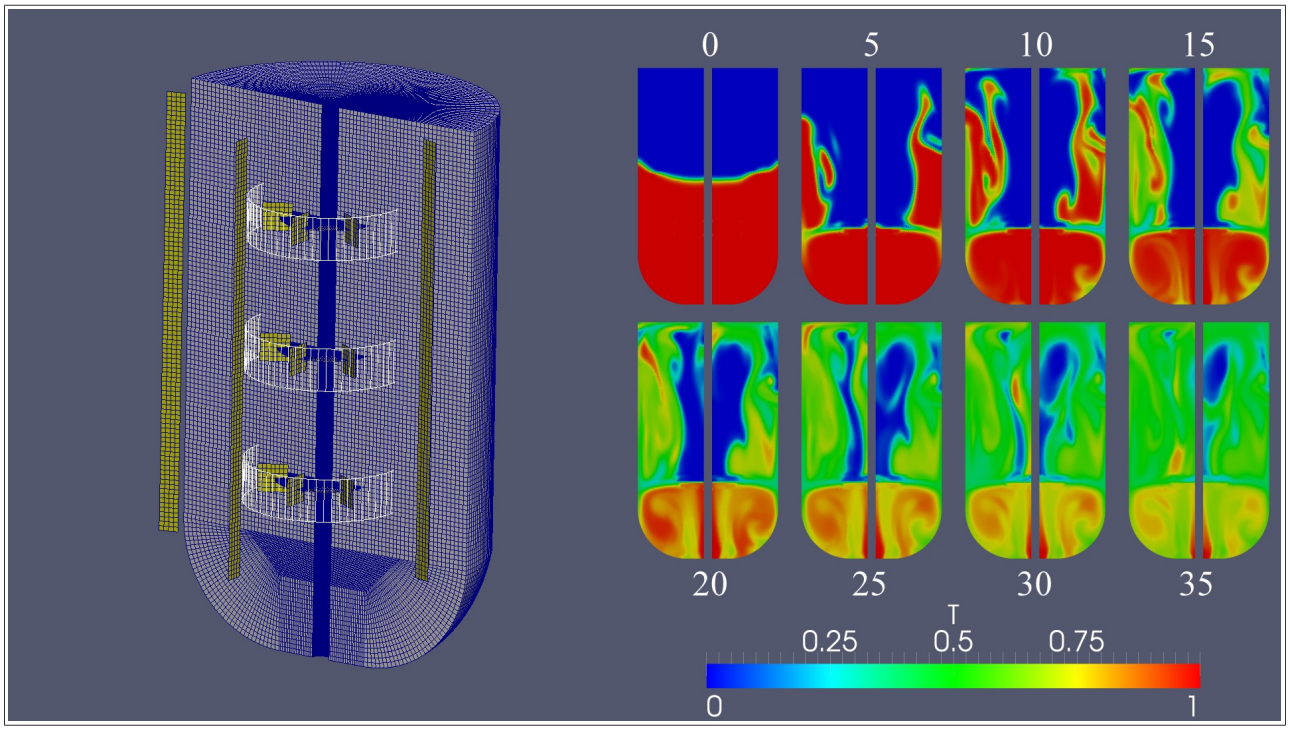
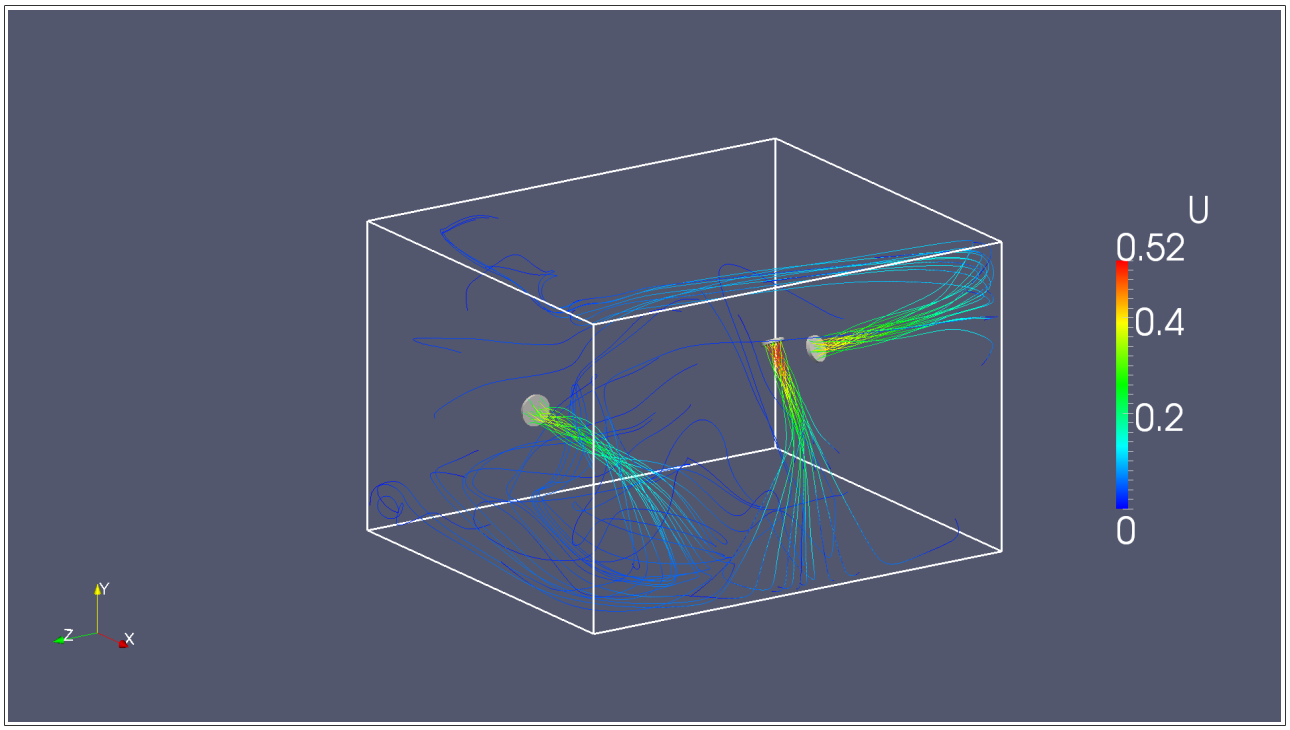
