Use of the Fast Fourier Transform in Solving Partial Differential Equations

B.K. Muite

http://kodu.ut.ee/~benson
http://en.wikibooks.org/wiki/Parallel_Spectral_Numerical_Methods

7 March 2018

Outline

- Motivation
- A warm up example: KdV equation
- Navier-Stokes equation
- Klein-Gordon equation
- Outlook

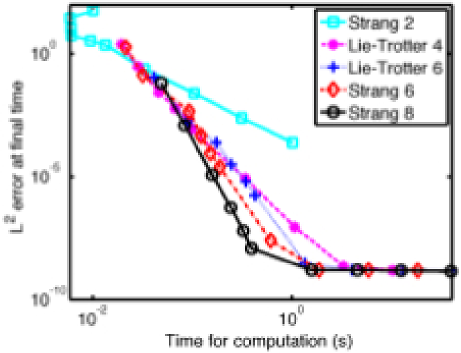
Motivation

- Kassam 2003 “Solving PDEs 10 times faster”
- Use FFT to allow investigation of solutions to partial differential equations
- Fast, accurate, easy to modify if appropriate infrastructure is available
- For exploration, ability to quickly program and run is important
- Usually want to start at desktop scale simulation and transition to supercomputer to allow for wider exploration of phenomena encoded in the differential equation
- Data storage, analysis and visualization can be a challenge for large scale simulations
- Most cases consider semi-linear partial differential equations

Kassam 2003 “Solving PDEs 10 times faster”

- Demonstrate work vs. accuracy for different time stepping schemes - original presentation included figure from Kassam (2003)
- Convergence of exponential time differencing fourth order Runge Kutta method for the 2D Gray-Scott equations from Kassam (2003) “Solving PDEs 10 times faster” `http://eprints.maths.ox.ac.uk/1192/1/NA-03-16.pdf`

Computational Efficiency for Gray Scott equations



- Computational efficiency for solving the Gray Scott equations using higher order time stepping schemes, M.T. Warnez and B.K. Muite “Reduced temporal convergence rates in high-order splitting schemes” `https://arxiv.org/pdf/1310.3901.pdf`

KdV-Burgers equation

- $$u_t = uu_x + \alpha u_{xx} - \beta u_{xxx}$$
- Simple case where can examine interplay between dispersion and dissipation
- $$\frac{\mathrm{d}}{\mathrm{d}t}\int \frac{u^2}{2}\mathrm{d}x = \int \frac{\mathrm{d}}{\mathrm{d}x}\frac{u^3}{3} - \alpha u_x^2 + \beta \frac{\mathrm{d}}{\mathrm{d}x}\frac{u_x^2}{2}\mathrm{d}x$$
- $$\frac{\mathrm{d}}{\mathrm{d}t}\int \frac{u^2}{2}\mathrm{d}x = -\alpha \int u_x^2 \mathrm{d}x$$
- Want numerical method to also reflect conservation laws
- For small values of α and ϵ can get small scale spatial and temporal features that require high resolution
- Want high order methods in space and time

Time stepping methods

- Usually 4th order Runge-Kutta
- Tend to prefer methods that satisfy energy like conserved quantities in the equation, implicit midpoint rule, implicit Runge-Kutta
- Give long time simulations that should be closer to a typical solution of the real equation
- Implicit time stepping requires iteration for nonlinear terms - not ideal for Fourier transform, though in many cases fixed point iteration is ok.

KdV-Burgers equation

- $$u_t = uu_x + \alpha u_{xx} - \beta u_{xxx}$$
- Implicit midpoint rule

$$\begin{aligned}\frac{u^{n+1}-u^n}{\delta t} =& \frac{\left(u^{n+1}+u^n\right)\left(u_x^{n+1}+u_x^n\right)}{4} \\ &+ \frac{\alpha}{2}\left(u_{xx}^{n+1}+u_{xx}^n\right) - \frac{\beta}{2}\left(u_{xxx}^{n+1}+u_{xxx}^n\right)\end{aligned}$$

KdV-Burgers equation

- $$u_t = uu_x + \alpha u_{xx} - \beta u_{xxx}$$
- Second order backward differentiation and second order extrapolation

$$\begin{aligned}\frac{3u^{n+1} - 2u^n + u^{n-1}}{\delta t} =& 2u^n u_x^n - u^{n-1} u_x^{n-1} \\ & + \alpha u_{xx}^{n+1} - \beta u_{xxx}^{n+1}\end{aligned}$$

