Computational Hypersonics Research at The University of Queensland

Rowan J. Gollan

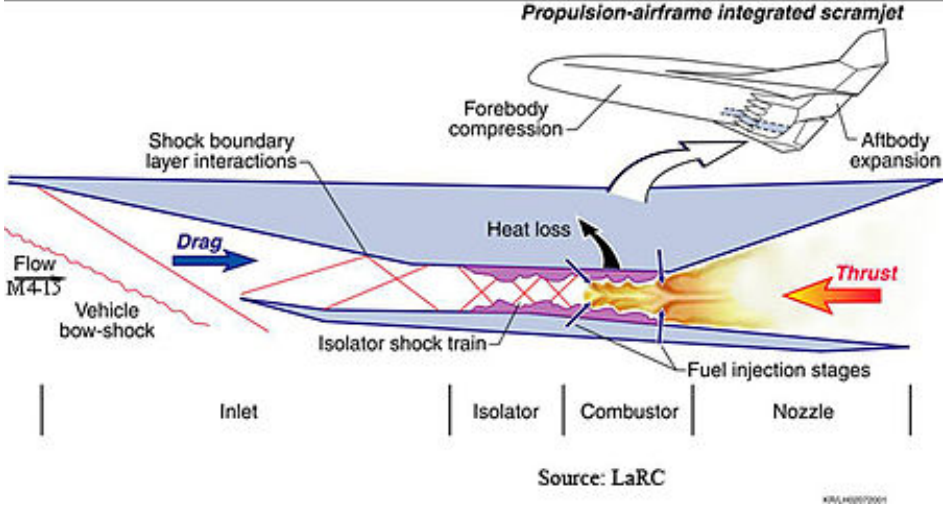
8 June 2018

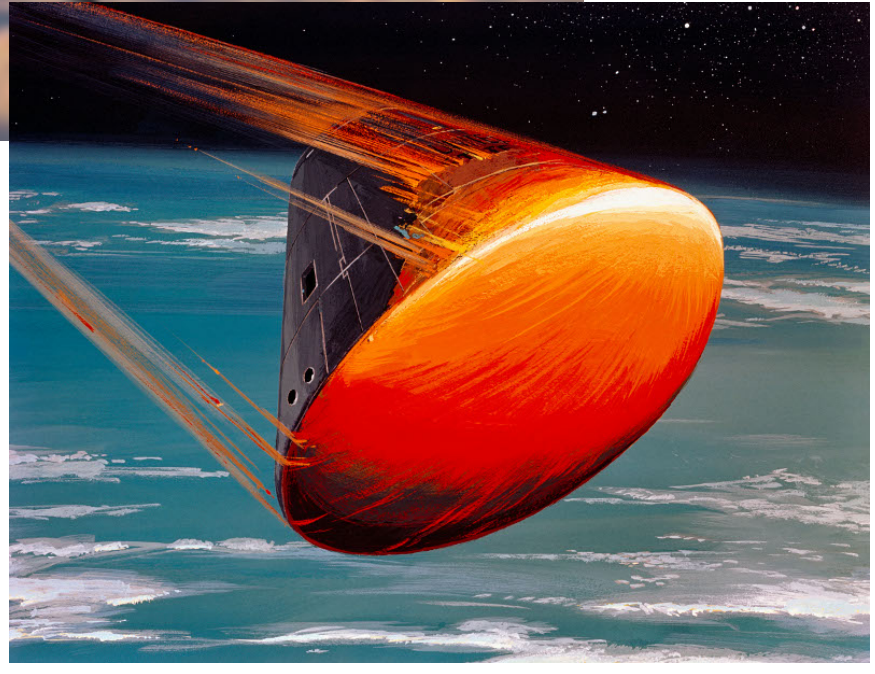
from The University of Queensland presenting at The University of Strathclyde

