ARBITRARY HIGH-ORDER DISCONTINUOUS GALERKIN METHOD FOR ELECTROMAGNETIC FIELD PROBLEMS*

K. Papke†, C. R. Bahls, U. van Rienen, Universität Rostock, IAE, IEF, 18059 Rostock, Germany

Abstract

In this paper, we present a time integration scheme applied to the Discontinuous Galerkin finite element method (DG-FEM, [1]) for the computation of electromagnetic fields in the interior of three-dimensional structures. This approach is also known as Arbitrary High-Order Derivative Discontinuous Galerkin (ADER-DG, [2, 3]). By this method, we reach arbitrary high accuracy not only in space but also in time. The DG-FEM allows for explicit formulations in time domain on unstructured meshes with high polynomial approximation order. Furthermore, the Discontinuous Galerkin method in combination with the arbitrary high order time integration scheme is well suited to be used on massively parallel computing architectures. Moreover the method can be extended for local time stepping to become more efficiently by reducing the computation time [4].

INTRODUCTION

For the design and optimization of Higher-Order-Mode Coupler (Fig. 1), used in RF accelerator structures, numerical computations of electromagnetic fields as well as scattering parameter are essential. These computations can be carried out in time domain. In this work the implementation and investigation of a time integration scheme based on the Discontinuous Galerkin Finite-Element Method (DG-FEM) with arbitrary order in space and time is demonstrated for solving 3-D electromagnetic problems in time domain.

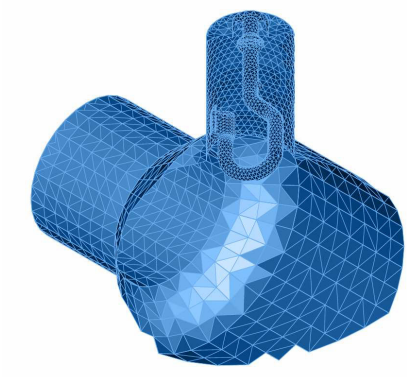

Figure 1: Tapered beam pipe with Higher-Order-Mode Coupler. The aspect ratio of such a grid is relatively high, due to the filigree structure of the antenna. Therefore, such a model is well suited for the application of local time stepping schemes.

THE NUMERICAL SCHEME

With known initially and boundary conditions it is sufficient to describe classic electromagnetic phenomena only by AMPERE's and FARADAY's law of MAXWELL equations. The partial differential equations can be written in the following general form:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \mathcal{A}_1 \frac{\partial \boldsymbol{u}}{\partial x} + \mathcal{A}_2 \frac{\partial \boldsymbol{u}}{\partial y} + \mathcal{A}_3 \frac{\partial \boldsymbol{u}}{\partial z} = 0, \qquad (1)$$

where

$$\boldsymbol{u}(x, y, z, t) \quad = \quad (E_x, E_y, E_z, H_x, H_y, H_z)^T. \quad (2)$$

The space-dependent Jacobian matrices $\mathcal{A}_i$ determine the physical behavior of the equations. They contain material properties as well as the curl operator applied to E- and H-field. To solve this ordinary partial differential equation, a physical initial condition as well as boundary conditions are still needed.

For the numerical scheme the computational domain $\Omega \in \mathbb{R}^3$ will be partitioned into conforming tetrahedral elements D^k. The approximate solution $\boldsymbol{u}_h^k$ of (1) inside each tetrahedron D^k is given by:

$$\boldsymbol{u}_h^k = \sum_{i=1}^{N_p} \hat{\boldsymbol{u}}_i^k(t) \cdot \Phi_i^k(\boldsymbol{x}). \qquad (3)$$

Here $\Phi_i^k(\boldsymbol{x})$ are the nodal ansatz functions and $\hat{\boldsymbol{u}}_i^k(t)$ are the time-dependent degrees of freedom which are allocated at the nodes of the element D^k.

