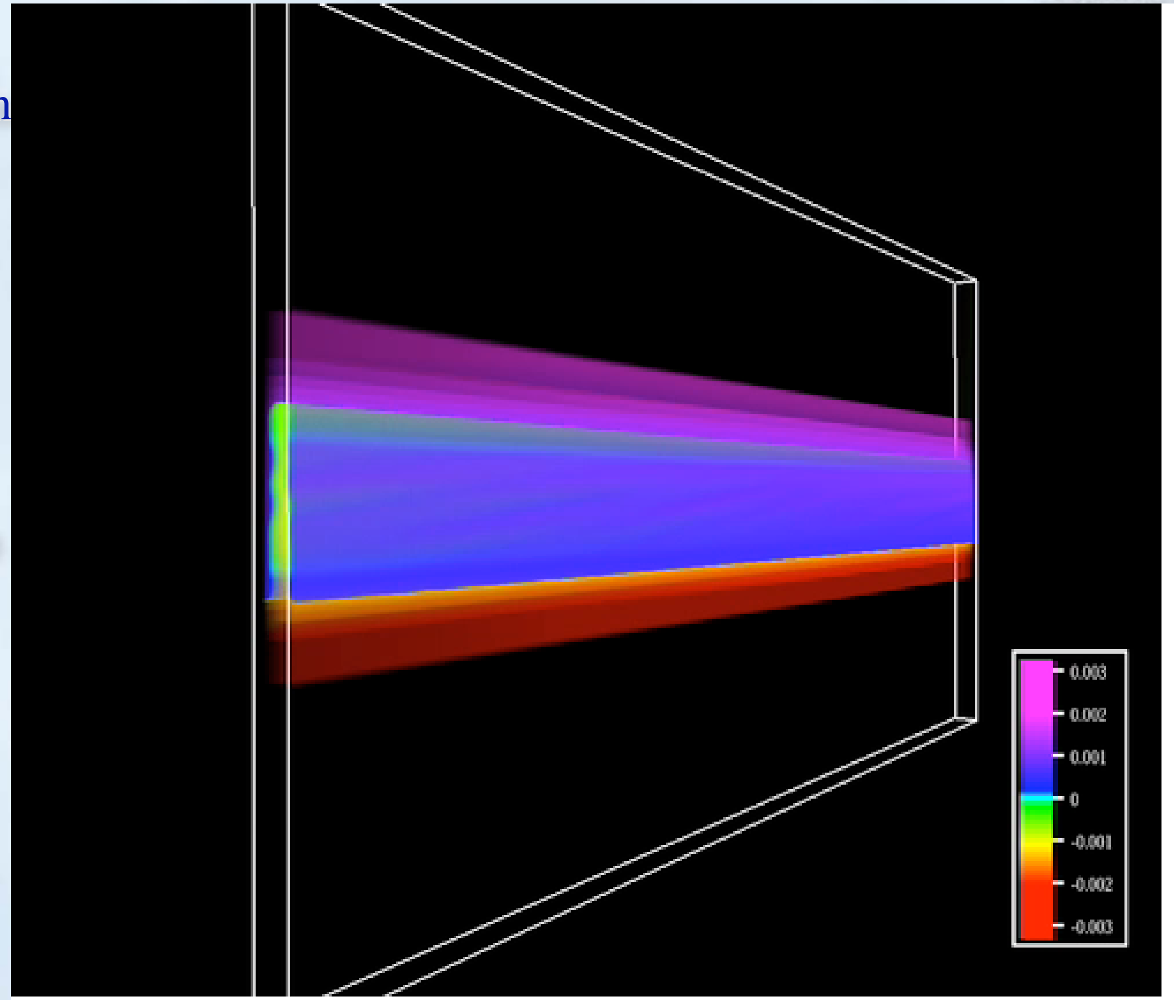
- Vastly improves performance on cache-based microprocessors
- Facilitates rapid ROI extraction
- Low order coefficient gathering reduces block boundary errors

VAPOR Demo
progressive data access - frequency truncation method

NCAR

Salt sheets and turbulence in double-diffusive shear layer

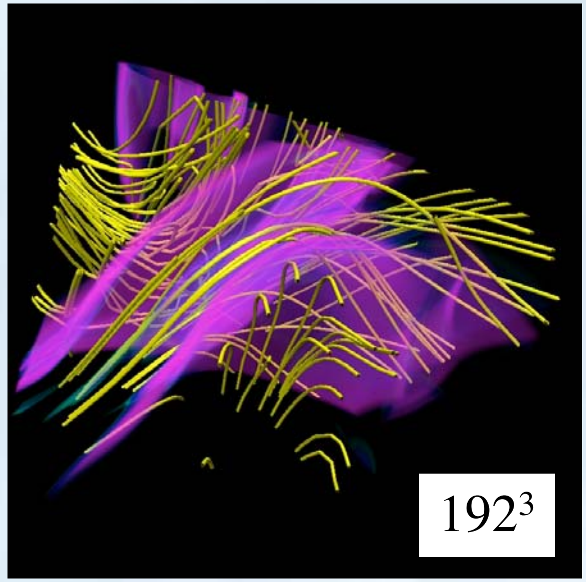
- 6144x144x3073 grid
- 12 GBs per field
- ~10 TB data saved
- 2007 NCAR Breakthrough Science (BTS) campaign
- 5 level wavelet hierarchy
  - 6144x144x3073
  - 3072x72x1536
  - 1536x36x768
  - 678x18x384
  - 384x9x192
- 32^3 wavelet blocks

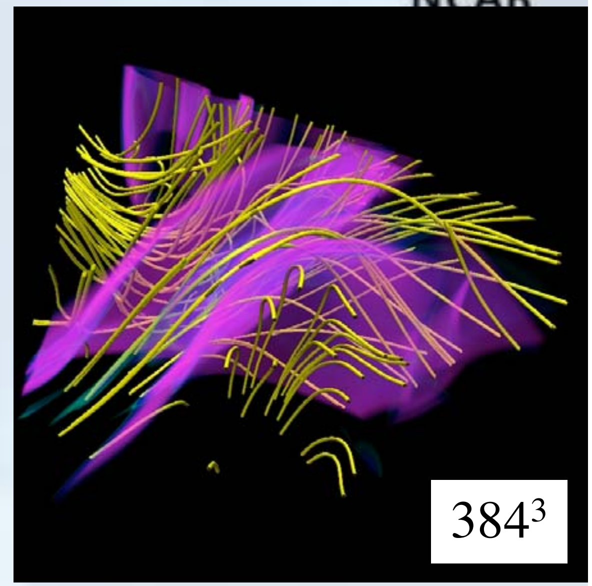


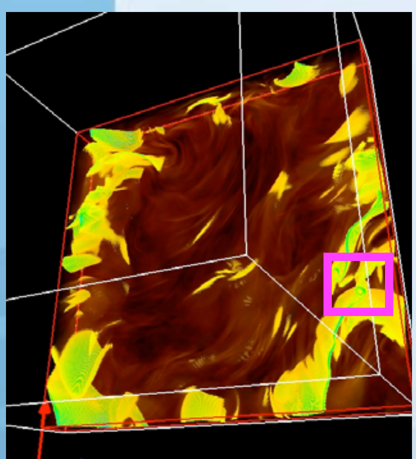
Bill Smyth and Satoshi Kimura, U. of Oregon

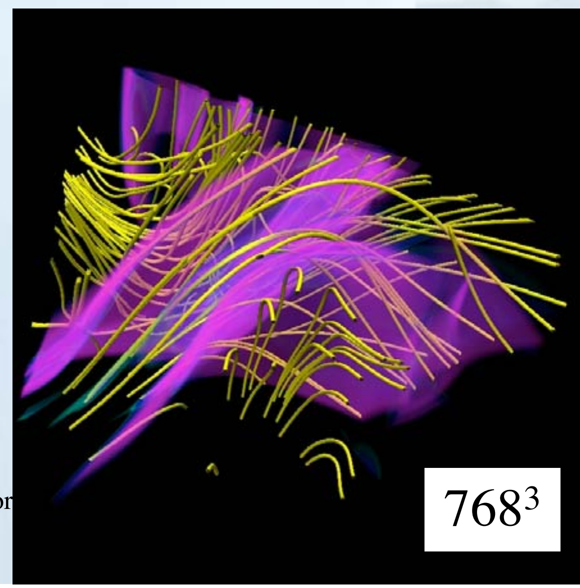
Magnetic field line integration resolution comparison

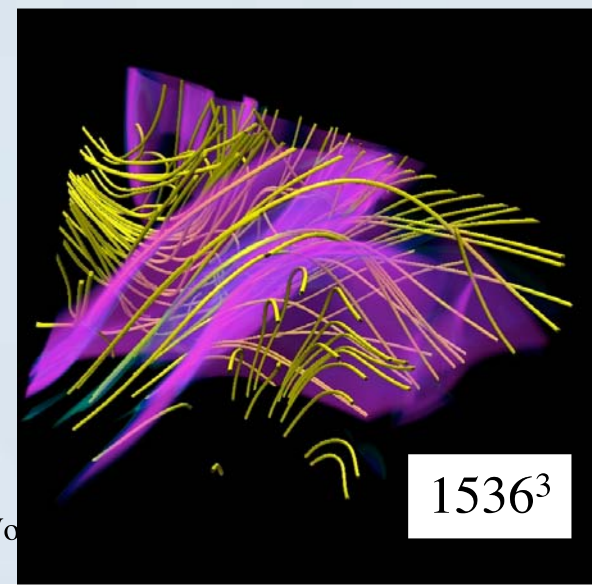
- 1536^3 MHD Simulation
- 4th order Runge-Kutte
- Mininni et al. (2007)











Wavelet based hierarchical data representation has been shown to enable powerful speed/quality tradeoffs in VAPOR. Data sets up to 2048^3 can effectively be analyzed with modest computing resources. But…

- Power-of-two reductions are limiting
- Not clear that current model will scale to petascale data sets

More aggressive data reduction required for petascale applications

Wavelet based progressive data access (2)
Coefficient prioritization method

- Goal: prioritize coefficients used in linear expansion

$$f(t) = \sum_{n=0}^{N-1} a_n u(t), \quad \text{original } f(t) \qquad \hat{f}(t) = \sum_{m=0}^{M-1} a_m u(t), \ (M < N), \quad \text{compressed } f(t)$$

