

## LA-UR-16-27385

Approved for public release; distribution is unlimited.

Title: Wavelet Decomposition using VTK-m

Author(s): Li, Shaomeng  
Sewell, Christopher Meyer

Intended for: Report

Issued: 2016-09-27

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Wavelet Transforms using VTK-m

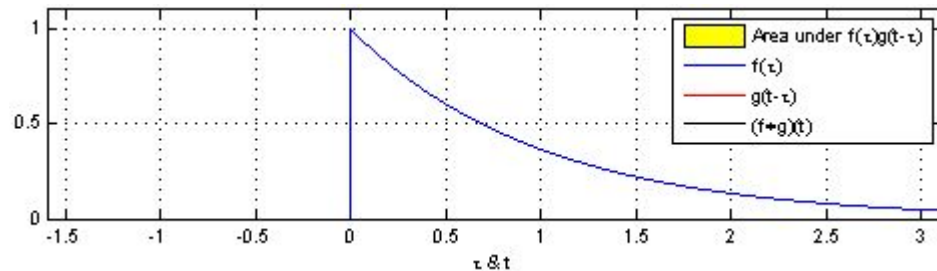
Samuel Li

# Outline

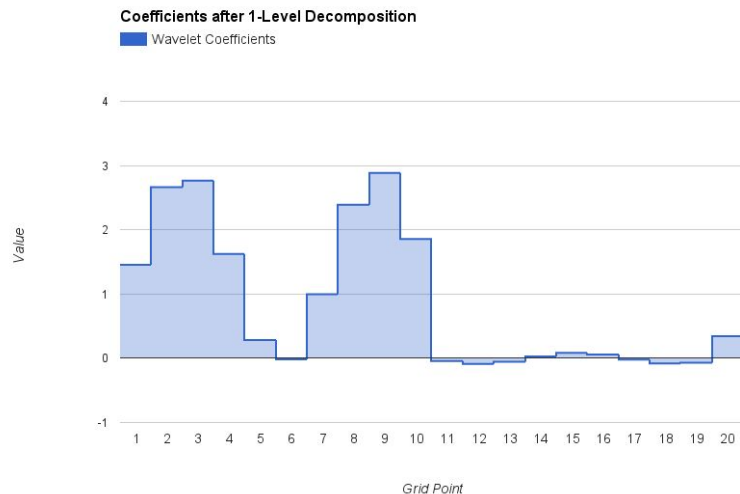
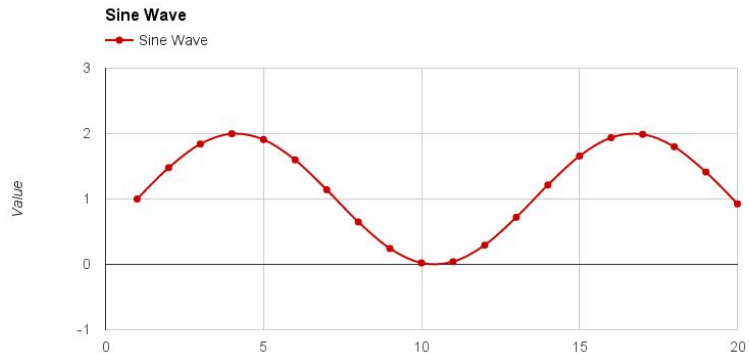
1. Wavelets 101
2. VTK-m 101
3. Wavelets + VTK-m: Performance
4. Wavelets + VTK-m: Accuracy
5. Lesson learned and conclusion

