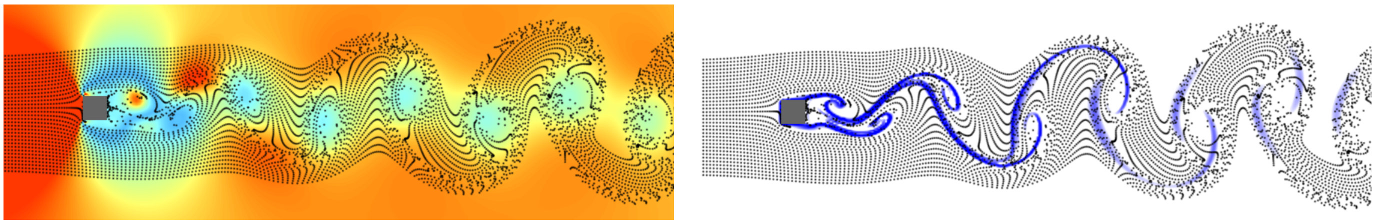
Flow Simulation and Visualization

Marco Ament



(a) (b)

Fig. 1: GPU-based Navier-Stokes simulation [3] with interactive manipulation of the boundary conditions like solid obstacles, in- and outflow conditions, and the viscosity of the fluid. The flow direction is from left to right and the obstacle leads to a Kármán vortex street. (a) Visualization of the pressure field (red means high, blue means low) combined with particle tracers. (b) Visualization of the finite-time Lyapunov exponent (blue means high, white means low) using backward-time integration.

Abstract—This report summarizes my research on real-time flow simulation and visualization. In particular, my research is concentrated on efficient GPU algorithms that can exploit the fine-granular thread parallelism of graphics hardware. Several established data structures and numerical solvers perform poorly on the GPU. For this reason, I study and develop new methods that are optimized for the underlying hardware architecture. In Eulerian flow simulations, the grid resolution plays an important role to capture turbulence effects on a small scale. To efficiently exploit the valuable computation and memory resources of a GPU, my work contributes an algorithm for dynamic grid refinement to interactively simulate and render smoke animations on the GPU. One of the most computationally expensive steps of a typical Navier-Stokes simulation is the numerical solution of the pressure Poisson equation. In this field, my work contributes a preconditioner for the conjugate gradient method that is optimized for the Poisson problem and for efficient GPU processing. In subsequent work, I exploited the gained performance benefit to spend more computation time for advanced visualization methods that can help control the flow by interactively manipulating boundary conditions and receive immediate feedback by visualizing Lagrangian coherent structures in real-time. For texture-based flow visualization, Semi-Lagrangian advection is often employed, which is susceptible to numerical diffusion; however, higher-order interpolation methods can be employed to reduce this effect. For this reason, I contributed an evaluation paper that studies the conservation of the frequency sprectrum for different interpolation methods.

Index Terms—Flow simulation, flow visualization, dynamic grid refinement, Poisson equation, Lagrangian coherent structures, higher-order interpolation

1 Introduction

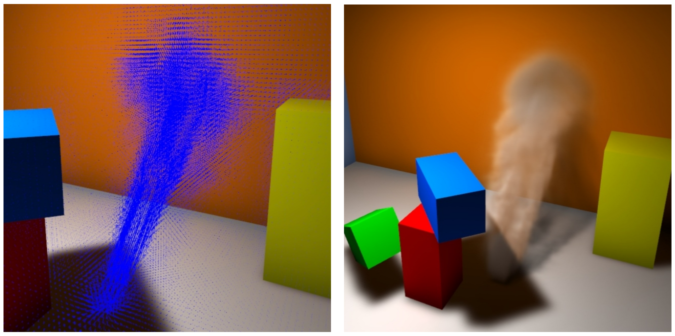
The simulation and visualization of flow dynamics play an important role in many scientific applications and in digital media. The increasing computational power of graphics processing units (GPUs) [2] offers scientists and digital artists novel opportunities for complex technical simulations and physically-based animations of fluids. In engineering, it is important to understand the phenomena behind the data to advance research and development in the application domain. For technical applications, the visual analysis of vector fields can be simplified by computing and visualizing derived scalar fields [1] like the finite-time Lyapunov exponent (FTLE), to extract meaningful coherent structures of the flow. In entertainment industry, the simulated vector field is employed for the realistic rendering and animation of fluids like water or smoke.

