DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Cloud Supercomputing with the ExaHyPE-Engine

Lukas Krenz

DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Cloud Supercomputing with the ExaHyPE-Engine

Wolken Hochleistungsrechnen mit der ExaHyPE-Engine

Author:	Lukas Krenz
Supervisor:	Univ.-Prof. Dr. Michael Bader
Advisor:	Leonhard Rannabauer
Submission Date:	February 15, 2019

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, January , 2019 Lukas Krenz

Acknowledgments

Quote!

(Booktitle)

Acknowledgements here.

Abstract

Abstract here!

Contents

CHAPTER 1
Introduction

Introduction here.

CHAPTER 2 Methods

Methods here.

Organize into chapters!

2.1. Conservative Form

Standard (hyperbolic) form of conservation law

$$\frac{\partial}{\partial_t} Q + \boldsymbol{\nabla} \cdot \cdot F(Q) = S(\mathbf{x}, t, Q) \tag{2.1}$$

extend to:

$$\frac{\partial}{\partial_t} Q + \boldsymbol{\nabla} \cdot \cdot F(Q, \boldsymbol{\nabla} Q) = S(\mathbf{x}, t, Q) \tag{2.2}$$

2.2. The ADER-DG Method

We describe the arbitrary derivative discontinous Galerkin (ADER-DG) method in this chapter.

We discuss the solution of a hyperbolic conservation law equation (2.1) with domain Ω and boundary $boundary$. In the discontinous Galerkin (DG) framework, we approximate this solution in the space

$$\Omega = \bigcup_i C_i \tag{2.3}$$

of disjoint quadrilateral cells C_i. Note that we do not distinguish between the approximation space and the domain, the use should be clear given its surrounding context. In the following we make use of the Einstein summation convention where summation over repeated indexes is implied.

Inside each cell C_i we represent the solution in terms of the basis function

$$u(\mathbf{x}, t^n)_{|C_i} = \hat{u}^n_{i,l} \Phi_l \mathbf{x}, \tag{2.4}$$

where l is a multi-index, containing one index per spatial dimension. For example,

dofs $\hat{u}$. local solution, etc.

$(l = (l_1, l_2))$ for the two dimensional case. This polynomial is interpolating, i.e. .... This choice of basis (called *nodal* basis) allows us to easily compute integrals over cells using Gaussian quadrature. In detail, we use a the Lagrange interpolation polynomials.

interpolating?

We now describe the derivation of the ADER-DG method. This scheme is a predictor-corrector method. We first compute a local solution of the cell in the predictor step and then connect with the neighbors in the corrector step. In the following we first describe the corrector step as it follows directly from the PDE. The predictor is derived shortly after.

Corrector

First, we multiply the system equation (2.1) by a test function Φ_i and integrate over the space-time volume $(C_i \times [t^n, t^{n+1}])$. We arrive at the so called weak formulation of the PDE

$$\int_{t^n}^{t^{n+1}} \int_{C_i} \Phi_k \frac{\partial \mathbf{Q}}{\partial t} \, dx \, dt + \int_{t^n}^{t^{n+1}} \int_{C_i} \Phi_k \left(\boldsymbol{\nabla} \cdot \mathsf{F}(Q, \boldsymbol{\nabla} Q) \right) dx \, dt = \int_{t^n}^{t^{n+1}} \int_{C_i} \Phi_k S(Q, x, t) \, dx \, dt \tag{2.5}$$

Include gradient everywhere!

Double check the following description

We now replace the solution $\mathbf{Q}$ with so called spacetime-predictor $\mathbf{q}_h(\mathbf{x}, t)$ into the weak form and write it as a polynomial using the representation of equation (2.4). Integrate first by parts in time (note basis here not defined over time), flux divergence by parts in space.

$$\begin{aligned}
&\left(\int_{C_i} \Phi_k \Phi_l \, dx \right) \left(\int_{t^n}^{t^{n+1}} \int_{\partial C_i} \text{dofs boundary time} \, dS \, dt \right) + \\
&\quad \left(\int_{t^n}^{t^{n+1}} \int_{C_i} \text{Riemann} \, dx \, dt \right) - \left(\int_{t^n}^{t^{n+1}} \int_{C_i} \text{Flux} \, dx \, dt \right) = \left(\int_{t^n}^{t^{n+1}} \int_{C_i} \text{Source} \, dx \, dt \right)
\end{aligned} \tag{2.6}$$

The first term motivates an orthogonal basis as this leads to a diagonal mass matrix.

