Simulation of the Advection Equation on Arbitrary Meshes using L^1-Optimized Slope Limiters

Maximilian Ahsmus
Leibniz-Institute for Tropospheric Research, Leipzig, Germany
ahsmus@tropos.de

Introduction

Why is Slope-Limiting necessary?

Numerical simulations are highly necessary in weather-modeling, but have the problem, that they are only approximations. In the advection equation this problem shows, by solving the discretized equation using Runge-Kutta methods. To minimize this errors there are several possibilities of slope limiting.
Another defiance of weather-modeling is, that it has to be efficient and flexible. This means it has to work on different meshes and the computer has to do as less as possible calculations.
For slope limiting there is shown a possibility how to do it with L^1-optimization, which works for arbitrary meshes.

The Optimization

The Optimization Problem

By fitting the manifold there are two different methods. One is the fitting with the values in the midpoint of the grid cell, the other is the fitting with the integral over the grid cell. For both we have an initial data set D of the form:

$$D = \{d_i \in \mathbb{R} : d_i = d(x_i, y_i), i \in I, (x_i, y_i) \in \mathbb{R}^2\}\,.$$

The fitted manifold should also hold the condition, that the value on the edges of the cell, we want to fit exact, is between the value of the cell and the surrounding cells. The problem that need to be optimized is:

$$\begin{aligned} \min \sum_{k=1}^{n} |d_k - f_k|\,, \\ d_0 \le f_{e_k} \le d_k : k \in \{1, \dots, m_1\} \subset \mathbb{N}\,, \\ d_0 \ge f_{e_k} \ge d_k : k \in \{m_1 + 1, \dots, m\} \subset \mathbb{N}\,, \\ d_0 = f_0\,. \end{aligned}$$

The Equivalent Linear Program

By transforming the absolute value $|d_k - f_k|$ to a new value β_k we get a problem with more restrictions $-\beta_k \le d_k - f_k \le \beta_k$ and by rewriting the restrictions in the form $Mx \le b$, we get the linear program:

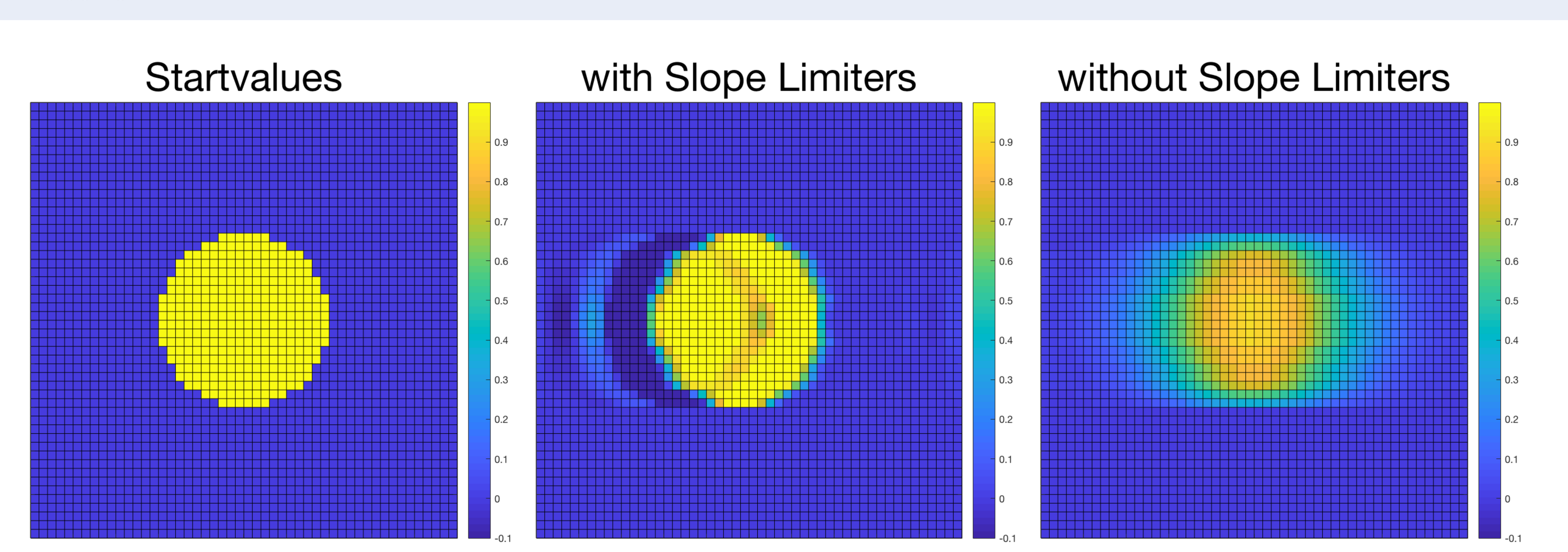
$$\min c^T \begin{pmatrix} \alpha \\ \beta \end{pmatrix}\,, \quad M \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \le b\,.$$

