PAUL SCHERRER INSTITUT PSI

ETH zürich

Veronica Montanaro :: AMAS Group, LSM

Improvements to an FFT solver in the Independent Parallel Particle Layer

Master thesis

November 16, 2023

Overview

Poisson equation: $\Delta\phi = \rho$.

$\rightarrow$ Can write the solution as a convolution:

$$\boxed{\phi(\vec{r}) = (G * \rho)(\vec{r}) = \int G(\vec{r} - \vec{r}')\rho(\vec{r}')d\vec{r}'}$$

$\implies$ can use **FFT-based** methods

$$\phi = h_x h_y h_z \mathrm{FFT}^{-1}\{\mathrm{FFT}\{\rho\} \cdot \mathrm{FFT}\{G\}\}$$

Periodic Boundary Conditions
$\rightarrow$ Can use directly

Open Boundary Conditions
$\rightarrow$ Make periodic (Hockney and Eastwood, 1988)

A novel method for Open boundaries: Vico et al. (2016)

Based on the standard Hockney trick (Hockney and Eastwood, 1988)

Hockney and Eastwood (1988)

- Double the domain $\implies (2N)^3$.
- On doubled grid, make periodic $\implies \rho_2, G_2$.
- Use FFT to compute convolution.
- Restrict solution to physical domain.

Vico et al. (2016)

Green's function in Fourier space:

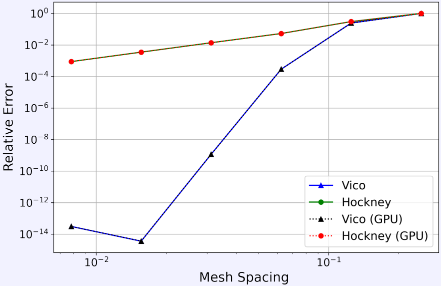
$$G_L(\vec{s}) = 2\left(\frac{\sin(L|\vec{s}|/2)}{|\vec{s}|}\right)^2,$$

where $L > \sqrt{L_x^2 + L_y^2 + L_z^2}$ is the length of the truncation window.

Novel Poisson solver: Problem

What do we gain?

Convergence for smooth Gaussian source:



Problem

Vico et al. (2016) requires **precomputation** of $G_L(\vec{s})$ on a $(4N)^3$ grid.

Currently

Code changes allowing mixed precision runs. **Algorithmic changes** to limit memory footprint to just needing a $(2N)^3$ grid.

These algorithmic changes require a Discrete Cosine Transform of type 1.

Mixed precision

Changes in Vico-Greengard solver

Before

All fields with same type as right hand side (ρ)

$\longrightarrow$

After

Pre-computed Green's function in frequency domain templated on left hand side ($\vec{E}$)

Choice of configuration

- ρ is kept in `double`. Having it in `float` $\implies$ errors in charge conservation, slower convergence.
- Consequence: solution ϕ also in `double`.
- $\vec{E}$ and $G_L(\vec{s})$ in `float` $\implies$ Largest data structure occupies half the space.

Memory testing

Setup

Test case

Gaussian test (Mayani, 2021)

- Runs five iterations of the solver with a Gaussian source.
- Grid points = from 32^3 to 128^3 $\rightarrow$ maximum size for single GPUs/CPU node. (Mayani, 2021)

CPU

- Two Intel Xeon Gold 6152 (44 cores, 380GB).
- Memory tracker: Kokkos Memory Highwater.

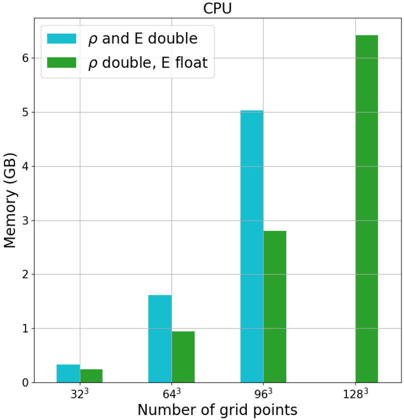
GPU

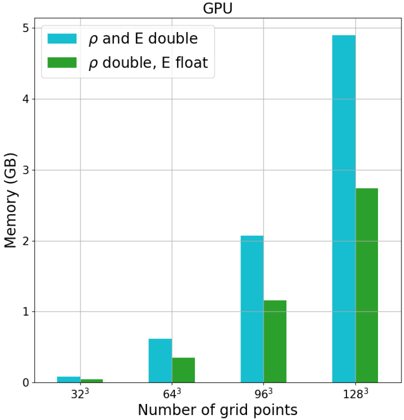
- One NVIDIA Ampere (40GB).
- Memory tracker: Kokkos Space Time Stack.