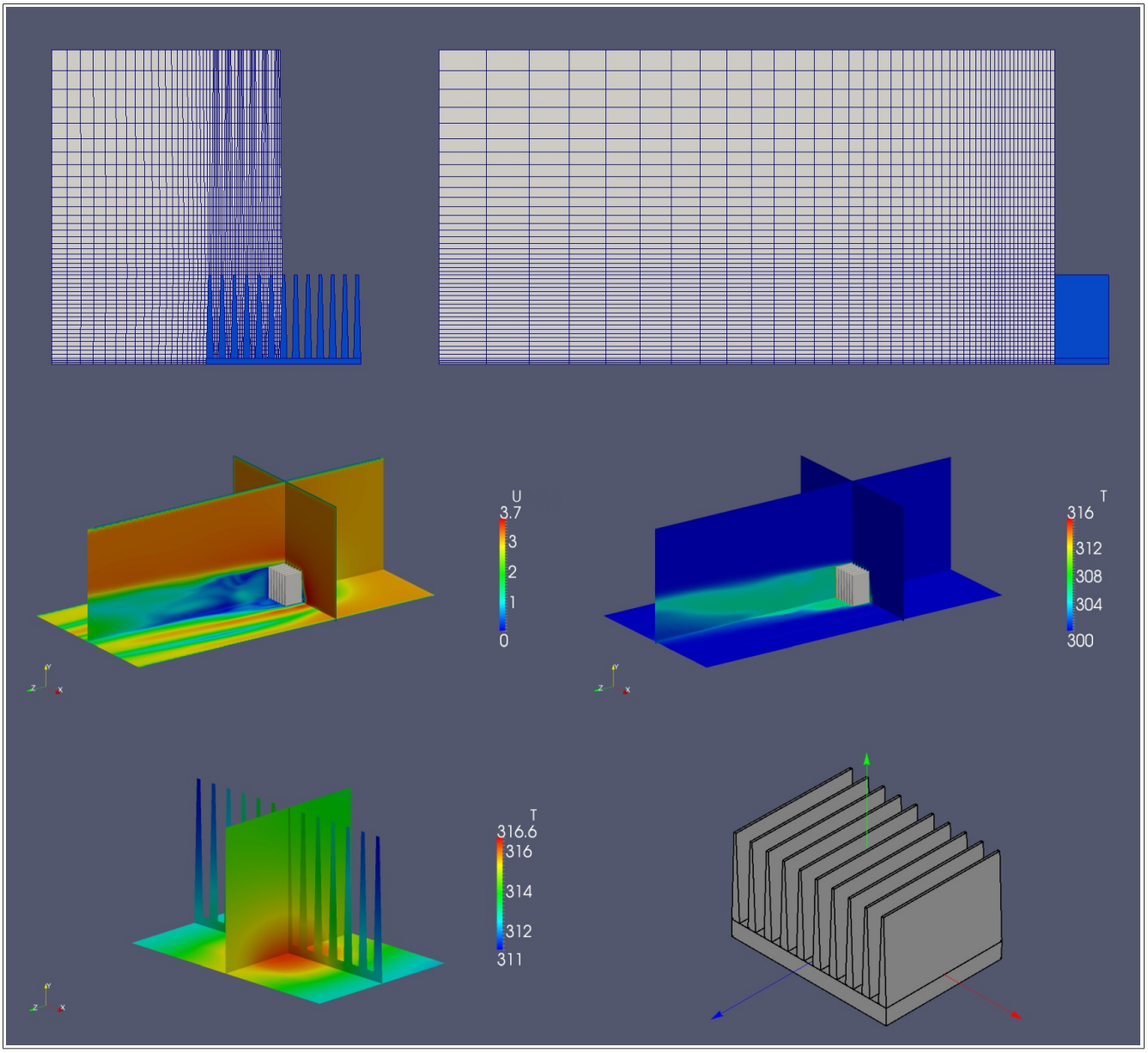
Computational Fluid Dynamics (OpenFOAM)