KdV-Burgers equation

- $$u_t = uu_x + \alpha u_{xx} - \beta u_{xxx}$$
- Second order Strang splitting

$$\frac{u^{n+1/3} - u^n}{0.5\delta t} = \frac{\left(u^{n+1/3} + u^n\right)\left(u_x^{n+1/3} + u_x^n\right)}{4}$$

$$\hat{u}^{n+2/3} = \exp\left[\left(\alpha k_x^2 - \beta k_x^3\right)\delta t\right]\hat{u}^{n+1/3}$$

$$\frac{u^{n+1} - u^{n+2/3}}{0.5\delta t} = \frac{\left(u^{n+1} + u^{n+2/3}\right)\left(u_x^{n+1} + u_x^{n+2/3}\right)}{4}$$

- Can use a finite volume scheme for Burgers equation $u_t = uu_x$

KdV-Burgers equation

- $$u_t = uu_x + \alpha u_{xx} - \beta u_{xxx}$$
- Carpenter Kennedy Runge Kutta

procedure RUNGE–KUTTA($\mathbf{u}$)
    $\mathbf{h} = \mathbf{0}$
    $\mathbf{u} = \mathbf{u}^n$
    for $k = 1 \to 5$ **do**
        $\mathbf{h} \leftarrow \mathbf{g}(\mathbf{u}) + \sigma_k \mathbf{h}$
        $\mu = 0.5\delta t(\zeta_{k+1} - \zeta_k)$
        $\mathbf{v} - \mu\mathbf{l}(\mathbf{v}) = \mathbf{u} + \gamma_k \delta t\mathbf{h} + \mu\mathbf{l}(\mathbf{u})$
        $\mathbf{u} \leftarrow \mathbf{v}$
    end for
    $\mathbf{u}^{n+1} = \mathbf{u}$
end procedure

- $\mathbf{l}(u) = \alpha u_{xx} - \beta u_{xxx}$
- $\mathbf{g}(u) = uu_x$

KdV-Burgers equation

- $$u_t = uu_x + \alpha u_{xx} - \beta u_{xxx}$$
- Fourth order implicit Runge Kutta

$$
\begin{aligned}
f(Y) &:= YY_x + \alpha Y_{xx} - \beta Y_{xxx} \\
Y_1 &= u^n + (\delta t)\left[\frac{1}{4} f(Y_1) + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right) f(Y_2)\right] \\
Y_2 &= u^n + (\delta t)\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right) f(Y_1) + \frac{1}{4} f(Y_2)\right] \\
u^{n+1} &= u^n + 0.5\,(\delta t)\,[Y_1 + Y_2]
\end{aligned}
$$

- use fixed point iteration and FFT

3D Navier-Stokes equation Equivalent Formulation

- Simplification of equation with periodic boundary conditions

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\Delta\mathbf{u} \quad (1)$$

$$\nabla\cdot\mathbf{u} = 0 \quad (2)$$

so

$$\nabla\cdot\left[\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)\right] = \nabla\cdot\left[-\nabla p + \mu\Delta\mathbf{u}\right] \quad (3)$$

$$\rho\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u}) = -\Delta p \quad (4)$$

$$p = \Delta^{-1}\left[\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u})\right] \quad (5)$$

so

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\rho\nabla\left(\Delta^{-1}\left[\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u})\right]\right) + \mu\Delta\mathbf{u} \quad (6)$$

3D Equivalent Formulation - Implicit Midpoint Time Discretization

- $$\rho\left[\frac{\mathbf{u}^{n+1,j+1}-\mathbf{u}^{n}}{\delta t}+\frac{\mathbf{u}^{n+1,j}+\mathbf{u}^{n}}{2}\cdot\nabla\left(\frac{\mathbf{u}^{n+1,j}+\mathbf{u}^{n}}{2}\right)\right]$$
$$=\rho\frac{\nabla\left[\Delta^{-1}\left(\nabla\cdot\left[(\mathbf{u}^{n+1,j}+\mathbf{u}^{n})\cdot\nabla(\mathbf{u}^{n+1,j}+\mathbf{u}^{n})\right]\right)\right]}{4}$$
$$+\mu\Delta\frac{\mathbf{u}^{n+1,j+1}+\mathbf{u}^{n}}{2},$$
- Video of Taylor Green Vortex
`http://vimeo.com/87981782`