In the latter domain, visual details play a fundamental role for a realistic appearance. At same time, high performance is crucial, especially for real-time applications. In Section 2, an algorithm for dynamic grid refinement is presented that automatically adapts the spatial resolution of a Eulerian-based fluid simulation. However, traditional tree-based data structures are not suitable for SIMD-based architectures. For this reason, the focus of this work is on the efficient implementation of a GPU-friendly hierarchical data structure.

In a typical Navier-Stokes simulation, the numerical solution of the discrete pressure Poisson equation plays a fundamental role and is often a computational bottleneck. Despite their high computational power, the fine grained parallelism of modern GPUs is problematic for certain types of numerical solvers. The conjugate gradient algorithm is one the most widely used solvers for such systems of linear equations. Typically, the method is used in conjunction with preconditioning to accelerate convergence. However, traditional preconditioners are not suitable for efficient GPU processing. Therefore, a novel approach is introduced in Section 3, specifically designed for the discrete Poisson equation. The benefit of fast solvers is that higher resolutions can be employed or that more computation time can be spent for rendering or visualization.

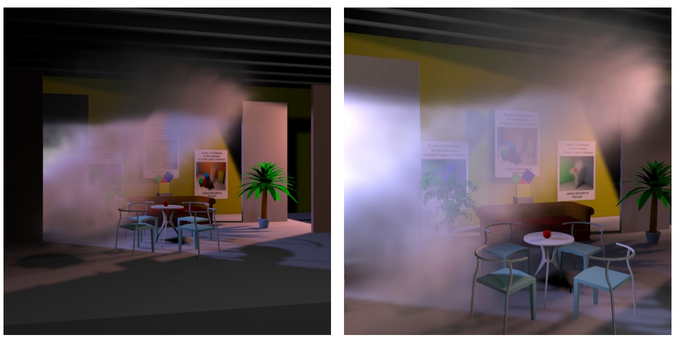
Many flow visualization methods provide only an instantaneous snapshot of a time-dependent vector field. In contrast, the FTLE allows one to visualize meaningful coherent structures over a time interval, which allows one to gain insight in time-dependent flows. However, computing the FTLE is expensive and usually requires integration of millions of particle trajectories over a long time period. In Section 4, the benefit of the previous Poisson solver is exploited to combine an interactive flow simulation with a FTLE visualization, which provides immediate visual feedback of the running simulation. In this way, computational steering is supported by visualizing regions of different behavior in a time-dependent vector field.

• *Marco Ament is with Karlsruhe Institute of Technology, Germany. E-mail: ament@kit.edu.*



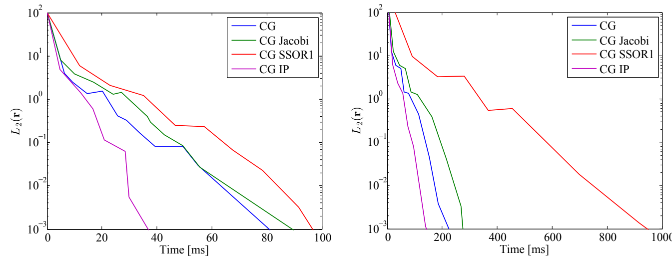
(a) (b)

Fig. 2: Interactive smoke simulation in the Cornell box. (a) Visualization of the velocity field with hedgehogs, showing the spatially varying resolution of the grid. (b) Corresponding rendering of the smoke with radiosity-based illumination.