stirred tank: MRF simulation



mixer: volumetric impulse sources



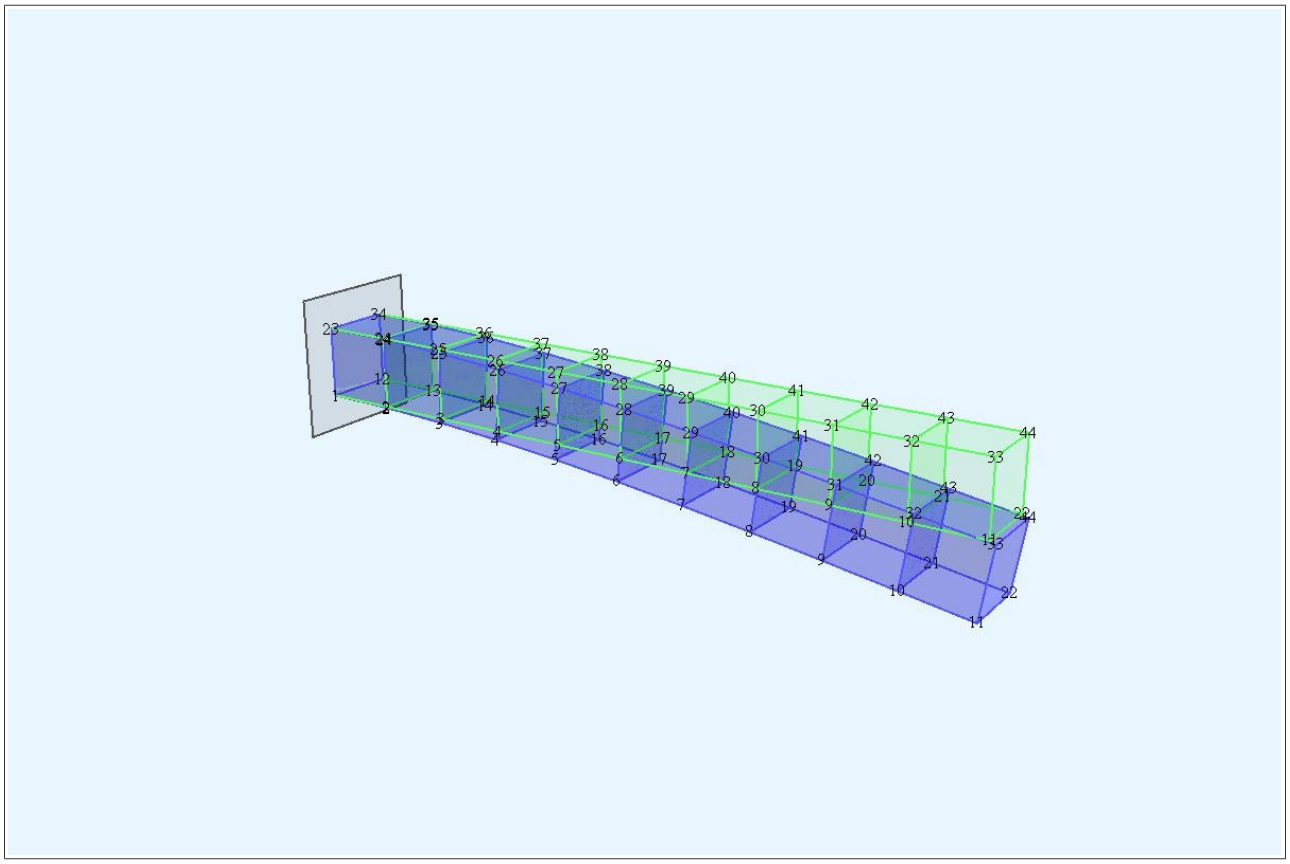
heatsink: solid – fluid heat transfer

Description: automatic CFD simulations in the OpenFOAM framework.

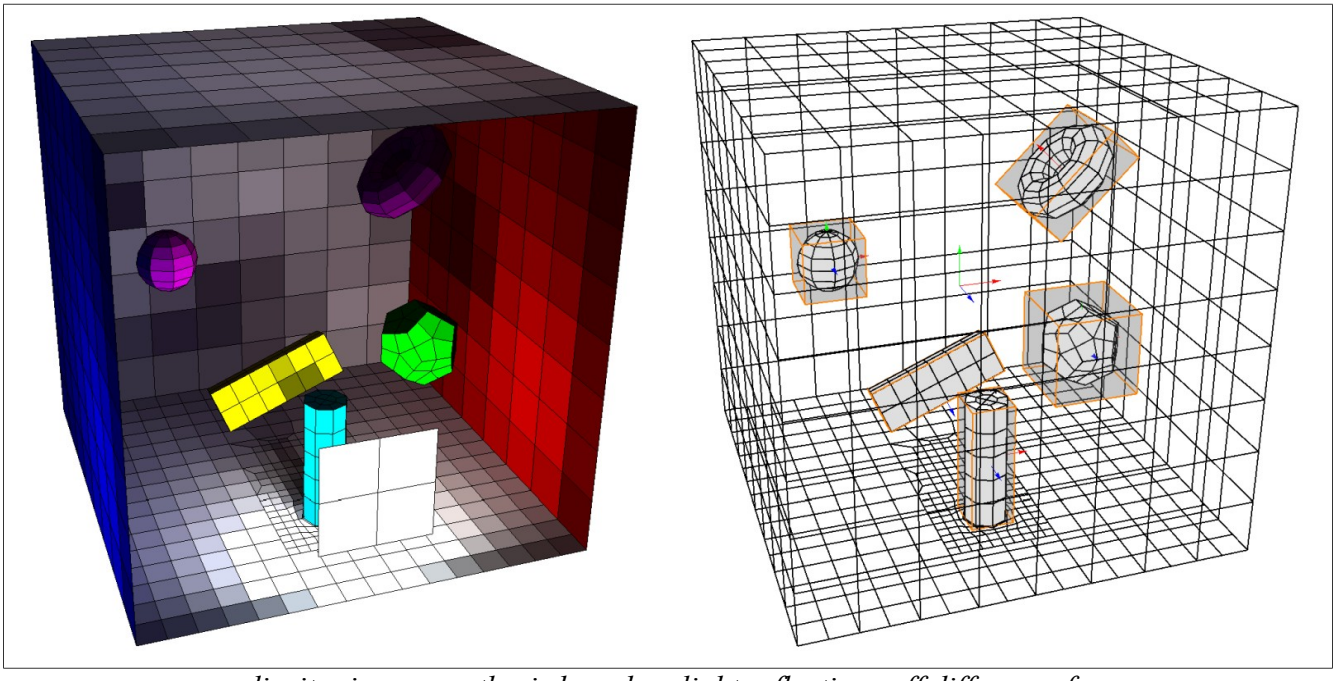
Features:

1) the cases don't need any assistance from the user but for the initialization (flow parameters, geometry), all the tools are open-source
2) meshing – adaptive size and grading (bash and Python)
3) multi-step solution – steady-state running, coarse to fine mesh interpolation, transient running
4) bash based case setup and running