# Wavelet 101

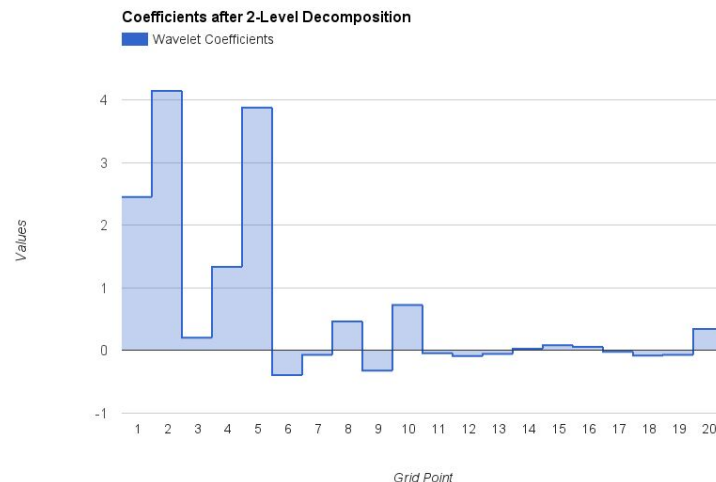
- Essence: a window sliding over the input array doing convolutions.  
(Animation from Wikipedia)
- Sliding windows == wavelet kernels
- Outcome: wavelet coefficients
  - approximate coeffs
  - detail coeffs
- Apply on X, Y, Z, and T dimensions to take advantage of coherence along each axis.



# Wavelet 101: Compression



- Wavelet transform concentrate information into few coefficients.
- Compression: only keep large coefficients (coefficient prioritization)
- Example on Sine wave:



# Outline

1. Wavelets 101
2. VTK-m 101
3. Wavelets + VTK-m: Performance
4. Wavelets + VTK-m: Accuracy
5. Lesson learned and conclusion

# VTK-m 101

- Next generation of VTK
- Massive parallelism on multiple backends: TBB, CUDA
- One code runs on all supported backends.
- A worklet is a basic computational unit doing calculations:
  - In my case, a worklet calculates the convolution at an index position.
  - Every core executes the same worklet at different index positions.
- (I'm using the parallel framework rather than vis algorithms)

