Experiences using 2decomp&fft to solve Partial Differential Equations using Fourier Spectral Methods

Sudarshan Balakrishnan, Abdullah H. Bargash, Gong Chen, Brandon Cloutier, Ning Li, Dave Malicke, Benson Muite, Michael Quell, Paul Rigge, Damian San Román Alerigi Mamdouh Solimani, Andre Souza, Abdulaziz S.Thiban, Mark Van Moer, Jeremy West

Tartu Ülikool
benson.muite@ut.ee
http://math.ut.ee/~benson
http://en.wikibooks.org/wiki/Parallel_Spectral_Numerical_Methods

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

Outline

- Project Aim
- Fourier Series
- The Heat Equation
- The Allen Cahn Equation
- The Gray-Scott Equations
- Nonlinear Schrödinger Equation
- Navier-Stokes Equation
- Maxwell's Equations
- Possible Further Work

Project Aim

- Teaching tool for use in mathematics and computer science courses from 1st year undergraduate to postgraduate level
- Research tool: investigate partial differential equations, investigate computer performance
- Help paper reproducibility and verifiability
- Reduce coding time
- Rely on 2decomp&fft for MPI parallelization (`http://2decomp.org`)
- Material has been tested in a course on Multivariable calculus and on an introduction to partial differential equations

Fourier series: Separation of Variables 1

$$
\begin{aligned}
\frac{dy}{dt} &= y &(1)\\
\frac{dy}{y} &= dt &(2)\\
\int \frac{dy}{y} &= \int dt &(3)\\
\ln y + a &= t + b &(4)\\
e^{\ln y + a} &= e^{t+b} &(5)\\
e^{\ln y} e^{a} &= e^{t} e^{b} &(6)\\
y &= \frac{e^{b}}{e^{a}} e^{t} &(7)\\
y(t) &= ce^{t}. &(8)
\end{aligned}
$$

Fourier series: Separation of Variables 2

- $$u_t = u_{xx} \tag{9}$$
- Suppose $u = X(x)T(t)$
- $$\frac{\frac{\mathrm{d}T}{\mathrm{d}t}(t)}{T(t)} = \frac{\frac{\mathrm{d}^2X}{\mathrm{d}x^2}(x)}{X(x)} = -C, \tag{10}$$
- Solving each of these separately and then using linearity we get a general solution
- $$\sum_n \alpha_n \exp(-C_n t) \sin(\sqrt{C_n} x) + \beta_n \exp(-C_n t) \cos(\sqrt{C_n} x) \tag{11}$$

- How do we find a particular solution?
- Suppose $u(x, t = 0) = f(x)$
- Suppose $u(0, t) = u(2\pi, t)$ and $u_x(0, t) = u_x(2\pi, t)$ then recall
- $$\int_0^{2\pi} \sin(nx)\sin(mx) = \begin{cases} \pi & m = n \\ 0 & m \neq n \end{cases}, \tag{12}$$
  $$\int_0^{2\pi} \cos(nx)\cos(mx) = \begin{cases} \pi & m = n \\ 0 & m \neq n \end{cases}, \tag{13}$$
  $$\int_0^{2\pi} \cos(nx)\sin(mx) = 0. \tag{14}$$

Fourier series: Separation of Variables 4

- So if

$$f(x) = \sum_n \alpha_n \sin(nx) + \beta_n \cos(nx). \tag{15}$$

- then

$$\alpha_n = \frac{\int_0^{2\pi} f(x)\sin(nx)\mathrm{d}x}{\int_0^{2\pi} \sin^2(nx)\mathrm{d}x} \tag{16}$$

$$\beta_n = \frac{\int_0^{2\pi} f(x)\cos(nx)\mathrm{d}x}{\int_0^{2\pi} \cos^2(nx)\mathrm{d}x}. \tag{17}$$

- and

$$u(x,t) = \sum_n \exp(-n^2 t)\left[\alpha_n \sin(nx) + \beta_n \cos(nx)\right] \tag{18}$$

- The Fast Fourier Transform allows one to find good approximations to α_n and β_n when the solution is found at a finite number of evenly spaced grid points