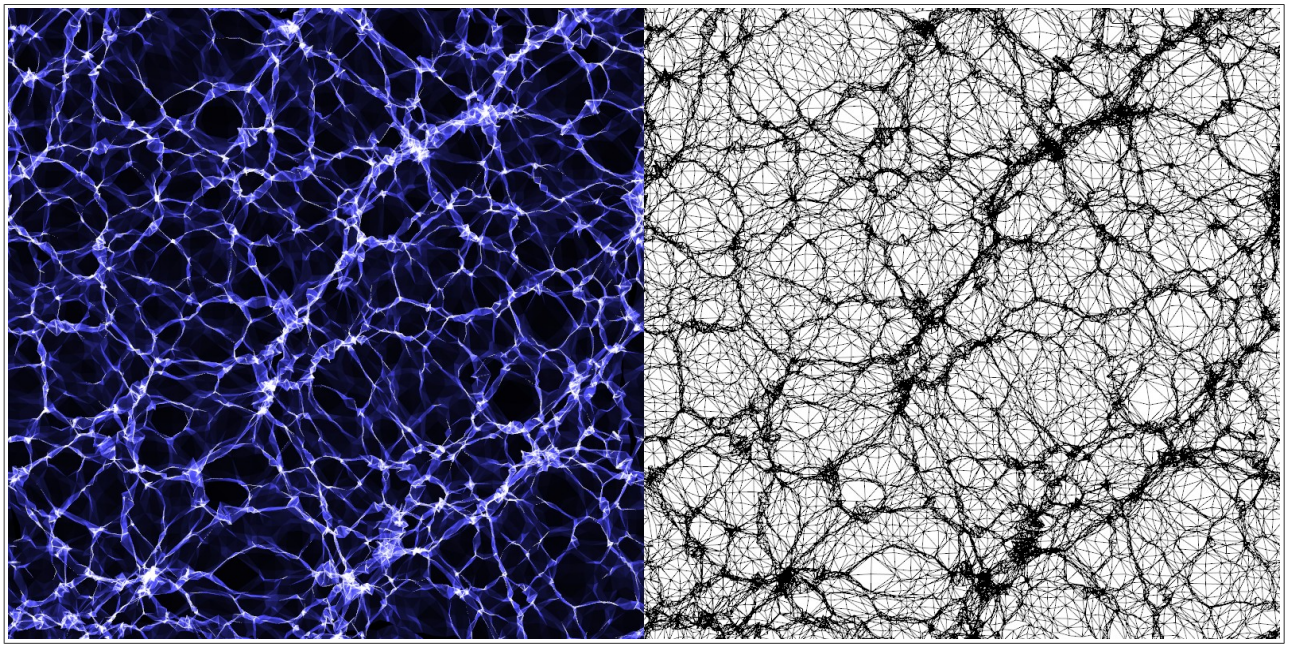
Miscellaneous projects (Wolfram Mathematica)