Seminar Outline

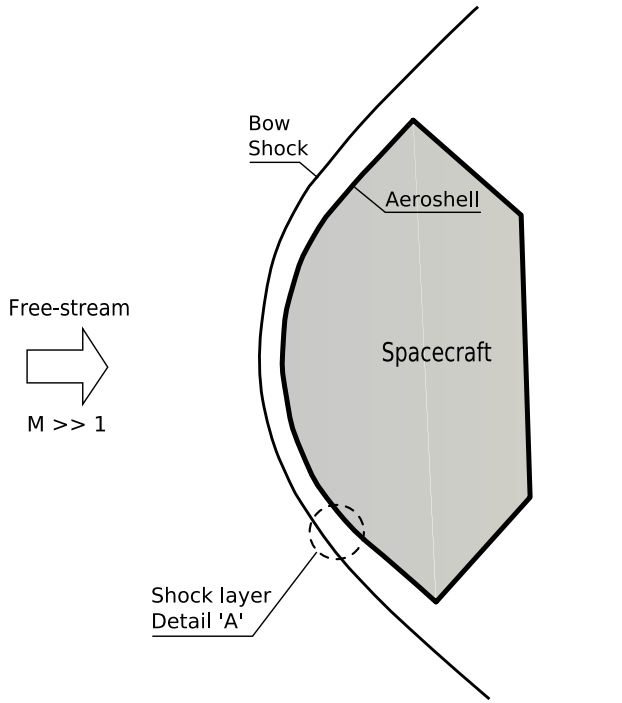
- Hypersonics research: why and how
- Computational Fluid Dynamics for hypersonic flows
- Eilmer: compressible flow simulator
  - implementation
  - features
- Verification and validation
- New developments
  - shock-fitting mode and moving grids
  - GPU acceleration for reacting flows
  - adjoint-based optimisation

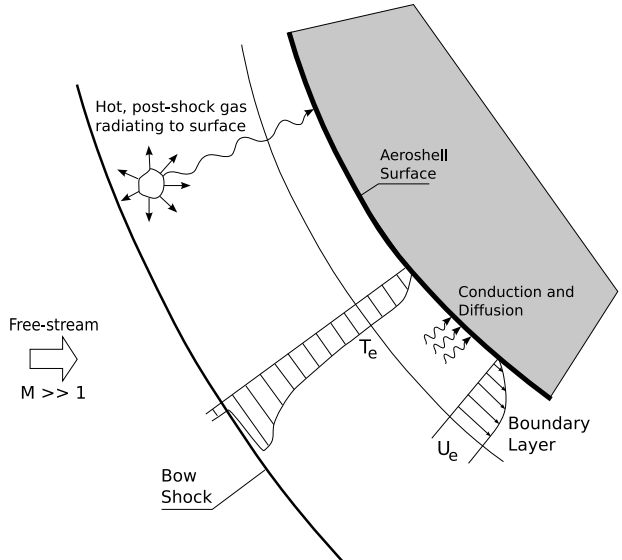
Why hypersonics?





Why computer simulation of hypersonic flows?





- The flow physics are modelled reasonably well, but the interactions are complex.
- Physical experimentation provides insights but has limitations, such as: scaling (time and length), boundary conditions, quantification of uncertainties, expense
- Computer simulation complements physical experiments, and vice versa.
- Analysis via computer simulation (might) substitute when we don't have suitable experience.
- Computer analysis good for 'what-if' studies, and design.

How hypersonics?