L^2 error given by: $L^2 = \left\| f(t) - \hat{f}(t) \right\|_2^2$

If *u(t)* (*φ(t)* and *ψ(t)* in case of wavelet expansion functions) are *orthonormal*, then

$$\text{orthonormal}: \langle u_k(t), u_l(t) \rangle = \int u_k(t) u_l(t) dt = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases}$$

$$L^2 = \sum_{i=M}^{N-1} \left(a_{\pi(i)}\right)^2 = \left\| f(t) - \hat{f}(t) \right\|_2^2, \text{ where } a_{\pi(i)} \text{ are discarded coefficients}$$

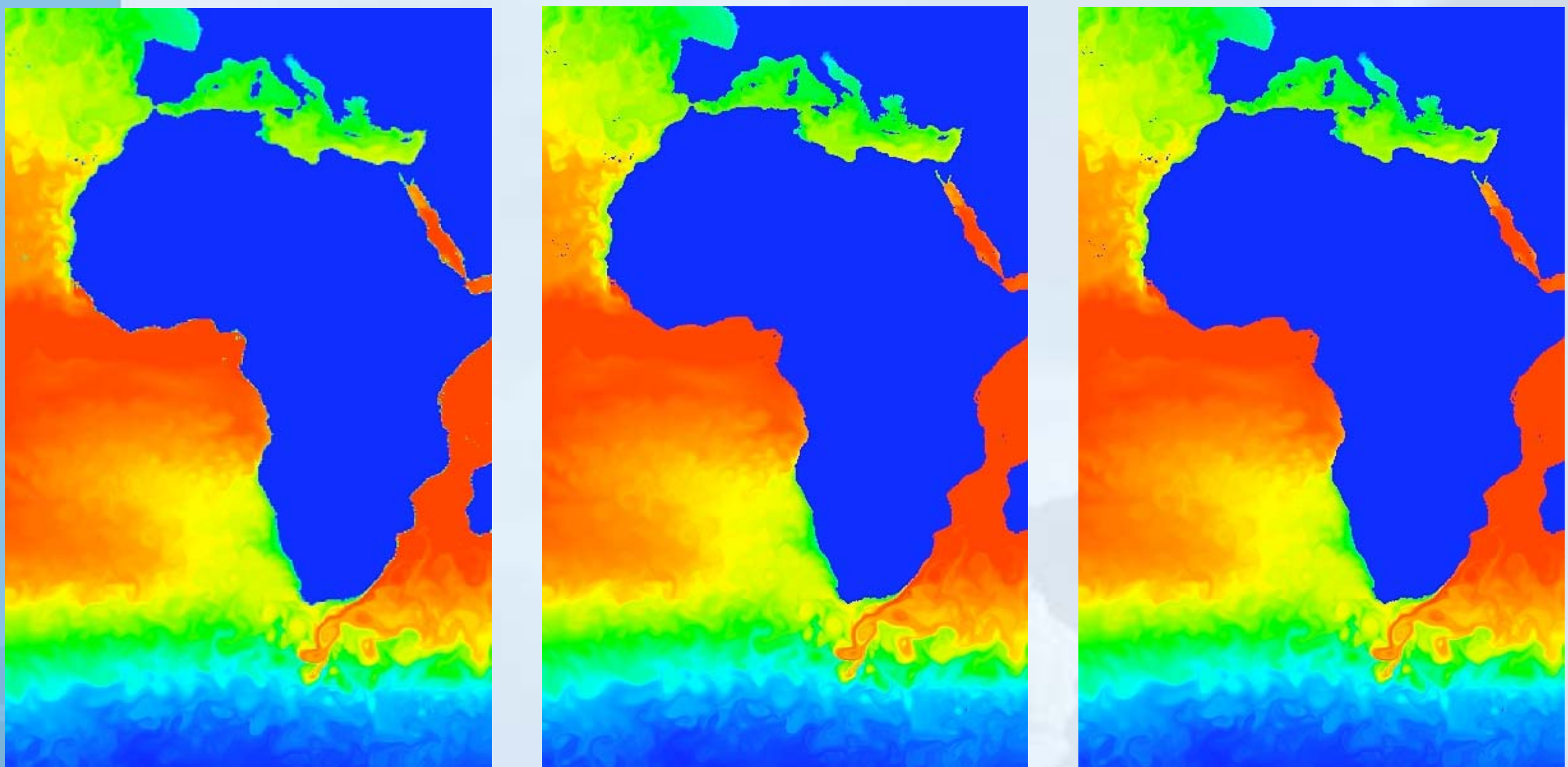
- The error is the sum of the squares of the coefficients we leave out!
- So to minimize the L^2 error, we simply discard (or delay transfer) the smallest coefficients!
- If discarded coefficients are zero, there is no information loss!

Wavelet based progressive data access (2)
Coefficient prioritization method

- Good
  - Approximation accuracy superior to frequency truncation method for a given compression rate
  - Arbitrary compression rates
  - Flexibility (numerous compression metrics possible)
    - Wavelet choices
    - Coefficient selection criteria
- Not so good
  - Algorithm complexity
  - Algorithm efficiency (both forward and inverse transform)
  - Coefficient coordinates not implicit

8:1 Compression - Global POP 1/10 degree ocean model

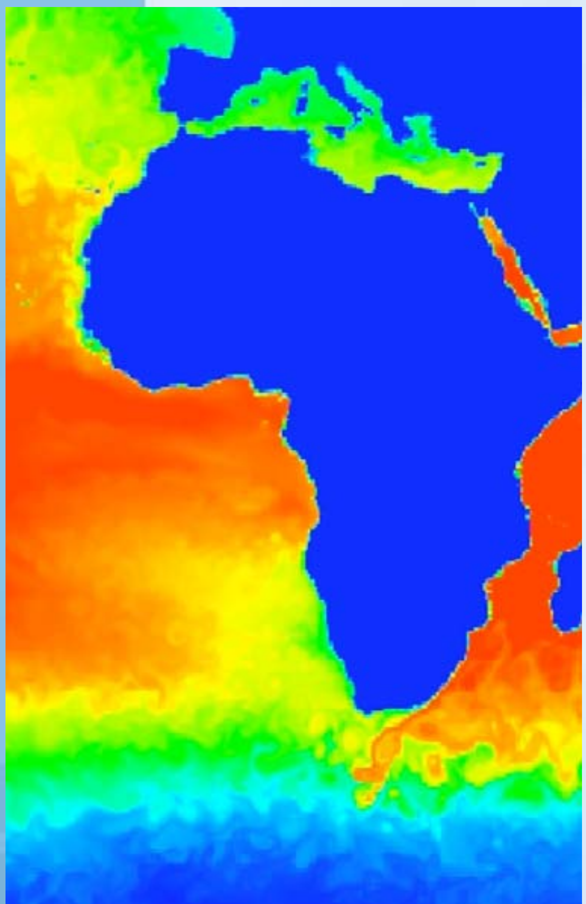
F. Bryan, 2006



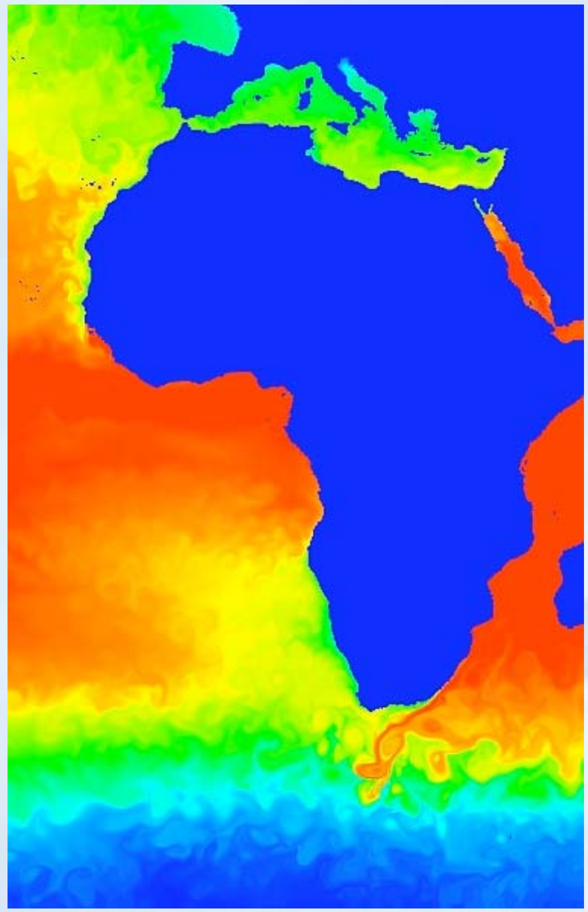
Frequency truncation | No compression | Coefficient prioritization