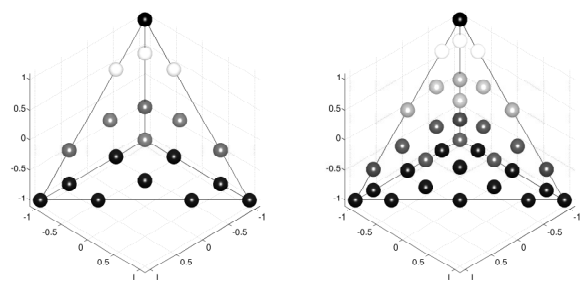

Figure 2: Reference element with the nodes on which the time-dependent degrees of freedom are defined. (left: 3rd order ansatz functions, right: 4th order ansatz functions)

* Work supported by Federal Ministry for Research and Education BMBF under contract 05K10HRC

† kai.papke@uni-rostock.de

ISBN 978-3-95450-116-8

Multiplication of Equation (1) by test functions $\Phi_j(\boldsymbol{x})$ and integration over the element D^k followed by integration by parts results in the following variational formulation [1, 2]:

$$\int_{D^k} \Phi_j \frac{\partial \boldsymbol{u}_h^k}{\partial t} dV + \int_{\partial D^k} \Phi_j \boldsymbol{F_h} dS - \int_{D^k} \left(\frac{\partial \Phi_j}{\partial x} \mathcal{A}_1 \boldsymbol{u}_h^k + \frac{\partial \Phi_j}{\partial y} \mathcal{A}_2 \boldsymbol{u}_h^k + \frac{\partial \Phi_j}{\partial z} \mathcal{A}_3 \boldsymbol{u}_h^k \right) dV = 0, \tag{4}$$

where we introduced the numerical flux that provides the coupling between the solutions of adjacent elements. This weak formulation allows for solutions in Ω which are discontinuous at the borders between the elements D^k. The surface and volume integrals over the ansatz- and test function, respectively their derivatives will be pre-computed for the reference element (Fig. 2) and adapted onto each element D^k by conformal mapping. Equation (4) has to be fulfilled for every test function Φ_j, which leads to a linear system of equations with the well known mass matrix $\mathcal{M}^k$ and stiffness matrices $\mathcal{S}_x^k$, $\mathcal{S}_y^k$ and $\mathcal{S}_z^k$:

$$\mathcal{M}_{(i,j)}^k = \int_{D^k} \Phi_i(\boldsymbol{r}) \Phi_j(\boldsymbol{r}) d\Omega = \langle \Phi_i | \Phi_j \rangle$$

$$\mathcal{S}_{x(i,j)}^k = \int_{D^k} \frac{\partial \Phi_i(\boldsymbol{r})}{\partial x} \Phi_j(\boldsymbol{r}) d\Omega = \langle \frac{\partial \Phi_i}{\partial x} | \Phi_j \rangle.$$

The resulting semi-discrete formulation corresponds to homogeneous differential equations with time derivations, which can be written for every element D^k as:

$$\frac{d\boldsymbol{u}_h^k}{dt} = \mathcal{L}^k(\boldsymbol{u}_h^k, t), \tag{5}$$

with the operator $\mathcal{L}^k$ that summarizes the mass- and stiffness matrices as well as the numerical flux. To get a fully discrete scheme in space and time we redraft Equation (5) in the integral form:

$$\boldsymbol{u}_h^k(t_{n+1}) = \boldsymbol{u}_h^k(t_n) + \int_{t_n}^{t_n+\Delta t} \mathcal{L}^k(\boldsymbol{u}_h^k, \tau) d\tau. \tag{6}$$

This leads to time integration schemes, where the problem lies now in the approximation of the integral in Equation (6). In the case of the MAXWELL equations the operator $\mathcal{L}^k$ behaves linear for $\boldsymbol{u}_h^k$ and is independently on the time. This enables the possibility to change the order of integration and application of the operator $\mathcal{L}^k$, what is advantageous but not necessary for the ADER approach. For the following formulas we remove the superscript k in $\boldsymbol{u}_h^k$ and all matrices to simplify the expressions. This means also that the following procedure for the time integration is carried out separately for each element.

ARBITRARY HIGH-ORDER DERIVATIVE TIME DISCRETIZATION

The basic idea of the ADER approach is the TAYLOR-expansion of the numerical solution in time, which allows for an analytical time integration.

$$\boldsymbol{u}_h(x, y, z, t) = \sum_{p=0}^{N} \frac{t^p}{p!} \frac{\partial^p}{\partial t^p} \boldsymbol{u}_h(x, y, z, t_0) \tag{7}$$

The foundation for this approach is the CAUCHY-KOWALEWSKAYA theorem that asserts a unique analytic solution for the considered boundary problem in the neighborhood of t_0. Here, N determines the convergence order of the resulting time integration scheme.

