Computer Aided Product Development
Modeling, Simulation and Rendering in the Industry

Markus Ihmsen
www.fifty2.eu / ihmsen@fifty2.eu

Follow us on

LinkedIn  YouTube
Product development - an iterative process
Question 1 – 2 Years Insights
// Problem

• Without simulation:
  • physical prototype required
  • complicated, expensive and unflexible tests

• Simulation tools:
  • require specialized knowledge by experts in the relevant simulation field
  • are slow
  -> complicated and time-consuming workflows
Traditional simulation methods

Workflow wading simulation (grid-based):

- 3 weeks meshing (manual work)
- 4 weeks simulation
- 1 week data analysis
Traditional simulation methods

Workflow wading simulation (grid-based):

- 3 weeks meshing (manual work)
- 4 weeks simulation
- 1 week data analysis
Product development - an iterative process

A waste of time
Purpose of FIFTY2

• Enable engineers and designers to craft the best version of their product together
  • Non simulation experts can describe and simulate the problem
  • Easy workflow and fast simulation
  • Results can be monitored and understood by engineers, designers, and decision makers at any time
Promise

- We build tools that:
  - Reduce time from question to insight (Q2I)
  - Maximize insights: quality and quantity of information
  - Are enjoyable and easy to use
Workflow wading simulation (PreonLab):

- Questions: 2 hours setup
- Days: 1 day simulation
- Insights: 3 hours analysis
// History

- 2007: Research
- 2013: New technology
- 2014: Technical partner
- 2015: Funding
- 2016: Foundation
- 2017: PreonLab 1.0
- 2018: PreonLab 2.0
- 2018: Sales partner
- 2018: PreonLab 3.0

AVL
// Watermanagement

Wading

Rain

Corrosion
// Raining
Rapid prototyping:
• Shorter cycles for design testing
• Allows progressive designs

• Duct 2 is a failure
• Continue with duct 1
// Oil

Gear box  Crank Case  Sloshing
Simulation der Ölverteilung

Simulationsergebnisse wurden von Versuchen bestätigt.
Particles carry physical quantities, e.g., temperature
Markets
„The future in hydraulic engineering belongs to 3D simulations. Due to its speed and usability, PreonLab has highest chances to be the standard solution in this market.“ (S. Corbe, CEO, TK Consult AG)

End customer:
- Governments
- Assurance companies
- City administrations
- Construction companies
Technical and theoretical background
Specify rubber material for the tires.
// PreonLab ingredients

- Physics: 3%
- Rendering: 7%
- Data structures: 11%
- Numerics: 7%
- Parallelization: 11%
- Data management: 18%
- GUI: 18%
- Testing: 14%
- Software design: 11%
For any point $x$ in fluid volume and time $t$ compute position $x(t)$. Same for any physical quantity $A(x,t)$

Requires:
- velocity $v$ at any point
- velocity change $dv$
CFD Solver – Navier-Stokes Equation

\[ \frac{D\vec{v}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{v} + \vec{g} \]

- Pressure term (solver)
- Viscosity term (explicit)
- Body forces / gravity (explicit)

602 billion trillion molecules per 18ml
250 916 670 000 000 000 terabytes
Fluid simulation

Discretize volume and time

40 megabytes
Spatial Discretization

- **Euler**
  - **Grid cells represent partial volumes**
    - Complete domain has to be discretized
    - Spatial derivatives efficiently computed
    - Advection expensive to compute

- **Lagrange**
  - **Particles represent partial volumes**
    - Only liquid phase has to be discretized
    - Advection efficiently computed
    - Spatial derivatives expensive to compute
// Spatial Discretization

• Euler
  • Grid cells represent partial volumes
    ➔ Align fluid cells with CAD model

• Lagrange
  • Particles represent partial volumes
    ➔ Fluid can move arbitrarily and aligns naturally
Numerical building blocks of a CFD Solver

