Distributed Parameter-less Adaptive DGFEM MHD Solver for Astrophysics

Lukas Korous[1], Pavel Karban[1]
[1]University of West Bohemia, Faculty of Electrical Engineering, Pilsen, Czech Republic

The discontinuous Galerkin (DG) method is a favorable alternative to the finite volume (FV) method, which is often used for conservation laws. DG methods offer higher order accuracy and reduced diffusion compared to the finite volume method while keeping the scheme highly parallelizable. We aim at making the implementation parameter-free, applicable to all possible problems. To achieve that, instead of using standard ways to achieve the zero divergence constraint on the magnetic field component, we use here an exactly divergence free basis, and for the numerical flux and flux limiting algorithms we offer several possibilities, mainly HLLD numerical flux, and Barth-Jespersen or Vertex-based flux limiting techniques, all working without parameters. The code works for 3 spatial dimensions in fully distributed manner, and is available from a public software repository.

Index Terms—**Numerical simulation, finite element method, MHD equations, adaptivity, discontinuous Galerkin method, astrophysics, solar flares, AMR, distributed computing.**

I. INTRODUCTION

The term magnetohydrodynamics (MHD) covers all physical phenomena that involve both electromagnetic (EM) field and a fluid that carries the EM field. Such phenomena are very interesting, yet very complex to study. The behavior of such a fluid is utilized in some industrial applications - liquid-metal cooling of nuclear reactors, magnetic fluid in dampers, sensors for precise measuring of angular velocities, etc. Such phenomena occur in nature as well - the most significant of which are the processes that take place inside and on the surface of stars.

II. MAGNETOHYDRODYNAMICS IN ASTROPHYSICS

There are several phenomena in the universe that we can look at as magnetohydrodynamic in nature - planets consisting of metals, interplanetary space, but mainly - stars. If we talk about the nearest star - and the only one we are able to study well enough - the Sun - these phenomena include those that occur in the Sun's photosphere (the layer of Sun that is visible): Sun spots, but also phenomena that occur above the Sun (further from the center of the Sun): in Sun's chromosphere, or even corona (solar flares) - even phenomena that originate from the Sun, but then spread through our solar system - solar winds, space weather. All these phenomena have a large impact on the lives of all of us. One such event is captured in1. For example solar flares (that are often followed by ejection of mass out of the Sun - the so called coronal mass ejections - CMEs) have impact on the Earth's magnetic field which in turn has impact on the electronic communication down on Earth (because the communication satellites used for transmissions may be damaged by the disturbances in the magnetic field). Also people operating at high altitudes, both in airplanes and manned space missions are exposed to the energetic particles coming from the Sun (this term is sometimes called *cosmic rays*). For all the above reasons, it is of great importance to understand the phenomena of space weather, and other MHD phenomena that occur in space.

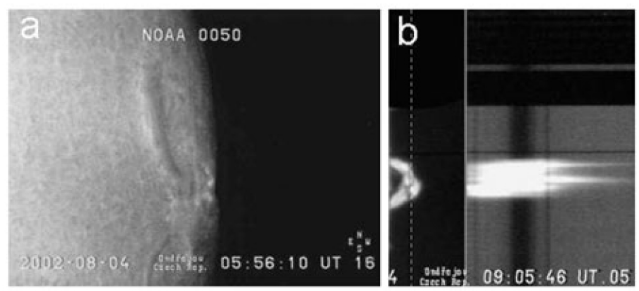

Fig. 1: Observation of a limb event above the sun's surface. $H\alpha$ slit-jaw is the middle (b) part, taken from [1].

