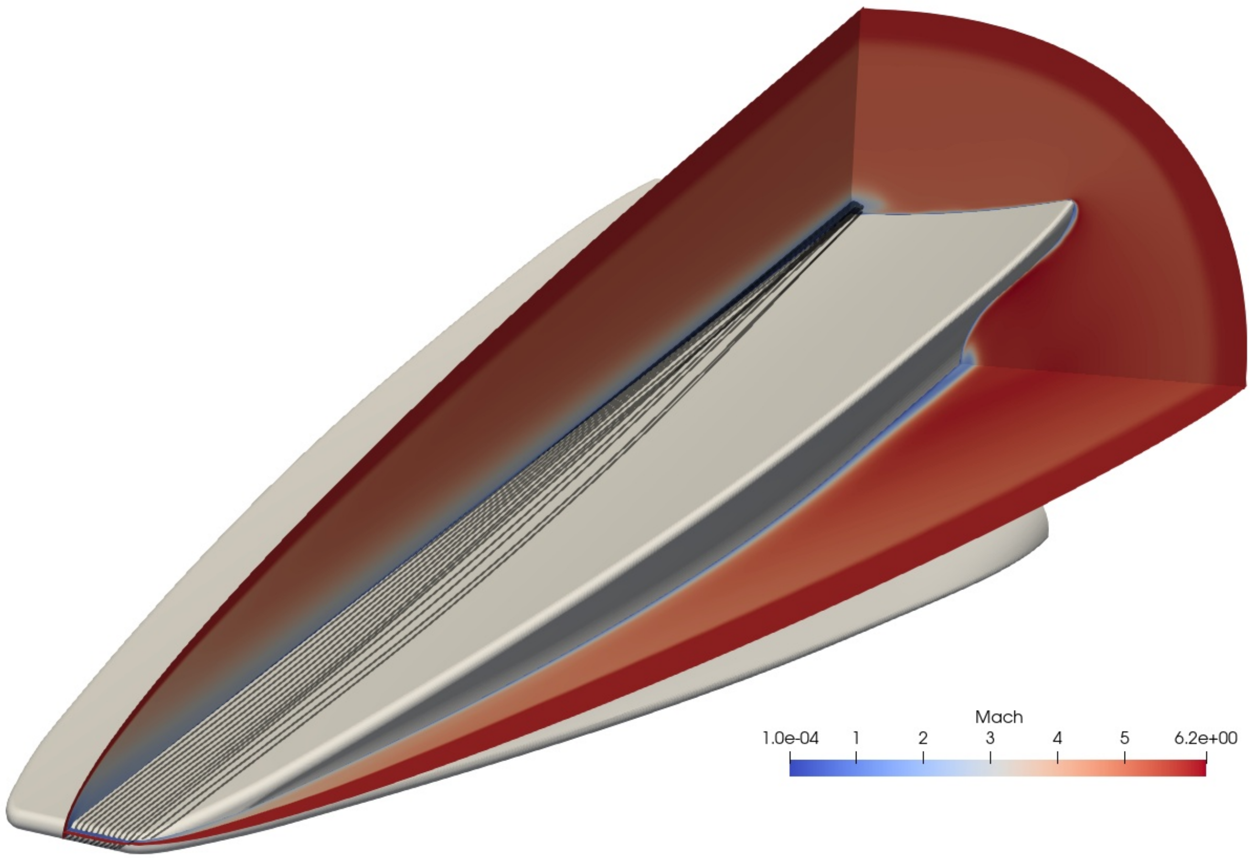
Introduction to Eilmer:
a multi-physics hypersonic flow solver