The 1D Heat Equation: Finding derivatives and timestepping

- Let

$$u(x) = \sum_k \hat{u}_k \exp(ikx) \tag{19}$$

- then

$$\frac{\mathrm{d}^\nu u}{\mathrm{d}x^\nu} = \sum (ik)^\nu \hat{u}_k. \tag{20}$$

- Consider $u_t = u_{xx}$, which is approximated by

$$\begin{aligned}
\frac{\partial \hat{u}_k}{\partial t} &= \alpha (ik)^2 \hat{u}_k && (21)\\
\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{h} &= \alpha (ik)^2 \hat{u}_k^{n+1} && (22)\\
\hat{u}_k^{n+1}(1 - \alpha h (ik)^2) &= \hat{u}_k^n && (23)\\
\hat{u}_k^{n+1} &= \frac{\hat{u}_k^n}{(1 - \alpha h (ik)^2)}. && (24)
\end{aligned}$$

The 2D Allen Cahn Equation

- Consider $u_t = \epsilon(u_{xx} + u_{yy}) + u - u^3$, which is approximated by

$$\frac{\partial \hat{u}}{\partial t} = \epsilon \left[(ik_x)^2 + (ik_y)^2 \right] \hat{u} + \hat{u} - \hat{u^3} \tag{25}$$

$$\frac{\hat{u}^{n+1} - \hat{u}^n}{h} = \epsilon \left[(ik_x)^2 + (ik_y)^2 \right] \hat{u^{n+1}} + \hat{u^n} - (\hat{u^n})^3 \tag{26}$$

The 3D Gray-Scott Equations

- $$\frac{\partial u}{\partial t} = D_u \Delta u + \alpha (1 - u) - uv^2, \quad (27)$$
  $$\frac{\partial v}{\partial t} = D_v \Delta v - \beta v + uv^2. \quad (28)$$
- Solved using a splitting method (More information on splitting for this at `http://arxiv.org/abs/1310.3901`)
- `http://web.student.tuwien.ac.at/~e1226394/`

The 2D nonlinear Schrödinger Equation

- $$i\psi_t + \psi_{xx} + \psi_{yy} = |\psi|^2\psi$$
- Solved using Fast Fourier Transform and splitting

The 2D nonlinear Schrödinger Equation

Table: Computation times in seconds for 20 time steps of 10^{-5} for a Fourier split step method for the cubic nonlinear Schrödinger equation on $[-5\pi, 5\pi]^2$.

Grid Size	GPU (Cuf)	GPU (Cuda)	GPU (OpenACC)	Xeon Phi (61 cores)	CPU (1 core)
256^2	0.00802	0.0116	0.0130	0.0122	0.442
512^2	0.0234	0.0315	0.0369	0.0291	1.94
1024^2	0.0851	0.105	0.132	0.118	12.7
2048^2	0.334	0.415	0.527	0.422	57.2
4096^2	1.49	2.02	2.30	1.626	329

The Real Cubic Klein-Gordon Equation

- $$u_{tt} - \Delta u + u = |u|^2 u \tag{29}$$
- $$E(u, u_t) = \int \frac{1}{2}|u_t|^2 + \frac{1}{2}|u|^2 + \frac{1}{2}|\nabla u|^2 - \frac{1}{4}|u|^4 \,\mathrm{d}\boldsymbol{x} \tag{30}$$
- $$\frac{u^{n+1} - 2u^n + u^{n-1}}{(\delta t)^2} - \Delta \frac{u^{n+1} + 2u^n + u^{n-1}}{4} + \frac{u^{n+1} + 2u^n + u^{n-1}}{4} \tag{31}$$
  $$= \pm \left|u^n\right|^2 u^n \tag{32}$$
- Parallelization done using 2decomp library for FFT and processing independent loops
- Other time stepping algorithms possible, including splitting

Simulations and Videos by Brian Leu, Albert Liu, and Parth Sheth

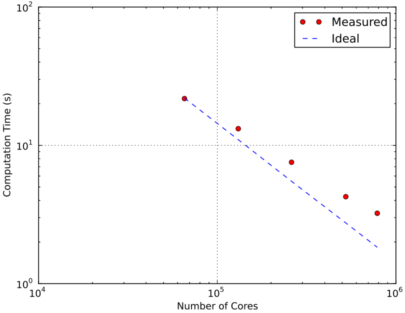

Figure: Strong scaling on Mira for a 4096^3 discretization.