Main goal that was set for this work is implementation of a software package, capable of numerically solving the magnetohydrodynamic equations with the following attributes:

- The code must implement mathematically correct and clean methods, preferably with no parameter-dependent algorithms
- The code must allow for a broad spectrum of solution attributes occurring (shocks, oscillations, high energies)
- The code must be able to capture very fine details of the solution through higher resolution, in reasonable computing time
- The code must be able to run a large-scale simulations, for which a single computer is not sufficient, and utilization of modern distributed computing approach is a must
- The code must be easy to use for the physicists, must be written using industry-standard modern object-oriented programming language

Manuscript received November 3, 2018; Corresponding author: L. Korous (email: lukas.korous@gmail.com).

III. Mathematical Model

The system of equations we are interested in solving is the system of ideal MHD equations in the form

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\nabla \cdot (\boldsymbol{\pi}), \\
\frac{\partial \boldsymbol{\pi}}{\partial t} &= -\nabla \cdot (\boldsymbol{\pi} \otimes \boldsymbol{u}) + \nabla \cdot \left(\frac{1}{\mu_0} \boldsymbol{B}\boldsymbol{B} - \frac{B^2}{2} \mathrm{I} \right) - \nabla p, \\
\frac{\partial U}{\partial t} &= -\nabla \cdot \left[\boldsymbol{\pi} \left(\frac{u^2}{2} + e \right) + p\boldsymbol{u} \right], \\
\frac{\partial \boldsymbol{B}}{\partial t} &= -\nabla \times (\boldsymbol{u} \otimes \boldsymbol{B} - \boldsymbol{B} \otimes \boldsymbol{u}).
\end{aligned}$$

Here, ρ is plasma density, $\boldsymbol{\pi}$ plasma momentum, $\boldsymbol{u}$ plasma velocity, p plasma pressure. U is the total energy, $\boldsymbol{B}$ is magnetic flux density, and e denotes internal energy.

IV. Numerical Solution

From the numerical perspective, to investigate such phenomena is to bring the complexity of multiple scales present in the physical world - magnetic reconnection occurs at substantially different scales than e.g. the solar flares. To handle this numerically with adequate resolution, and reasonable computational costs, the implementation must be able to limit the number of degrees of freedom used for the discretization to such that yield the largest accuracy increase - namely, in this work, this is achieved via a state-of-the-art AMR approach.

We look at the Discontinuous Galerkin method (DG, [2], [3]) - which was chosen in order to deliver the first and second of the above requirement (handling sharp fronts, discontinuities and the like with mathematical cleanliness), and give the entire process from the integral equations down to algorithms performing basic operations. Pitfalls of the numerical solution of the MHD equations (divergence constraint, slope limiting), and chosen way of solving them is given as well, while still keeping the above requirements in mind.

We start with the ideal MHD equations in the conservative form

$$\frac{\partial \boldsymbol{\Psi}}{\partial t} + \nabla \cdot \mathbf{F}(\boldsymbol{\Psi}) = \mathbf{S}, \tag{1}$$

where we have

$$\mathbf{F}_i = \left(\begin{array}{c}
\pi_i \\
\frac{\pi_1 \pi_i}{\rho} - B_1 B_i + \frac{1}{2}\delta_{1i}(p + U_m) \\
\frac{\pi_2 \pi_i}{\rho} - B_1 B_i + \frac{1}{2}\delta_{2i}(p + U_m) \\
\frac{\pi_3 \pi_i}{\rho} - B_1 B_i + \frac{1}{2}\delta_{3i}(p + U_m) \\
\frac{\pi_i}{\rho}\left(\frac{\gamma}{\gamma - 1} p + U_k \right) + \frac{2}{\rho}\varepsilon_{ijk}(\pi_k B_i - \pi_i B_k) B_j \\
\frac{\pi_i B_1 - \pi_1 B_i}{\rho} \\
\frac{\pi_i B_2 - \pi_2 B_i}{\rho} \\
\frac{\pi_i B_3 - \pi_3 B_i}{\rho}
\end{array} \right), \tag{2}$$

$$\boldsymbol{\Psi} = (\rho, \pi_1, \pi_2, \pi_3, U, B_1, B_2, B_3)^T, \tag{3}$$

$$\mathbf{S} = (0, \rho g_1, \rho g_2, \rho g_3, \boldsymbol{\pi} \cdot \boldsymbol{g}, 0, 0, 0)^T. \tag{4}$$

where ε_{ijk} is the Levi-Civita symbol, and δ_{ij} the Kronecker delta.