K. A. Damm

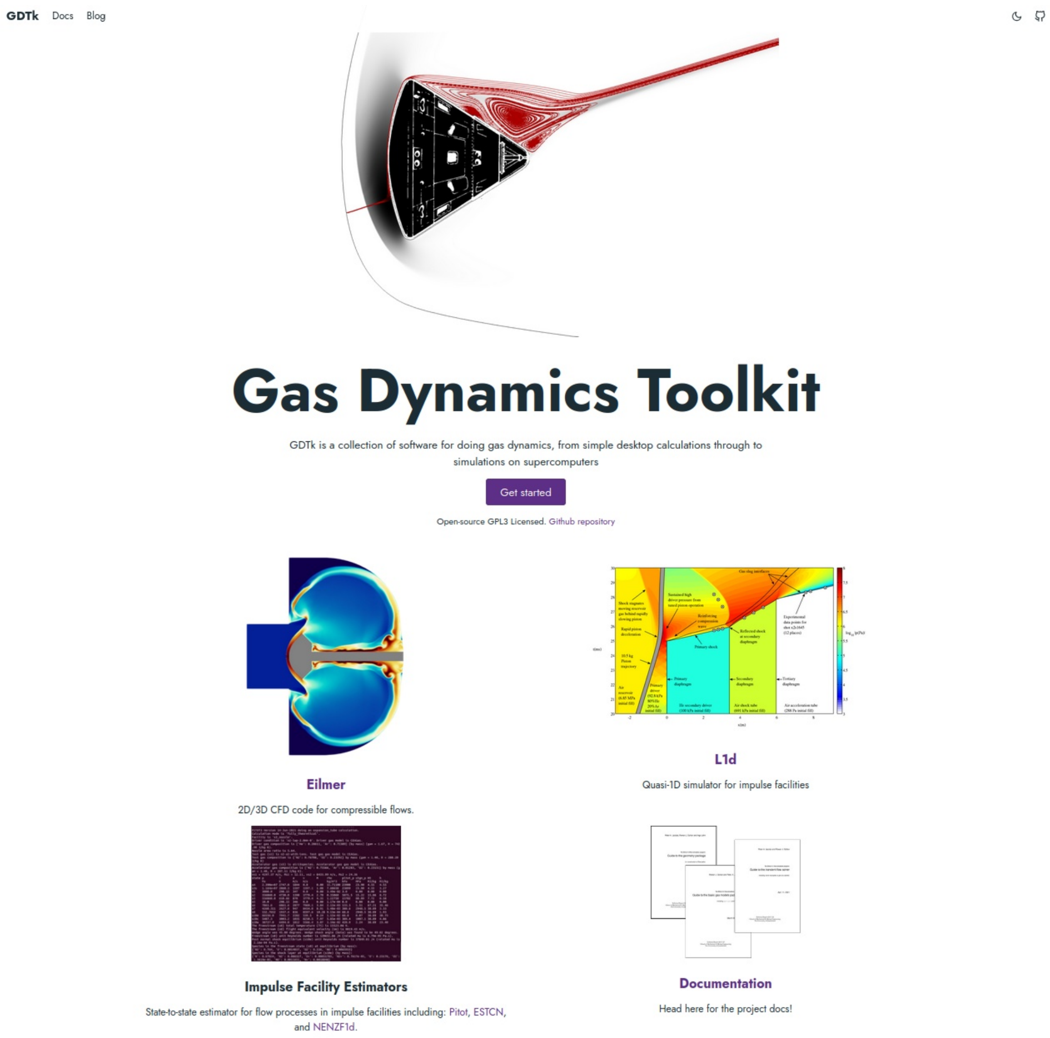
20th May 2022

- **Simulation tools covering a range of fidelities**
  - Open-source code hosted on GitHub
  - 30+ years of development, primarily at UQ
- **Development team:**
  - Peter Jacobs
  - Rowan Gollan
  - Kyle Damm
  - Nick Gibbons
  - and many grad student contributions
- **User base:**
  - local at University of Queensland
  - University of Southern Queensland
  - University of New South Wales
  - CalTech
  - Purdue
  - Oxford
- **User support:**
  - monthly meet-ups
  - issue tracker
  - email (point-to-point with developers)
- **Documentation:**
  - user guides
  - reference manuals
  - technical notes
  - catalogue of examples

