

Visualization of the Solution of Advection-Diffusion-Reaction Equations

V.T. Luan¹, M. Ritter² and G. Ritter³

¹ Institute of Mathematics, ² Institute of Basic Sciences in Civil Engineering, University of Innsbruck, Austria

³ Centre for Atmospheric and Instrumentation Research (CAIR), University of Hertfordshire, Hatfield, AL10 9AB, U.K.

ADR Equations in 2D and Numerical Approach

Introduction:

We solve the spatial advection-diffusion-reaction (ADR) models numerically using standard FDM leading to a stiff ODE system. We compute the numerical solution of this system based on exponential integrators and visualize the results using 3D volume rendering and ortho-slices in the VISH visualization shell.

Mathematical model:

Consider an ADR model in 2D for $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^N$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(a\mathbf{u}) - \Delta(d\mathbf{u}) = R(\mathbf{u}), \quad \mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0$$

on a smoothly bounded domain $\Omega \subset \mathbb{R}^2$, with the boundary $\partial\Omega$ subject to homogeneous Neumann boundary conditions. The advection field a and the diffusion coefficient d are given. The model problem can be reformulated as an abstract ODE:

$$\mathbf{u}'(t) = F(\mathbf{u}(t)), \quad F(\mathbf{u}) = A\mathbf{u} + R(\mathbf{u}), \quad \mathbf{u}(t_0) = \mathbf{u}_0,$$

where $A = -\nabla(a.) + \Delta(d.)$

Numerical method: Exponential Rosenbrock methods (see [3])

Let $\mathbf{u}_n \approx \mathbf{u}(t_n)$, $\mathbf{u}_{n+1} \approx \mathbf{u}(t_{n+1})$, $t_{n+1} = t_n + h$

Linearizing the above problem in each step at \mathbf{u}_n to get

$$\mathbf{u}'(t) = \mathbf{J}_n \mathbf{u}(t) + \mathbf{g}_n(\mathbf{u}(t)), \quad \mathbf{J}_n = \frac{\partial F}{\partial \mathbf{u}}(\mathbf{u}_n), \quad \mathbf{g}_n(\mathbf{u}) = F(\mathbf{u}) - \mathbf{J}_n \mathbf{u}.$$

Scheme:

$$\mathbf{U}_{ni} = \mathbf{u}_n + \mathbf{c}_i h \varphi_1(\mathbf{c}_i h \mathbf{J}_n) F(\mathbf{u}_n) + h \sum_{j=2}^{i-1} \mathbf{a}_{ij} (h \mathbf{J}_n) \mathbf{D}_{nj},$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + h \varphi_1(h \mathbf{J}_n) F(\mathbf{u}_n) + h \sum_{i=2}^s \mathbf{b}_i (h \mathbf{J}_n) \mathbf{D}_{ni},$$

$$\mathbf{D}_{ni} = \mathbf{g}_n(\mathbf{U}_{ni}) - \mathbf{g}_n(\mathbf{u}_n)$$

Experiment:

- + Parameters taken from [3]
- + The advection field: $a=10$
- + The diffusion coefficient: $d=1/100$
- + Domain: $\Omega = [0, 1]^2$
- + Initial value:

$$\mathbf{u}(\mathbf{x}_1, \mathbf{x}_2, 0) = 0.3 + 256(\mathbf{x}_1(1 - \mathbf{x}_1)\mathbf{x}_2(1 - \mathbf{x}_2))^2$$

To compute the numerical solution, we use the new exponential Rosenbrock **exprb54s5** (order 5) which is proposed in [4]. For a comparison of visualization, we also use the old scheme **exprb43** (order 4) in [3].