Weak formulation reads

$$\int_{\Omega_t} \frac{\partial \boldsymbol{\Psi}}{\partial t} \mathbf{v} - \int_{\Omega_t} \mathbf{F}(\boldsymbol{\Psi})(\nabla \cdot \mathbf{v}) + \int_{\partial \Omega_t} (\mathbf{F}(\boldsymbol{\Psi}) \cdot \boldsymbol{n}) \mathbf{v} = \int_{\Omega_t} \mathbf{S}\mathbf{v}, \tag{5}$$

where the terms $\mathbf{F}(\boldsymbol{\Psi})(\nabla \cdot \mathbf{v})$; $(\mathbf{F}(\boldsymbol{\Psi}) \cdot \boldsymbol{n}) \mathbf{v}$ are meant as a component-wise multiplication, Ω_t is a domain occupied by the fluid at time t, and $\mathbf{v}$ are test functions from a suitable Sobolev space.

We employ the standard *broken Sobolev space*, obtaining thus a finite dimensional space of basistest functions defined on hexahedral elements. Since we are interested in large scale problems, the domain is decomposed using standard approaches to do so - using the MPI communication layer implemented in the deal.II [4] and Trilinos [5] frameworks. Cubical domain Ω with color-coded processor-owned elements

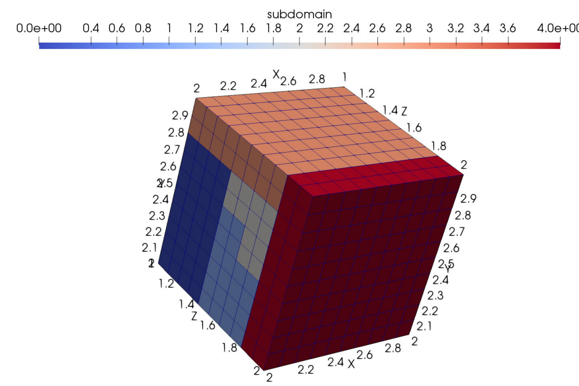

Fig. 2: Cubical domain Ω with color-coded processor-owned elements.

is visible on Fig. 2.

Divergence-free FE Space

The divergence-free constraint of the magnetic field, $\nabla \cdot \boldsymbol{B} = 0$ (Gauss's law) is not enforced by the solution of the problem (5). Therefore, we need to perform additional work to be sure that we do not have a non-physical solution in the sense that the constraint is not satisfied. There are two

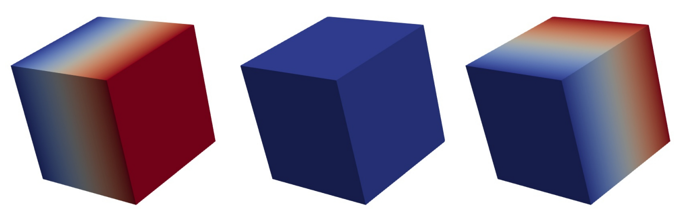

Fig. 3: One of the divergence-free basis functions.

often used approaches to handle this problem - the Constraint-Transport (CT) method, and divergence cleaning. The first one is not suitable for this work, as it constraints the triangulation in such a way, that implementing Adaptive Mesh Refinement would be very complicated, if possible at all. The second approach, the divergence cleaning methods need additional postprocessing step which may be omitted for the sake of calculation efficiency. The approach taken in this work is to replace the standard FE space with such basis functions for the magnetic field component (B) with a vector-valued (3-dimensional) space V_h^B of functions that have exactly

$$\nabla \cdot \mathbf{v}_h^B = 0, \quad \mathbf{v}_h^B \in V_h^B, \tag{6}$$

where these functions are as before discontinuous on element interfaces Γ_{ij}. An example of such a function is in Fig. 3.