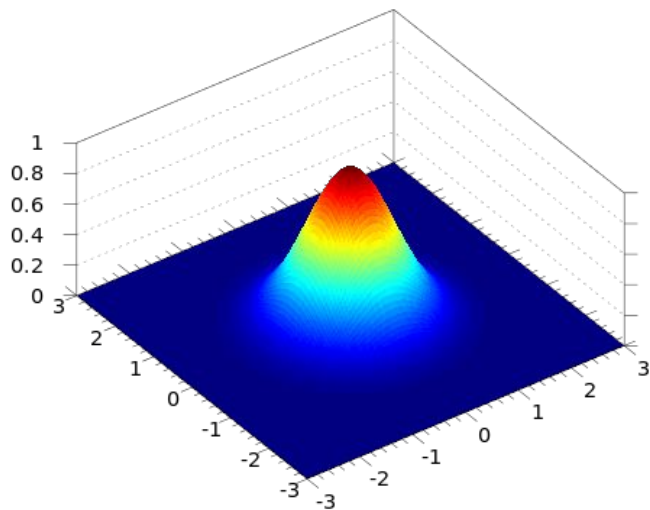


# Outline

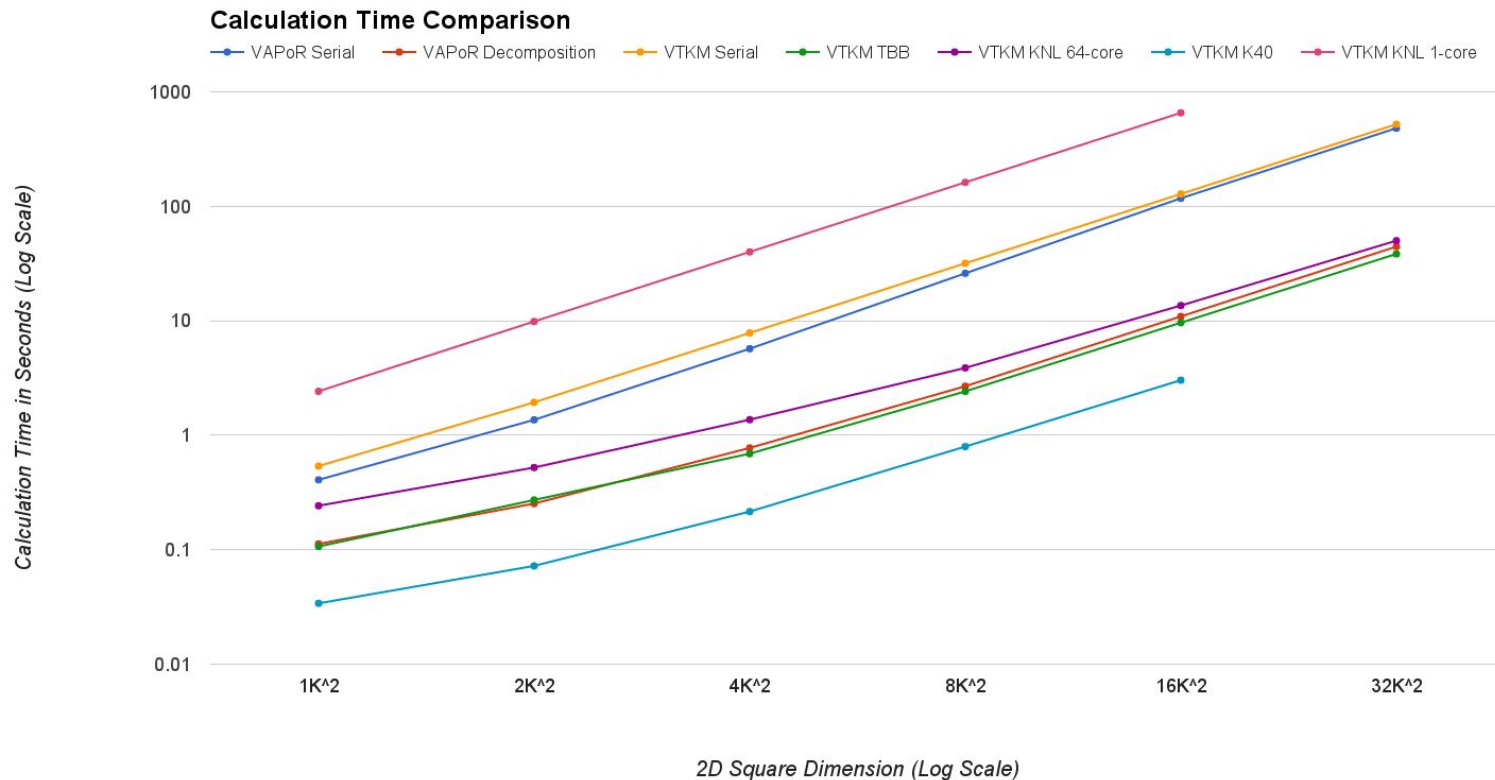
1. Wavelets 101
2. VTK-m 101
3. Wavelets + VTK-m: Performance
4. Wavelets + VTK-m: Accuracy
5. Lesson learned and conclusion

# Wavelets + VTK-m: Performance Comparison

- Parallelism strategy for wavelet transforms:
  - Spatial decomposition + (pthreads, OpenMP, etc.)
  - VTK-m
- Test data: 2D Gaussian distribution in double precision (image from Wikipedia)
- Five techniques to compare:
  - Serial VAPoR (reference implementation)
  - Serial VTK-m
  - Spatial decomposition with VAPoR (OpenMP, 16 cores)
  - VTK-m with TBB (16 cores)
  - Tesla K40 GPU
- Increase 2D square size from  $1,024^2$  to  $32,768^2$



# Wavelets + VTK-m: Performance Comparison



# Outline

1. Wavelets 101
2. VTK-m 101
3. Wavelets + VTK-m: Performance
4. Wavelets + VTK-m: Accuracy
5. Lesson learned and conclusion