64:1 Compression - Global POP 1/10 degree ocean model

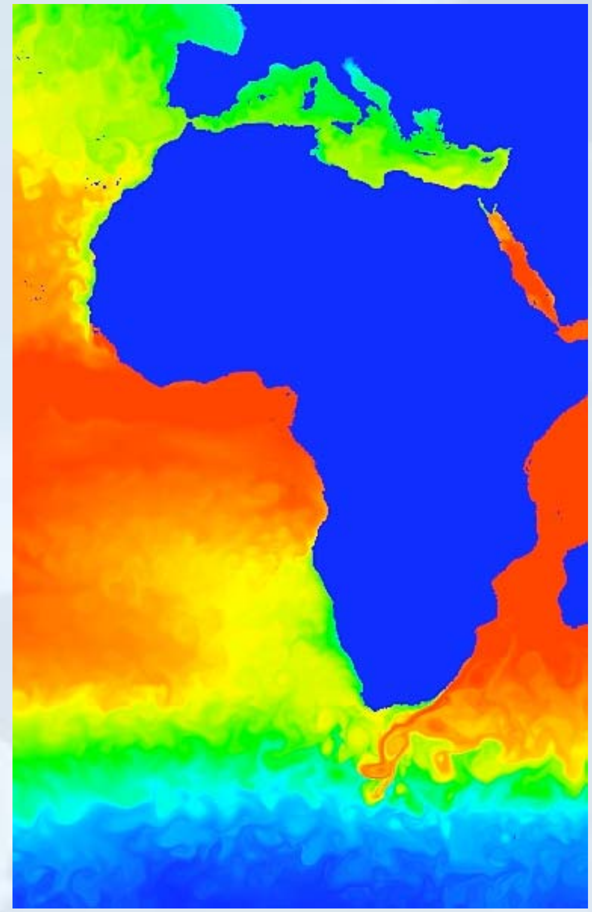
F. Bryan, 2006



Frequency truncation



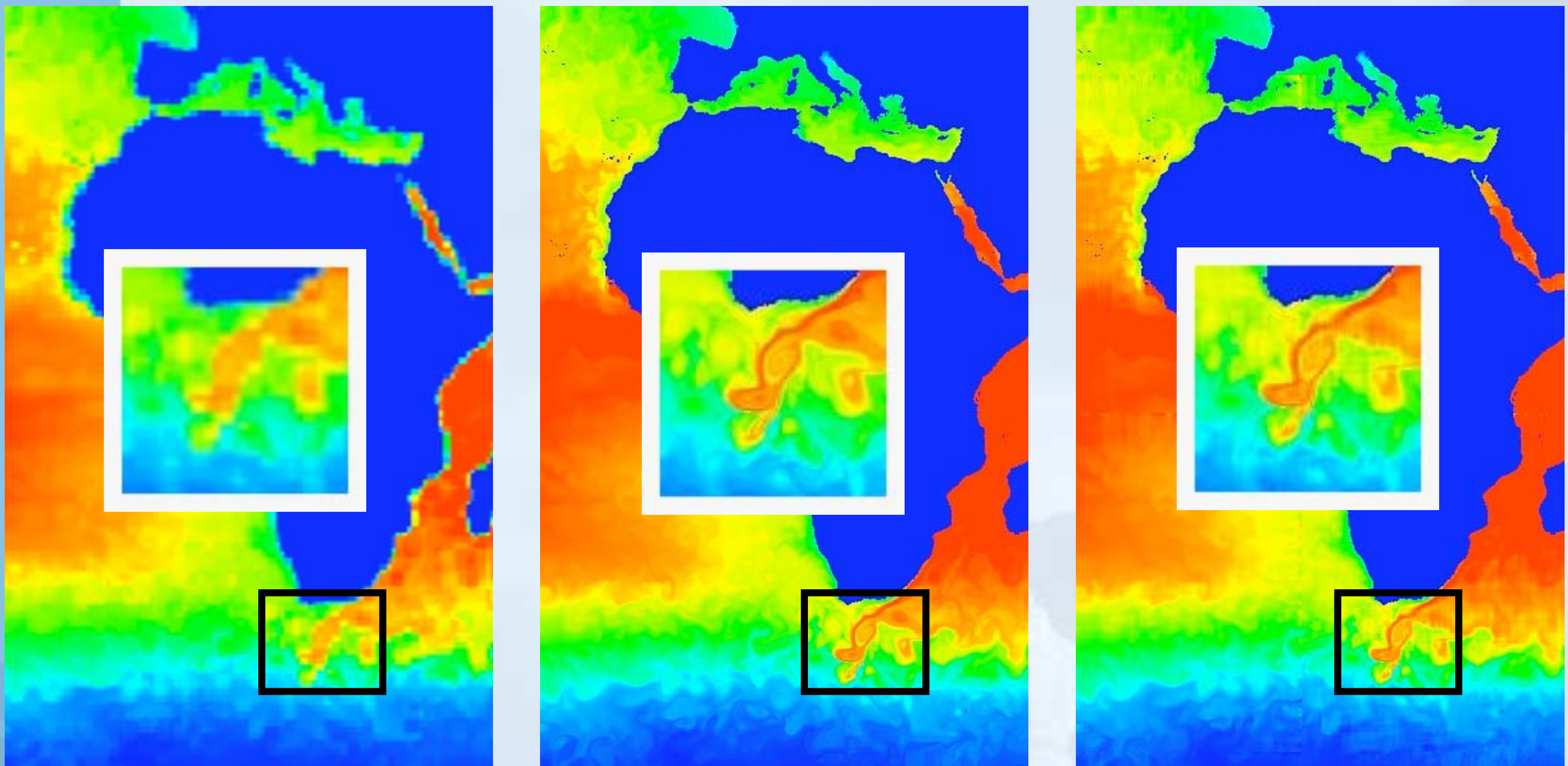
No compression



Coefficient prioritization

512:1 Compression - Global POP 1/10 degree ocean model

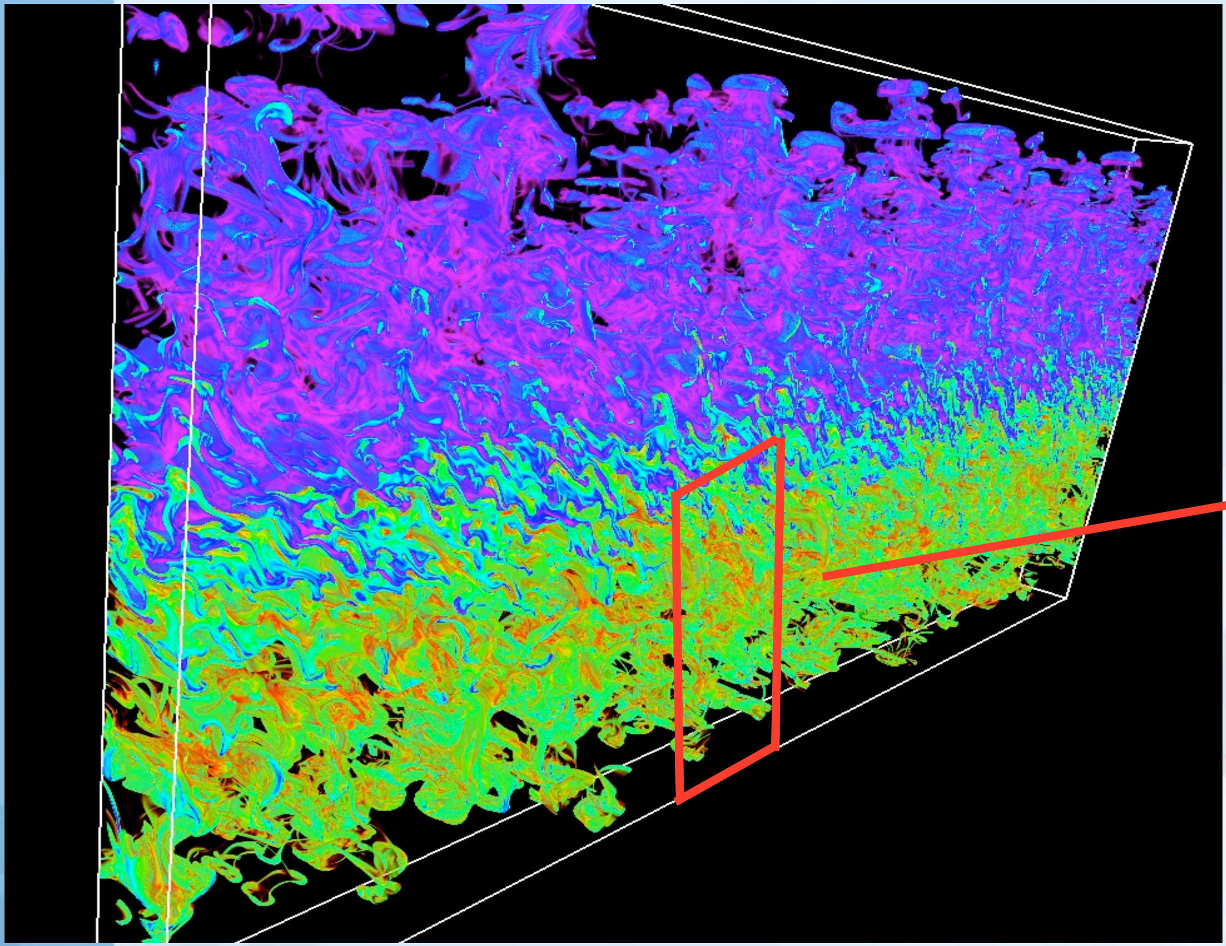
F. Bryan, 2006

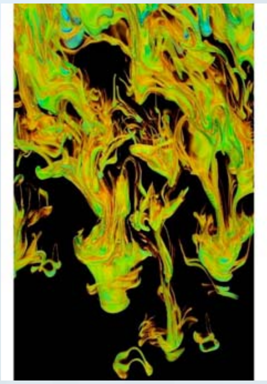


Frequency truncation | No compression | Coefficient prioritization

Seawater turbulence on a 6144x144x3073 grid

W. Smyth & S. Kimura, 2007

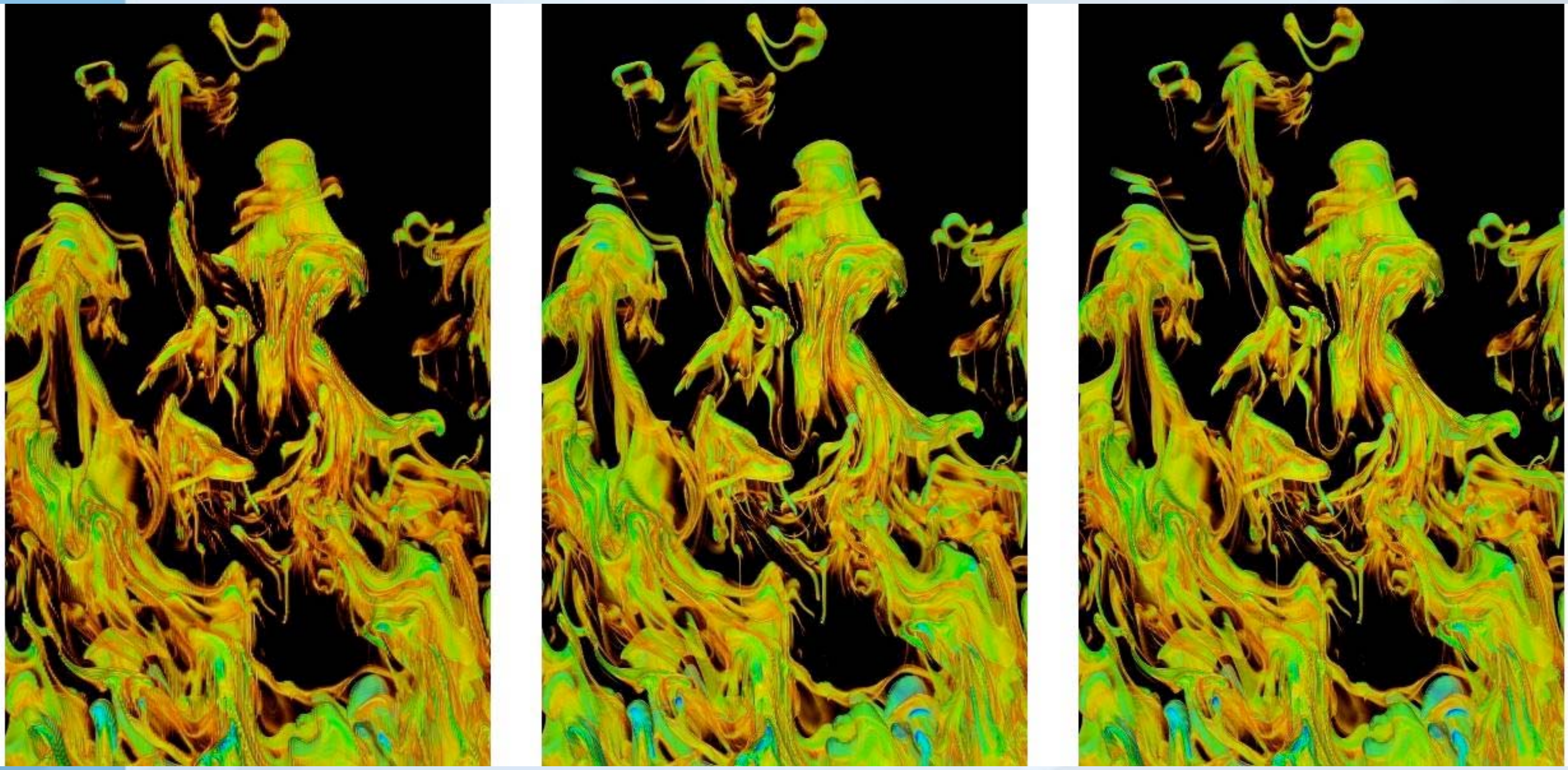




