No. 645 **June 2021**

A Proof of Concept for Very Fast Finite Element Poisson Solvers on Accelerator Hardware

D. Ruda, S. Turek, D. Ribbrock, P. Zajac

ISSN: 2190-1767

A Proof of Concept for Very Fast Finite Element Poisson Solvers on Accelerator Hardware

Dustin Ruda[1,*], **Stefan Turek**[1,**], **Dirk Ribbrock**[1], and **Peter Zajac**[1]

[1] Chair of Applied Mathematics and Numerics, TU Dortmund University, 44227 Dortmund, Germany

It is demonstrated that modern accelerator hardware specialized in AI, e.g., "next gen GPUs" equipped with Tensor Cores, can be profitably used in finite element simulations by means of a new hardware-oriented method to solve linear systems arising from Poisson's equation in 2D. We consider the NVIDIA Tesla V100 Tensor Core GPU with a peak performance of 125 TFLOP/s, that is only achievable in half precision and if operations with high arithmetic intensity, such as dense matrix multiplications, are executed, though. Its computing power can be exploited to a great extent by the new method based on "prehandling" without loss of accuracy. We obtain a significant reduction of computing time compared to a standard geometric multigrid solver on standard x64 hardware.

1 Introduction

Graphic cards with AI components, particularly Tensor Cores (TC) specialized in accelerating dense matrix multiplications, have recently gained importance in computer systems. As an example, the NVIDIA Tesla V100 SXM2 promises 125 TFLOP/s in half precision (HP) due to its TCs (31 TFLOP/s in HP without TCs, 16 TFLOP/s in single (SP) and 8 TFLOP/s in double precision (DP)) according to the manufacturer specifications[1]. It is of course desirable to exploit this high performance in HP in basic components of matrix-based finite element (FE) simulations. FE discretizations generally result in very sparse stiffness matrices. If matrix-vector or matrix-matrix products with the Poisson matrix stored in standard CSR format are computed on the V100, the observed performance (10–100 GFLOP/s) is far below the peak rates, as the results in [1, 2] show. If, however, dense matrices are used, 1 up to 100 TFLOP/s in HP are attained [1, 2]. In short, there is a large gap between actual and theoretical performance but also a huge potential if the hardware is appropriately used, e.g., by the hardware-oriented method explained below. We stick with Poisson's equation (in 2D), which is an important component in many flow simulations, and consider the case of one and many (i.e., a matrix valued linear system $AX = B$) right-hand sides. Due to currently examined time-simultaneous [3] Navier–Stokes solvers, that allow for solving the pressure Poisson problems for each time step at once, the latter is relevant in practice.

2 Prehandling of Finite Element Matrices and a Direct Poisson Solver

To construct a solver that is capable of exploiting the hardware mentioned above, there are two conditions to be met. Firstly, the condition numbers of involved matrices need to be reduced because the condition number $\mathcal{O}\left(h^{-2}\right)$ (if h denotes the grid width) of the standard FE Poisson matrix causes high computational errors that prohibit the use of low precision, which is necessary to achieve the V100's peak flop rate in HP, though. To this end, we introduced the concept of "prehandling" [4], an explicit preconditioning without excessively increasing the density of the matrix. The hierarchical finite element method [5], that is based on successive refinement of an initial coarse grid, proved to be successful in the 2D case because it reduces the condition number to $\mathcal{O}\left(\log^2 \frac{1}{h}\right)$ while preserving sparsity and also accuracy in lower precision as shown in [1, 2, 4].

Secondly, the solver needs to consist of multiplications with small, dense matrices (or large, sparse ones containing such as blocks) to exploit the TCs of the V100. For this purpose, the nodes of the hierarchical mesh are grouped into three types, the prehandled matrix is renumbered accordingly and a two-step Schur complement is applied (for details see [2]). Consequently, we obtain a direct solver, which we also refer to as PSC method (Prehandling Schur Complement), for Poisson's equation that is based on dense matrix-vector or matrix-matrix products (for many right-hand sides) and thus tailored for next gen GPUs.

3 Numerical Results Compared to Standard Solvers on Standard Hardware

As a proof of concept, we consider Poisson's equation on the unit square discetized by bilinear finite elements (Q1) on an equidistant mesh of grid width $h \in \left\{\frac{1}{256}, \frac{1}{512}, \frac{1}{1024}\right\}$ (the latter leads to $N \approx 10^6$ unknowns). The performance of the direct PSC method on the V100 GPU measured in GFLOP/s is shown in Fig. 1a. The case of many right-hand sides allows for

* Corresponding author: e-mail Dustin.Ruda@math.tu-dortmund.de
** Corresponding author: e-mail Stefan.Turek@math.tu-dortmund.de

[1] see https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf.