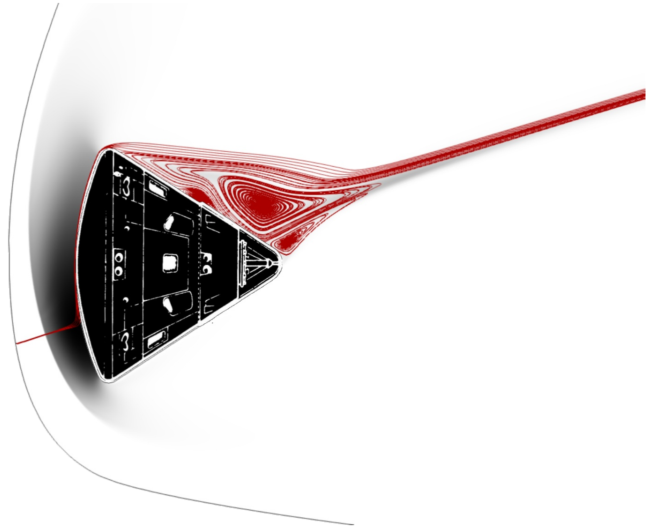
Single-Block Navier-Stokes Integrator *

P. A. Jacobs
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23665

July 1991 †

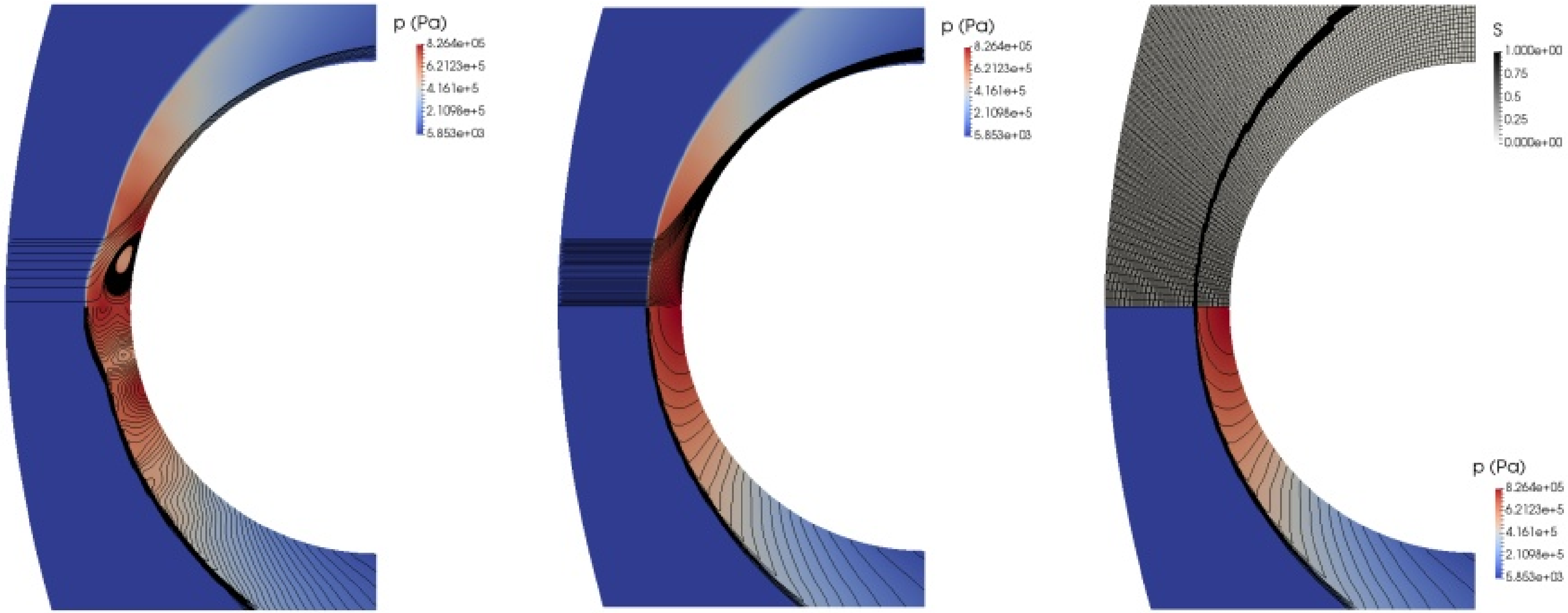


Eilmer: in-house CFD code



- **Core code written in D programming language**
  - Object-oriented, statically typed, with C-like syntax
- **2D/3D finite volume compressible flow solver**
- **Several gas models:**
  - ideal, thermally perfect, multi-temp., state-specific
- **Euler (inviscid), NS (laminar), RANS (turbulent)**
- **Finite-rate chemistry**
- **Solid domains with conjugate heat transfer**
- **Moving grid capability based on GCL:**
  - User-controlled
  - Shocking-fitting for blunt body flows
- **User-defined customisations:**
  - Boundary conditions and source terms
  - pre- and post-processing
- **Parallel computation using shared (D) or distributed memory (MPI)**