614x144x1536 ROI

8:1 Compression - Seawater turbulence on a 6144x144x3073 grid

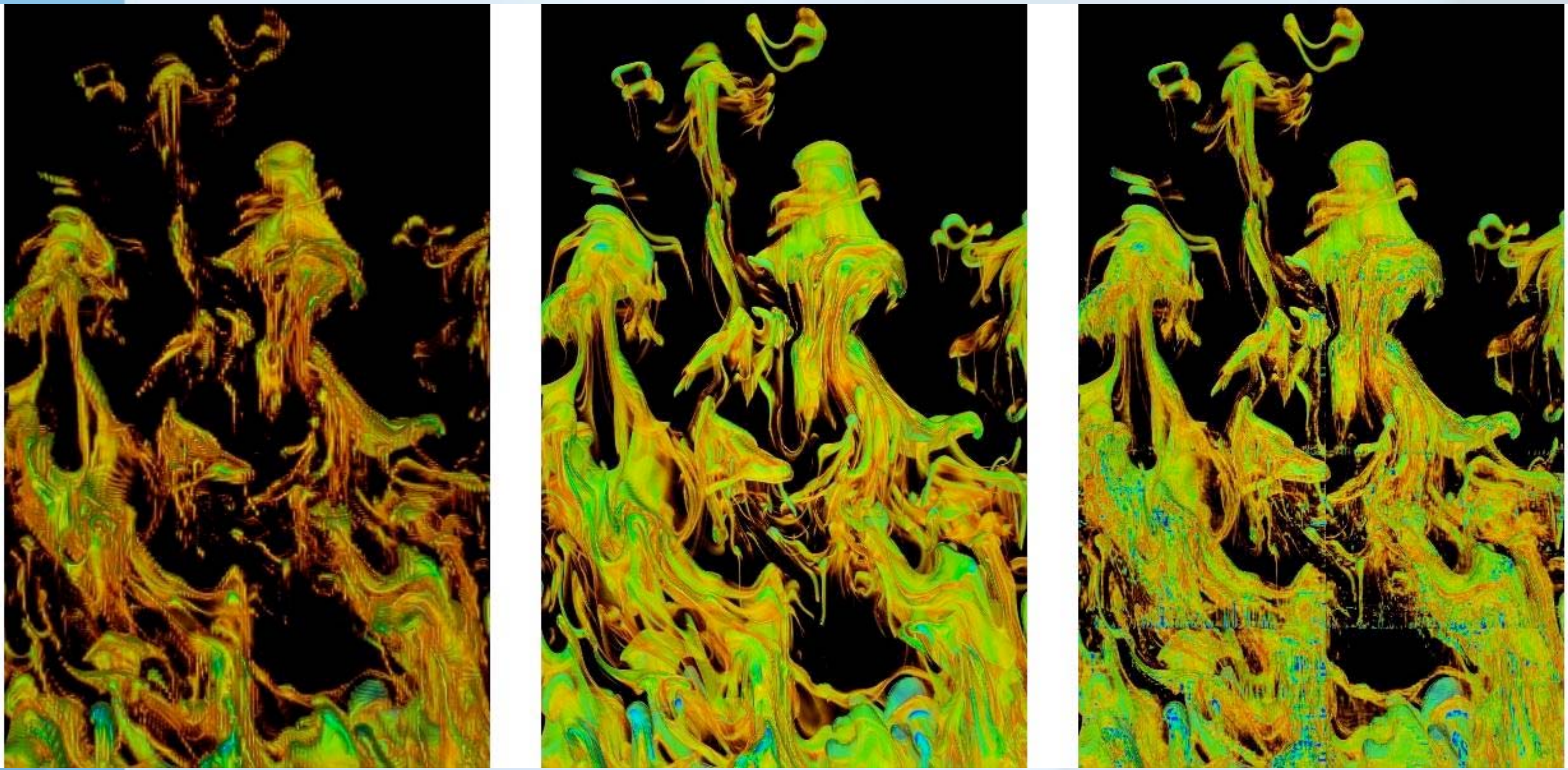
W. Smyth & S. Kimura, 2007



Frequency truncation | No compression | Coefficient prioritization

64:1 Compression - Seawater turbulence on a 6144x144x3073 grid

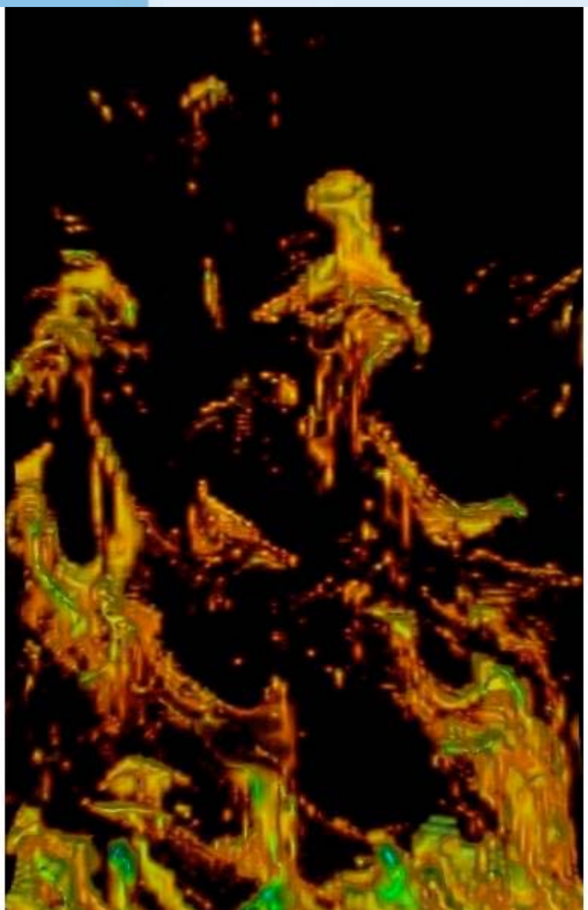
W. Smyth & S. Kimura, 2007



Frequency truncation | No compression | Coefficient prioritization

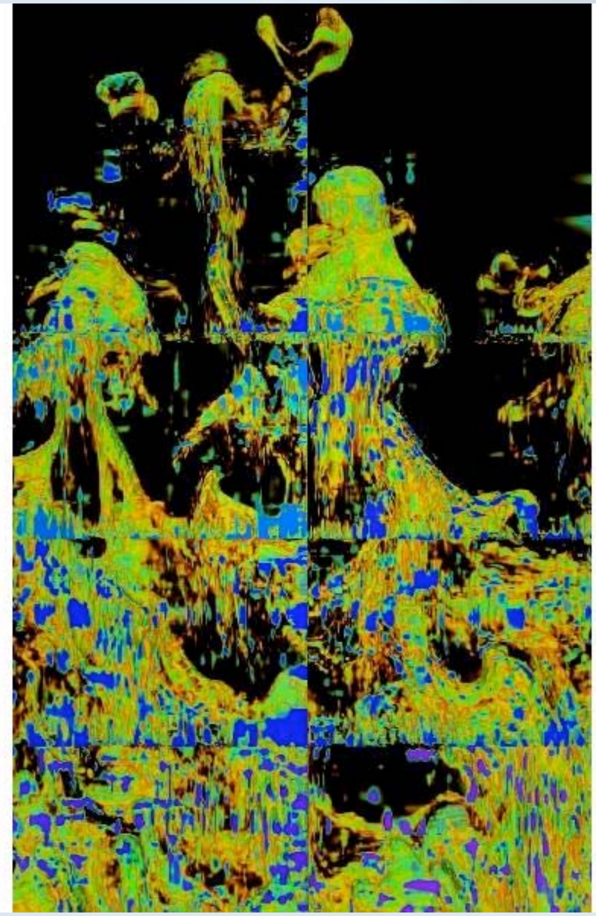
512:1 Compression - Seawater turbulence on a 6144x144x3073 grid