References

- [1] Bengner W., Ritter G., Heinzl R. (2007). *The concepts of vish*, 4th High-End Visualization Workshop, Obergurgl, Austria, Jun 18-21 (pp.26-39): Berlin, Lehmann Media-LOB.de
- [2] Engel K., Hadwinger M., Kniss J.M., Rezk-Salama C., Weiskopf D. (2006). *Real-Time VOLUME GRAPHICS*: A K Peters Ltd
- [3] M. Hochbruck, A. Ostermann and J. Schweiter, *Exponential Rosenbrock-type methods*, SIAM J. Numer. Anal., 47 (2009), pp. 786–803.
- [4] V. T. Luan and A. Ostermann, *An exponential Rosenbrock method of order five - derivation, analysis and numerical comparison*, preprint.
- [5] Ritter M. (2009). *Introduction to HDF5 and F5*, CCT Technical Report Deries, Baton Rouge, Louisiana: CCT-TR-2009-13

Techniques for the Solution Visualization and Results

Strategies:

- Utilize visualization shell VISH [1], a flexible module based on framework for interactive scientific visualization
- Interpret time as space (z-axis)
- Ortho-slices and volume-renderings
- Difference 3D plot of $u(\mathbf{x}, t)$ as a function of \mathbf{x} and t

Data conversion:

- Stored the data of the numerical solution in MATLAB binary
- Octave plugin to general purpose scientific format F5 [5]

Results:

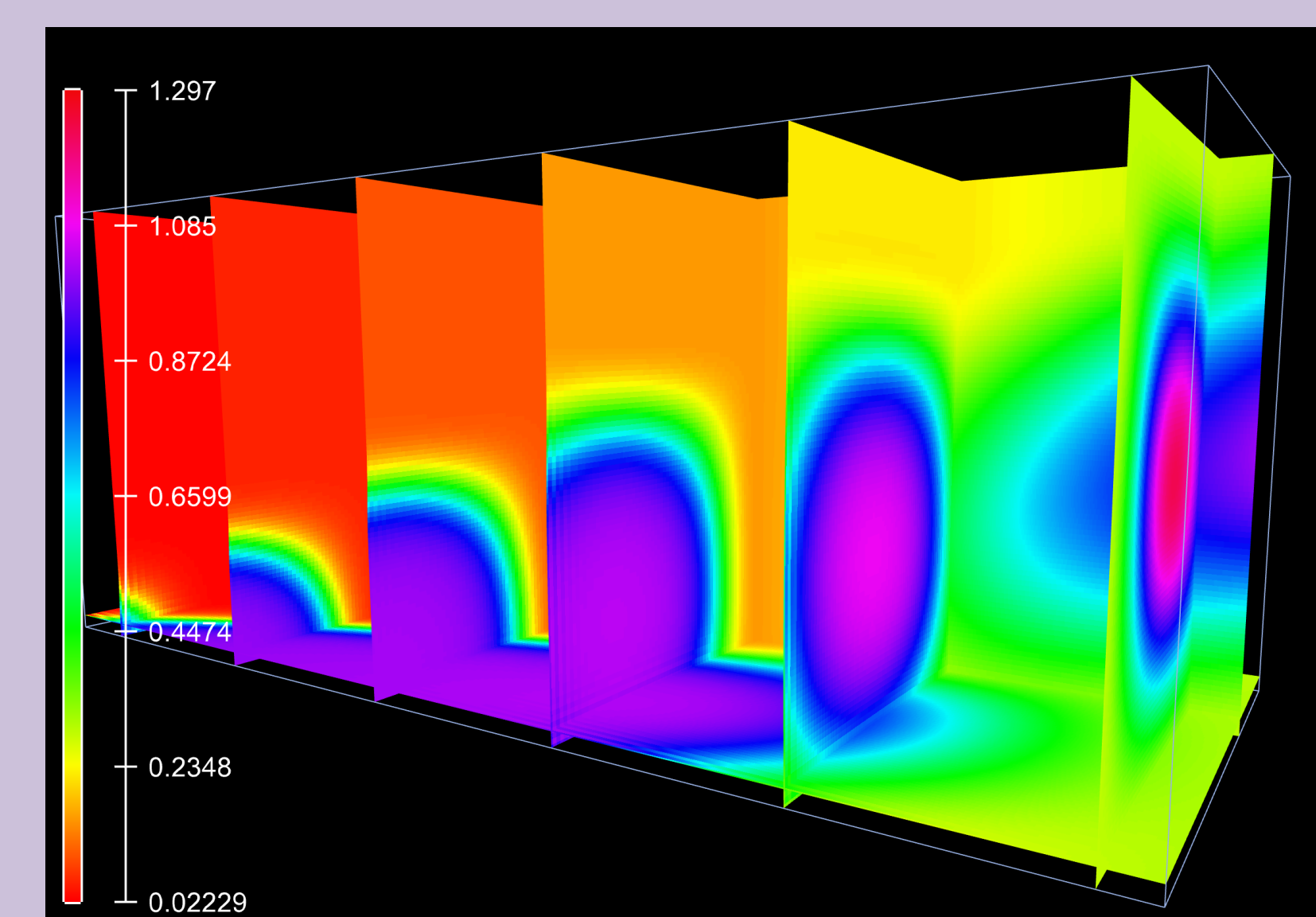


Fig.1: Color-mapped ortho-slices. z-axis runs from right to left illustrating the time evolution of the 2D initial condition. The slices show some linear artifacts close to the border as well as some jittering in the z-orthogonal slices.

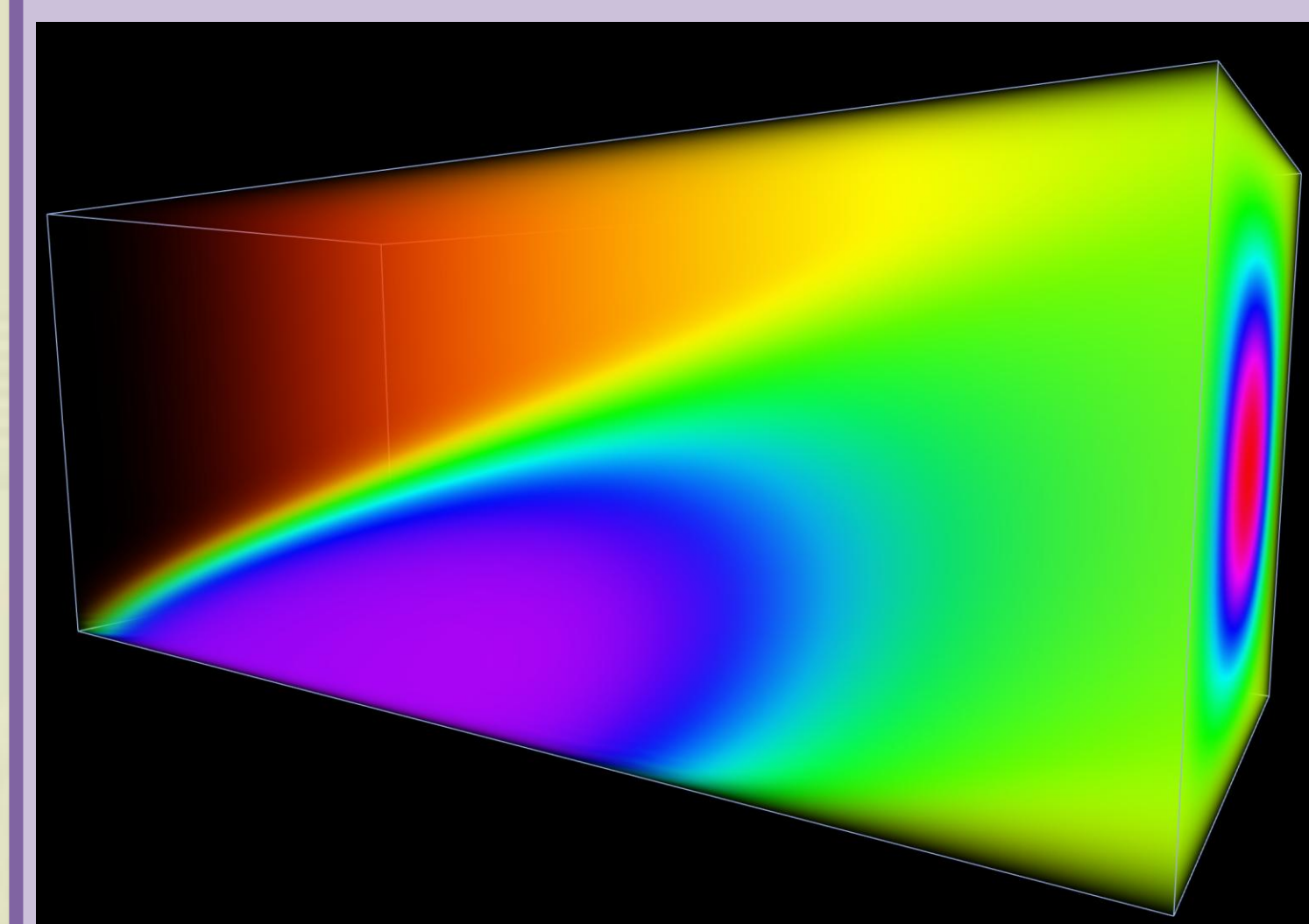


Fig.2: The full texture based volume rendering [2] illustrates the full domain but also occludes itself..

