Transient Simulation of Nonlinear Electro-Quasistatic Field Problems Accelerated by Multiple GPUs

Christian Richter[1], Sebastian Schöps[2], and Markus Clemens[1], *Senior Member, IEEE*

[1]Chair of Electromagnetic Theory, University of Wuppertal, 42119 Wuppertal, Germany, *christian.richter@uni-wuppertal.de*
[2] Graduate School of Computational Engineering and Institut für Theorie Elektromagnetischer Felder, Technische Universität Darmstadt, 64285 Darmstadt, Germany, *schoeps@gsc.tu-darmstadt.de*

This paper focuses on the acceleration of the numerical linear algebra for solving large nonlinear electroquasistatic problems in time domain as they occur in high-voltage applications utilizing nonlinear electric field stress grading materials. An algebraic multigrid (AMG) scheme is executed on multiple GPUs as a preconditioner for the conjugate gradients method. The speedup is further increased by exploiting the fact that repeated solutions of similar shaped systems must be obtained in the Newton-Raphson iterations of the implicit time-stepping scheme. A tailored update algorithm for the AMG preconditioner is proposed that clearly reduces the communication between GPUs and host.

***Index Terms*—multigrid, electroquasistatic, parallelism, GPUs**

I. Introduction

The Finite-Element-Based simulation of electromagnetic fields is an established technique for the design and optimization of devices such as high-voltage insulators. To further improve the accuracy, large scale 3D models are considered where the simulation takes into account both large homogeneous regions and small geometric details. For solving these models in acceptable time a fast solution of the resulting large scale discrete problems is of high importance.

II. Problem Formulation

For the low frequency applications mentioned above the electro-quasistatic (EQS) approximation of Maxwell's equations is sufficiently accurate. It can be formulated as the following partial differential equation

$$\nabla\cdot\left(\kappa\left(\vec{r}\right)\nabla\varphi\left(\vec{r},t\right)\right)+\nabla\cdot\left(\varepsilon\left(\vec{r}\right)\nabla\partial_t\varphi\left(\vec{r},t\right)\right)=f\left(\vec{r},t\right)$$

where φ is the scalar electrical potential, ε is the electrical permittivity, κ is the electrical conductivity with adequate initial and boundary conditions. The system can be discretized in space using Finite Elements (FE) and in time using for example Singly Diagonal Implicit Rung-Kutta (SDIRK) methods. The nonlinearity is taken into account by using a nonlinear solver in each time-step, e.g. a Newton-Raphson scheme. As a result, many sparse linear algebraic systems of equations have to be solved. For this the conjugate gradient method is applicable which is typically accelerated by a preconditioner, e.g. an algebraic multigrid method scheme (AMG-CG) [1].

III. Solving electromagnetic problems on GPUs

When computing large FE problems most of the time is spend solving the linear systems. Unfortunately, this operation is carried out in each time step and within each nonlinear iteration. AMG-CG can be divided into two parts: in the *setup-phase* the preconditioner is initiated. In case of AMG this corresponds to creating multiple grid levels and their interconnection operators. This process is time consuming, but has to be performed only once during the solution of an individual linear system. The *solve-phase* consists of vector-vector and sparse matrix-vector operations. This can be performed very fast on GPUs. Furthermore, as long as the system matrix remains constant the preconditioner can be reused and the setup can be avoided. This is for example the case when solving transient linear EQS problems without field stress grading material. When more complex materials are considered, the system matrix and the Jacobian used in the Newton-Raphson scheme will change depending on the local electric field distribution.

For accelerating sparse linear algebra system solutions with GPUs the CUSP library [2] is available. It provides sparse matrix and vector operations and linear solvers including AMG-CG based on smoothed aggregation. However, solving large scale problems may exceed the global memory of a single GPU, especially when using AMG preconditioners that have high memory demands due to multiple grid levels. To overcome this limitation a multi-GPU AMG-CG linear solver has been presented as an add-on to CUSP [3]. It creates the preconditioner on the CPU host computer, where it can be set up using available main memory [4]. Afterwards, it is distributed across all GPUs. The CUSP add-on involves multi-GPU vector and matrix classes, SpMV as well as BLAS level 1 operators and inter-device communication routines.

IV. Accelerating an electro-quasistatic solver with multiple GPUs

The in-house FEM code MEQSICO is capable of solving large nonlinear electro-quasistatic problems [5]. They are discretized in space by FEM and in time using an SDIRK3(2) time stepping scheme [6]: in each time step the nonlinear contributions are reassembled and the system matrix and right-hand-side vectors are constructed. This is already efficiently parallelized using OpenMP. However, this approach can be improved to interact efficiently with the GPU-accelerated AMG-CG proposed above.

In the nonlinear setting the Jacobian matrix changes and thus the preconditioner must be constructed repeatedly. To circumvent this problem two different options are applicable: either using a simplified Newton scheme, i.e., freezing the Jacobian or freezing the preconditioner in the CG algorithm. Both approaches result in an increase of the Newton-Raphson or CG iterations, respectively. Here we follow the second approach. Further optimization is possible by not using the old Jacobian, but the one of the current time step for the finest level the preconditioner. Thus only the coarser levels are not changing and the error introduced is damped in the post-smooth function on the finest grid. Another possibility is to update only the values of the preconditioner but not the algebraic grids. This implies that the restriction and prolongation operators are not recalculated as the number or position of the matrix entries remains unchanged. This reduces the Jacobian matrix update routine to a Galerkin product for each update.