- Reference frame: Eulerian, Lagrangian
- Navier-Stokes equations: incompressible, compressible
- Spatial derivatives: FD, FVM, kernel-interpolation, FEM, SPH, MPS, FPM, MLS
- Force term: implicit, explicit
- Time-stepping: implicit, explicit
Implicit formulation – Discretizing time

\[ \frac{Dv_i}{Dt} = -\frac{\nabla p_i}{\rho_i,0} + \nu_i \nabla^2 v_i + \nabla \Phi_i + f_i^b \]

1. Explicit calculation of all terms except the pressure term

\[ v_i^* = v_i + \Delta t \left( \nu_i \nabla^2 v_i + \nabla \Phi_i + f_i^b \right) \]

2. Implicit computation of the pressure field

\[ \nabla^2 p_i = \frac{\rho_i,0 - \rho_i^*}{\Delta t^2} \quad \rho_i^* = \rho_i - \Delta t \rho_i,0 \nabla \cdot (v_i^*) \]

\[ v_{i,t+\Delta t} = v_i^* - \Delta t \frac{\nabla p_i}{\rho_i,0} \]

\[ x_{i,t+\Delta t} = x_i + \Delta t v_{i,t+\Delta t} \]
Kernel weighting – Discretizing space

- Value interpolation at position \( \mathbf{x} \)
- Over finite local support domain
- Weighting is color coded
- Riemann sum

\[
A(\mathbf{x}) = \int A(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}'
\]

\[
\langle A(\mathbf{x}) \rangle = \int_{\Omega} A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'
\]

\[
\langle A(x_i) \rangle = \sum_j V_j A(x_j) W(x_i - x_j, h)
\]
// Simulation loop

1. Find neighbors
2. Compute explicit forces
3. Solve for implicit forces
4. Update velocities, positions and other physical quantities of partial volume (particles)
Performance – Neighbor search

- Typically 30 to 40 particles in influence domain
- 1 million particles: 20ms

- Reduce number of queries
  - Compact hashing
  - Z-index sorting
  - kd-trees
Performance – System of equations

- Pressure term
  - Enforce volume conservation of fluid
  - Solve for unknown pressure field
  - Pressure gradient results in forces
- Viscosity term
  - Minimize strain-rate tensor
  - Changes the velocity gradient

\[ D = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \]
Performance – System of equations

- Simulation with 100 million particles
- Linear system of size 100M x 100M
  - Conjugate gradient
  - Relaxed Jacobi
  - Gauss-Seidel
- System is sparsely filled
  - Matrix-free and compact implementation

\[ Ax = b \]
Performance - Parallelization

- Expectable speed-up obeys Amdahl’s law
// Performance - Parallelization

- Expectable speed-up obeys Amdahl’s law
- Reduce latencies
  - SIMD: Optimized data structures
// Performance - Parallelization

- Expectable speed-up obeys Amdahl’s law
- Reduce latencies
  - Single Instruction Multiple Data
  - SMP: Increase cache coherency
// Performance - Parallelization

- Expectable speed-up obeys Amdahl’s law
- Reduce latencies
  - SMP: Increase cache coherency
  - SIMD: Optimized data structures
  - MMP: Load balancing
Performance – Hybrid parallel implementation

73500 liter
157 Million particles
40 sec
// Rendering

Simulation:
• Discrete representation

Visualization:
• Continuous volume
• Simulate absorption, reflection and transmission of light
// Classical approaches

- Create triangle mesh from particles
- Isosurface generation via
  - Marching cubes
  - Dual contouring
  - Adaptive dual marching cubes

- Mesh files can be used
  - By any raytracing program

- But
  - Time-consuming to construct
  - Consume a lot of memory
// Preon renderer

• Renders fluid as smooth surface without generating an explicit mesh
• CPU-based ray tracer which works on systems without graphic cards / cluster
• Fast generation by using neighbor search data structure
• Implicit foam rendering
• Output is image per frame
// Preon renderer

- Physically-based stochastic raytracer
- Photon mapping
- Fluid properties can be colorized on virtual surface and sensors can be visualized
What’s next?
Product development

Still a waste of time!

Include more physics:
• Improve quality
• New applications

Optimizations:
• Performance
• Workflow
Particle-based rigid body solver
slow motion: 5x
Highly viscous fluids