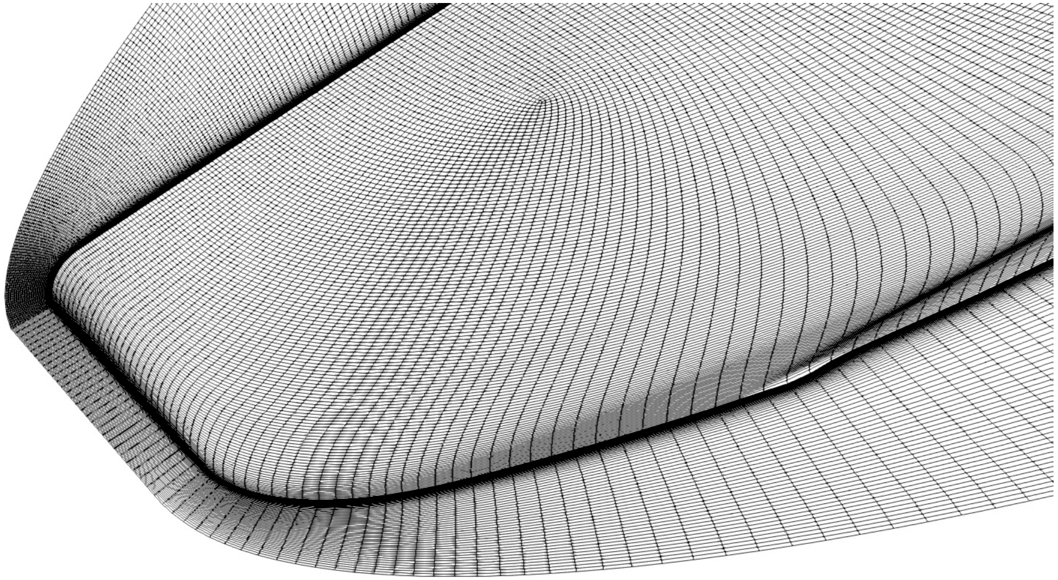
Eilmer: Flux evaluation methods



- **Inviscid fluxes:**
  - 1D flux calculators
  - High dissipation methods: ***Hanel, LDFSS0, HLLE, EFM***
  - Standard low dissipation methods: ***AUSMDV, LDFSS2, HLLC***
  - Special low dissipation methods: ***ASF***
  - Carbuncle fix via blending (shock detector: change in normalised velocity)

- **Viscous fluxes:**
  - Divergence theorem (2D) at either faces, vertices
  - Weighted least squares (2D/3D) at either faces, vertices, **cell-centers**

Eilmer: Structured grid solver

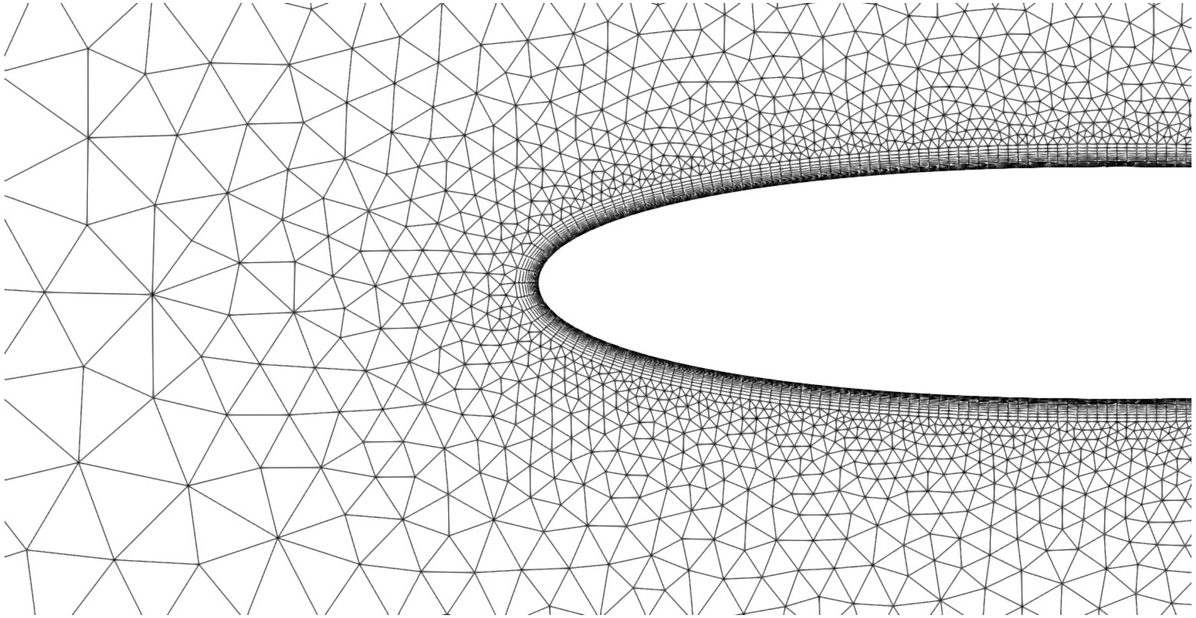


- **Quadrilateral (2D) and Hexahedral (3D) meshes**
  - Native grid generation
  - Import GridPro format
  - Import Plot3D format