# Wavelets + VTK-m: Accuracy Comparison

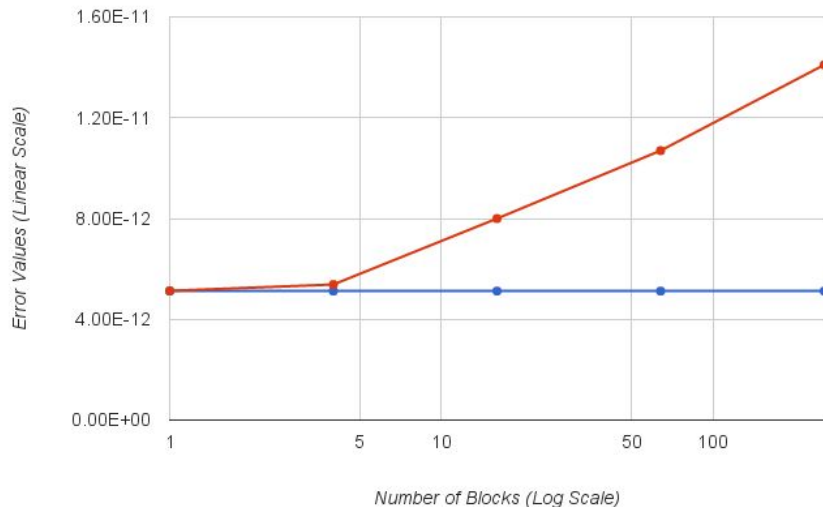
- Accuracy measurements:
  - **L-infinity norm**: maximum point-wise difference
  - **Root Mean Square Error (RMSE)**: average point-wise difference (similar to SNR)
  - Normalize both L-infinity and RMSE measurements by the data range.
- Reminder: compression strategy is to threshold smaller coefficients.  
E.g. 100:1 means to keep a hundredth largest coefficients.
- Parallel techniques may have impacts on accuracy:
  - VTK-m keeps largest coefficients **globally**
  - Spatial decomposition keeps largest coefficients of each block, or **locally**
- Assumption: VTK-m provides better accuracy, especially when spatial decomposition has too many blocks.

# Wavelets + VTK-m: Accuracy Comparison

- Compression level: 100:1.
- Spatial Decomposition use number of blocks: 1, 4, 16, 64, 256.
- VTK-m has one accuracy number.