Conservation of mass:

$$\frac{\partial}{\partial t}\rho + \nabla \cdot \rho \mathbf{u} = 0 \tag{1}$$

Conservation of species mass:

$$\frac{\partial}{\partial t}\rho_i + \nabla \cdot \rho_i \mathbf{u} = -\left(\nabla \cdot \mathbf{J}_i\right) + \dot{\omega}_i \tag{2}$$

Conservation of momentum:

$$\frac{\partial}{\partial t}\rho \mathbf{u} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla p - \nabla \cdot \left\{-\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\dagger}) + \tfrac{2}{3}\mu(\nabla \cdot \mathbf{u})\boldsymbol{\delta}\right\} \tag{3}$$

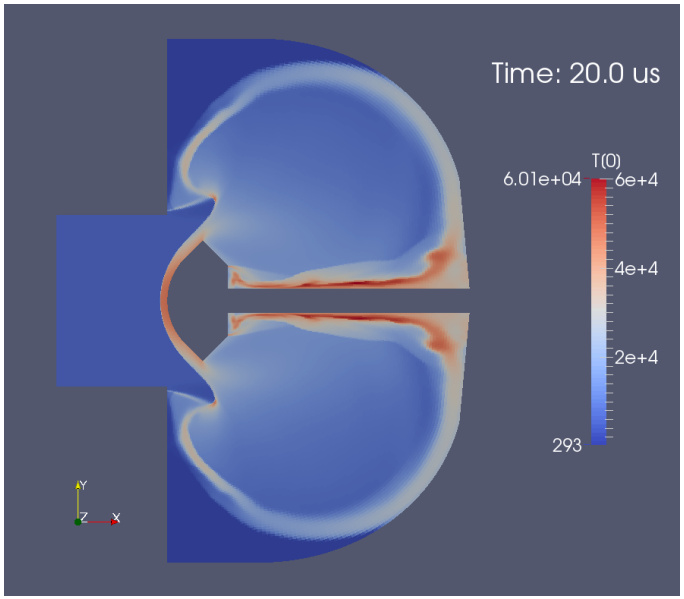
Conservation of total energy:

$$\begin{aligned}\frac{\partial}{\partial t}\rho E + \nabla \cdot (e + \frac{p}{\rho})\mathbf{u} = \nabla \cdot [k\nabla T + \sum_{s=1}^{N_v} k_{v,s}\nabla T_{v,s}] + \nabla \cdot \left[\sum_{i=1}^{N_s} h_i \mathbf{J}_i\right] \\ - \left(\nabla \cdot \left[\left\{-\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\dagger}) + \tfrac{2}{3}\mu(\nabla \cdot \mathbf{u})\boldsymbol{\delta}\right\} \cdot \mathbf{u}\right]\right) - Q_{\text{rad}}\end{aligned} \tag{4}$$

Conservation of vibrational energy:

$$\frac{\partial}{\partial t}\rho_i e_{v,i} + \nabla \cdot \rho_i e_{v,i}\mathbf{u} = \nabla \cdot [k_{v,i}\nabla T_{v,i}] - \nabla \cdot e_{v,i}\mathbf{J}_i + Q_{T-V_i} + Q_{V-V_i} + Q_{\text{Chem}-V_i} - Q_{\text{rad}_i} \tag{5}$$

Eilmer features – 1/2



- 2D/3D compressible flow simulation.
- Gas models include ideal, thermally perfect, equilibrium.
- Finite-rate chemistry.
- Multi-temperature and state-specific thermochemistry.
- Inviscid, laminar, turbulent (k-ω) flow.
- Solid domains with conjugate heat transfer in 2D.
- User-controlled moving grid capability, with shock-fitting method for 2D geometries.
- Dense-gas thermodynamic models and rotating frames of reference for turbomachine modelling.

Eilmer features – 2/2

- Transient, time-accurate, using explicit Euler, PC, RK updates.
- Alternate steady-state solver with implicit updates using Newton-Krylov method.
- Parallel computation via shared-memory on workstations, and using MPI on a cluster computer.
- Multiple block, structured and unstructured grids.
- Native grid generation and import capability.
- Unstructured-mesh partitioning via Metis.

- `en.wikipedia.org/wiki/Eilmer_of_Malmesbury`
- Gas model calculator and compressible flow relations.