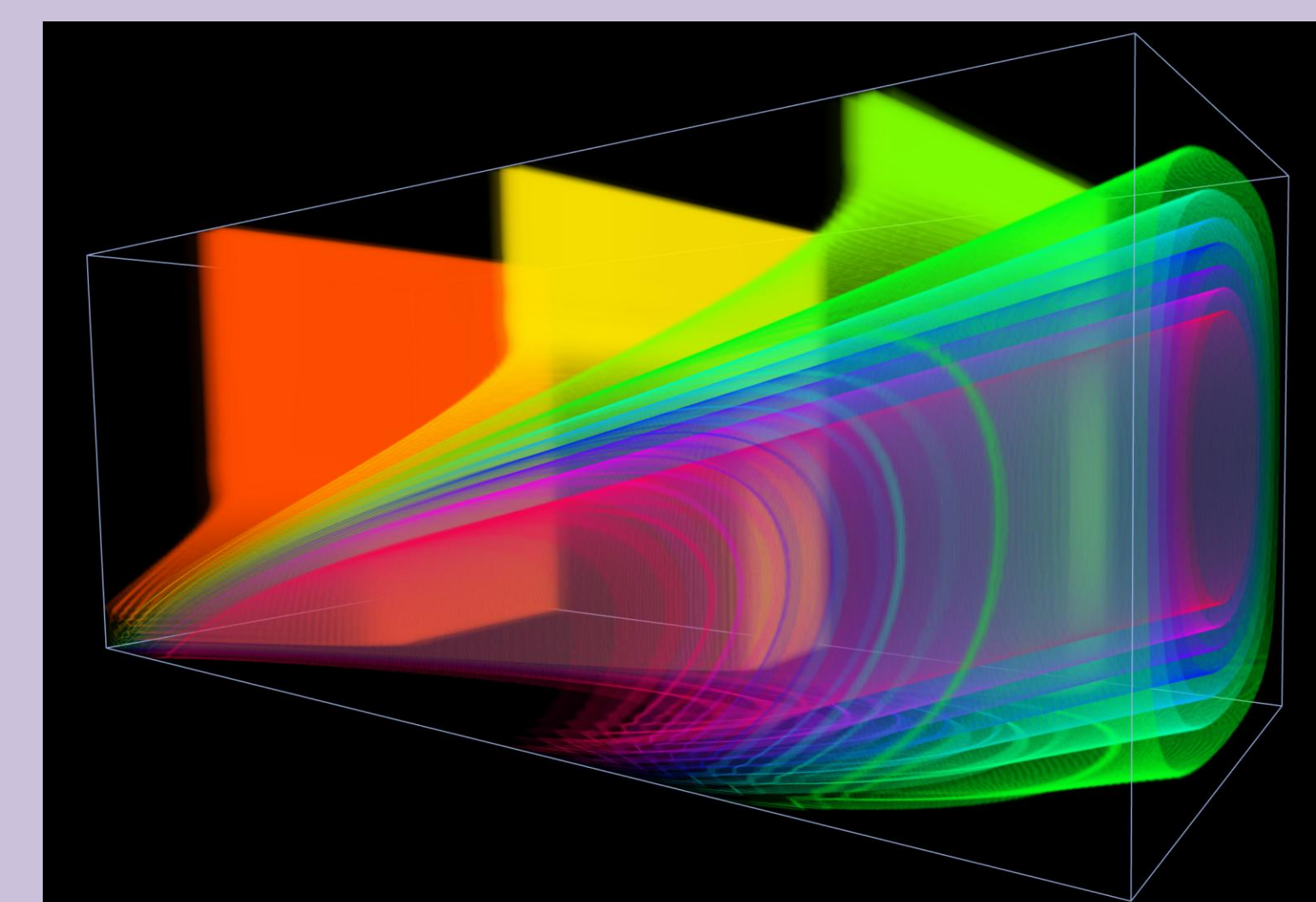


Fig.3: A transfer function scales the transparency and shapes iso-surfaces enhancing the full volume rendering.