Memory testing

Results

About 50% less memory used.





Mixed precision

Hockney-Eastwood problem

Problem

Hockney's Green's function depends on data structure used for ρ.

$+$

HeFFTe does not support mixed precision.

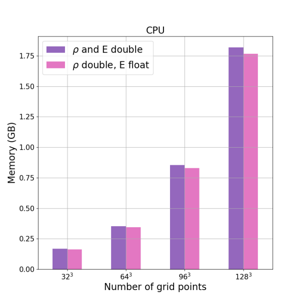
$\downarrow$

We can't have a transformed Green's function with a different type.

$\longrightarrow$

Consequence

No advantage memory-wise from mixed precision.



One of the reasons why we would need mixed precision in heFFTe.

DCT-I

Setting up

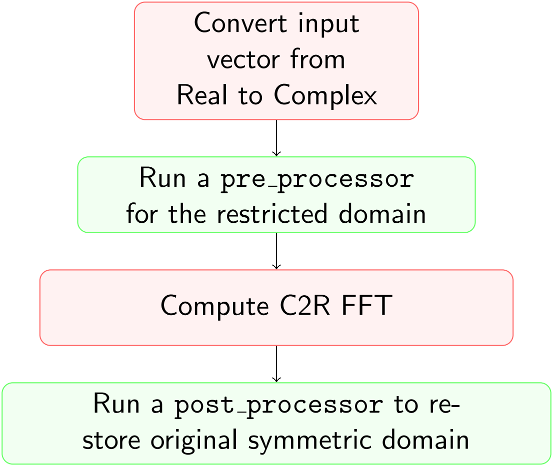
My main focus: **executor** class.

$\Longrightarrow$ In particular, `r2r_executor`

- The class contains methods for the execution of Sine and Cosine transforms.
- Currently: only in 1-D and only type II and III Cosine transform.

Only fftw has native DCTs of type I,II and III. All other back-ends make use of this class.

R2R executor workflow



CuFFT Plan

- Add `cufft_cos1` back-end
- Add `gpu_cos1_pre/post_processor` in a similar way as already done in fftw
- Add scaling factor of DCT-I for CuFFT to be used with cuda::scale.
- Repeat for 1-D, 2-D and 3-D

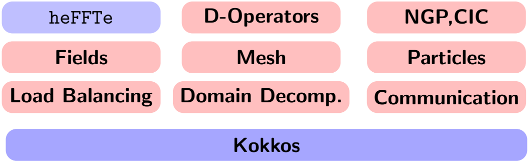
Summary

Motivations for DCT-I

IPPL $\Longrightarrow$ Independent Parallel Particle Layer framework (Muralikrishnan et al., 2021)

Inside of the library, we implemented a performance portable Poisson solver based on both methods.

> We use heFFTe (Ayala et al., 2020) to perform the FFTs.



Summary

Motivations for DCT-I (Cont'd)

Why memory reduction?

- Vico et al. obtains better accuracy with a coarser grid than with Hockney-Eastwood.
- Especially significant for GPUs because of memory constraints.

Moreover: Publication of a paper on the performance portable Poisson Solver is planned.

Thanks

Thanks for your attention.
Open for questions.

References

Adelmann, A., Calvo, P., Frey, M., Gsell, A., Locans, U., Metzger-Kraus, C., Neveu, N., Rogers, C., Russell, S., Sheehy, S., Snuverink, J., and Winklehner, D. (2019). OPAL a Versatile Tool for Charged Particle Accelerator Simulations. *arXiv:1905.06654 [physics]*. arXiv: 1905.06654.

Ayala, A., Tomov, S., Haidar, A., and Dongarra, J. (2020). heFFTe: Highly Efficient FFT for Exascale. In Krzhizhanovskaya, V. V., Závodszky, G., Lees, M. H., Dongarra, J. J., Sloot, P. M. A., Brissos, S., and Teixeira, J., editors, *Computational Science – ICCS 2020*, Lecture Notes in Computer Science, pages 262–275, Cham. Springer International Publishing.

Hockney, R. and Eastwood, J. W. (1988). *Computer Simulation Using Particles*. CRC Press.

Mayani, S. (2021). A performance portable poisson solver for the hose instability.

Muralikrishnan, S., Frey, M., Vinciguerra, A., Ligotino, M., Cerfon, A. J., Stoyanov, M., Gayatri, R., and Adelmann, A. (2021). IPPL GitLab wiki. `https://gitlab.psi.ch/OPAL/Libraries/ippl/-/wikis/home`.

Vico, F., Greengard, L., and Ferrando, M. (2016). Fast convolution with free-space Green's functions. *Journal of Computational Physics*, 323:191–203.