A similar transformation is shown in [1]. The differences between the fitting of the manifold with the midpoint value and the integral over the grid cell, is in the calculation of matrix M [2]. The matrix has in both cases the same dimensions.

Accuracy Comparison

The Advection Equation with and without Slope Limiting

As we can see the absolute Value of the differences between the start values and the final values is better with slope limiting. Using another Runge-Kutta method for the advection equation does not lead to a better solution. E is the error between the start values and the final values. The index s means with slope limiters and w without slope limiters.

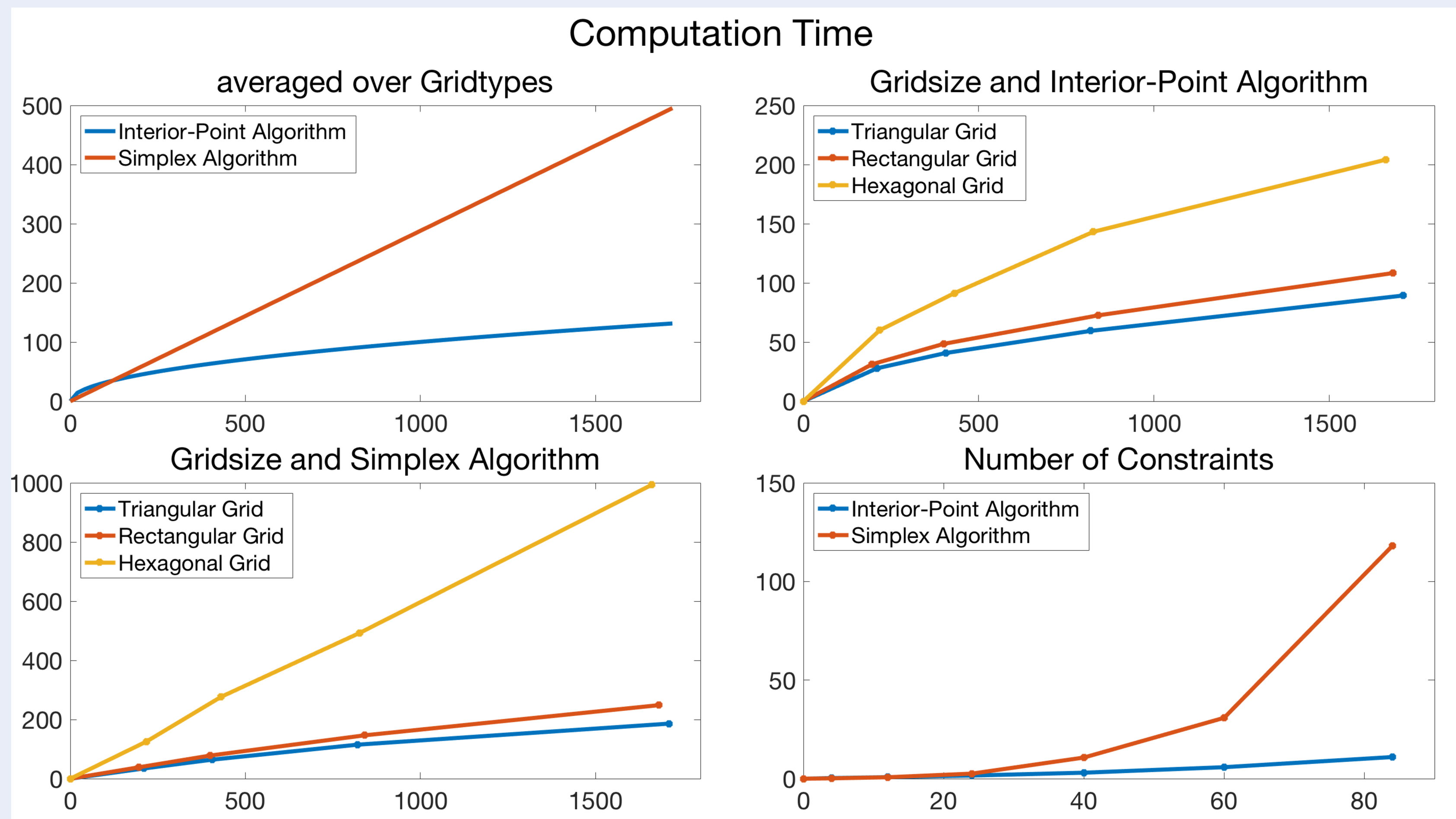


For the errors we get: $E_s = 129.9984$ and $E_w = 189.4537$.

The Algorithms

The Computation Time

The interior-point algorithm is implemented as in [3] and the simplex algorithm as in [4]. If the algorithm solves the problem it is faster than the simplex algorithm because of the sparsity of the matrix M.



Possibility for faster Computation

For the interior-point and the simplex algorithm there could be an increase of speed, when a warm start is implemented like in [5] explained. The warm start simplex and warm start interior-point algorithm could be faster than the interior-point algorithm. Initializing a warm start, the matrix M can be expanded. This leads to an initial basis made out of the basis from the previous optimization of the cell and an added 1 as additional basis variable. So the following linear program has to be solved:

$$\min \left(c^T 0\right) \begin{pmatrix} \alpha \\ \beta \\ 1 \end{pmatrix}\,, \quad \left(M\ (b - b_p)\right) \begin{pmatrix} \alpha \\ \beta \\ 1 \end{pmatrix} \le b_p\,.$$

This possible formulation of the problem needs also a minimal amount of memory because only the last solution is cached.

Conclusion

As shown this sort of implementation leads to an increase of accuracy. The efficiency is better with the interior-point algorithm especially for big grids. But the efficiency could be better, if a warm start is done, which will be done next. The flexibility is given because of the different types of cells, that can all be calculated using the same algorithms.
Farther it needs to be checked, where the differences between the L^1- and L^2-optimization show up and especially if the computation time is not so different between these two types of optimization.

References

[1] C. Gersbacher. "Higher-Order Discontinuous Finite Element Methods and Dynamic Model Adaptation for Hyperbolic Systems of Conservation Laws". PhD thesis. Albert-Ludwigs-University Freiberg im Breisgau, 2017.
[2] P. H. Lauritzen et al. *Numerical Techniques for Global Atmospheric Models*. Heidelberg: Springer, 2011.
[3] J. Nocedal and S. J. Wright. *Numerical Optimization*. New York: Springer, 2006.
[4] N. Ploskas and N. Samaras. *Linear Programming Using MATLAB*. Cham: Springer, 2017.
[5] E. Yildirim and S. Wright. "Warm-Start Strategies in Interior-Point Methods for Linear Programming". In: *SIAM Journal on Optimization* 12.3 (2002). DOI: 10.1137/S1052623400369235.