- **Flow state reconstruction:**
  - Piecewise parabolic (3rd order accurate)
  - Face-based limiters: Modified Van Albada and Min-mod

- **Domain decomposition:**
  - Manual specification of split points
  - Load balancing pre-processor (balances load but not communication)

Eilmer: Unstructured grid solver



- **Quad., triangle (2D) and Hexahedral, Tetrahedral, Prism, Pyramid (3D) meshes**
  - Native quadrilateral/hexahedral grid generation in unstructured format
  - Import SU2 format

- **Flow state reconstruction:**
  - Least squares (2nd order accurate)
  - Cell-based limiters: Barth, Venkat (optional pressure-based augmentation), Park

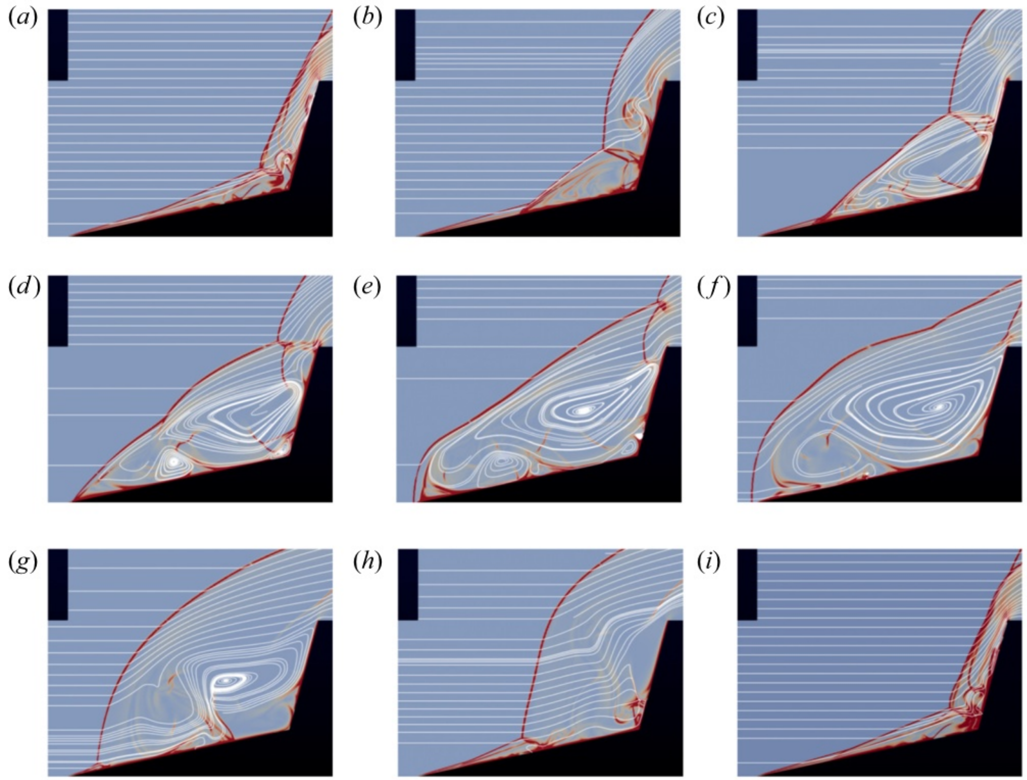
- **Domain decomposition:**
  - METIS graph partitioning software
  - Local Reverse Cuthill-McKee ordering

Eilmer: Time integration methods

- **Transient, time-accurate updates ysing Runge-Kutta family of integrators**:
  - Euler (first order)
  - Predictor-corrector (second order)
  - RK3 (third order)

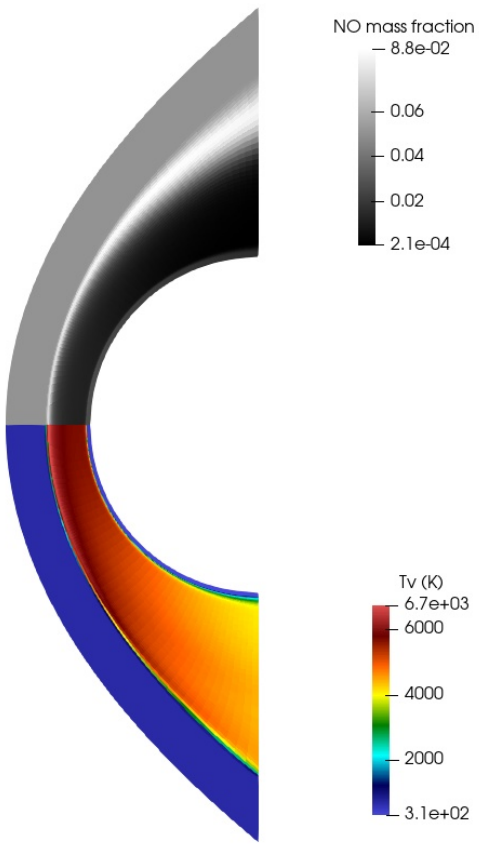
- **Point implicit time integration**
  - Backward Euler (first order)
  - Implicit RK1 (first order)

