Interactive flow simulation with Common Lisp

Nicolas Neuss

Angewandte Mathematik
Friedrich-Alexander-Universität Erlangen-Nürnberg

Differential equations

- Differential equations appear when a phenomenon is determined by local interactions in a continuum

- Ordinary differential equations (ODEs) have the form

$$\frac{du}{dt}(t) = f(t, u(t)), \quad t \in]0, T[\,.$$

and (mostly) describe phenomena depending only on time.

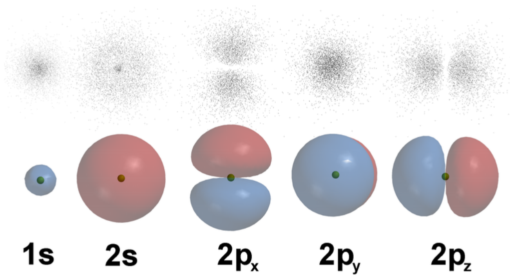
- Partial differential equations (PDEs) have the form

$$\frac{\partial u}{\partial t}(t) = f(t, x, u(t,x), \frac{\partial u}{\partial x}(t,x)), \quad t \in]0, T[,\ x \in \Omega$$

and describe phenomena depending on space and time.

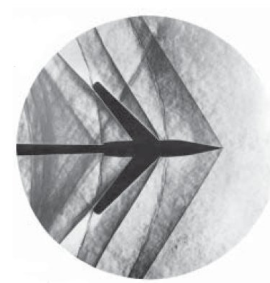
Applications

PDEs are ubiquitous on all scales



Schrödinger

. . .



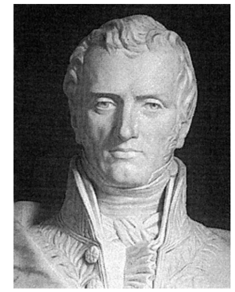
Navier-Stokes

. . .

Einstein

The Navier-Stokes equation





- The work of these people gave us the Navier-Stokes equation

$$
\begin{aligned}
\rho\frac{\partial \vec{v}}{\partial t} + \rho(\vec{v}\cdot\nabla)\vec{v} + \nabla p &= \vec{f} + \mu\Delta\vec{v} + (\lambda+\mu)\nabla(\nabla\cdot\vec{v}) \\
&= \vec{f} + \mu\Delta\vec{v}
\end{aligned}
$$

- The simplification in the second line is valid for incompressible flow with

$$\nabla\cdot\vec{v} = 0$$

- The parameter μ describes the (kinematic) viscosity.

Mathematics of the NSE

- In $d = 2$ space dimensions a unique solution exists for all times.
- In $d = 3$ space dimensions this is known only for simple flows with

$$\text{Reynolds} = \frac{U \cdot D}{\mu} \quad \text{small } (\lesssim 100)$$

U: characteristic velocity, D: characteristic length, μ: viscosity

- Apropos: Proving long-time existence and uniqueness of NSE solutions in the case $d = 3$ and large Reynolds numbers would win you 1,000,000$!

Bad news

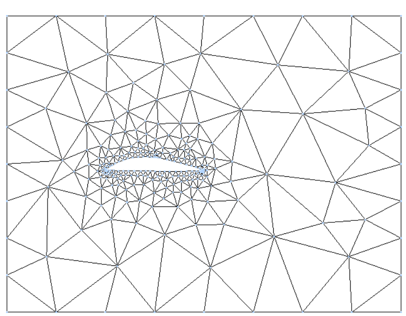
- For practically relevant flow problems Reynolds $= 10^5 \ldots 10^8$
- "Turbulent flow": Vortices of different sizes down to a scale of Reynolds$^{-\frac{3}{4}}$
- $\rightsquigarrow$ Currently active research on turbulence models

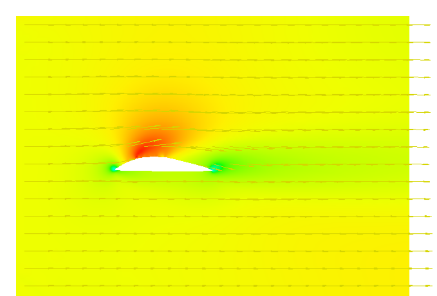
Interactive Simulation: Flow around an airfoil

- "Long Night of the Sciences 2017" at the FAU Erlangen-Nürnberg
- Interactive demo:
  - Visitors draw an airfoil
  - A flow is simulated around the airfoil
  - Lift and drag are computed
  - Result is ranked according to the ratio lift:drag in a high-score table
- Common Lisp inside
  - SBCL
  - Femlisp (uses lparallel and other libraries . . .)
  - Hunchentoot and cl-who

More detailed description

- `index.html` $\rightsquigarrow$ Paper.js (handles drawing of airfoil in browser)
- `calculate-drag-lift` (with parameters name/curve) $\rightsquigarrow$
  - Fit scaled curve in a rectangular channel, triangulate, discretize, solve
  - Calculate the force on the airfoil and the lift/drag
  - Write status report and images to data directory.
- `show-scores` $\rightsquigarrow$ Show score table and Top-10 list
- `show-result` $\rightsquigarrow$ Show results of a single calculation





Finally

Improvements

- Better physical model
- Polar curve (all angles of attack simultaneously)
- Faster/more accurate calculation
- Instantaneous feedback, computational steering?

References

- Jürg Lehni und Jonathan Puckey: Paper.js
- Florian Sonner: `index.html` (Adaption of Paper.js)
- Nicolas Neuss: Femlisp (`http://www.femlisp.org`)
- Wikipedia: Some pictures in this talk

Try it yourself...

- `http://131.188.56.128:8080/index.html`
- `http://131.188.56.128:8080/show-scores`