Slope Limiting

It is well known that the Discontinuous Galerkin method exhibits nonphysical spurious oscillations in the vicinity of sharp discontinuities. Methods to amend that can be categorized according to multiple aspects. Out of these, two are important from the perspective of this work. First categorization is whether the approach changes the equations by introducing additional term that 'smoothes' the solution near the sharp front. In this work, such an approach is not preferred, as we aim at implementing a generally usable solver, where extensive analysis of the impact of a change in the governing equations for the particular problem is not possible.

Introduced by D. Kuzmin in [6], the Vertex-based limiter aims at being an improvement over the Barth-Jespersen limiter. It considers the solution in the form

$$u_h\left(x\right) = u_c + \alpha_e \left(\nabla u\right)_c \cdot \left(x - x_c\right), \; 0 \leq \alpha_e \leq 1, \tag{7}$$

but the definition of the correction factor α_K reads

$$\alpha_K = \min_i \begin{cases} \min\left\{1, \frac{u_i^{max} - u_c}{u_i - u_c}\right\}, & if \quad u_i - u_c > 0, \\ 1, & if \; u_i - u_c = 0, \\ \min\left\{1, \frac{u_i^{min} - u_c}{u_i - u_c}\right\}, & if \quad u_i - u_c < 0, \end{cases} \tag{8}$$

where in this case u_i^{min}, u_i^{min} are defined in such a way that for each of the vertices they are initialized with a small and a large constant, respectively, and then in the loop over all elements that contain the $i-$th vertex, the values are updated as:

$$u_i^{max} = \max\left\{u_c, u_i^{max}\right\}, \quad u_i^{min} = \min\left\{u_c, u_i^{min}\right\}. \tag{9}$$

This slope limiting technique proves to have all the required attributes from the perspective of this work. A comparison between an unlimited and limited solution is given in Figures (4) and (5), respectively.

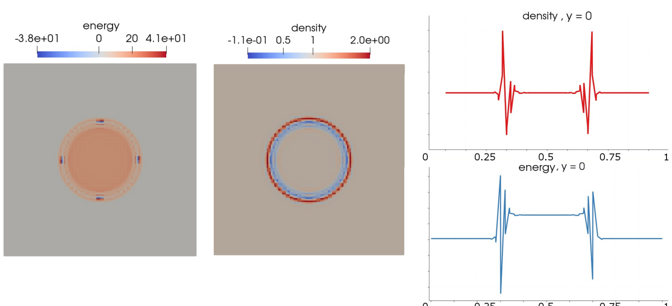

Fig. 4: Unlimited solution.

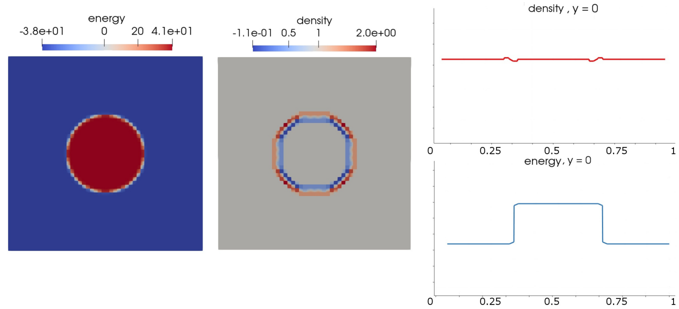

Fig. 5: Limited solution.

Automatic Mesh Refinement

Since the biggest challenge is the satisfaction of the requirement of both the fine resolution of the solution, and the capability of running large-scale simulations on modern distributed computing architecture - while still providing all the other attributes of the implemented code. The approach to this, which involves mainly the Adaptive Mesh Refinement technique (AMR) together with Domain Decomposition technique, is presented and implemented. And finally the verifications, and benchmarks and actual usage on an astrophysical problem are presented.

