Performance auto-tuning of a fluid simulation on heterogeneous devices

Olafs Vandans, s1139243

Goal

- A fluid simulation system that adjusts its parameters dynamically to achieve real-time performance
- This will require auto-tuning according to the underlying computing hardware (CPU, GPU, FPGA...)
- Using OpenCL for computing, OpenGL for rendering
- Adjustable parameters are the resolution, time-step, anything else that will come with understanding the fluid simulation code better

Fluid simulation

Used in computer games, animation, visual effects, scientific computing and the industry

- Useful for liquids as well as gases, smoke, fire
- Recently: fluid dynamics could explain some quantum phenomena

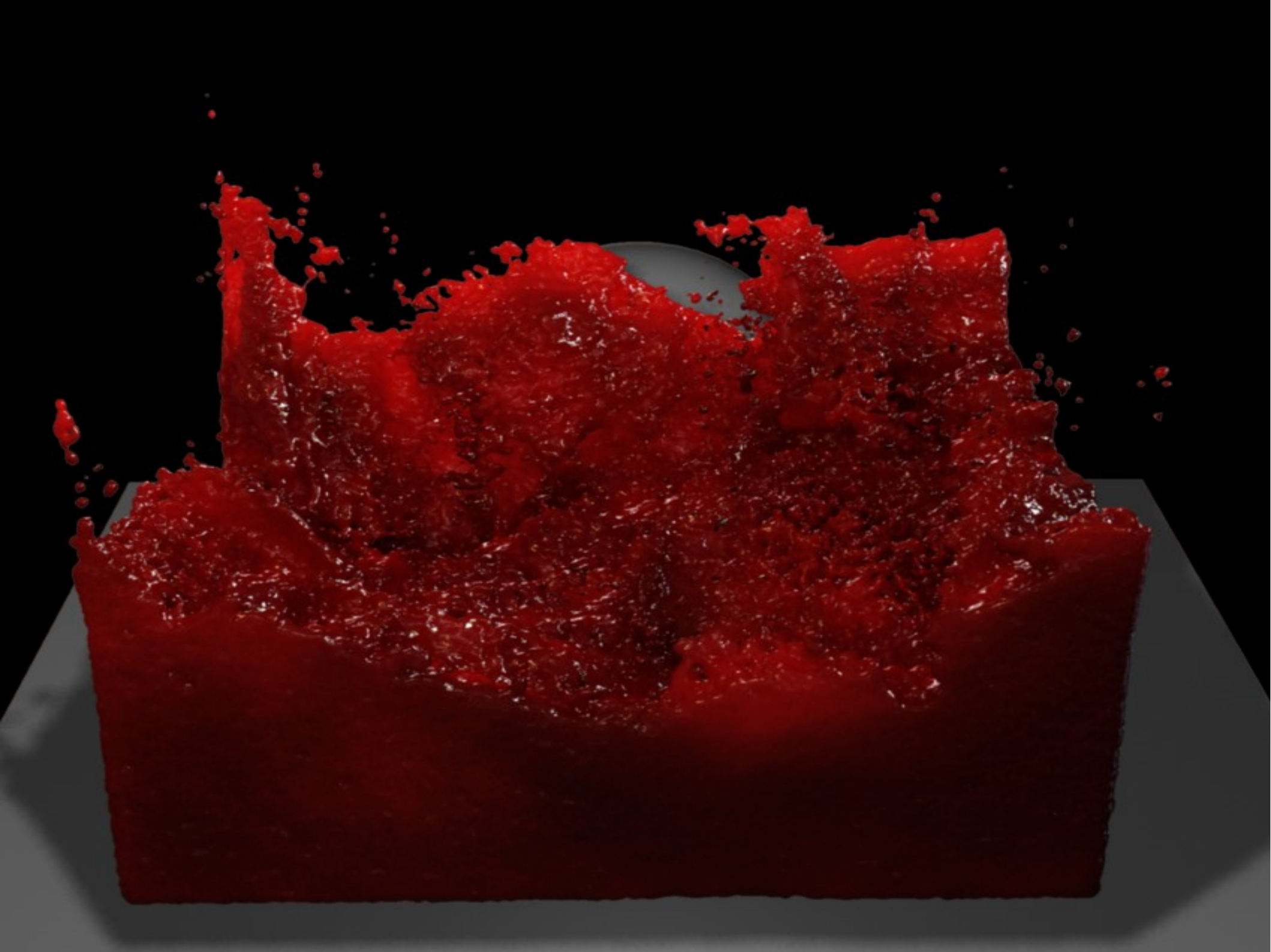
  http://www.wired.com/2014/06/the-new-quantum-reality/
- I will use the stable, mass-conserving method by Jos Stam

Fluid simulation

- Modelled on a vector field in 2D or 3D
- Highly parallelizable
- Dimensions → nested loops
- Solving linear equations – iterative (good for tuning)
- Many data accesses

Code example

```
for ( k=0 ; k<20 ; k++ ) {
	for ( i=1 ; i<=N ; i++ ) {
		for ( j=1 ; j<=N ; j++ ) {
			for ( l=1 ; l<=N ; l++ )
			{
				x[IX(i,j,l)] =
				(x0[IX(i,j,l)] + a*(x[IX(i-1,j,l)]+x[IX(i+1,j,l)]+
						x[IX(i,j-1,l)]+x[IX(i,j+1,l)]+
						x[IX(i,j,l-1)]+x[IX(i,j,l+1)]))/(1+6*a);
			}
		}
	}
	set_bnd ( N, b, x );
}
```



Current progress

- Basic environment (running OpenCL)
- 2D implementation ready
- 3D implementation 90% ready (bugs)
- Started to identify parallelizable parts

Roadmap

- This weekend: fully working 3D implementation
- Next week: tests suggested by Christophe
- End of October: running on OpenCL to some degree
- November: intensive parallelization and optimization
- December: Auto-tuning component, testing
- January – March: Report, optimizing the parallelization, nice demos

Testing on hardware

- Benchmarking
- Ideally: GPU computer, mobile device, HPC
- Testing just the parallelization
- Testing with auto-tuning