- http://www-personal.umich.edu/~alberliu/
- http://www-personal.umich.edu/~brianleu/
- http://www-personal.umich.edu/~pssheth/

The 2D and 3D Navier Stokes Equation

- Consider incompressible case only
- $$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\Delta\mathbf{u} \qquad (33)$$
  $$\nabla\cdot\mathbf{u} = 0. \qquad (34)$$
- p pressure, μ viscosity, ρ, density
- 2D $\mathbf{u}(x,y) = (u(x,y), v(x,y))$
- 3D $\mathbf{u}(x,y,z) = (u(x,y,z), v(x,y,z), w(x,y,z))$

2D Vorticity-Streamfunction Formulation

- $$\omega = \nabla \times \mathbf{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = -\Delta\psi$$

- $$\rho\left(\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y}\right) = \mu\Delta\omega \tag{35}$$

  and

  $$\Delta\psi = -\omega. \tag{36}$$

Time Discretization

- $$\rho \left[\frac{\omega^{n+1,k+1} - \omega^n}{\delta t} \right. \tag{37}$$
  $$+ \frac{1}{2} \left. \left(u^{n+1,k} \frac{\partial \omega^{n+1,k}}{\partial x} + v^{n+1,k} \frac{\partial \omega^{n+1,k}}{\partial y} + u^n \frac{\partial \omega^n}{\partial x} + v^n \frac{\partial \omega^n}{\partial y} \right) \right]$$
  $$= \frac{\mu}{2} \Delta \left(\omega^{n+1,k+1} + \omega^n \right),$$

  and

  $$\Delta \psi^{n+1,k+1} = -\omega^{n+1,k+1}, \tag{38}$$
  $$u^{n+1,k+1} = \frac{\partial \psi^{n+1,k+1}}{\partial y}, \quad v^{n+1,k+1} = -\frac{\partial \psi^{n+1,k+1}}{\partial x}. \tag{39}$$
- Fixed point iteration used to obtain nonlinear terms

Example Videos

- `http://www-personal.umich.edu/~cloutbra/research.html`
- Simulations on a single NVIDIA Fermi GPU about 20 times faster than a 16 core CPU

3D Equivalent Formulation

- Simplification of equation with periodic boundary conditions

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\Delta\mathbf{u} \tag{40}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{41}$$

so

$$\nabla\cdot\left[\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)\right] = \nabla\cdot\left[-\nabla p + \mu\Delta\mathbf{u}\right] \tag{42}$$

$$\rho\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u}) = -\Delta p \tag{43}$$

$$p = \Delta^{-1}\left[\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u})\right] \tag{44}$$

so

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\rho\nabla\left(\Delta^{-1}\left[\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u})\right]\right) + \mu\Delta\mathbf{u} \tag{45}$$

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 and $\delta t = 0.01$ for 256^3 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	256^3	512	1000	4060	0.578
IMR	512^3	1024	500	9899	5.68
CK	512^3	4096	2000	7040	4.0

Table: 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

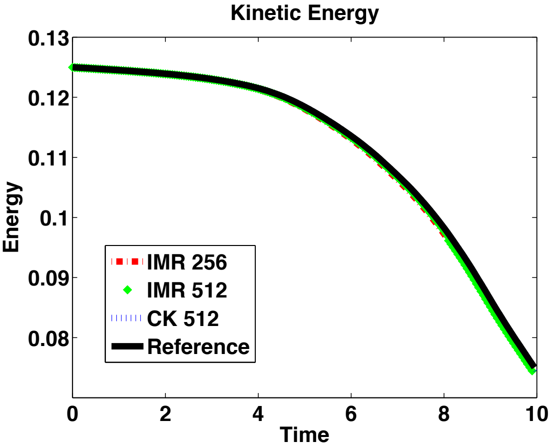

Figure: KE of solutions are so close they are almost indistinguishable

Kinetic Energy Dissipation Rate

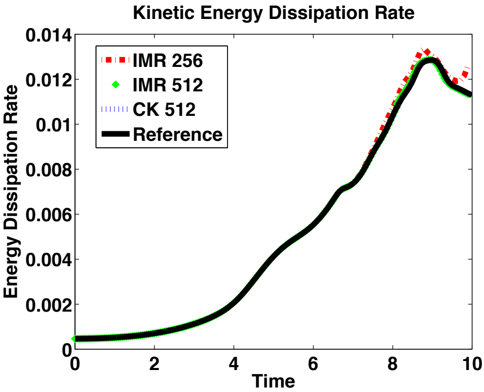

Figure: Plot during the initial stage, where flow is essentially inviscid and laminar. Fully developed turbulent flow is observed around $t_{max} \approx 8$.