Predictor

To derive the predictor, we again take an approximation of our solution in a nodal basis, but now consider polynomials that are defined both in space and time.

$$\ldots \tag{2.7}$$

We know multiply the conservation law again by a test function (of the same function space as the basis) and arrive at the weak formulation

$$\ldots \tag{2.8}$$

Similar to the derivation of the corrector, we again integrate the first term by parts in time and the flux divergence in space. This time we do not use the Riemann solver for the flux boundary term but rather use the discrete solution at time t. Note that this neglects the interaction with neighbouring cells; this is corrected in the corrector step.

$$\dots \tag{2.9}$$

Inserting **??** results in a local systems of equations that can be solved in a fixed point iteration scheme. For details and proof of convergence for the linear case, see (todo). Write about initial guess (we use naive one anyway, right?)

Extrapolate unknowns to boundary, convert from our basis to basis on faces. Then Riemann solver and so on

Boundary extrapolation and stuff

Maybe write algorithm?

Full Scheme

Actually write this section. . .

2.3. Compressible Euler

Vector of conserved quantities:

$$Q = (\rho, \rho \boldsymbol{v}, \rho E) \tag{2.10}$$

Flux:

$$F(Q) = \begin{pmatrix} \rho \boldsymbol{v} \\ \boldsymbol{v} \otimes \rho \boldsymbol{v} + p\mathbf{I} \\ \boldsymbol{v} \cdot (\mathbf{I}\rho E + \mathbf{I}p) \end{pmatrix} \tag{2.11}$$

$$p = (\gamma - 1)\left(\rho E - 0.5\left(\boldsymbol{v} \cdot \rho \boldsymbol{v}\right)\right) \tag{2.12}$$

2.4. Compressible Navier Stokes

Following [Dum10] by Dumbser.

The full compressible Navier Stokes equations with heat transfer can be described by

$$F(Q, \nabla Q) = \begin{pmatrix} \rho \boldsymbol{v} \\ \boldsymbol{v} \otimes \rho \boldsymbol{v} + \mathbf{I}p + \boldsymbol{\sigma}(Q, \nabla Q) \\ \boldsymbol{v} \cdot (\mathbf{I}\rho E + \mathbf{I}p + \boldsymbol{\sigma}(Q, \nabla Q)) - \kappa \nabla T \end{pmatrix}, \tag{2.13}$$

where p, $\boldsymbol{\sigma}$ and $(\kappa \nabla T)$ denote the pressure, stress tensor and heat flux respectively. Temperature is denoted by T.

We use the equation of state of an ideal gas

$$p = (\gamma - 1)\left(\rho E - 0.5\left(\boldsymbol{v} \cdot \rho \boldsymbol{v}\right)\right). \tag{2.14}$$

The temperature T relates to pressure and density by the ideal gas law

$$\frac{p}{\rho} = RT, \tag{2.15}$$

where R is the specific gas constant.

Write more text about constants, cite idolikecfd or sth

$$c_v = \frac{1}{\gamma - 1} R \tag{2.16}$$

$$c_p = \frac{\gamma}{\gamma - 1} R \tag{2.17}$$

$$R = c_p - c_v \tag{2.18}$$

$$\gamma = \frac{c_p}{c_v} \tag{2.19}$$

Heat conduction coefficient κ

$$\kappa = \frac{\mu\gamma}{\Pr} \frac{1}{\gamma - 1} R \tag{2.20}$$

with R gas constant and $\Pr$ Prandtl number

The viscous effects are modeled by the stress tensor

$$\boldsymbol{\sigma}(Q, \nabla Q) = \left({}^{2}\!/\!{}_{3} \mu \nabla \cdot \boldsymbol{v} \right) - \mu \left(\nabla(\boldsymbol{v}) + \nabla(\boldsymbol{v})^{T} \right). \tag{2.21}$$

2.4.1. Boundary conditions

To close the system we need to impose boundary conditions.