(a) (b)

Fig. 3: Interactive smoke simulation in the stage scene. (a) View from a distance. Large parts of the scene do not contain any smoke. (b) Close-up view of the smoke.



(a) (b)

Fig. 4: Convergence plots of different preconditioners for the CG algorithm running on a single GPU. The plots show how fast the L_2-norm of the residual vector $\mathbf{r}$ is minimized with the pure CG method, the Jacobi preconditioner, the SSOR preconditioner, and the novel IP preconditioner. The problem size is (a) 32^3 and (b) 64^3 grid points.

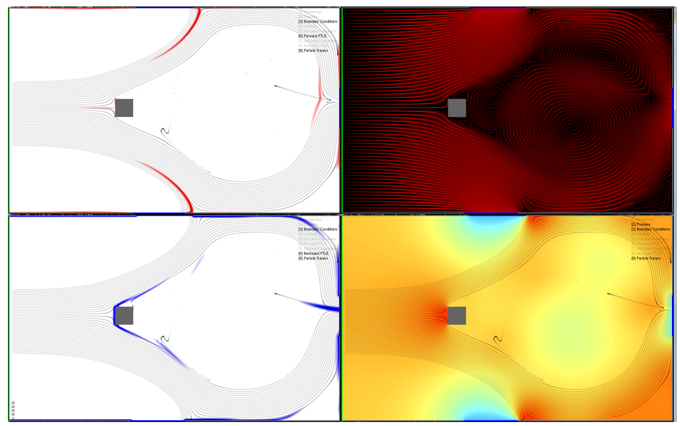


Fig. 5: Interactive flow steering with four different views that are connected with brushing and linking. A user can manipulate boundary conditions and receives immediate visual feedback in all views, showing forward-time FTLE with particles (top left), backward-time FTLE with particles (bottom left), path lines (top right), and pressure with particles (bottom right).

Texture advection is an important method in the flow visualization tool box. The aim is to solve the advection equation accurately so that important features, such as edges or spectrum of the resulting image, are conserved. But due to discretization and interpolation an error occurs while solving the equation. This can lead to numerical diffusion and consequently to a loss of the feature properties. Especially iterative methods implicitly implement a low-pass filter while providing a very fast computation. In Section 6, higher-order interpolation schemes are evaluated with respect to their conservation of the frequency spectrum. Therefore, radial frequency sum diagrams are introduced to compare different time steps of the advection processes.

2 Dynamic Grid Refinement for Fluid Simulations

The paper *Dynamic Grid Refinement for Fluid Simulations on Parallel Graphics Architectures* [5] presents a physically-based fluid simulation that is suitable for efficient processing on GPUs. The irregular and dynamic structure of an adaptive grid requires sophisticated memory access patterns as well as a decomposition of the problem for parallel processing and the distribution of tasks to multiple threads. In this paper, a problem decomposition is presented that takes advantage of the specific properties of the hardware while reducing expensive hierarchy traversals. Moreover, the data representations of the fluid's quantities greatly affect the overall performance. Therefore, the technique distinguishes between the storage of the velocity field for the advection step and the storage of the pressure field for the Poisson solver to account for the different memory access patterns. The method is employed to create flexible smoke animations that can handle complex scene geometries and that adapt itself dynamically according to local refinement conditions. The smoke simulation is combined with radiosity-based rendering, which achieves interactive frame rates while walking through the scene. The technique was published in a conference paper at the Eurographics Symposium on Parallel Graphics and Visualization 2009 [5].

Figures 2(a) and (b) show a smoke simulation in the Cornell box. In Figure 2(a), the velocity field is visualized with blue hedgehogs. In areas of high velocity and vorticity, the spatial resolution is higher than in the outer regions. Figure 2(b) shows the same time step of the simulation, but the smoke is rendered with radiosity-based illumination. Figure 3(a) and (b) show smoke simulation in a large stage scene. Large parts of the scene do not contain any smoke as shown in the overview in Figure 3(a). Therefore, the spatial resolution of the grid is very low in the outer regions. Figure 3(b) shows a close-up of the smoke from a slightly different viewing angle.

