GPU-Optimized Tridiagonal and Pentadiagonal System Solvers for Spectral Transforms in QuiCC

Dmitrii Tolmachev[1*], Philippe Marti[1], Giacomo Castiglioni[1], Andrew Jackson[1], Daniel Ganellari[2]
[1] EPM Group, ETH Zurich, Switzerland, [2] ETH Zurich / CSCS, Switzerland

* dmitrii.tolmachev@erdw.ethz.ch

Introduction

QuiCC is a code under development designed to solve the equations of magnetohydrodynamics in a full sphere and other geometries. It uses a fully spectral approach to the problem, with the Jones-Worland (JW) polynomials as a radial basis and Spherical Harmonics (SH) as a spherical basis. We present an alternative to the quadrature approach to their evaluation – the polynomial connection approach, which is more accurate and requires less memory. In this work, we demonstrate an efficient GPU implementation of this algorithm, based on the efficient tridiagonal and pentadiagonal GPU solvers developed with the new single-warp GPU programming approach.

Polynomial connection approach

Based on recurrence relations between Jacobi polynomials, which allow to construct high order Jacobi polynomials from low order β:

$$P_n^{(\alpha,\beta)} = \gamma_n^{(\alpha,\beta)} P_n^{(\alpha,\beta+1)} + \zeta_n^{(\alpha,\beta)} P_{n-1}^{(\alpha,\beta+1)}$$

$$(1+x)P_n^{(\alpha,\beta)} = \mu_n^{(\alpha,\beta)} P_n^{(\alpha,\beta-1)} + \nu_n^{(\alpha,\beta)} P_{n+1}^{(\alpha,\beta-1)}$$

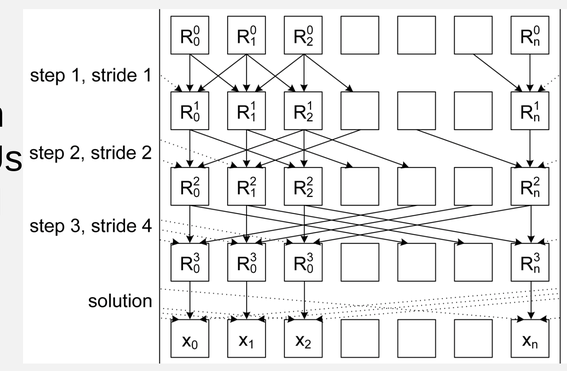
Polynomial connections – bidiagonal multiplications and solves.

Expansion on $P_n^{(\pm\frac{1}{2},\pm\frac{1}{2})}$ − Discrete Cosine Transforms with VkFFT.

Connection $P_n^{(\pm1,\pm1)} - P_n^{(\pm\frac{1}{2},\pm\frac{1}{2})}$ − Chebyshev–Legendre Transform.

Parallel Cyclic Reduction algorithm

Backward Substitution is a sequential algorithm, so we use Parallel Cyclic Reduction algorithm for banded system solving on GPUs Originally developed for tridiagonal systems, it is optimized for bidiagonal and generalized for pentadiagonal systems.



Performance analysis

- We solve bidiagonal systems in FP64 with optimized PCR on Nvidia A100 and AMD MI250 GPUs for different range of batches.
- Bandwidth is estimated as 2x system buffer size (minimal transfer size, excluding coefficients) divided by the kernel execution time.
- Batch size of 10 is too small for GPU execution, 100 has performance comparable to CPU, 1000+ use 50-100% of GPU.
- PCR algorithm solves systems as the next power of 2, linear scaling between them means it is balanced in compute/memory.
- For sizes > 1024, AMD benefits from having a larger warp size, as Nvidia can only use 255 registers/thread and spills some of them.

Conclusions

- Verified validity of the single-warp GPU programming approach.
- Implemented efficient bi-, tri- and pentadiagonal GPU solvers.
- Small batch number is needed to achieve 50-100% GPU utilization.
- The same machine precision accuracy as Backward Substitution.

Single-warp GPU programming approach

Warp – the basic unit of execution, a collection of threads (32 on Nvidia, 64 on AMD), that are executed simultaneously on single SM. We propose a new GPU code design paradigm:

- One warp per kernel per solved system design. Number of kernels is equal to the number of solved systems.
- Fast data transfers between threads via shuffle instructions.
- No shared memory usage – all calculations in the registers.
- No synchronizations means low number of threads is sufficient to cover latencies.
- Runtime autotuned for GPU and system size with the help of the code generation platform based on the VkFFT library design.

References

P22 - Efficient Data Managment in Fully Spectral Dynamo Simulations on Heterogeneous Nodes, *PASC '23*.

P. Marti, A. Jackson. Accurate and efficient Jones-Worland spectral transforms for planetary applications. *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '21*. doi: 10.1145/3468267.3470620.

P. Marti and A. Jackson. A fully spectral methodology for magnetohydrodynamic calculations in a whole sphere. *Journal of Computational Physics, 305:403–422, 2016*. doi: 10.1016/j.jcp.2015.10.056.

D. Tolmachev. VkFFT and beyond - a platform for runtime GPU code generation. *Proceedings of the 2023 International Workshop on OpenCL, IWOCL '23.* doi: 10.1145/3585341.3585357

Accuracy analysis

- GPU PCR accuracy is compared to CPU Backward Substitution in FP64 for JW connection matrices. Results are verified in FP128.
- PCR and BS are accurate if the absolutes of the main diagonal elements are bigger than the respective sub-/super diagonal elements – it is true for the polynomial connection matrices.
- Top left plot confirms that for fixed order, PCR and BS have almost fixed accuracy, close to machine precision (PCR is slightly worse).
- Top right plot confirms that for small β, both PCR and BS experience similar accuracy drop – sub-/super diagonal elements become close in value to the main diagonal elements.
- Solution: use double-double FP128 emulation for them. It will soon be implemented as a data type in the code generation platform.

Financial Support

This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme grant agreement No 833848 (UEMHP) and from PASC for the AQUA-D software development project.