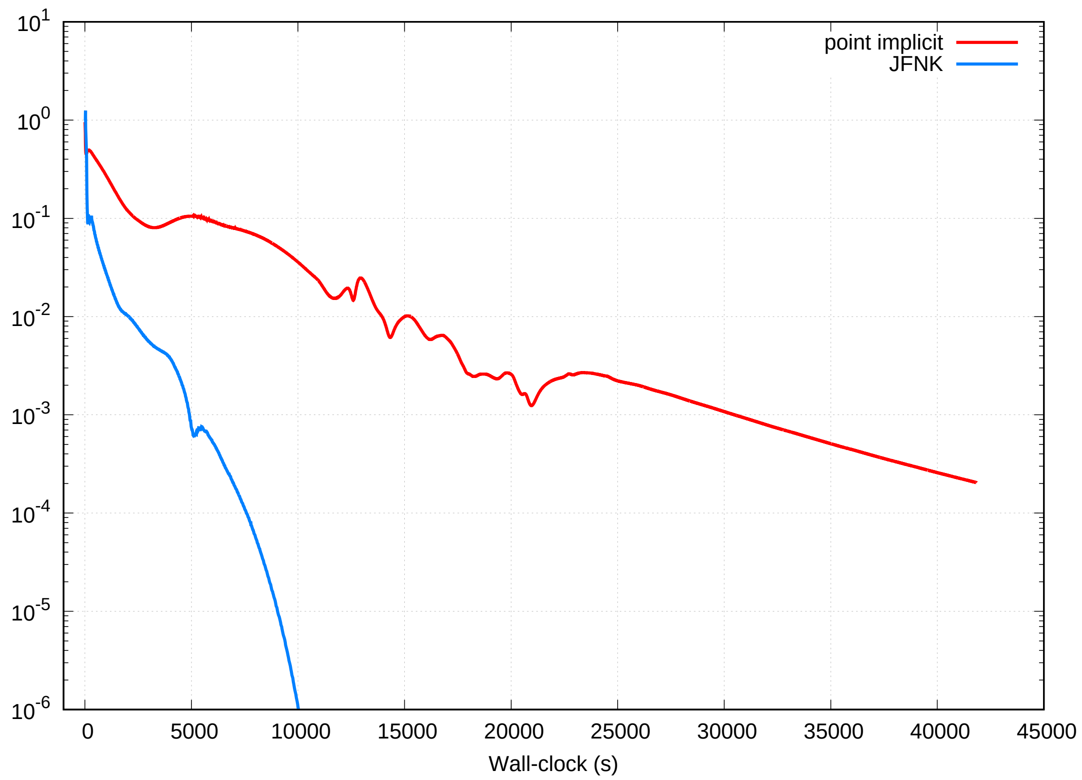
- **Super-Time-Stepping (experimental)**
  - Runge-Kutta-Legendre family of methods
  - RKL1 (first order)
  - RKL2 (second order)
  - Accelerates viscous dominated flows