3 Poisson Preconditioned Conjugate Gradients

The paper *A Parallel Preconditioned Conjugate Gradient Solver for the Poisson Problem on a Multi-GPU Platform* [4] introduces a novel preconditioning technique for the Poisson problem that is well suited for efficient multi-GPU processing. The discretization of the Poisson problem leads to a large and sparse system of linear equations, which is often solved with the preconditioned conjugate gradient (CG) algorithm. It is shown that performance of traditional preconditioners is strongly lowered by the hardware characteristics of GPUs. The

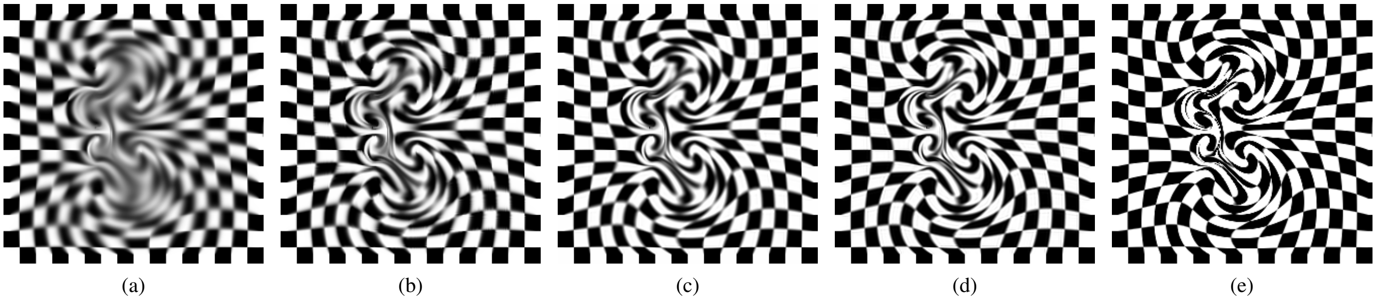


Fig. 6: Texture-based flow visualizations with different interpolation methods. Semi-Lagrangian advection with (a) bilinear interpolation, (b) back and forth error compensation and correction (BFECC), (c) bicubic interpolation, and (d) biquintic interpolation. (e) Ground truth without interpolation.

novel preconditioner is a sparse approximate inverse of the matrix and requires only GPU-friendly matrix-vector products. Furthermore, a multi-GPU algorithm is presented that builds on asynchronous computation and data transfer to improve performance scalability. The technique was published in a paper at the Euromicro Conference on Parallel, Distributed, and Network-based Processing 2010 [4].

Figures 4(a) and (b) show convergence plots for solving the discrete Poisson equation in 3D with different preconditioners on a single GPU for problem sizes of 32^3 and 64^3 grid points, respectively. On the GPU, commonly employed preconditioners like Jacobi or symmetric successive overrelaxation (SSOR) perform slower than no preconditioning at all due to the inherently serial forward and backward substitution steps in the CG algorithm. The novel incomplete Poisson (IP) preconditioner avoids these serial steps and achieves significant speedups compared to the pure CG method.

4 Flow Simulation Steering

The paper *GPU-based Two-dimensional Flow Simulation Steering using Coherent Structures* [3] exploits the previously introduced preconditioner as part of a numerical Navier-Stokes solver and presents an integrated environment for interactive steering and visualization of a 2D flow simulation. Such integrated environments require computational resources not only for the simulation but also for the visualization, which can be expensive as well. Therefore, efficient solvers play a crucial role in this case because the solution of the pressure projection is the most expensive part of the fluid simulation. A fast solver allows one to spend more computation time for advanced flow visualization techniques, for example, to compute the FTLE for the visualization of Lagrangian coherent structures (LCS). The integrated framework was published in a paper at the Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering 2011 [3].