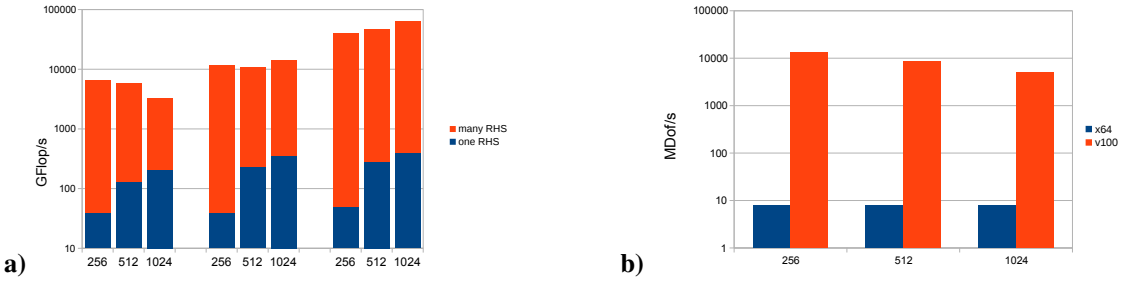

Fig. 1: a) GFLOP/s for PSC with one and many right-hand sides (RHS) depending on h^{-1} on the V100 GPU in DP, SP and HP with TCs (left, middle and right three columns, respectively). **b)** Comparison of MDof/s between MG on the AMD CPU in DP (left columns) and PSC in HP with TCs on the V100 (right columns), both for many RHS, depending on h^{-1}.

more dense matrix-matrix multiplications and thus yields the highest performance, due to the TCs especially in HP, namely 60 TFLOP/s if $h = \frac{1}{1024}$. This clearly shows the outstanding performance achieved by exploiting the TCs, which would not be available with common ALUs in HP.

However, one can only limitedly draw conclusions on the efficiency of the method from this high performance. Hence, we use the measure "million degrees of freedom solved per second" (MDof/s) that enables a comparison with a standard approach. As such, we consider a FE geometric multigrid (MG) method (implemented in the FEAT3[2] software) including sparse matrix-vector operations and in DP – due to the high condition number – on the x64 (multicore) AMD EPYC 7542 CPU[3]. The results of both approaches for many right-hand sides are depicted in Fig. 1b. The MG method requires $\mathcal{O}(N)$ (or as a rule of thumb for the meshes at hand approx. $1,000N$) operations, whereas, according to an estimate in [2], approx. $12N^{\frac{3}{2}}$ operations are needed for the PSC method if the variable coarse grid width is chosen optimally; an amount of $N = 1$ million unknowns thus yields $12,000N$ operations. The results confirm that the 12 times higher complexity is compensated by the remarkable hardware exploitation of the new method on the V100 GPU leading to lower computing times. In the case of one right-hand side the solution works 8 times and for a matrix of right-hand sides 600 times faster than the standard MG approach while accuracy is preserved.

In preliminary numerical tests on the successor model of the V100, the NVIDIA Ampere A100 TC GPU promising 300 TFLOP/s in HP[4], we were able to further accelerate the solution by approx. 50% using the direct PSC method [2].

4 Conclusion and Outlook

We provided a proof of concept of an efficient algorithm resulting from knowledge of modern hardware that benefits from the high performance of GPUs powered by TCs. The presented studies only treat Poisson problems on a simple mesh so far, but this equation is crucial within flow simulations and, according to recent tests, the PSC method is also applicable in the case of partially unstructured, refined triangular coarse grids.

Further research will be focused on extensions to the 3D case, other differential operators and finite element spaces and less storage intensive semi-direct variants of the PSC method.

Acknowledgements All comparative results have been created using the FEM software package FEAT3[2], whereby calculations have been carried out on the LiDO3 cluster at TU Dortmund University.

References

[1] S. Turek, D. Ruda, D. Ribbrock, and P. Zajac, Eine Machbarkeitsstudie zu schnellen FEM–Poisson–Lösern auf Beschleunigerhardware als Beispiel für Hardware–orientierte Numerik, Rundbrief GAMM 1/2021, pp. 4–12 (2021).
[2] D. Ruda, S. Turek, D. Ribbrock, and P. Zajac, Very Fast FEM Poisson Solvers on Lower Precision Accelerator Hardware: A "Proof-of-Concept" Study for NVIDIA Tesla V100 (2021, in preparation).
[3] J. Dünnebacke , S. Turek, C. Lohmann, A. Sokolov, and P. Zajac, Increased space-parallelism via time-simultaneous Newton-multigrid methods for nonstationary nonlinear PDE problems, The International Journal of High Performance Computing Applications 35(3), pp. 211–225 (2021).
[4] D. Ruda, S. Turek, P. Zajac, and D. Ribbrock, The Concept of Prehandling as Direct Preconditioning for Poisson–Like Problems, in: Numerical Mathematics and Advanced Applications Enumath 2019, F. Vermolen, and C. Vuik, Lecture Notes in Computational Science and Engineering (Springer, 2020), pp. 1011–1019.
[5] H. Yserentant, On the Multi-Level Splitting of Finite Element Spaces, Numerische Mathematik 49, pp. 379–412 (1986).

[2] see http://www.mathematik.tu-dortmund.de/~featflow/en/software/feat3.html.
[3] 32 cores per CPU, 128 MB L3-Cache, 1.5 TFLOP/s in DP, 3 TFLOP/s in SP, see https://www.amd.com/de/products/cpu/amd-epyc-7542.
[4] see https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/a100-80gb-datasheet-update-nvidia-us-1521051-r2-web.pdf.