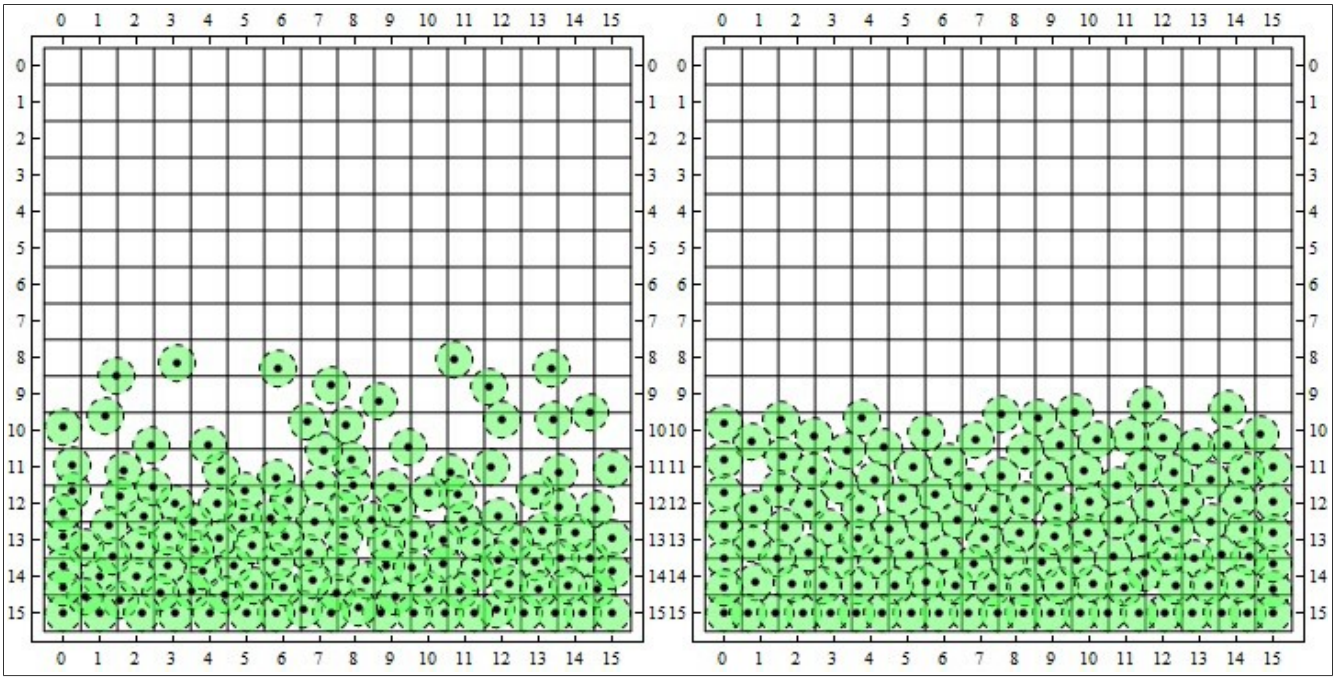
FEM: beam bending



radiosity: image synthesis based on light reflections off diffuse surface

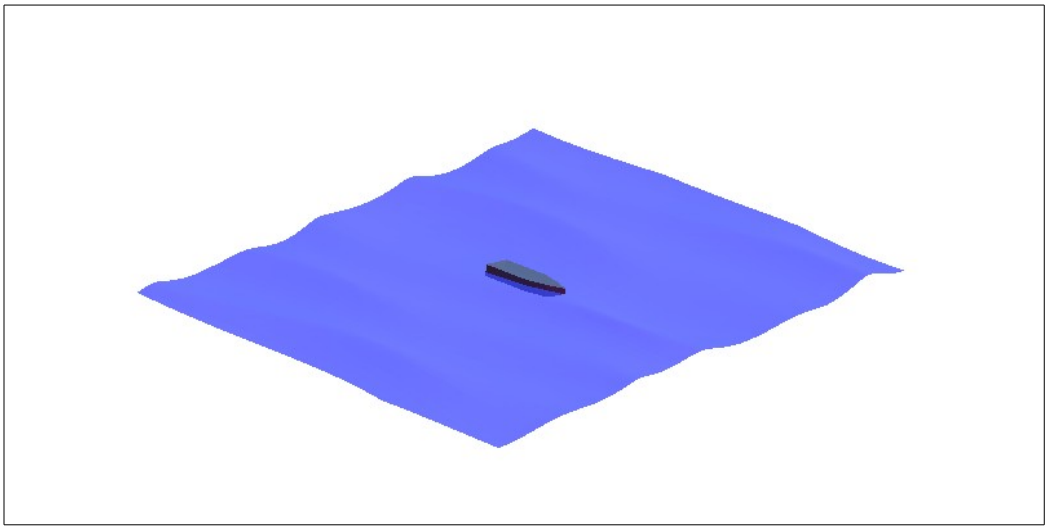


periodic caustics pattern in space and time



2D DEM based on the CUDA SDK "Particle" sample

Buoyancy Force Calculation

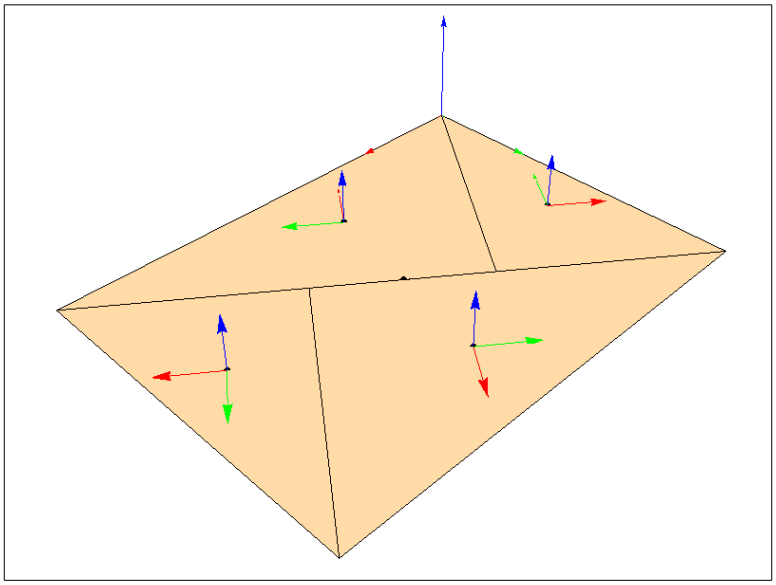


Description: simple boat simulation

Algorithm Overview: calculating hydrostatic forces on immersed bodies represented by a triangular mesh