3D Equivalent Formulation - Carpenter-Kennedy Discretization

- **procedure** RUNGE–KUTTA($\mathbf{u}$)
  - $\mathbf{h} = \mathbf{0}$
  - $\mathbf{u} = \mathbf{u}^n$
  - **for** $k = 1 \to 5$ **do**
    - $\mathbf{h} \leftarrow \mathbf{g}(\mathbf{u}) + \beta_k \mathbf{h}$
    - $\mu = 0.5\delta t(\alpha_{k+1} - \alpha_k)$
    - $\mathbf{v} - \mu \mathbf{l}(\mathbf{v}) = \mathbf{u} + \gamma_k \delta t \mathbf{h} + \mu \mathbf{l}(\mathbf{u})$
    - $\mathbf{u} \leftarrow \mathbf{v}$
  - **end for**
  - $\mathbf{u}^{n+1} = \mathbf{u}$
- **end procedure**

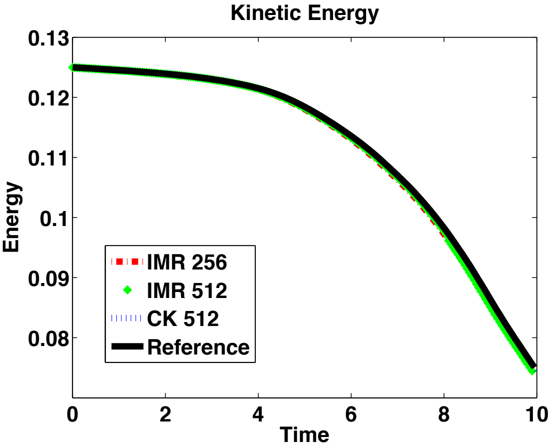
Performance

- $\delta t = 0.005$ for 512^3 grid points.
- For IMR scheme, fixed point iteration procedure was stopped once the difference between two successive iterates was less than 10^{-10} in l^∞ norm of velocity fields.

Method	Grid Size	Cores	Time Steps	Time (s)	$\frac{\text{Core Hours}}{\text{Timestep}}$
IMR	512^3	1024	500	9899	5.68
CK	512^3	4096	2000	7040	4.0

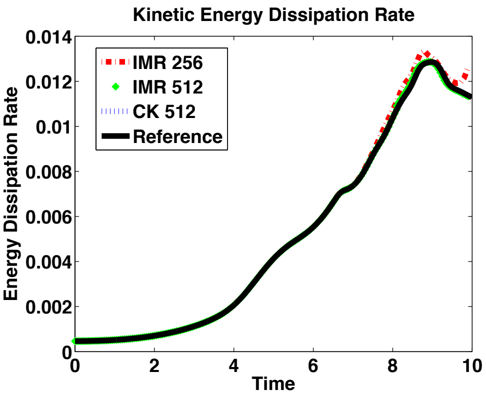
Performance of Fourier pseudospectral code on Shaheen. IMR is an abbreviation for implicit midpoint rule and CK is an abbreviation for Carpenter–Kennedy.

Kinetic Energy Evolution



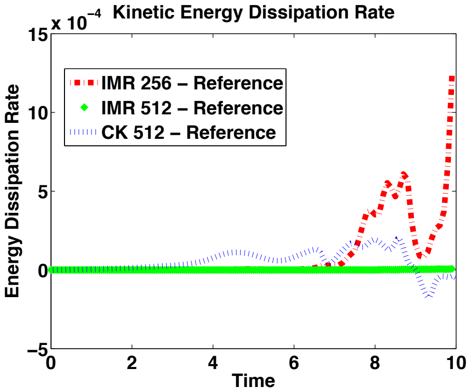
KE of solutions are so close they are almost indistinguishable

Kinetic Energy Dissipation Rate



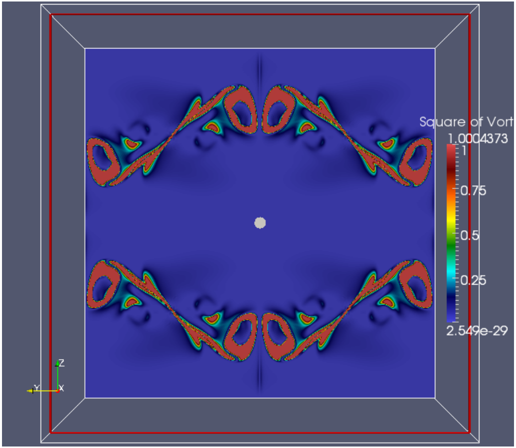
Plot during the initial stage, where flow is essentially inviscid and laminar. Fully developed turbulent flow is observed around $t_{max} \approx 8$.