For some scenarios we use Cauchy boundary conditions. In most cases, we would like to impose periodic boundary conditions, due to the inner workings of ExaHyPE this is not possible. Instead we use the analytical solution of our problems at the boundary, imposing both value and gradients of the conservative variables. Note that this leads to an error when our problem does not posses an exact analytical solution. This is the case for test cases that are analytical solutions to the incompressible Navier Stokes equations but do not satisfy the compressible equation set.

As a physical boundary condition we limit ourselves to the no-slip boundary condition, where we assume that the fluid has a velocity of zero near the wall. We enfore this by setting

Check if this is the correct physical description!

$$\rho^{o} = \rho^{i}, \tag{2.22}$$

$$\rho\boldsymbol{v}^{o} = -\rho\boldsymbol{v}^{i}, \tag{2.23}$$

$$\rho E^{0} = \rho E^{i}, \tag{2.24}$$

$$(\nabla Q)^{o} = (\nabla Q)^{i}, \tag{2.25}$$

where a superscript of o and i denotes the values outside and inside of the boundary respectively.

2.4.2. Diffusive Flux stuff

Max eigenvalue of convective part $|\lambda_c^{\max}|$,i.e. of $(\partial \mathbf{F}/\partial \mathbf{Q}) \cdot \mathbf{n}$, and viscous part $\mathtt{maxViscEigen}$, i.e. of $(\partial \mathbf{F}/\partial (\nabla \mathbf{Q} \cdot \mathbf{n})) \cdot \mathbf{n}$ is

$$|\lambda_c^{\max}\| = \|\boldsymbol{v}\| + c \tag{2.26}$$

$$|\lambda_v^{\max}\| = \max\left(\frac{4}{3}\frac{\mu}{\rho}, \frac{\gamma\mu}{\Pr\rho}\right) \tag{2.27}$$

with speed of sound $c = \sqrt{\gamma RT}$

Maximum timestep (CFL)

$$\Delta t = \frac{\text{CFL}}{2N+1} \frac{h}{|\lambda_c^{\max}| + 2|\lambda_v^{\max}| \frac{2N+1}{h}} \tag{2.28}$$

with N polynomial order and h characteristic length scale of elements

Numerical flux:

$$G(q_h^-, \nabla q_h^-; g_h^+, \nabla q_h^+) \cdot \mathbf{n} = \frac{1}{2}\left(F(q_h^+, \nabla q_h^+) + F(q_h^-, \nabla q_h^-)\right) - \frac{1}{2} s_{\max}(q_h^+ - q_h^-) \tag{2.29}$$

with

$$s_{\max} = \max\left(|\lambda_c(q_h^-)|, |\lambda_c(q_h^+)\right) + 2\eta \max\left(|\lambda_v(q_h^-)|, |\lambda_v(q_h^+)\right) \tag{2.30}$$

and

$$\eta = \frac{N+1}{h} \tag{2.31}$$

2.5. Scenarios

2.5.1. Clouds

The scenario is described in terms of potential temperature θ

$$\theta = T \frac{p_0}{p}^{R/c_p}, \tag{2.32}$$

where $p_o = 10 \times 10^5$ Pa is the reference pressure. Solving for T leads to

$$T = \theta(\frac{p}{p_0})^{R/c_p}. \tag{2.33}$$

To allow for an easier description of the initial conditions we split the θ into a background state $\overline{\theta}$ and a pertubation θ'

$$\theta = \overline{\theta} + \theta'. \tag{2.34}$$

The initial conditions for pressure and density can be computed directly from the PDE once few assumptions have been given. We assume that $\overline{\theta}$ is constant over the entire domain. Inserting the definition of potential temperature, given by equation (2.33), into the equation of state (see equation (2.14)) leads us to

$$\rho(z) = \frac{p_0^{\frac{R}{c_p}} p^{\frac{1}{\gamma}}(z)}{R\overline{\theta}}. \tag{2.35}$$