W. Smyth & S. Kimura, 2007



Frequency truncation

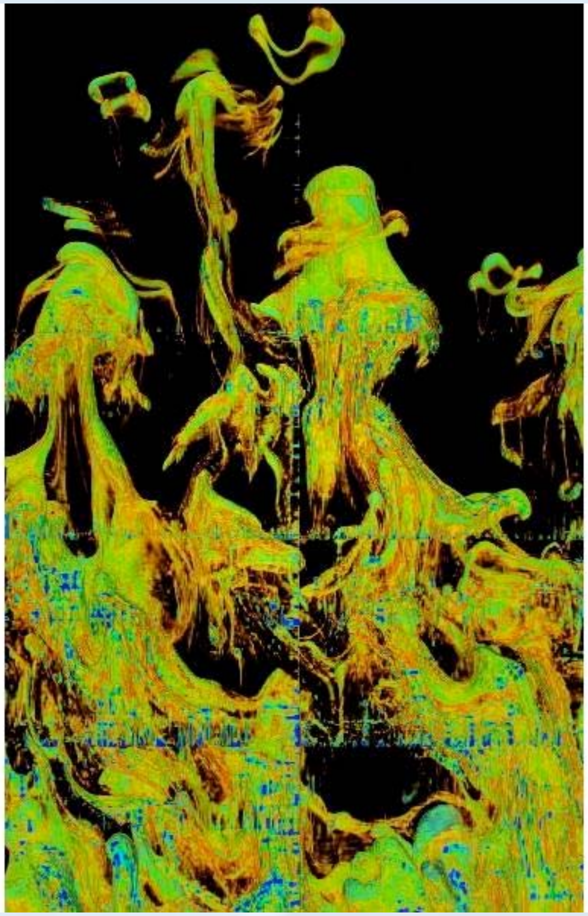
No compression



Coefficient prioritization

Coefficient prioritization method permits arbitrary compression rates not possible with frequency truncation method