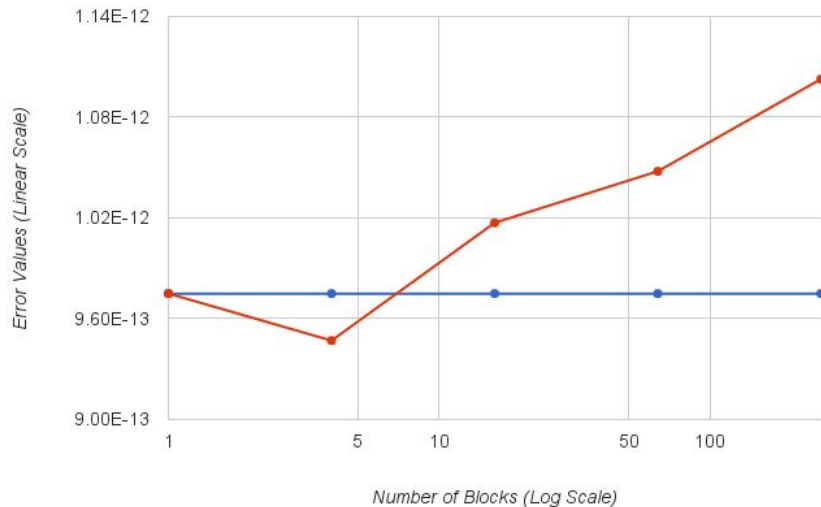
L-infinity norm at 100:1

— VTKM — Spatial Decomposition



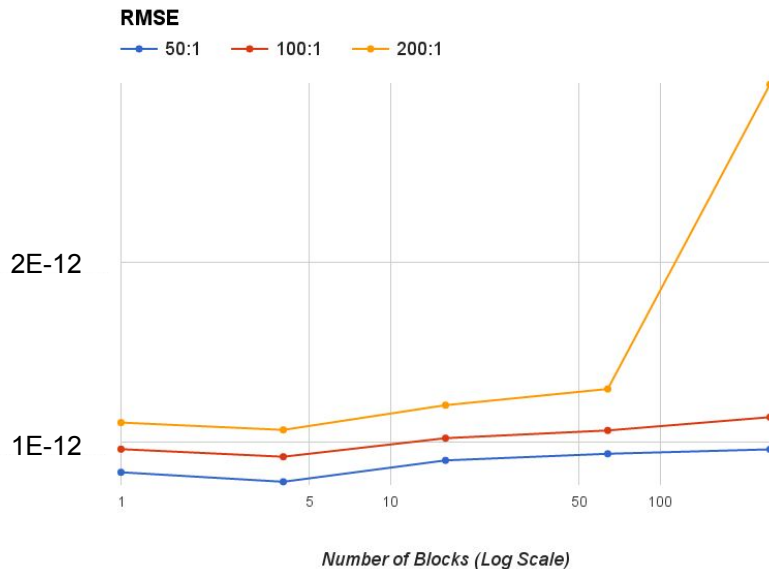
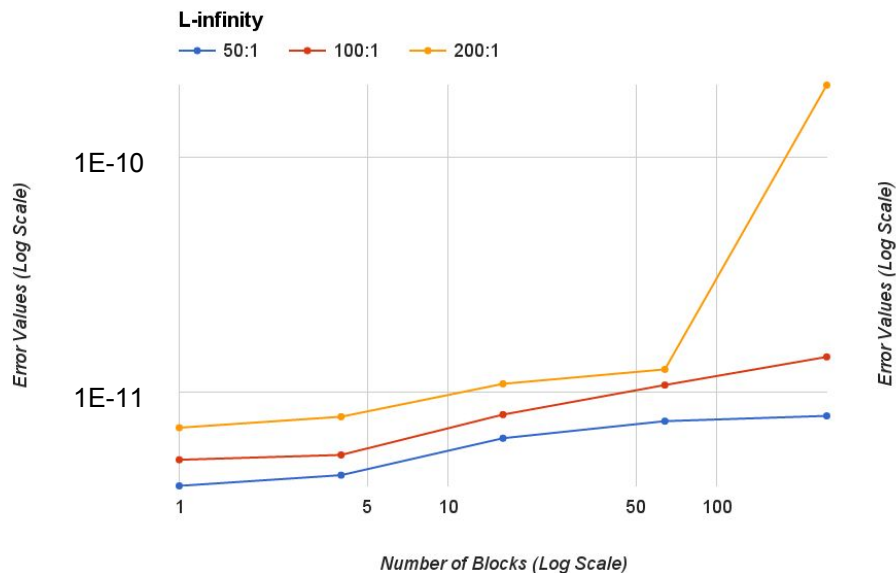
RMSE at 100:1

— VTKM — Spatial Decomposition



# Impact on Accuracy by Spatial Decomposition

- Impact of spatial decomposition at different compression ratios  
--- 50:1, 100:1, and 200:1
- Number of blocks are 1, 4, 16, 64, 256.



# Outline

1. Wavelets 101
2. VTK-m 101
3. Wavelets + VTK-m: Performance
4. Wavelets + VTK-m: Accuracy
5. Lesson learned and Conclusion



# Lessons Learned

- Launching worklets is expensive.
- Natural logic of performing 2D wavelet transform:
  - Repeat the same 1D wavelet transform on every **row**;
  - Repeat the same 1D wavelet transform on every **column**.
  - Invoke the 1D wavelet worklet every time: **num\_rows x num\_columns**
- VTK-m approach of performing 2D wavelet transform:
  - Create a worklet for 2D that handles both rows and columns
  - Invoke this new worklet only one time.
- Fast calculation, but cannot reuse 1D implementations.

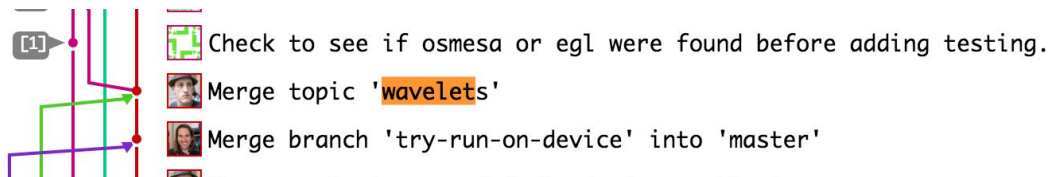
# Wavelets + VTK-m: Conclusion

- ++ One code runs on GPUs as well (and a lot faster)
- ++ Better accuracy compared to spatial decomposition
- +/- Comparable performance on multi-core CPUs.
- -- There's a Learning curve to program with VTK-m

# What is done, and what is next

- What's done:

- Wavelet Transform in 1D gets merged to VTK-m mainstream.
- Wavelet Transform in 2D is ready to merge.



- What's next

- Finish Wavelet Transform in 3D
- Write a paper to report these results.