Origins

- in the late 1980s, the state of the art for scramjet simulations involving reactive flow was JP Drummond *SPARK* code
- Flow solver component based on Bob McCormack's (1969) finite-difference shock-capturing technique.
- All configuration hard-coded into the Fortran source code and compiled to run on a Cray supercomputer.
- In the 1980s, a new CFD technology (upwind flux) was being developed by the applied mathematics people and parallel computing environments were being developed by the computer science people (cluster computers).
- Dec 1990: following a CFD lesson on the chalk-board from Bob Walters and Bernard Grossman, *cns4u* was started with the intention to be like SPARK but with new technology

Development of Eilmer

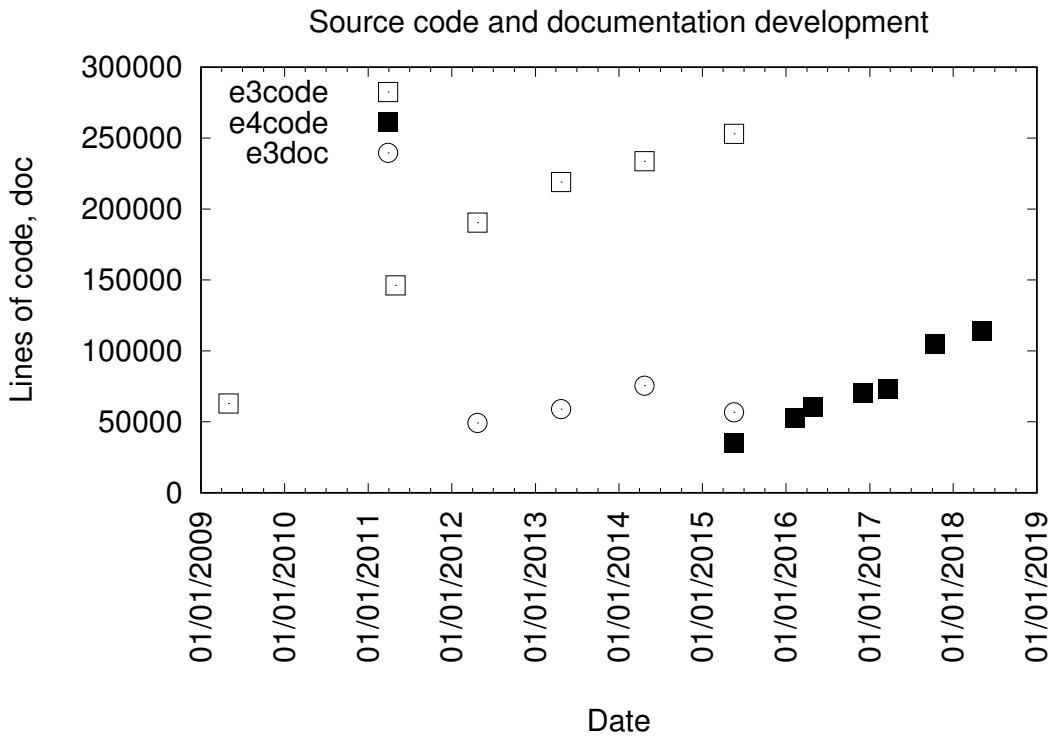
- 1993 built *sm3d*, a space-marching code for 3D scramjet flows
- 1995 through 1999: the postgrad years expanded scope of experimentation and application
- 1996: code reformulation around fluxes (frequent discussions with Mike Macrossan); all code still in C with a preprocessor having a little command interpreter built in.
- 1997: discovered scripting languages Tcl and Python
- May 2003: *scriptit.tcl* provided fully programmable environment for simulation-preparation.
- Aug 2004: *Elmer* began as a hybrid code using Python and C.
- Jun 2005: rewrite of *Elmer(2)* in C alone.
- Jul 2006: rewrite *Elmer2* in C++ and, in 2008, call it *Eilmer3*. Class-based implementation was easier to extend.