The AMR is a very important algorithm in the overall numerical solution, handling the multi-scale aspect of the studied problems. The general schema of any AMR algorithm is described in the Algorithm 1. The *solution acceptance criteria* is usually either spatial error estimate threshold, or number of elements, etc.

Algorithm 1: Generic AMR algorithm

Data: Mesh T_0
Result: A mesh T_n and a solution $\boldsymbol{y}_n$ on this mesh satisfying the solution acceptance criteria
i = 0
while *true* **do**
    obtain solution $\boldsymbol{y}_i$ on T_i
    evaluate solution $\boldsymbol{y}_i$ acceptance criteria
    if *solution acceptance criteria satisfied* **then**
        return
    else
        identify subset T_i^r of all elements $K \in T_i$ to be refined, $T_i^r \subseteq T_i$
        obtain T_{i+1} by refining (at least) all $K \in T_i^r$
        i = i + 1

V. Results

Several benchmarks (MHD Blast, Orszag-Tang vortex) - 7, 8, as well as a simulation of a flux-rope based on Titov-Demoulin model (9) have been done. All simulations show correct and well resolved solution patterns, and no non-physical oscillations appear in the solution. The automatic mesh refinement (AMR) framework clearly resolves the areas with sharp gradients well, and the selected element refinements

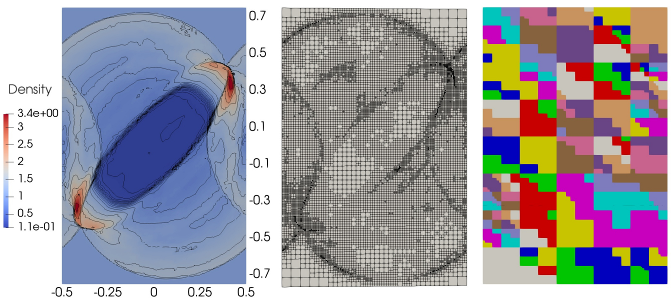

Fig. 6: MHD Blast problem, obtained ρ distribution, $t \approx 0.2$ (left), $t \approx 1.0$ (right).

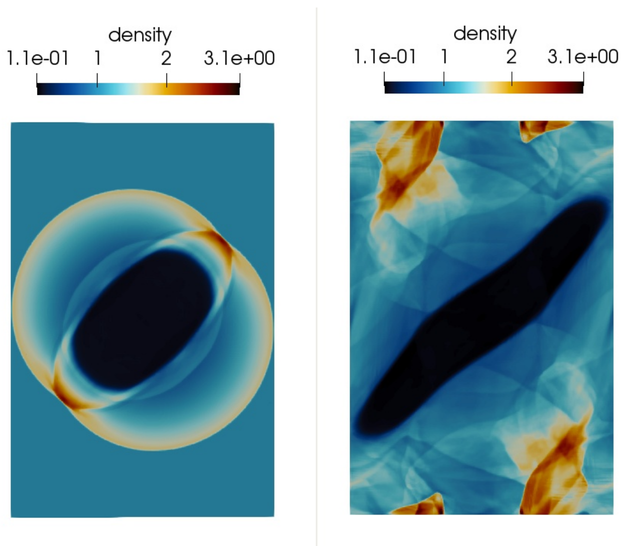

Fig. 7: MHD Blast problem, obtained ρ distribution, $t \approx 0.2$ (left), $t \approx 1.0$ (right).

lead to a dramatic reduction of the computation cost while maintaining the solution resolution.

VI. Conclusion

The goal of this work has been accomplished, and a software package implementing the discontinuous Galerkin method for MHD equations, together with AMR functionality, and without presence of artificial parameters in the implemented algorithms, was implemented. Further work in employing the software for solution of more MHD problems in astrophysics is being done.

Acknowledgment

This research has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the RICE New Technologies and Concepts for Smart Industrial Systems, project No. LO1607.