Eilmer: Steady-state accelerator

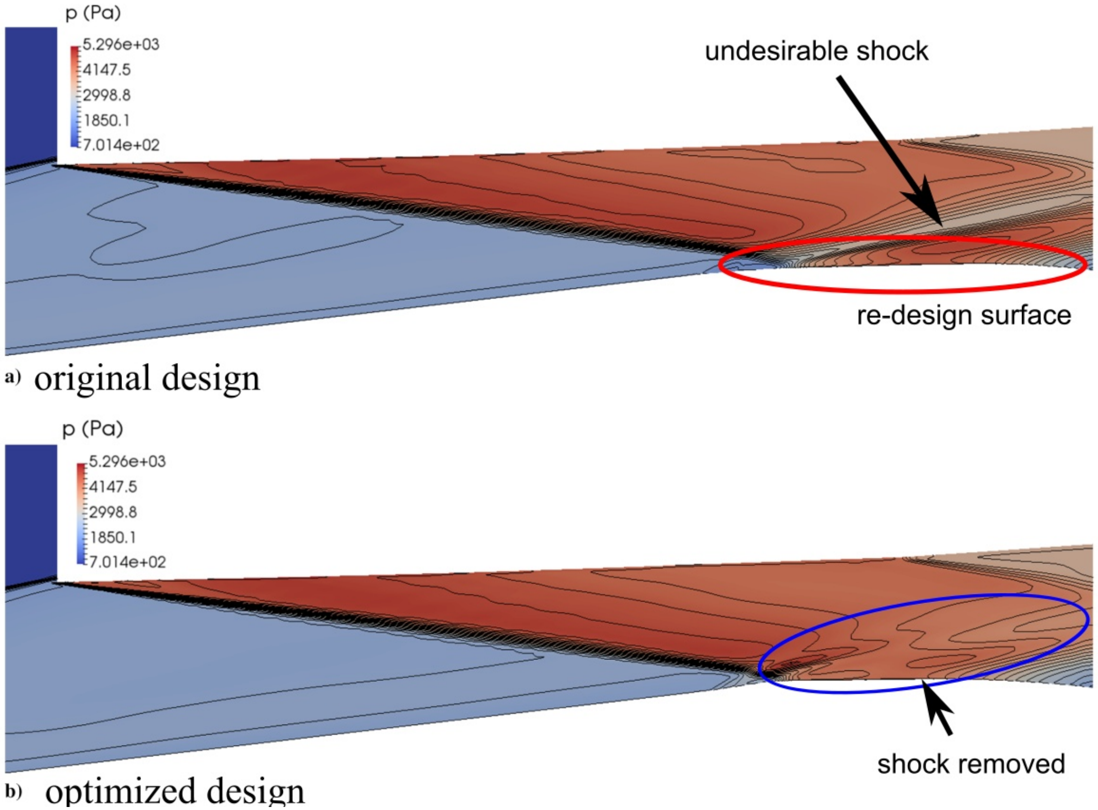
- **Jacobian-Free Newton-Krylov method for accelerating convergence**:
  - Frechet derivative to approximate matrix-vector products
  - Preconditioners: Block-Jacobi, SGS, SGS relaxation, ILU0, ILU(k)
  - Auto-differentation of flow solver routines via a complex-step method