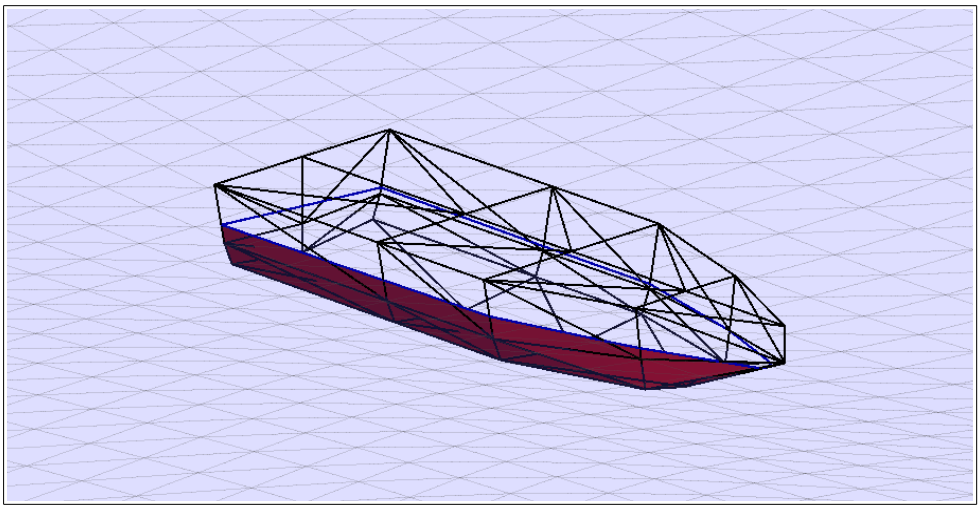
Steps:

1) moment of inertia (rotational inertia) calculation for triangle mesh (hull) – triangles are approximated as plates with infinitely small thickness
   only calculated once, can be calculated before the whole simulation
   - divide every triangle into 2 right-angled triangle
   - with the help of parallel axis theorem (inertia tensor) sum the contribution of the right-angled triangles to the moment of inertia at the center of mass

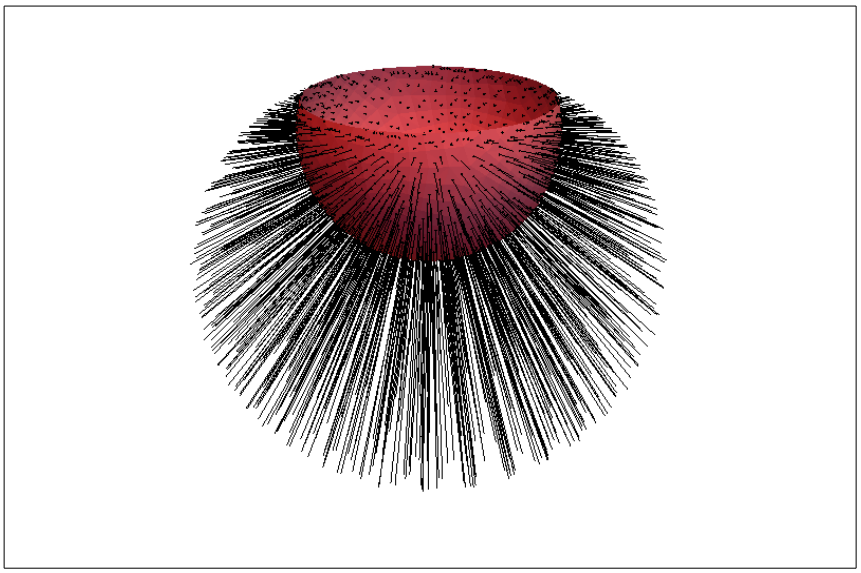


moment of inertia validation – rectangle assembled from 4 right-angled triangles

2) separate and tessellate boat hull according to water height for buoyancy force calculation – divide triangles that are only partially submerged into fully submerged and non-submerged triangles



3) calculate the point of application and magnitude of the hydrostatic force for every triangle resulted in step 2 and accumulate it at the center of mass of the boat



pressure distribution on a partially submerged sphere

4) use the calculated force to update the position of the boat – render image
5) update water height, go back to step 2