The time derivatives will be replaced by spatial derivatives based on the problem (1) itself.

$$\frac{\partial^p \boldsymbol{u}_h}{\partial t^p} = (-1)^p \left(\mathcal{A}_1 \frac{\partial}{\partial x} + \mathcal{A}_2 \frac{\partial}{\partial y} + \mathcal{A}_3 \frac{\partial}{\partial z} \right)^p \boldsymbol{u}_h \tag{8}$$

Since the ansatz functions Φ_i are only defined for a reference element, which is given in the $(\xi\eta\zeta)$- coordinate system, the spatial derivatives have to be carried out with respect to ξ, η and ζ. Equation (8) can be rewritten with a linear transformation from the reference element to every individual tetrahedral element D^k as:

$$\frac{\partial^p \boldsymbol{u}_h}{\partial t^p} = (-1)^p \left(\mathcal{A}_1^* \frac{\partial}{\partial \xi} + \mathcal{A}_2^* \frac{\partial}{\partial \eta} + \mathcal{A}_3^* \frac{\partial}{\partial \zeta} \right)^p \boldsymbol{u}_h. \tag{9}$$

For the MAXWELL equations, for example the transformed Jacobi matrice $\mathcal{A}_1^*$ offers the following structure:

$$\mathcal{A}_1^* = \begin{pmatrix} 0 & \mathcal{C}_\xi \\ \mathcal{C}_\xi^T & 0 \end{pmatrix}, \text{ with } \mathcal{C}_\xi = \begin{pmatrix} 0 & -\frac{\partial \xi}{z} & \frac{\partial \xi}{y} \\ \frac{\partial \xi}{z} & 0 & -\frac{\partial \xi}{x} \\ -\frac{\partial \xi}{y} & \frac{\partial \xi}{x} & 0 \end{pmatrix}. \tag{10}$$

Clearly, these matrices differ from element to element. Together with Equation (9), the analytical expression for the approximate solution $\boldsymbol{u}_h$ in Equation (7) can be rewritten as:

$$\boldsymbol{u}_h(\xi, \eta, \zeta, t) = \sum_{p=0}^{N} \frac{t^p}{p!} (-1)^p \left(\mathcal{A}_1^* \frac{\partial}{\partial \xi} + \mathcal{A}_2^* \frac{\partial}{\partial \eta} + \mathcal{A}_3^* \frac{\partial}{\partial \zeta} \right)^p \boldsymbol{u}_h(\xi, \eta, \zeta, t_0) \tag{11}$$

Similar to finite element methods, this approximation will be projected onto each test function in order to get an approximation of the evolution of the degrees of freedom $\hat{\boldsymbol{u}}_j(t)$ during one time step from t_n to t_{n+1}.

$$\hat{\boldsymbol{u}}_j(t) \langle \Phi_n | \Phi_j \rangle = \langle \Phi_n | \sum_{p=0}^{N} \frac{t^p}{p!} (-1)^p \times \left(\mathcal{A}_1^* \frac{\partial}{\partial \xi} + \mathcal{A}_2^* \frac{\partial}{\partial \eta} + \mathcal{A}_3^* \frac{\partial}{\partial \zeta} \right)^p | \Phi_i \rangle \hat{\boldsymbol{u}}_i(t_0) \tag{12}$$

ISBN 978-3-95450-116-8

On the left hand side of Equation (12) we find again the elements of the mass matrix $\mathcal{M}_{(n,j)}$, which is already used for the DG-FEM. Furthermore, after expansion of the right hand side we obtain also the following projections:

$$\mathcal{S}^{\lambda\mu\nu}_{(n,i)} = \langle \Phi_n | \frac{\partial^p}{\partial\xi^\lambda \partial\eta^\mu \partial\zeta^\nu} \Phi_i \rangle, \tag{13}$$

where $\lambda + \mu + \nu = p$ and $\lambda, \mu, \nu \geq 0$. All these projection matrices can be pre-computed for the reference element by a computer algebra system once and then be stored. In the case of nodal ansatz functions, that result from an interpolation on orthonormal polynomials it is not necessary to compute the projections. Instead, we can use the following relation from [1]:

$$(\mathcal{M}^{-1}\mathcal{S}^{\lambda\mu\nu})_{(i,j)} = \frac{\partial^p}{\partial\xi^\lambda \partial\eta^\mu \partial\zeta^\nu} \Phi_j \Big|_{\xi_i,\eta_i,\zeta_i}. \tag{14}$$

By using only these derivation matrices we directly obtain an explicit expression for the time dependent degrees of freedom $\hat{\boldsymbol{u}}_j(t)$ in Equation (12), due to the contained multiplication with the inverse mass matrix.