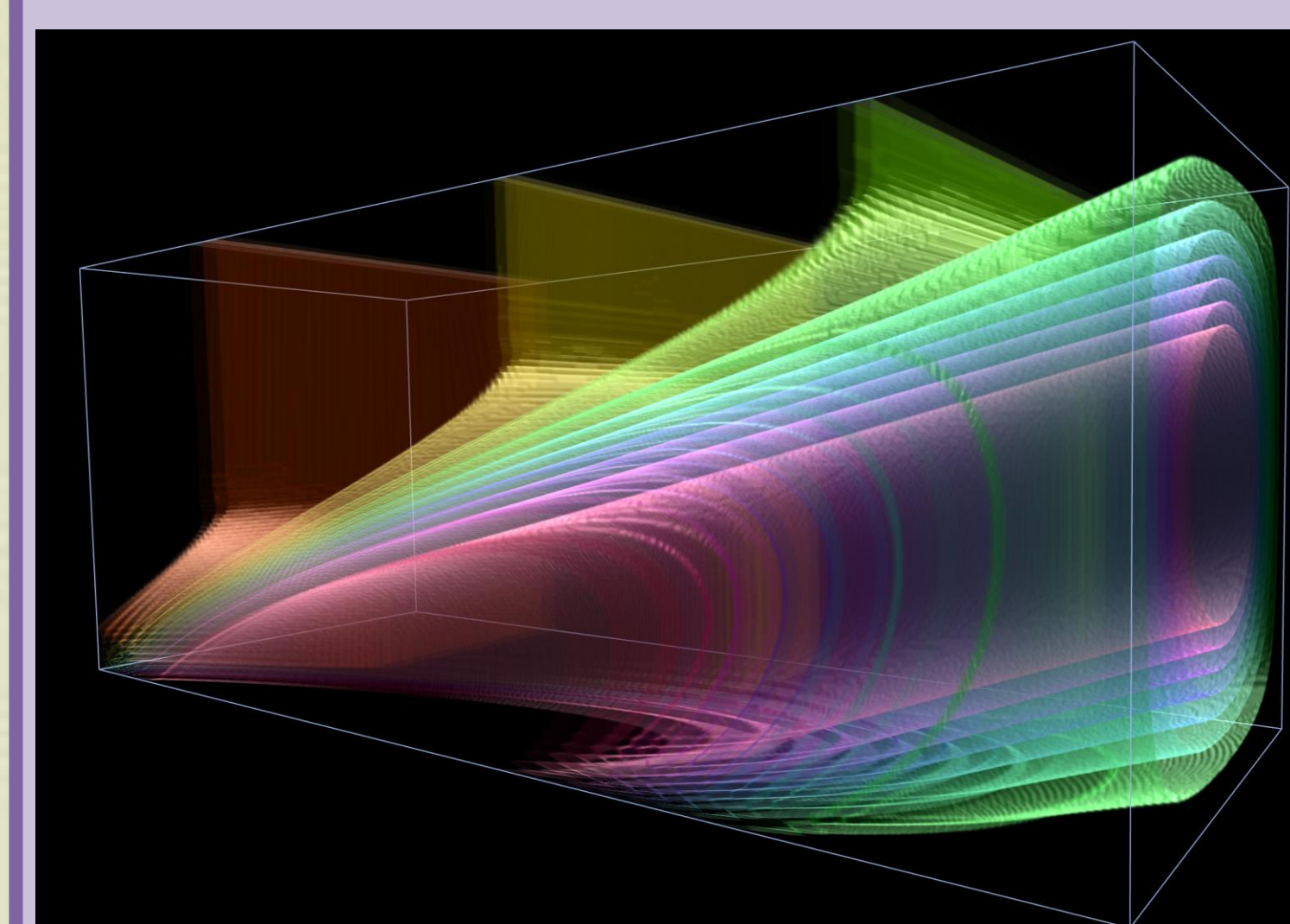


Fig.4.: Gradient of scalar field used as normal vector for illumination.