Kinetic Energy Dissipation Rate



Difference in kinetic energy dissipation rates between the current discretizations and the reference solution.

Vorticity



Square of the vorticity in the plane centered at $(\pi, 0, 0)$ with normal vector $(1, 0, 0)$.

Discrete energy equality for midpoint rule

- $$\|u(t=T)\|_{l^2}^2 - \|u(t=0)\|_{l^2}^2 = -\mu \int_0^T \|\nabla u\|_{l^2}^2 \mathrm{d}t$$

- $$\|u^N\|_{l^2}^2 - \|u^0\|_{l^2}^2 = -\frac{\mu}{4} \sum_{n=0}^{N-1} \left\| \nabla \left(U^n + U^{n+1} \right) \right\|_{l^2}^2 \delta t.$$

Conclusion on Navier Stokes Equations

- At almost the same computational cost, both 2nd-order accurate IMR and 4th-order Carpenter-Kennedy time stepping method, capture same amount of detail of the flow for 512^3.

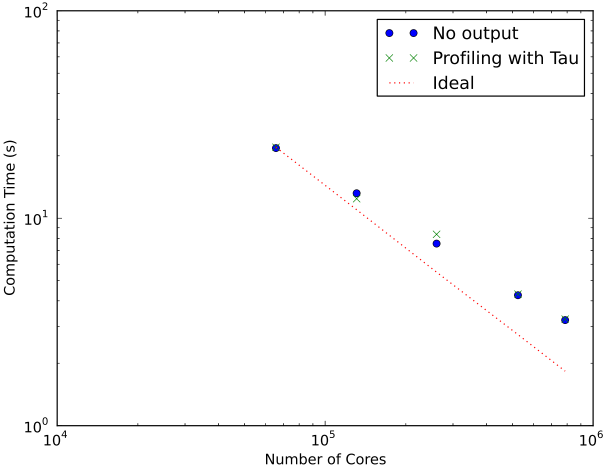
The Real Cubic Klein-Gordon Equation

- $$u_{tt} - \Delta u + u = |u|^2 u$$
- Full application benchmark
- Depending on temporal discretization, can incorporate solution of linear system
- Can also incorporate accuracy
- With an instrumented reference code, can obtain a large number of system characteristics
- $$E(u, u_t) = \int \frac{1}{2}|u_t|^2 + \frac{1}{2}|u|^2 + \frac{1}{2}\left|\nabla u\right|^2 - \frac{1}{4}\left|u\right|^4 \mathrm{d}\boldsymbol{x}$$