Kinetic Energy Dissipation Rate

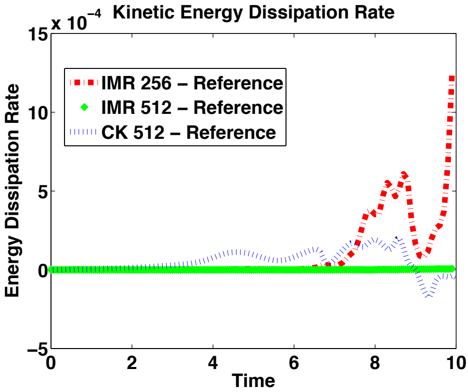

Figure: Difference in kinetic energy dissipation rates between the current discretizations and the reference solution.

Vorticity

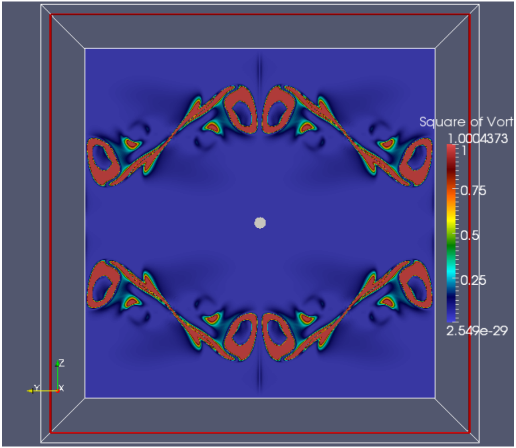

Figure: Square of the vorticity in the plane centered at $(\pi, 0, 0)$ with normal vector $(1, 0, 0)$.

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.
- Simulations with 256^3 grid points resulted in poor spatial convergences.

The 3D Maxwell's Equations

- $$\vec{D}_t - \nabla \times \vec{H} = 0$$
- $$\vec{B}_t + \nabla \times \vec{E} = 0$$
- with the relation between the electromagnetic components given by the constitutive relations:
- $$\vec{D} = \varepsilon_o(x, y, z)\vec{E}$$
- $$\vec{B} = \mu_o(x, y, z)\vec{H}$$
- Maxwell Simulation `http://vimeo.com/71822380`

Further Work

- Integration with other codes or simple examples for other spatial discretizations (one other example https://code.google.com/p/incompact3d/)
- Uniform interface for users with no programming background
- “Use MPI” vs. “Include mpif.h”
- C/C++ examples
- Better archiving and documentation procedure – currently wikibooks + github
- Integration with accelerators
- Integration with visualization tools
- Magnetohydrodynamics

Conclusion

- Easy to program numerical method which can be used to study semilinear partial differential equations
- Method parallelizes well on hardware with good communications so a good tool to introduce parallel programming ideas
- Research tool to investigate and provide conjectures for behavior of solutions to partial differential equations
- Research tool to investigate computer hardware performance and correctness
- Better user interface and integration with visualization would help make it easier for those without strong programming backgrounds

Acknowledgements

- This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- This research used the resources of the Supercomputing Laboratory at King Abdullah University of Science and Technology.
- This research used resources of the Keeneland Computing Facility at the Georgia Institute of Technology, which is supported by the National Science Foundation under Contract OCI-0910735.
- This material is based upon work supported by the National Science Foundation under Grant Number 1137097 and by the University of Tennessee through the Beacon Project.

Acknowledgements and References

- W. Auzinger, O. Batrashev, K. Hillewaert, D. Ketcheson, O. Koch, B. Leu, A. Liu, B. Palen, M. Parsani, W. Schlag, P. Sheth, M. Warnez
- The Blue Waters Undergraduate Petascale Education Program administered by the Shodor foundation
- The Division of Literature, Sciences and Arts at the University of Michigan

- Carpenter M.H. and Kennedy C. A. “Fourth-Order 2N-Storage Runge–Kutta schemes” NASA Langley Research Center Technical Memorandum 109112, (1994).
- `http://en.wikibooks.org/wiki/Parallel_Spectral_Numerical_Methods`
- `http://2decomp.org`
- Beamer `https://en.wikipedia.org/wiki/Beamer_(LaTeX)`

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