Figure 5 shows a simple 2D box with a cubic obstacle inside, one large inflow boundary condition (green) on the left, and three small pressure boundary conditions (blue) on the top, the right, and the bottom side. The solid walls and the obstacle (gray) are modeled with no-slip boundary conditions. The four views show different visualizations of the same flow simulation. The views are connected with brushing and linking and a user receives immediate visual feedback in all views when the boundary conditions are changed.

Figure 5(top left) shows the forward-time FTLE field computed from 80 time steps of the simulation and mapped to red color. The repelling LCS close to the obstacle and the outlets clearly show where the flow of the particles is separated. In addition, black particles are advected with the flow to visualize parts of the vector field as well. Similarly, Figure 5(bottom left) shows a visualization of the backward-time FTLE field, also computed from 80 time steps and mapped to blue color. The attracting LCS indicate areas where flow of the particles is merging, for example, close to the right outlet, where the upper and lower currents are joining before flowing out of the domain. In Figure 5(top right), the flow is visualized with path lines to provide a dense overview of trajectories. Figure 5(bottom right) employs color mapping to visualize the pressure field.

5 Higher-order Texture Advection

The paper *Spectral Analysis of Higher-Order and BFECC Texture Advection* [6] presents a spectral analysis of higher-order texture advection in combination with Back and Forth Error Compensation and Correction (BFECC). Semi-Lagrangian techniques exhibit high numerical diffusion, which acts as a low-pass filter and tends to smooth out high frequencies. In the spatial domain, numerical diffusion leads to a loss of details and causes a blurred image. To reduce this effect, higher-order interpolation methods or BFECC can be employed. In the paper, different compositions of higher-order interpolation schemes with and without BFECC are analyzed with radial power spectra for different advection times and input textures to evaluate the conservation of the frequency spectrum up to fifth-order polynomials.

Figure 6 shows flow visualizations of a time-dependent vector field with a checkerboard texture and Semi-Lagrangian advection after 100 time steps. In Figure 6(a), bilinear interpolation is employed, which leads to strong blurring due to numerical diffusion. In Figure 6(b), BFECC interpolation reduces the loss of high frequencies, but blurring is still significant. With bicubic interpolation, in Figure 6(c), smoothing is further reduced, while biquntic interpolation in Figure 6(d) leads only to small additional improvements. Figure 6(e) shows ground truth that does not exhibit any numerical diffusion by integrating a full path line backward in time to determine the location of each pixel in the original image at time step 0.

References

[1] M. Ament. *Computational Visualization of Scalar Fields*. PhD thesis, University of Stuttgart, 2014.

[2] M. Ament, S. Frey, C. Müller, S. Grottel, T. Ertl, and D. Weiskopf. GPU-accelerated visualization. In E. W. Bethel, H. Childs, and C. Hansen, editors, *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, pages 223–260. Chapman and Hall/CRC, 2012.

[3] M. Ament, S. Frey, F. Sadlo, T. Ertl, and D. Weiskopf. GPU-based two-dimensional flow simulation steering using coherent structures. In *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, pages 18:1–18:18, 2011.

[4] M. Ament, G. Knittel, D. Weiskopf, and W. Strasser. A parallel preconditioned conjugate gradient solver for the Poisson problem on a multi-GPU platform. In *Proceedings of the Euromicro Conference on Parallel, Distributed, and Network-based Processing*, pages 583–592, 2010.

[5] M. Ament and W. Strasser. Dynamic grid refinement for fluid simulations on parallel graphics architectures. In *Proceedings of the Eurographics Symposium on Parallel Graphics and Visualization*, pages 9–15, 2009.

[6] R. Netzel, M. Ament, M. Burch, and D. Weiskopf. Spectral analysis of higher-order and BFECC texture advection. In *Proceedings of the Workshop on Vision, Modeling, and Visualization*, pages 87–94, 2012.