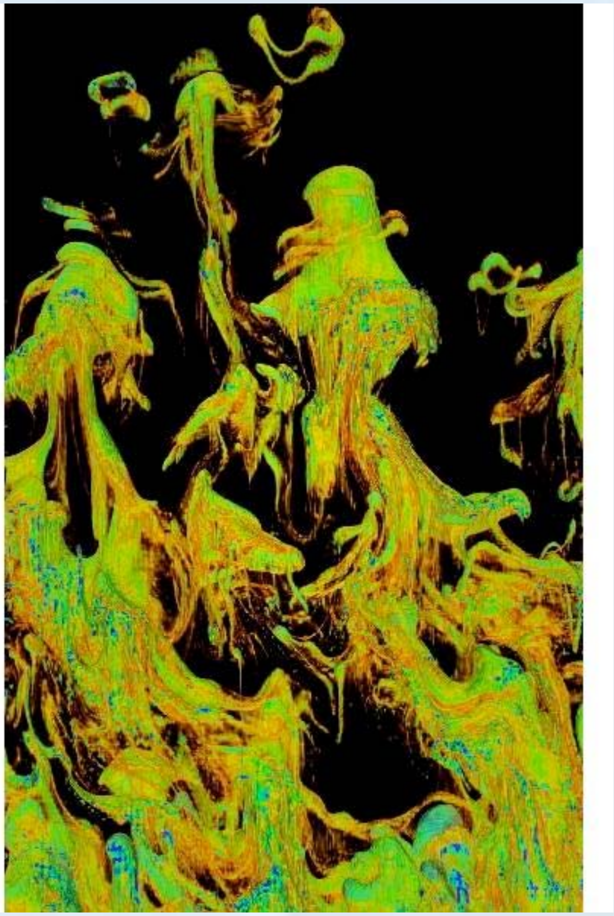
No compression



100:1 compression

100:1 compression without blocking

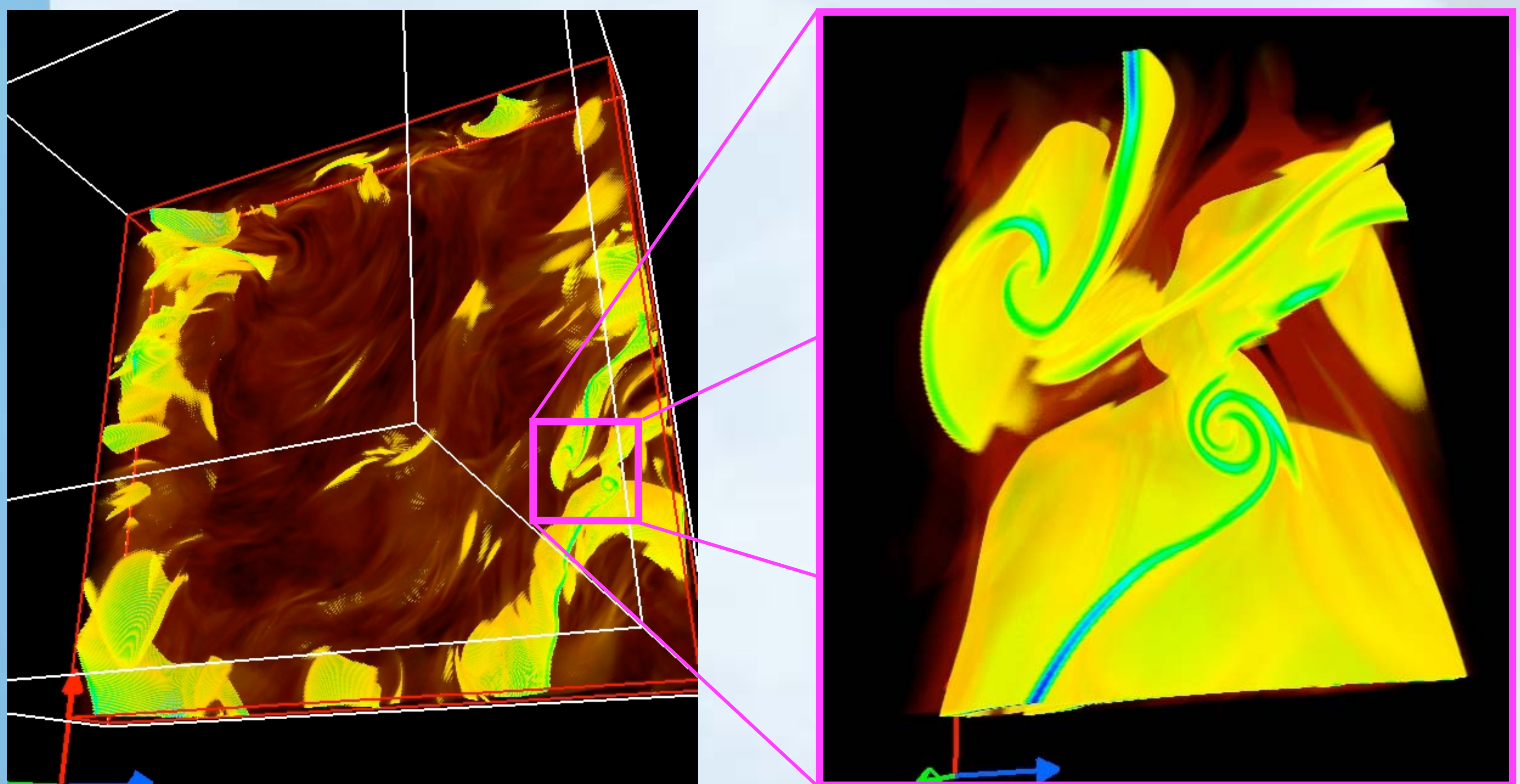
No compression



100:1 compression

512:1 Compression - 1536^3 MHD Decay Simulation

Mininni et al., PRL ***97****, 244503 (2006)*

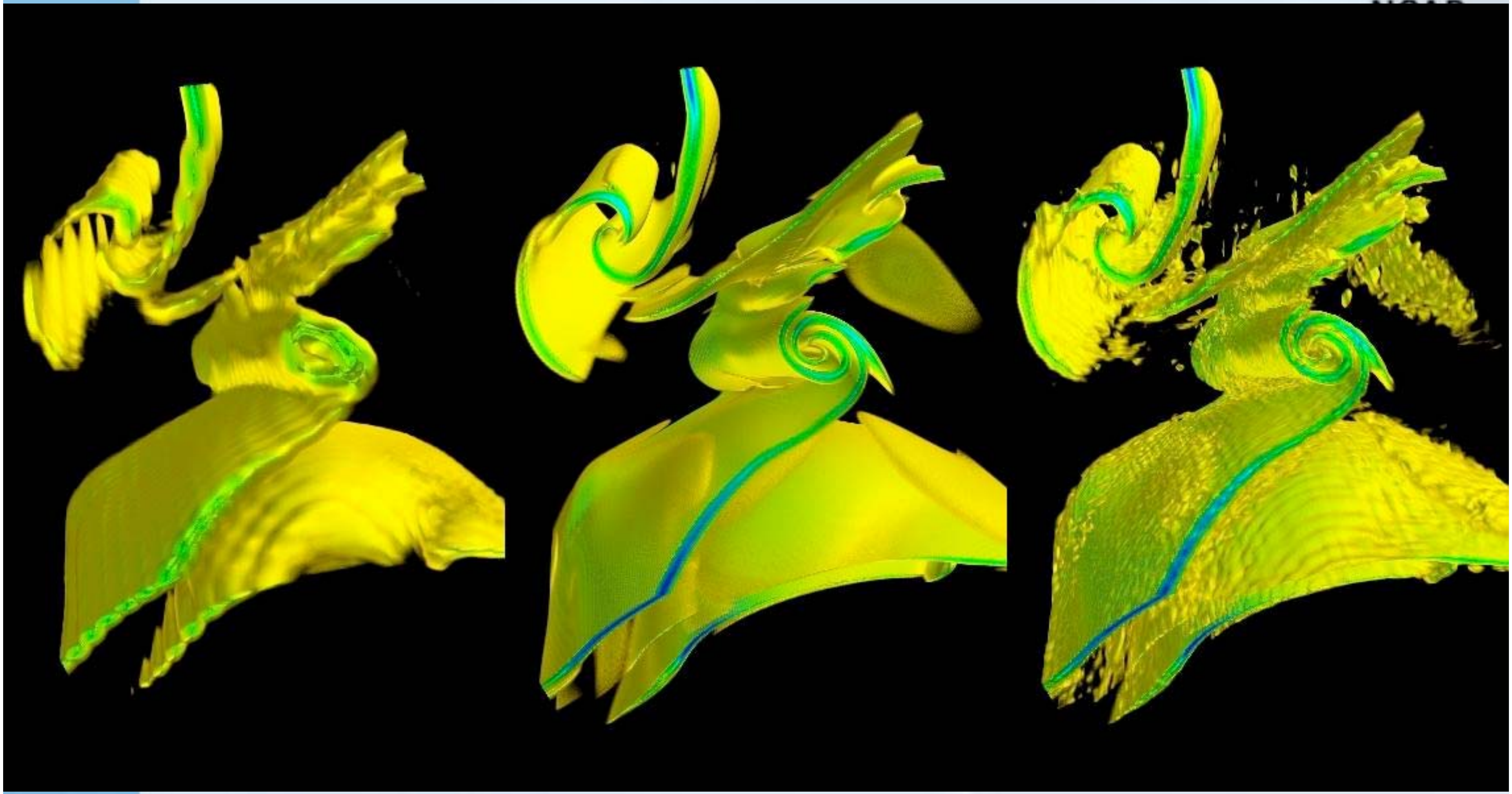


Full 1536^3 domain

140x300x100 ROI

512:1 Compression - 1536^3 MHD Decay Simulation

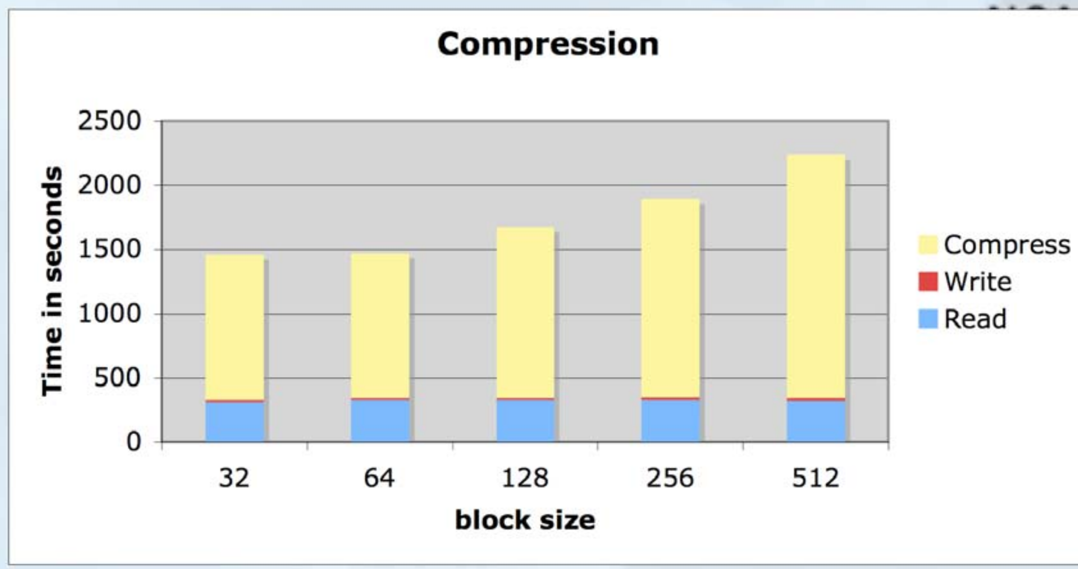
*Mininni et al., PRL **97**, 244503 (2006)*

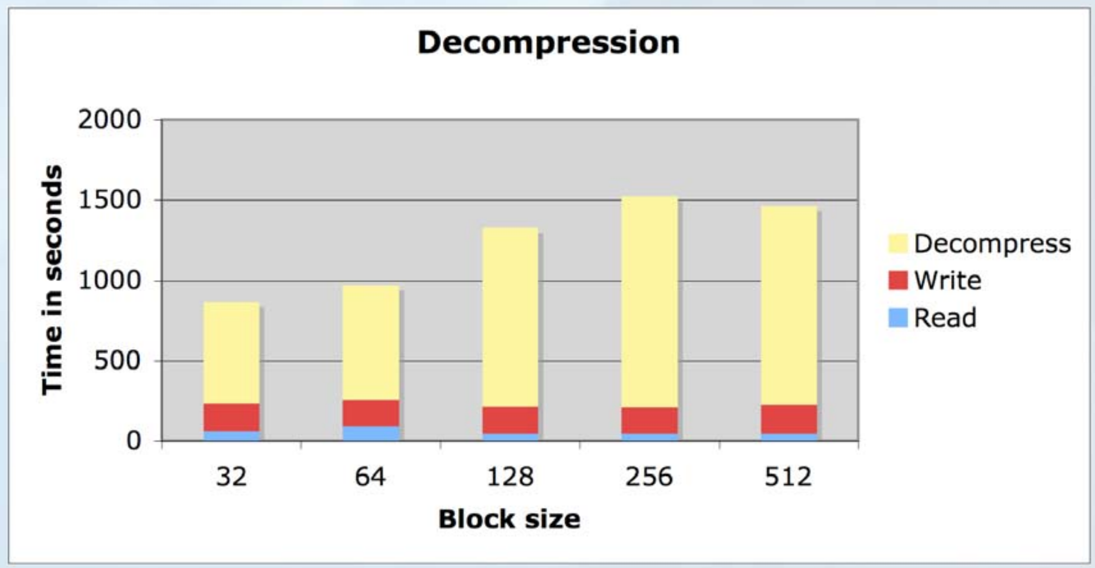


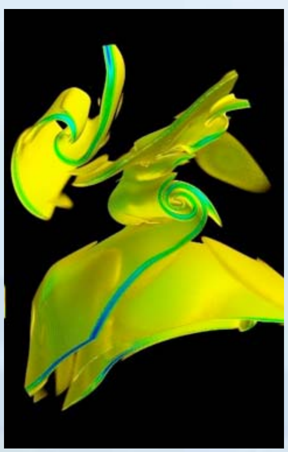
Frequency truncation | No compression | Coefficient prioritization

Serial timings - coefficient prioritization

- Compress (decompress) file and write it back to disk
- 1536^3 MHD Simulation
- 512:1 compression
- Lifting 4,4 wavelet

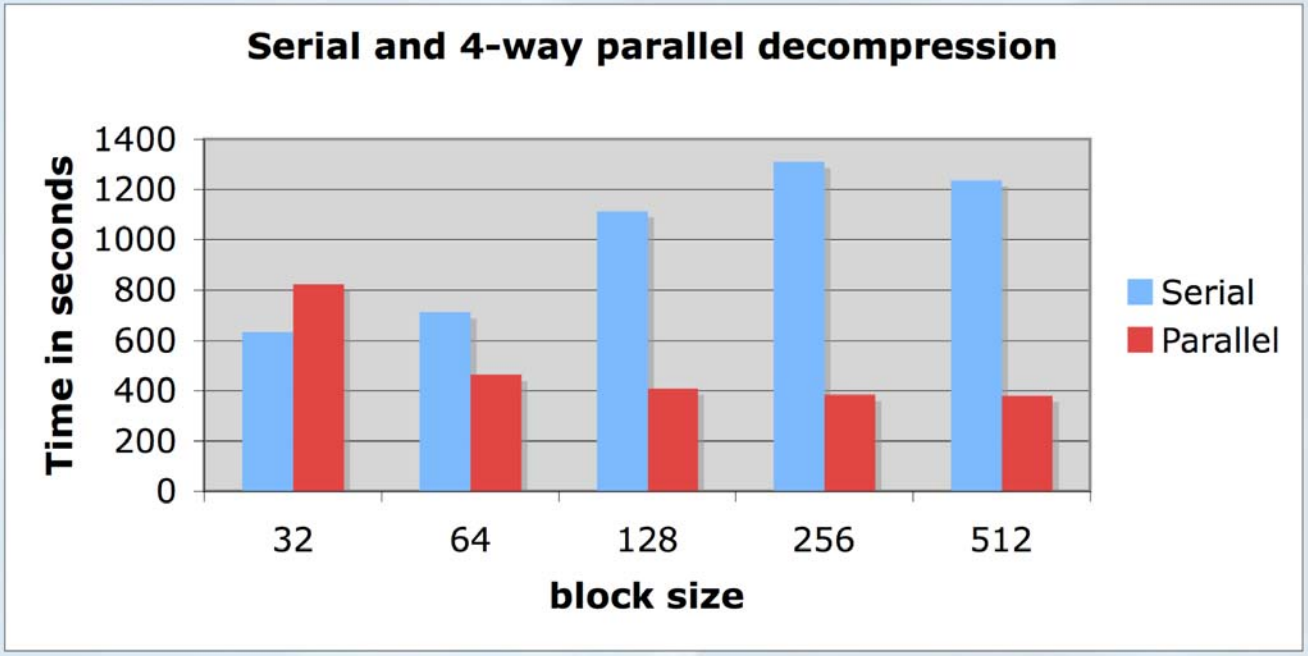


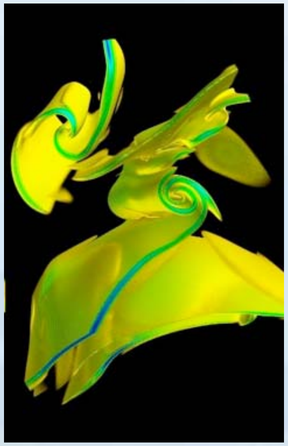




Parallel wavelet decoding

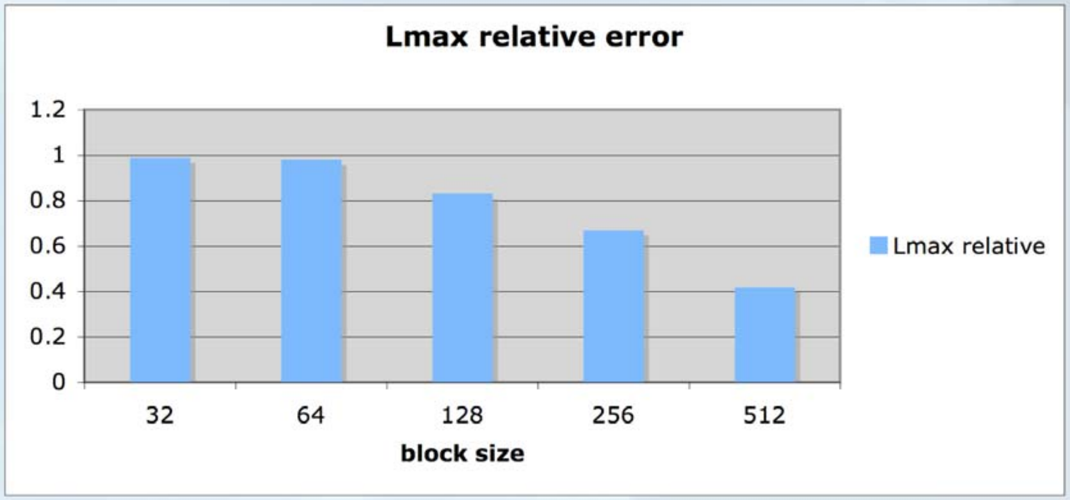
- Compress (decompress) file and write it back to disk
- 1536^3 MHD Simulation
- 512:1 compression
- Lifting 4,4 wavelet