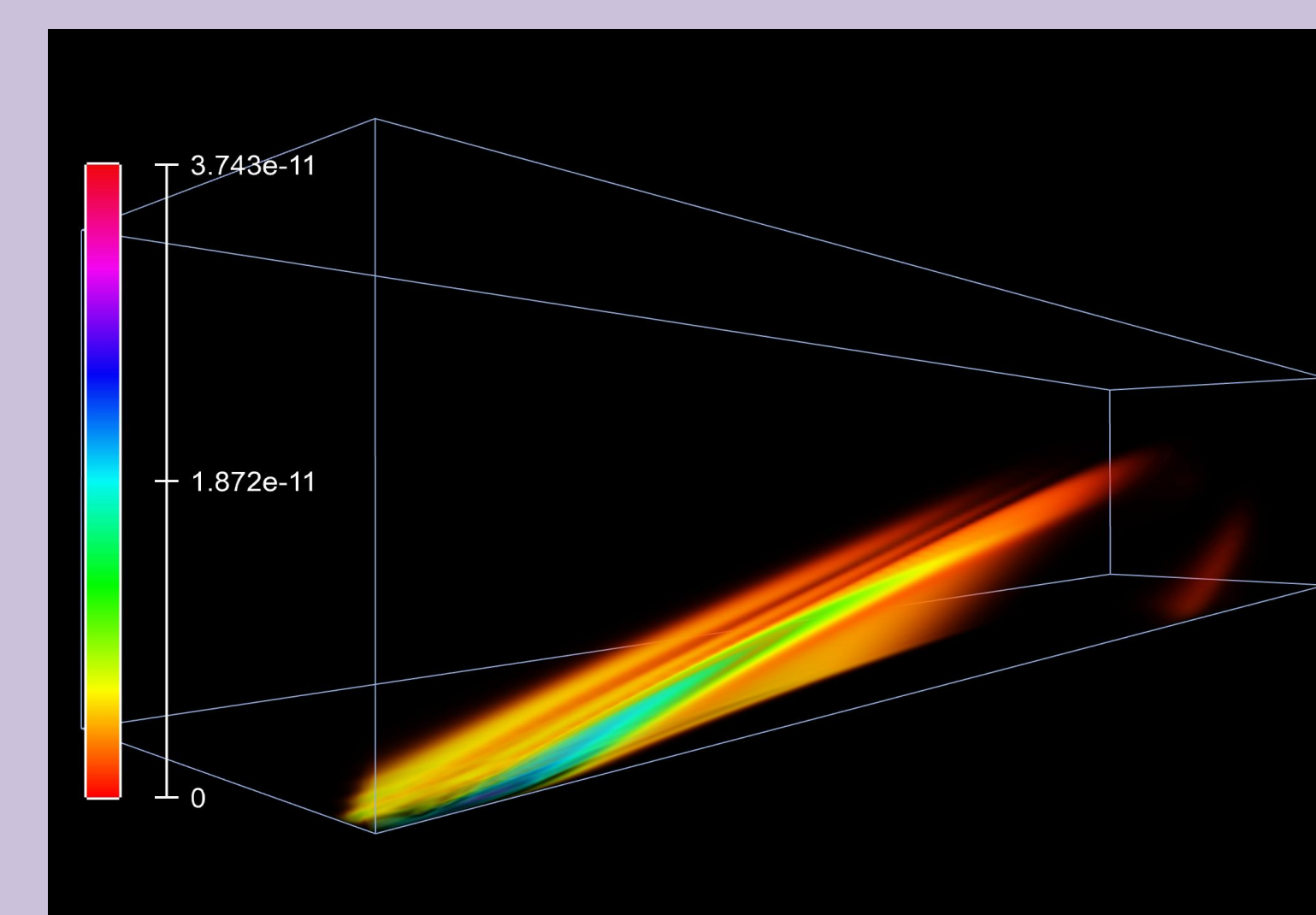


Fig.5: Volume-rendering of the difference of methods: **exprb54s5** and **exprb43**. The difference increases in time (right to left) and concentrates in the shell like structure.

Future Work:

- Improve volume rendering by pre-integrated textures
- Add modules for binary operations on fields (e.g. differences)
- Stabilize F5 plugin for direct Matlab export (binary releases)