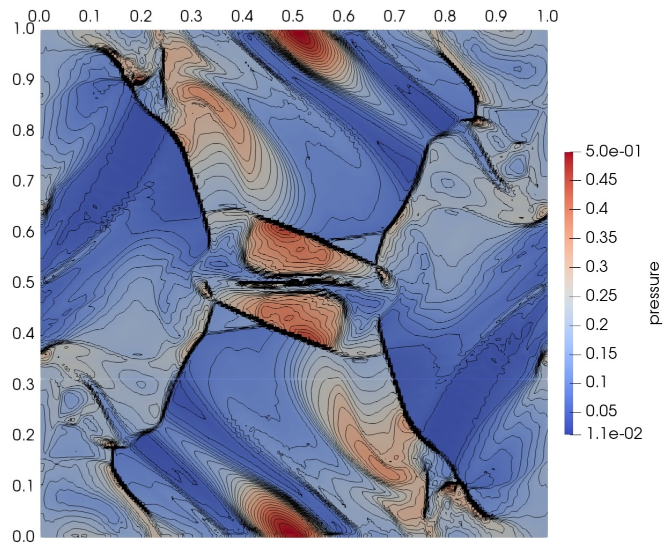

Fig. 8: Orszag-Tang problem, obtained p distribution at time $t \approx 0.5$.

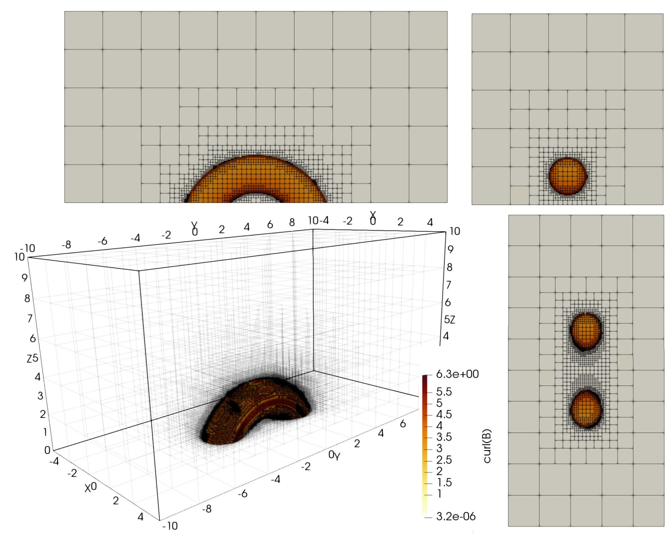

Fig. 9: Titov-Demoulin model, AMR result.

References

[1] P. Kotrc, M. Barta, P. Schwartz, Y. Kupryakov, L. Kashapova, and M. Karlicky, "Modeling of ha eruptive events observed at the solar limb," vol. 284, 06 2012.

[2] J. A. Rossmanith, "High-Order Discontinuous Galerkin Finite Element Methods with Globally Divergence-Free Constrained Transport for Ideal MHD," *ArXiv e-prints*, Oct. 2013.

[3] P. Mocz, M. Vogelsberger, D. Sijacki, R. Pakmor, and L. Hernquist, "A discontinuous Galerkin method for solving the fluid and magnetohydrodynamic equations in astrophysical simulations," *mnras*, vol. 437, pp. 397–414, Jan. 2014.

[4] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and D. Wells, "The `deal.II` library, version 8.4," *preprint*, 2015.

[5] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, "An overview of the trilinos project," *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 397–423, Sep. 2005. [Online]. Available: http://doi.acm.org/10.1145/1089014.1089021

[6] D. Kuzmin, "A vertex-based hierarchical slope limiter for p-adaptive discontinuous galerkin methods," *Journal of Computational and Applied Mathematics*, vol. 233, no. 12, pp. 3077 – 3085, 2010, finite Element Methods in Engineering and Science (FEMTEC 2009).