Videos by Brian Leu, Albert Liu, Michael Quell and Parth Sheth

- `http://www-personal.umich.edu/~brianleu/`
- `http://www.michaelquell.at/`

Scaling study Brian Leu, Albert Liu, and Parth Sheth

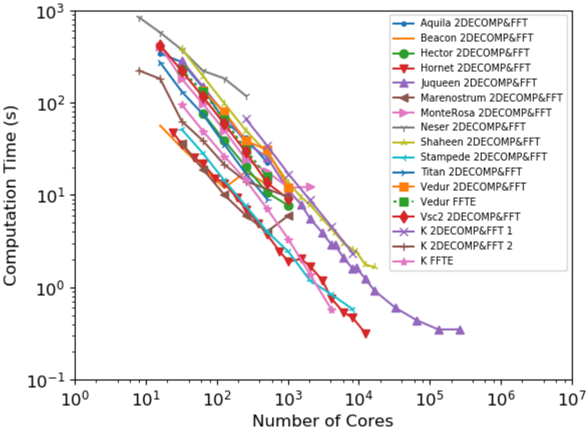


- Strong scaling on Mira for a 4096^3 discretization

Numerical Scheme

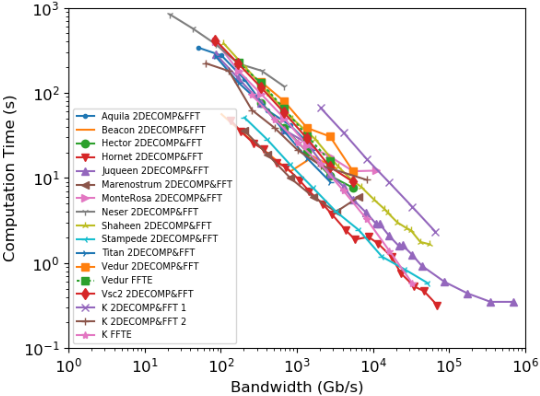
- $$\frac{u^{n+1}-2u^n+u^{n-1}}{\delta t^2}-\Delta\frac{u^{n+1}+2u^n+u^{n-1}}{4}+\frac{u^{n+1}+2u^n+u^{n-1}}{4}=\left|u^n\right|^2u^n$$
- $u^n \approx u(n\delta t, x, y, z)$
- Time stepping takes place in Fourier space
- Solution of linear system of equations is easy in Fourier space, though can also be done by iterative methods in real space
- Two FFTs per time step

Scaling with Cores



- Scaling results showing computation time for 30 time steps as a function of the number of processor cores. A discretization of 512^3 points was used.

Scaling with Cores



- Scaling results showing computation time for 30 time steps as a function of total on chip bandwidth defined as the maximum theoretical bandwidth from RAM on a node multiplied by the number of nodes used. A discretization of 512^3 points was used.

A Runtime Estimation Model

- d_1, d_2, d_3 system and implementation dependent constants
- N number of grid points in each dimension, assumed to be the same in all three dimensions
- L_n minimum network latency, B_c average bandwidth to a core from RAM
- p number of processes
- Assume a hypercube network - speed optimal for FFT
- $$\frac{d_1 \times N^3 + d_2 \times [N \log(N)]^3}{B_c \times p} + 2L_n + d_3 \log(p)$$

A Ranking

Rank	Machine Name	Time (s)	Cores used	Manufacturer and Model	Node Type	Total Cores
1	Hornet	0.319	12,288	Cray XC40	2x12 core Intel Xeon 2.5 GHz E5-2680v3	94,656
2	Juqueen	0.350	262,144	IBM Blue Gene/Q	16 core 1.6 GHz Power PC A2	458,752
3	Stampede	0.581	8,162	Dell Power Edge	2x8 core Intel Xeon 2.7 GHz E5-2680	462,462
4	Shaheen	1.66	16,384	IBM Blue Gene/P	4 core 0.85 GHz PowerPC 450	65,536
5	K computer	2.346	8,192	Fujitsu	1x8 core SPARC64 VIIIFX	663,552
6	MareNostrum III	4.00	64	IBM DataPlex	2x8 core Intel Xeon 2.6 GHz E5-2670	48,384
7	Hector	7.66	1024	Cray XE6	2x16 core AMD Opteron 2.3 GHz 6276 16C	90,112
8	VSC2	9.03	1024	Megware	2x8 core AMD Opteron 2.2 GHz 6132HE	21,024
9	Beacon	9.13	256	Appro	2x8 core Intel Xeon 2.6 GHz E5-2670	768
10	Monte Rosa	11.9	1,024	Cray XE6	2x16 core AMD Opteron 2.1 GHz 6272	47,872
11	Titan	17.0	256	Cray XK7	16 core AMD Opteron 2.2 GHz 6274	299,008
12	Vedur	18.6	1,024	HP ProLiant DL165 G7	2x16 core AMD Opteron 2.3 GHz 6276	2,560
13	Aquila	22.4	256	ClusterVision	2x4 core Intel Xeon 2.8 GHz E5462	800
14	Neser	118.7	128	IBM System X3550	2x4 core Intel Xeon 2.5 GHz E5420	1,024