Finally, the integral of the time dependent degrees of freedom from t_n to t_{n+1} can be written as:

$$\int_{t_0}^{t_0+\Delta t} \hat{\boldsymbol{u}}^k_j(t)dt = \mathcal{M}^{-1}_{(n,j)} \langle \Phi_n | \sum_{p=0}^{N} \frac{\Delta t^{(p+1)}}{(p+1)!} (-1)^p \left(\mathcal{A}^*_1 \frac{\partial}{\partial\xi} + \mathcal{A}^*_2 \frac{\partial}{\partial\eta} + \mathcal{A}^*_3 \frac{\partial}{\partial\zeta} \right)^p |\Phi_i\rangle \hat{\boldsymbol{u}}^k_i(t_0). \tag{15}$$

With the $\mathcal{L}$ operator applied on these resulting matrices, the fully time integration in Equation (6) is performed with arbitrary high order accuracy N.

RESULTS

The numerical method for the spatial discretization is implemented in NUDG++, an open source framework written in C++. The time integration procedure is currently in progress, but runs already on CPU. It follows the implementation for GPU's, which accelerates the algorithm extensive. At this time the algorithm is tested only for a first simple example, the lossless rectangular cavity (Fig. 3).

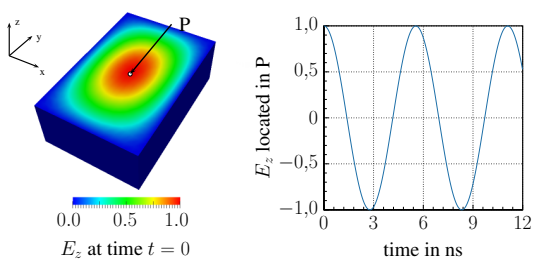

Figure 3: Left: Electric field distribution of the E_z component in a rectangular resonator. Right: The corresponding time evolution of E_z in the center of the cavity.

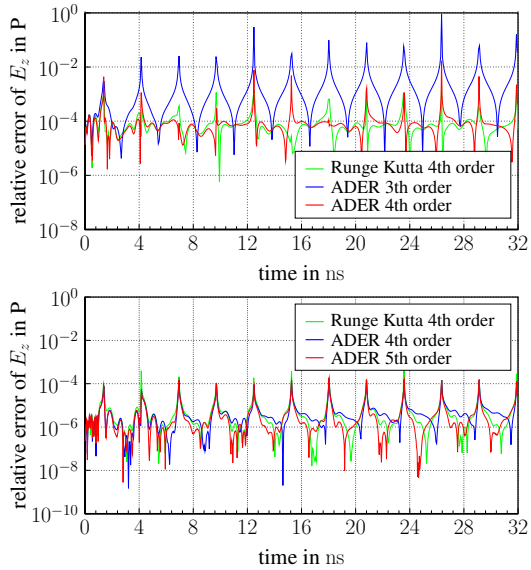

Figure 4: Comparison between ADER time integration scheme and a 4th order RUNGE-KUTTA scheme. Top: 3rd order ansatz functions, Bottom: 4th order ansatz functions.

In Figure 4 the relative error is shown for the ADER scheme compared with a 4th order RUNGE-KUTTA scheme. For the same convergence order the errors of both algorithms lies as expected in the same order of magnitude.

CONCLUSIONS AND OUTLOOK

By using only one CPU core the 4th order RUNGE-KUTTA scheme is $\sim$ 3 times faster then the ADER approach in the same convergence order (3rd order ansatz functions). This relation factor reduces in the case of 4th order ansatz functions to 2. An implementation on parallel architectures probably improves this ratio related to the ADER aproach. Therefore we are currently focused on an implementation for GPUs with the Cuda or OpenCL framework.

REFERENCES

[1] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods*, (New York: Springer, 2007).

[2] M. Dumbser and M. Käser, "An arbitrary High-Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes – II. The Three-Dimensional Isotropic Case," Geophys. J. Int., 2006.

[3] A. Taube, M. Dumbser et al. "Arbitrary High-Order Discontinuous Galerkin Schemes for the Magnetohydrodynamic Equations," Journal of Scientific Computing Vol. 30, 2007.

[4] N. Gödel, S. Schomann et al. , "GPU Accelerated Adams-Bashforth Multirate Discontinuous Galerkin FEM Simulation of High Frequency Electromagnetic Fields," IEEE Transactions on Magnetics Vol. 46, 2010.

ISBN 978-3-95450-116-8