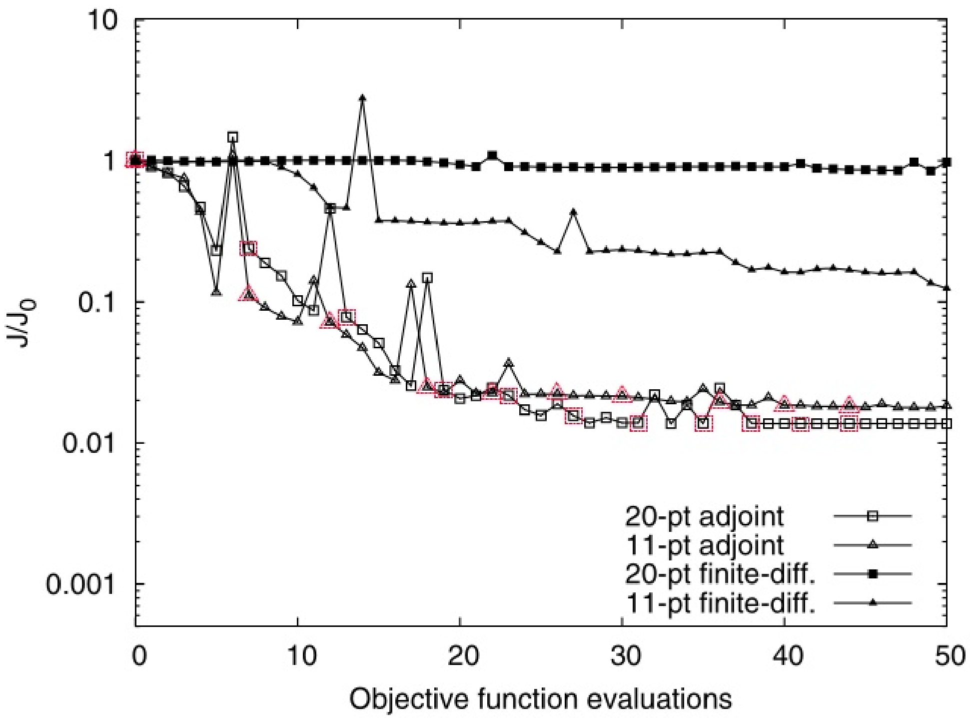




Advanced feature: Adjoint Optimization

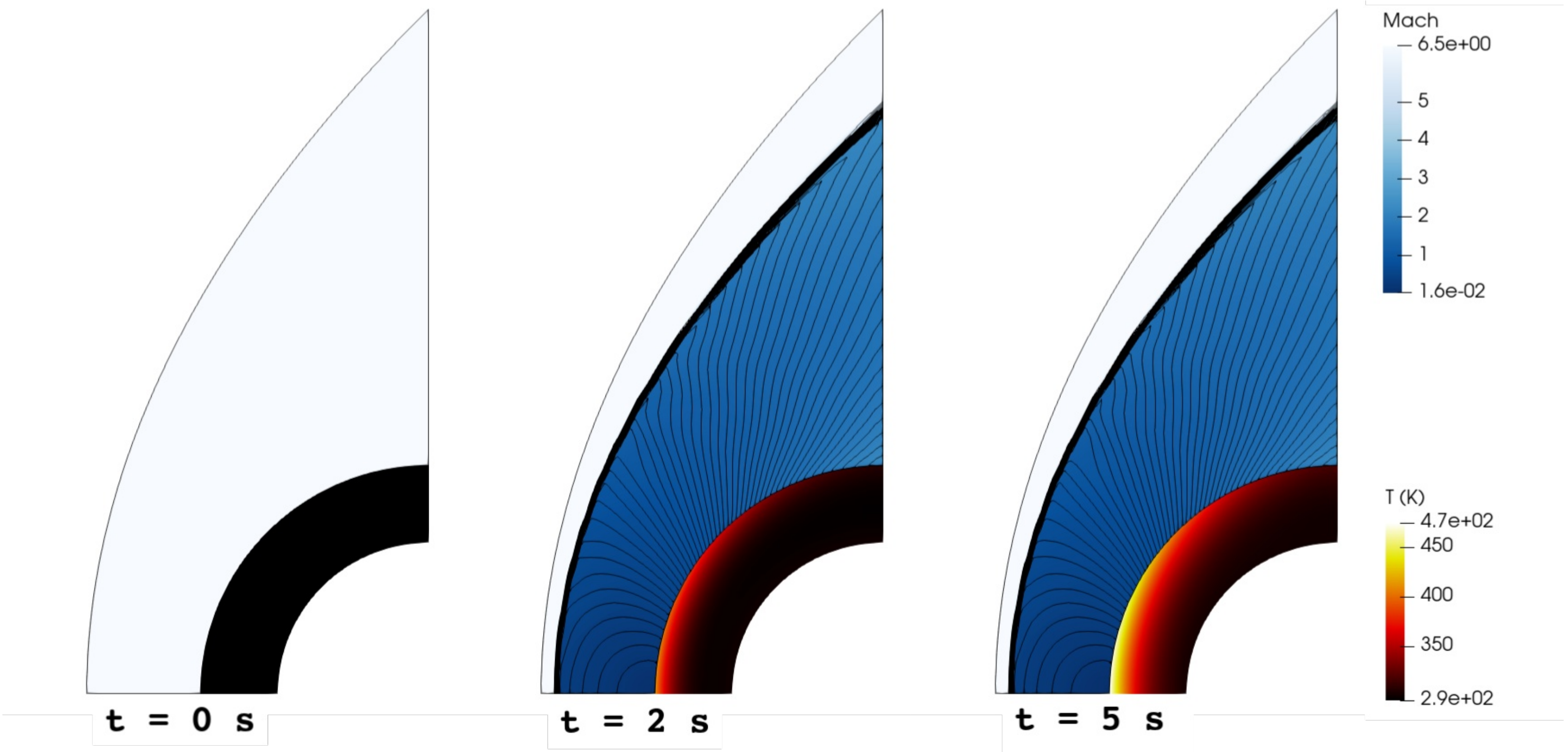
- **Efficient multi-parameter aerodynamic shape optimization**
  - In-built state-of-the-art adjoint solver
  - Sensivities evaluated via same complex-step method as flow solver





Advanced feature: CHT solver

- **Couple flow solver and solid solver**
  - Tight coupling using time-accurate integration schemes from flow solver
  - Loose coupling using JFNK flow solver and Super-Time-Stepping solid solver



Advanced feature: CHT solver

- **Couple flow solver and solid solver**
  - Tight coupling using time-accurate integration schemes from flow solver
  - Loose coupling using JFNK flow solver and Super-Time-Stepping solid solver