A Ranking

Rank	Machine Name	Time (s)	Total Cores	Interconnect	1D FFT Library	Chip Bandwidth Gb/s	Theoretical Peak TFLOP/s
1	Hornet	0.319	94,656	Cray Aries	FFTW 3	68	3,784
2	Juqueen	0.350	458,752	IBM 5D torus	ESSL	42.6	5,872
3	Stampede	0.581	462,462	FDR infiniband	Intel MKL	51.2	2,210
4	Shaheen	1.66	65,536	IBM 3D torus	ESSL	13.6	222.8
5	K computer	2.346	663,552	Fujistu Tofu	FFTW 3	64	10,620
6	MareNostrum III	4.00	48,384	FDR10 infiniband	Intel MKL	51.2	1,017
7	Hector	7.66	90,112	Cray Gemini	ACML	85	829.0
8	VSC2	9.03	21,024	QDR infiniband	FFTW 3	42.8	185.0
9	Beacon	9.13	768	FDR infiniband	Intel MKL	51.2	16.0
10	Monte Rosa	11.9	47,872	Cray Gemini	ACML	85	402.1
11	Titan	17.0	299,008	Cray Gemini	ACML	85	2,631
12	Vedur	18.6	2,560	QDR infiniband	FFTW 3	85	236
13	Aquila	22.4	800	DDR infiniband	FFTW 3	12.8	8.96
14	Neser	118.7	1,024	Gigabit ethernet	FFTW 3	10.7	10.2

Outlook

- For numerical solution of partial differential equations, want to find solution accuracy and time to solution as a function of computational cost or computational energy
- For computer scientist, algorithm efficiency is easier to measure
- Some effort needs to be made to relate solution accuracy and time to solution in addition to computational efficiency
- For large simulations, visualization and I-O are also bottlenecks
- For FFT overlapping computation and communication, both within FFT and the overall computation may enable better time to solution
- Spectral element method would better fit current architectures, however increased code complexity a limiting factor. Accuracy also a limiting factor – more floating point computations typically implies more floating point errors.

Acknowledgements

The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by:

- RIKEN
- The Beacon Project at the University of Tennessee;
- The UK's national high-performance computing service;
- The Barcelona Supercomputing Center - Centro Nacional de Supercomputación;
- The Swiss National Supercomputing Centre (CSCS);
- The Texas Advanced Computing Center (TACC) at The University of Texas at Austin;
- The KAUST Supercomputer Laboratory at King Abdullah University of Science and Technology (KAUST);
- The Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory;
- The Aquila HPC service at the University of Bath;
- The Vienna Scientific Cluster (VSC);
- The PRACE research infrastructure resources in Germany at HLRS and FZ Jülich;
- The High Performance Computing Center of the University of Tartu;
- Kraken at the National Institute for Computational Science;
- Trestles at the San Diego Supercomputing Center;
- the University of Michigan High Performance Computing Service FLUX;
- Mira at the Argonne Leadership Computing Facility at Argonne National Laboratory;

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding bodies or the service providers.

Acknowledgements and References

- S. Aseeri, S. Andersson, A. Bauer, B. Cloutier, D. DeMarle, C. Doering, H. Johnston, D. Ketcheson, D. Keyes, R. Krasny, M. Parsani, D. Pekurovsky, M. Pippig, P. Rigge, M. Srinivasan, D. Takahashi, E. Vainikko, J. Whitehead, M. Winkel, B. Wylie, H. Yi and R. Yokota

- S. Aseeri, O. Batrašev, M. Icardi, B.Leu, A. Liu, N. Li, B.K. Muite, E. Müller, B. Palen, M. Quell, H. Servat, P. Sheth, R. Speck, M. Van Moer, J. Vienne, ”Solving the Klein-Gordon equation using Fourier spectral methods: A benchmark test for computer performance” `arXiv:1501.04552` also in 23rd High Performance Computing Symposium (HPC 2015) held in Conjunction with 2015 Spring Simulation Multi-Conference, April 2015.
- Beamer `https://en.wikipedia.org/wiki/Beamer_(LaTeX)`

- European Union Regional Development Fund — Investing in your future — EXCS ESTONIAN COMPUTING — KAUST — MICHIGAN

References

- A.-K. Kassam, "Solving PDEs 10 times faster" `http://eprints.maths.ox.ac.uk/1192/1/NA-03-16.pdf`
- B. Cloutier, B.K. Muite and M. Parsani, "Case 3.5: Fourier Psuedo-Spectral Method." Second International Workshop on High-Order CFD Methods, Germany, May 2013
- P. Rigge and B.K. Muite, "Precision effects on Numerical Solutions to the Sine-Gordon equation" `http://www-personal.umich.edu/%7Eriggep/media/files/sg_talk12.pdf`
- M.T. Warnez and B.K. Muite "Reduced temporal convergence rates in high-order splitting schemes" `https://arxiv.org/pdf/1310.3901.pdf`

- European Union Regional Development Fund — Investing in your future — EXCS ESTONIAN COMPUTING — KAUST — MICHIGAN