V. Numerical example

As an example the electrical field of a high voltage insulator is calculated using 2nd order FE ansatz functions. It is both solved with constant materials (linear EQS problem) and coated with a microvaristor-layer as an example for a nonlinear problem, respectively. It solved on a compute server equipped with two Xeon 2560 eight-core CPUs and four attached nvidia Tesla K20X GPUs using Cuda 6.5, Thrust 1.8, CUSP 0.4 and GCC 4.4.7. Optimization level 2 is turned on. The problem dimensions are given in Tab.1.
This comparison is obviously not fair since hardware costs are not regarded but this hardware configuration reflects the situation of many computational laboratories.

TABLE I
DEGREES OF FREDDOM AND NON-ZERO MATRIX ENTRIES FOR THE NUMERICAL EXAMPLES

Problem	Degrees of Freedom	Non-Zero Matrix Entries
High-voltage insulator	12e06	320e6
High-voltage insulator with nonl. microvaristor-layer	14e06	387e6

Fig. 1 compares the average computational time for one system, i.e., assembly, setup of the linear system and the inversion of the linear system. Here the setup consists of setting up the right-hand-side and the Jacobian. One can see that especially the time consuming solution of the linear system is reduced. In this part the speedup is up to 25 times. In this case the TrilinosML preconditioner is more effective on the host as CUSP's AMG, especially in terms of iteration steps (45 compared to 101). However, the multi-GPU CUSP AMG-CG implementation outperforms both CPU-based options.
The average amount of time for one iteration of the nonlinear solver is shown in Fig. 2. Here the calculation time due to matrix assembly is obviously more relevant. Furthermore, the Newton-Raphson scheme has to be applied which requires more calculations in the setup of the linear system. Even then, the linear solver requires the dominating time slice. This part, however, is clearly accelerated by the multi-GPU AMG-CG linear system solver. When using the multi-GPU code, the solution of the linear system reduces from the most time consuming part to the least time consuming part. In addition, for the given nonlinear example no significant rise of the number of iterations can be noticed.

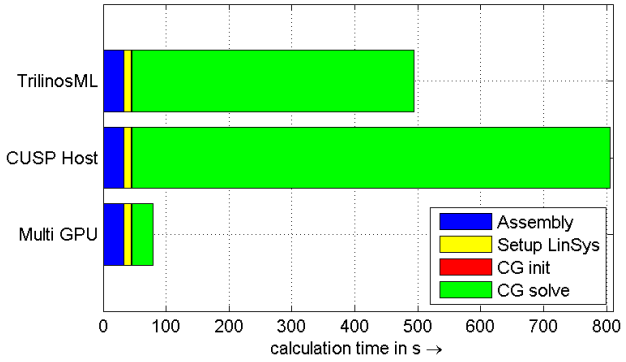

Fig. 1: Calculation time for one time stage in the linear eqs-simulation.

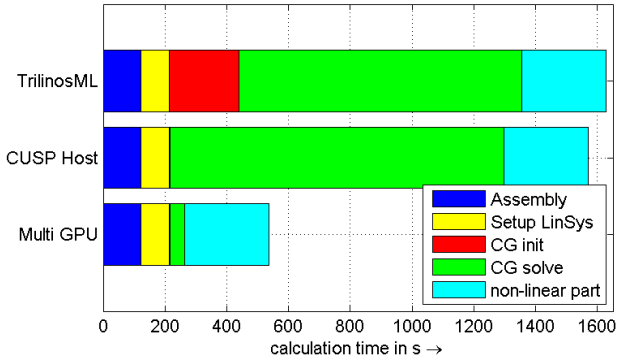

Fig. 2: Calculation time for one time stage in the nonlinear EQS simulation.

VI. Conclusion

We presented a method for solving nonlinear electro-quasistatic problems in time domain accelerated by a multi-GPU linear solver. It is optimized for solving linear problems repeatedly with small changes in the system matrix as they occur in nonlinear problems. Therefore a specific multi-GPU AMG preconditioner was proposed that updates the preconditioner without additional calculations.

VII. References

[1] K. Stüben, *"Algebraic multigrid (AMG): An introduction with applications,"* GMD. Report 53, 1999.

[2] N. Bell and M. Garland, "*"Efficient sparse matrix-vector multiplication on CUDA,"* NVIDIA Corporation, NVIDIA Technical Report NVR-2008-004, 2008.

[3] C. Richter, S. Schöps, and M. Clemens, "*Multi-GPU acceleration of algebraic multigrid preconditioners for elliptic field problems*." *IEEE Trans. Magn.*, vol. 50, no. 2, 2014.

[4] C. Richter, S. Schöps, and M. Clemens, "*GPU acceleration of algebraic multigrid preconditioners for discrete elliptic field problems*." *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 461-464, 2014.

[5] T. Steinmetz, M. Helias, G. Wimmer et al, "*Electro-quasistatic field simulations based on a discrete electromagnetism formulation," IEEE Trans. Magn.*, vol. 42, no. 4, pp. 755-758, 2006.

[6] F. Cameron, R. Piché, and K. Forsman, "*Variable step size time integration methods for transient eddy-current problems*," *IEEE Trans. Magn.*,vol. 34, pp. 3319–3323, Sept. 1998.