Flow in hydrostatic equilibrium can now be described by the ordinary differential equation(ODE)

$$\frac{d}{dz} p(z) = -g\rho(z) = -\frac{g p_0^{\frac{R}{c_p}} p^{\frac{1}{\gamma}}(z)}{R\overline{\theta}} \tag{2.36}$$

$$p(0) = p_0 \tag{2.37}$$

where we use the reference pressure p_0 as the pressure on ground level. This ODE can be now simply solved by separation of variables. After simplifying, we arrive at the initial condition for pressure

$$p(z) = \left(\left(1 - \frac{1}{\gamma}\right) \left(C - \frac{g p_0^{\frac{R}{c_p}} z}{R\overline{\theta}}\right) \right)^{\frac{c_p}{R}}, \tag{2.38}$$

with a constant of integration

$$C = \frac{c_p p_0^{\frac{R}{c_p}}}{R}. \tag{2.39}$$

We now compute the pertubated potential temperature θ and use equation (2.33) to convert it to temperature. We then evaluate the density $\rho(\mathbf{x})$ with equation (2.15). The energy can be computed by inserting the previously computed pressure into the equation of state (2.14).

2.6. Convergence

We first define the L_p norms for $p > 0$ by

$$\|f(x)\| = \left(\int_K |f(x)|^p \, d\mu \right)^{1/p}. \tag{2.40}$$

We start by a known analytical solution $f(x)$ and compare it to our approximation $\hat{f}(x)$. Let Q and $\hat{Q}$ denote the analytical and approximate solution at node. We start by computing the point-wise error, and compute the cell-wise error from this. Observe that for our broken space Ω

$$\|f(x)\| = \sum_{K \in \Omega} \|f_K(x)\|_p. \tag{2.41}$$

We are now ready to integrate the error for each cell. The output contains the position and conserved variables of each point. Each cell consists of $(N+1)^d$ nodes, each associated with a quadrature weight w_i. The quadrature weights are associated with the reference cuboid. We thus need to map our element to the reference element, before we can compute the integral.

Let V denote the volume (or area) of each cell. For example, in three dimensions $V = \Delta x \, \Delta y \, \Delta z$. The final result is then

$$\|f_K(x)\| = \left(V \sum_i |f(\mathbf{x_i})|^p w_i \right)^{1/p}, \tag{2.42}$$

where we used the mapping to the reference triangle (integration by substitution).

Following [Dum10], the scenario can be described in primitive variables by setting

$$p = p_0 \cos(\mathbf{k}\mathbf{x} - \omega t) + p_b, \tag{2.43}$$

$$\rho = \rho_0 \sin(\mathbf{k}\mathbf{x} - \omega t) + \rho_b, \tag{2.44}$$

$$\mathbf{v} = \mathbf{v_0} \sin(\mathbf{k}\mathbf{x} - \omega t). \tag{2.45}$$

We set the constants to $\left(p_0 = 0.1, p_b = \gamma^{-1}, \rho_0 = 0.5, \rho_b = 1, \mathbf{v_0} = 0.25(1,1)^\mathsf{T}, \mathbf{k} = \pi/5(1,1)^\mathsf{T}\right)$.

We derive a source term by inserting this solution into the PDE using the symbolic math toolkit *SymPy*.

CHAPTER 3
Conclusion

Conclusion here.

APPENDIX A

Placeholder…

Bibliography

[Dum10] M. Dumbser. “Arbitrary high order PNPM schemes on unstructured meshes for the compressible Navier–Stokes equations.” In: *Computers & Fluids* 39.1 (2010), pp. 60–76 (see pp. 4, 8).