References:
Water interaction model for boats in video games by Jacques Kerner

2D Fluid Simulation

Description: 2D “stable fluid” simulation

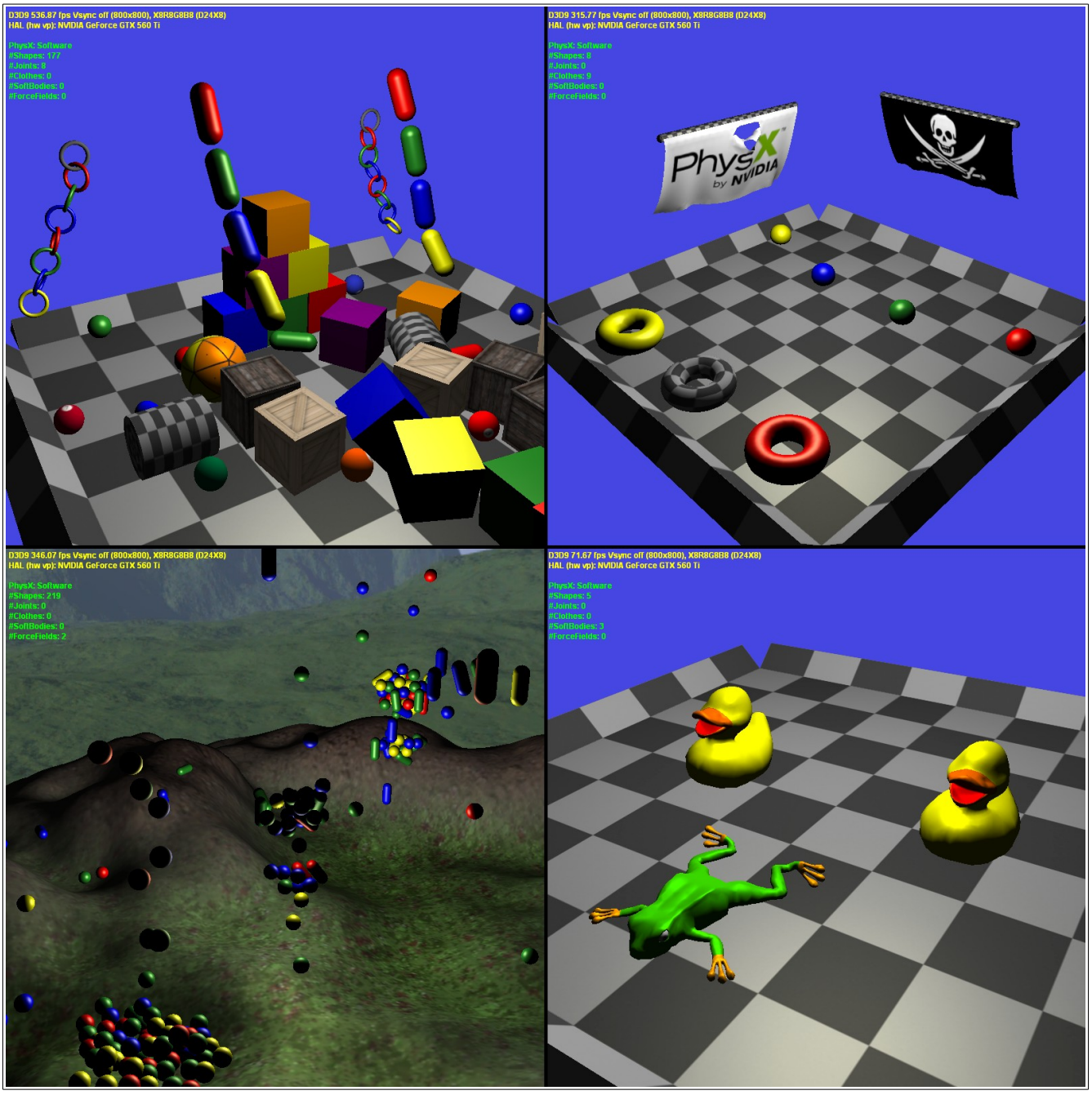
Features:

1) implementation of the renowned algorithm of Jos Stam
2) obstacles
3) vorticity confinement
4) monotonic cubic interpolation, MacCormack advection

Reference:

Jos Stam – Stable Fluids
GPU Gems – Fast Fluid Dynamics Simulation on the GPU
GPU Gems 3 – Real-Time Simulation and Rendering of 3D Fluids

NVIDIA PhysX 2.x Engine



Description: game like physics sandbox application, with data driven "levels" featuring the core components of the PhysX 2.X SDK

Features:

1) simple OO entity hierarchy
2) xml based scene description (TinyXML loader) – meshes, textures, materials, actors, PhysX descriptors, scenes
3) PhysX features – rigidbody, cloth + tearing, pressure cloth, softbody, joints, terrain, forcefields