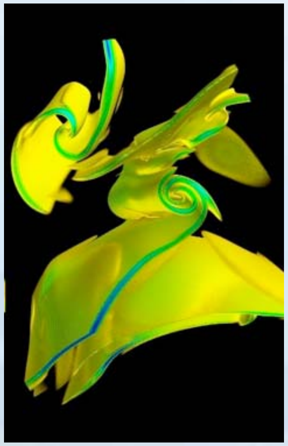


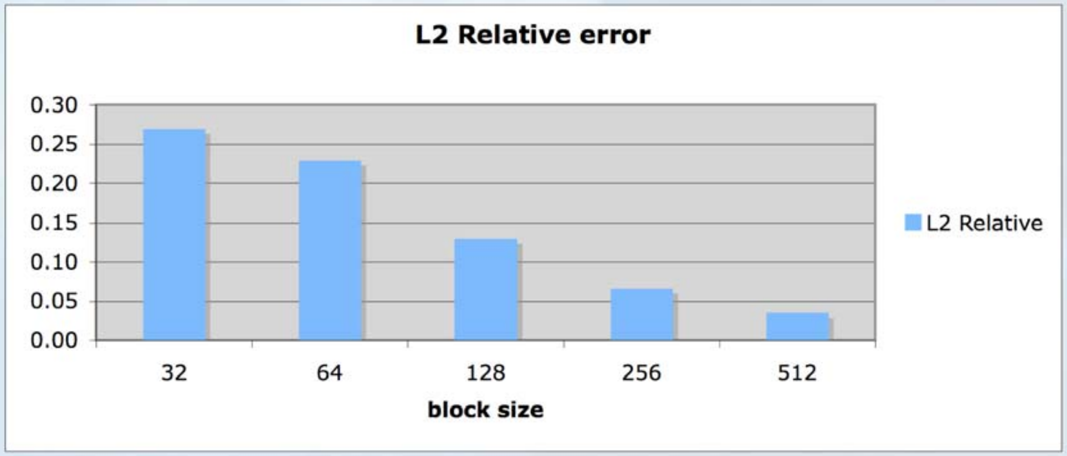


L2 and Lmax errors - coefficient prioritization

- Compress (decompress) file and write it back to disk
- 1536^3 MHD Simulation
- 512:1 compression
- Lifting 4,4 wavelet







Coefficient Prioritization Compression Research Challenges

- Block boundary artifacts
  - Low order coefficient gathering (as done with hierarchical progressive access)
  - Asymmetric wavelets
- Efficient coefficient coordinate encoding
  - Present schemes (e.g octrees, zerotrees) don't scale
- Performance
  - Efficient <u>in situ</u> encoder implementation on petatflop systems
  - Efficient decoder for smaller, interactive systems
- Fully decompressed data can overwhelm resources of analysis platform
  - Perform analysis/visualization in wavelet space
  - On-the-fly regridding
- Choice of wavelet family
- Coefficient prioritization scheme (L2 error minimization may not be best choice)
- Developing meaningful error metrics

Final remarks

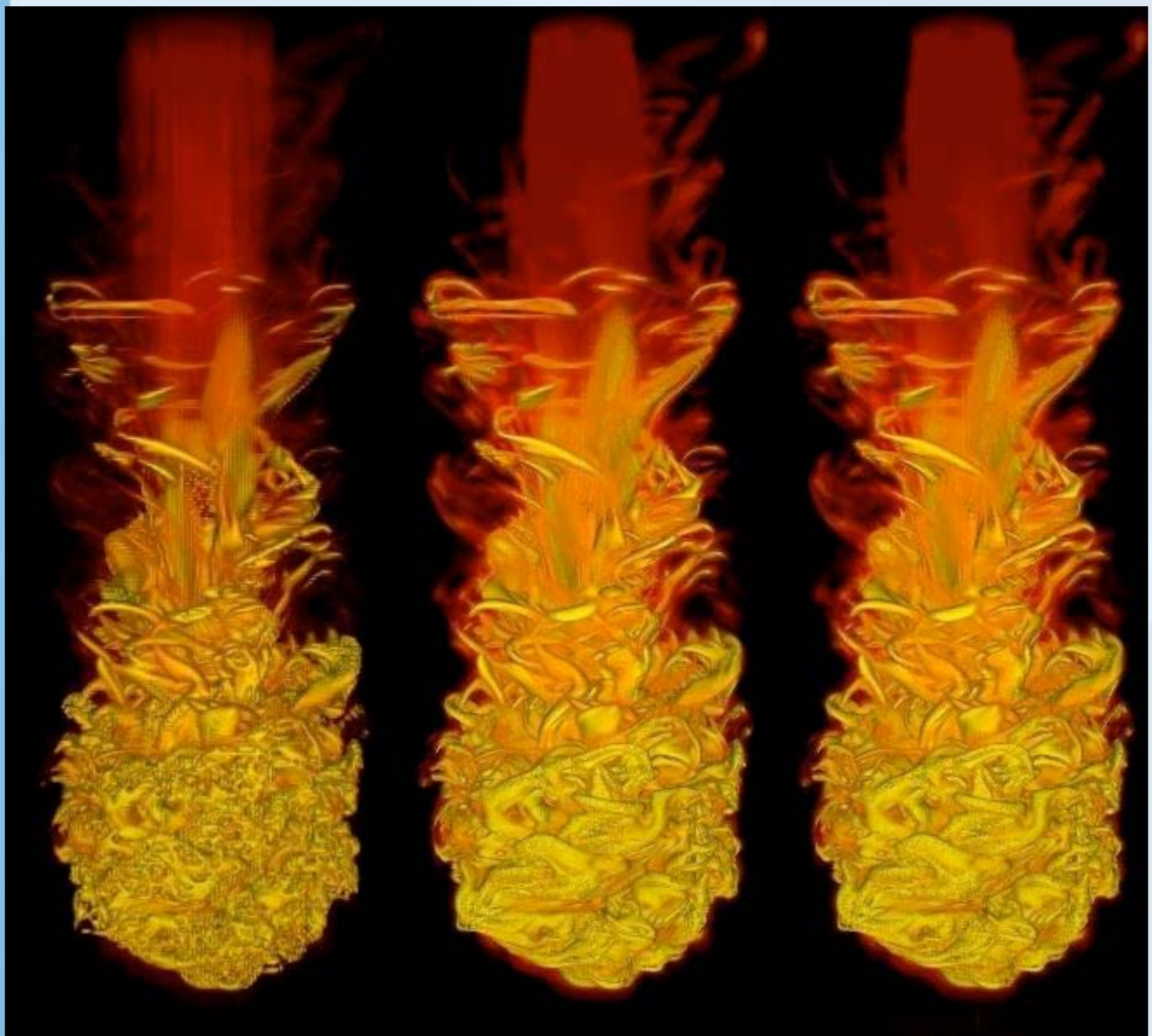
- Progressive data access != compression
  - Compression: loss of information
  - Progressive data access: transforming data to a space where they can be accessed more intelligently
- Limits of compression are application and data dependent
- Opportunities exist for rapid hypothesis testing using compressed data that may subsequently be validated with native data
- Consider value of saving some timesteps at reduced fidelity
- Moore's law does not apply to all computing technologies
  - We are entering the era of the Petaflop, not the Petabyte-per-second!

Questions???

VAPOR: www.vapor.ucar.edu

64:1 compression - 512x512x2048 Thermal Starting Plume

M. Rast, 2003

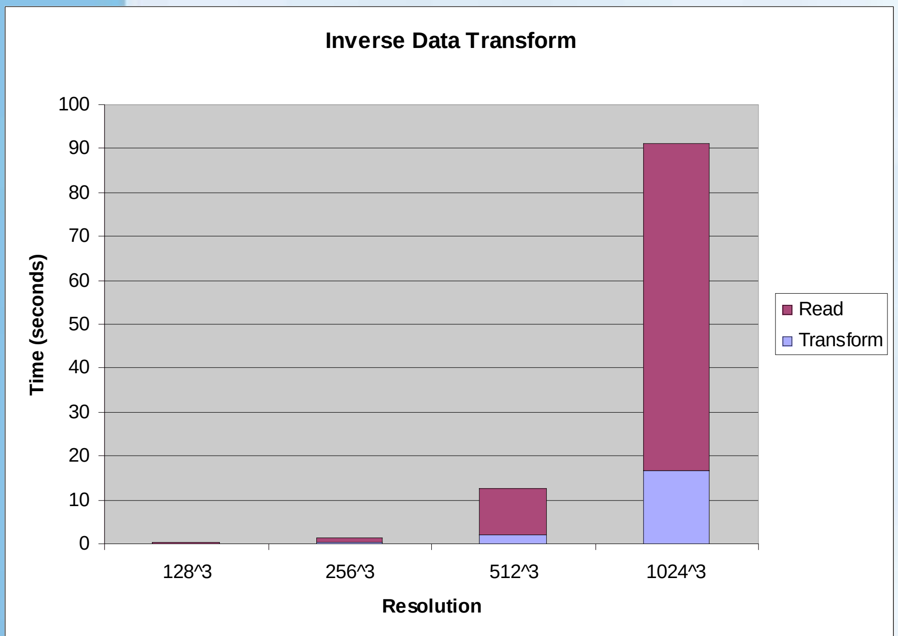


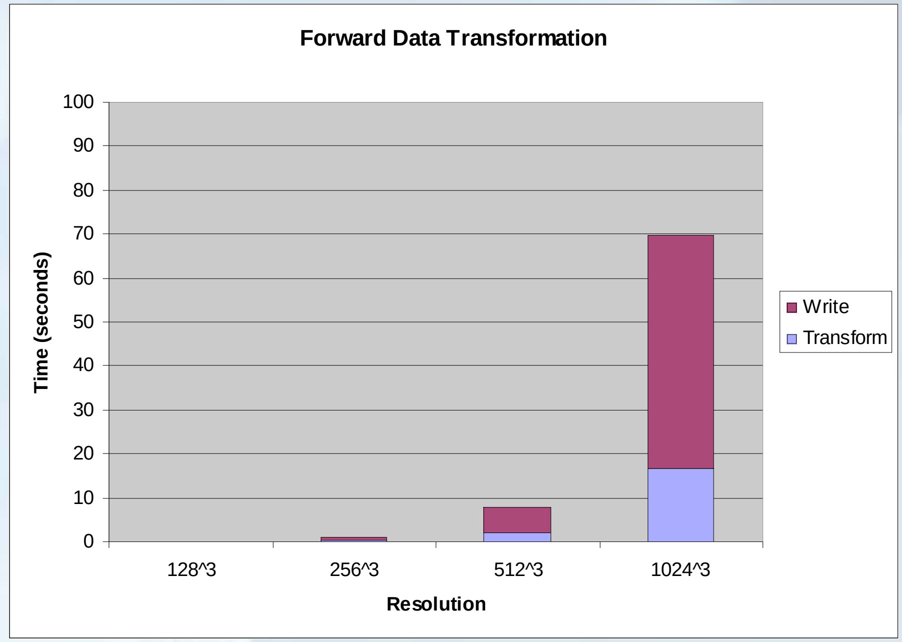
Frequency truncation | No compression | Coefficient prioritization

Blocking

- Good
  - Necessary for good performance on cache coherent microprocessors
  - Facilitate parallel implementation
  - Smaller memory footprint
  - Facilitate ROI extraction
- Bad
  - Boundary artifacts

Performance of forward and inverse Haar wavelet transform





Data

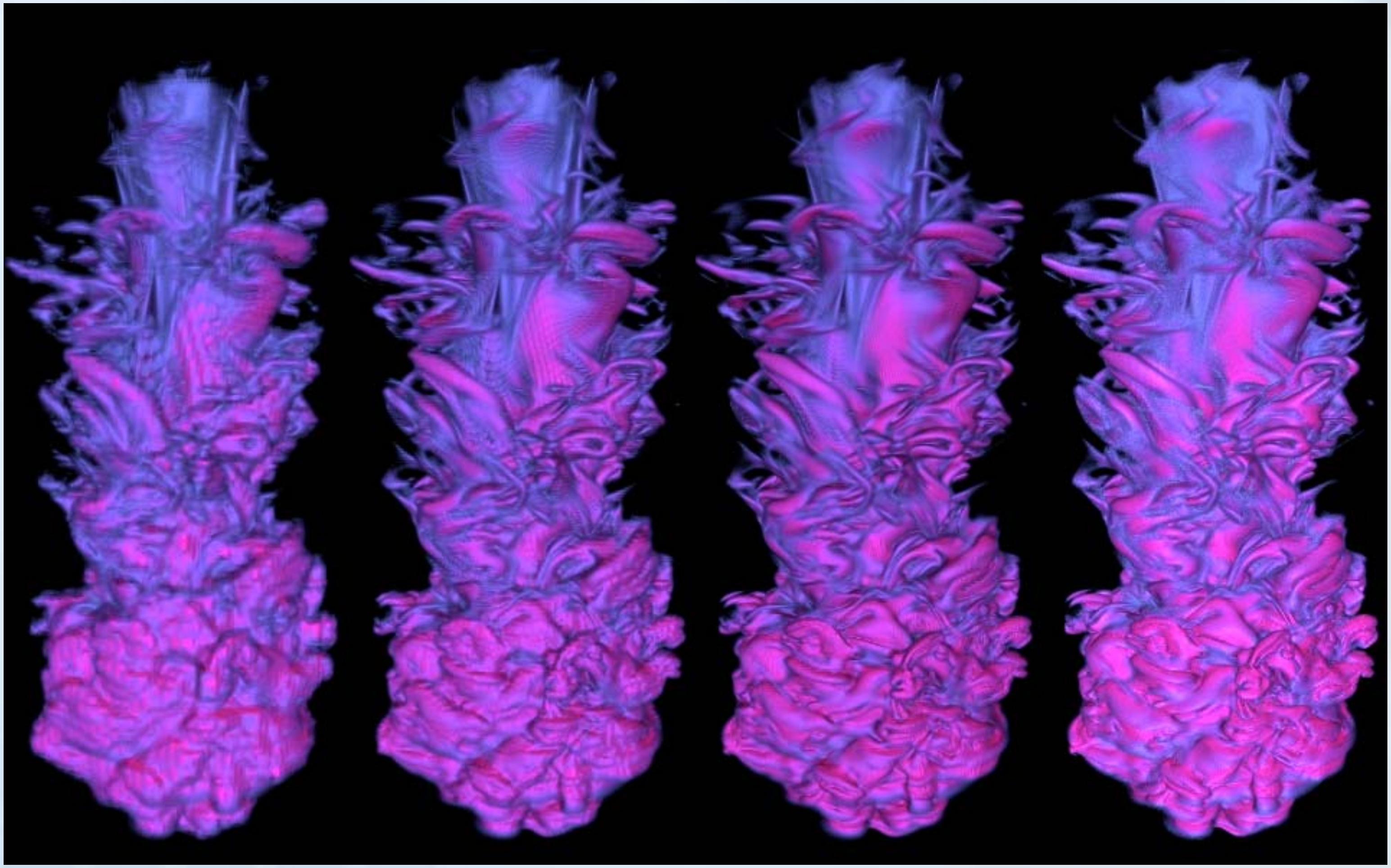
- Scalar
- Single precision

System

- Linux RHEL 3.0
- 2 x Intel 3.4 GHz Xeon EMT64
- 8 GBs RAM
- 1Gb/sec Fibre Channel storage

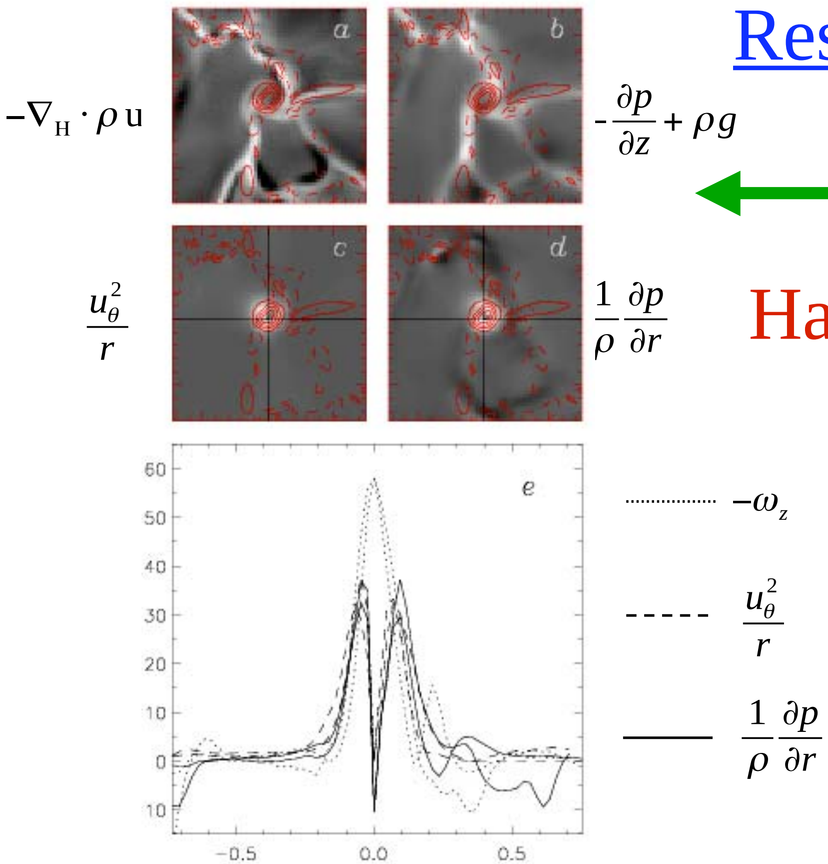
Gains in microprocessor technology enable transforms at very low cost

Solar thermal plume at varying resolutions (compressions) under frequency truncation method



63^2x256	126^2x512	252^2x1024	504^2x2048
(512:1)	(64:1)	(8:1)	(native)

A test of multiresolution analysis: Force balance in supersonic downflows

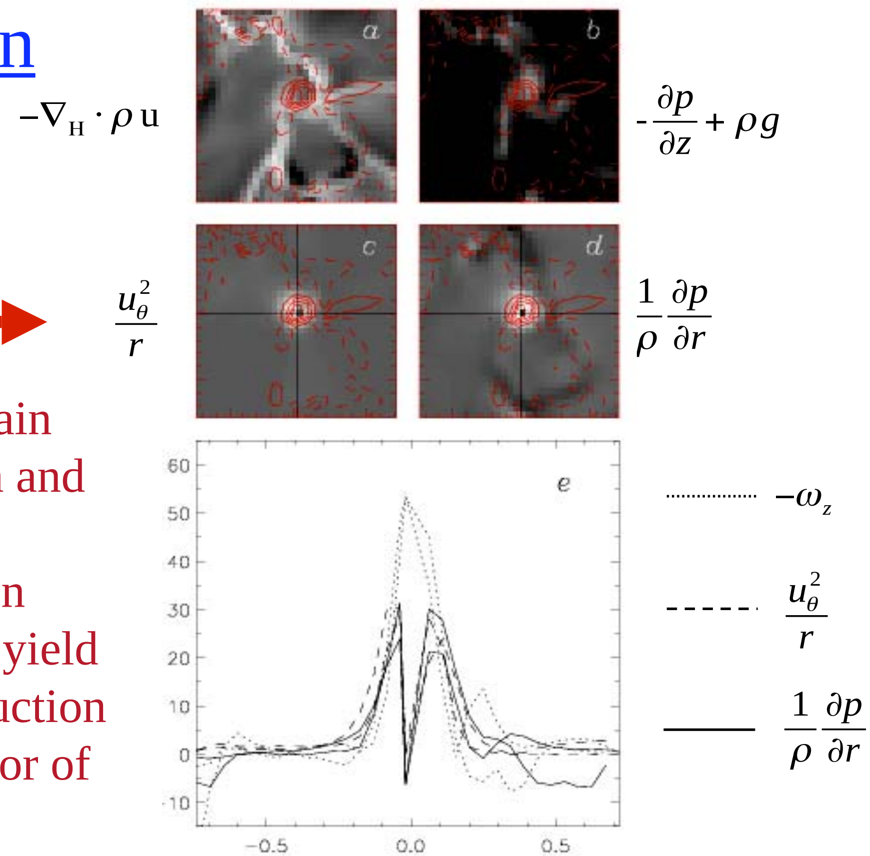


Resolution

Full

Half

Subdomain selection and reduced resolution together yield data reduction by a factor of 128!!!



Sites of supersonic downflow are also those of very high vertical vorticity. The cores of the vortex tubes are evacuated, with centripetal acceleration balancing that due to the inward directed pressure gradient. Buoyancy forces are maximum on the tube periphery due to mass flux convergence.

The same interpretation results from analysis at half resolution.

Courtesy Mark Rast, 2004