Eilmer4 – think big, but control the complexity

- Jun 2015+: rebuild in the D and Lua programming languages.
- Heather Muir worked on the unstructured-grid generator. based on the paving algorithm.

Implementation

D language data storage and solver, with embedded Lua interpreters for preprocessing, user-controlled run-time configuration in boundary conditions and source terms and thermochemical configuration.



At 60 lines per page, the Eilmer4 code is equivalent to a 1200 page document.

Verification & Validation

Quality Control: Verification & Validation

To quote Blottner:

> *Verification: are we solving the equations right?*
> *Validation: are we solving the right equations?*

Verification:

- comparison to exact solutions
- comparison to manufactured solutions
- order of accuracy test
- code-to-code comparison

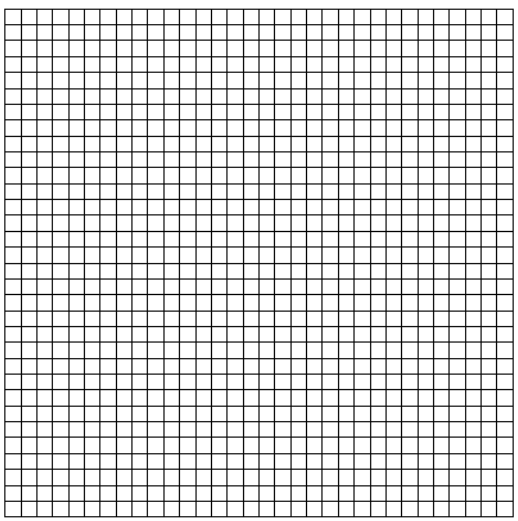
Validation:

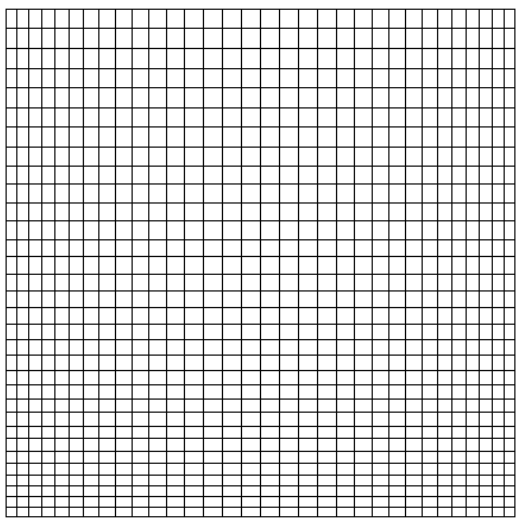
- are our models of flow physics correct
- comparison to experimental measurements

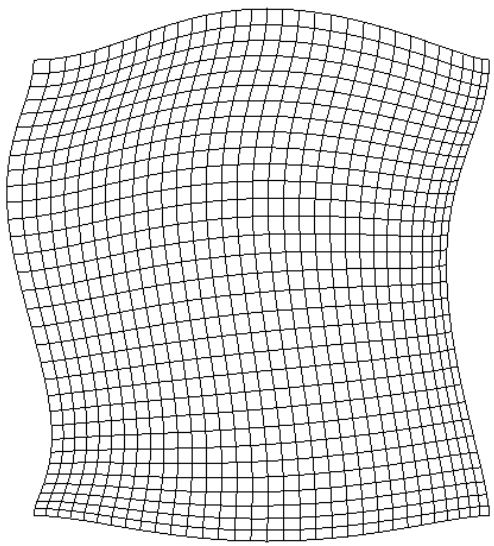
Verification for invisicd flows

For inviscid flows, we have verified the code using:

- Roy's manufactured solution for supersonic invisicd flow. Demonstrated 2nd order spatial accuracy on regular, stretched, and distorted grids.
- Power & Stewarts's exact solution for an oblique detonation wave. Demonstrated 1st order spatial accuracy in the presence of shocks.

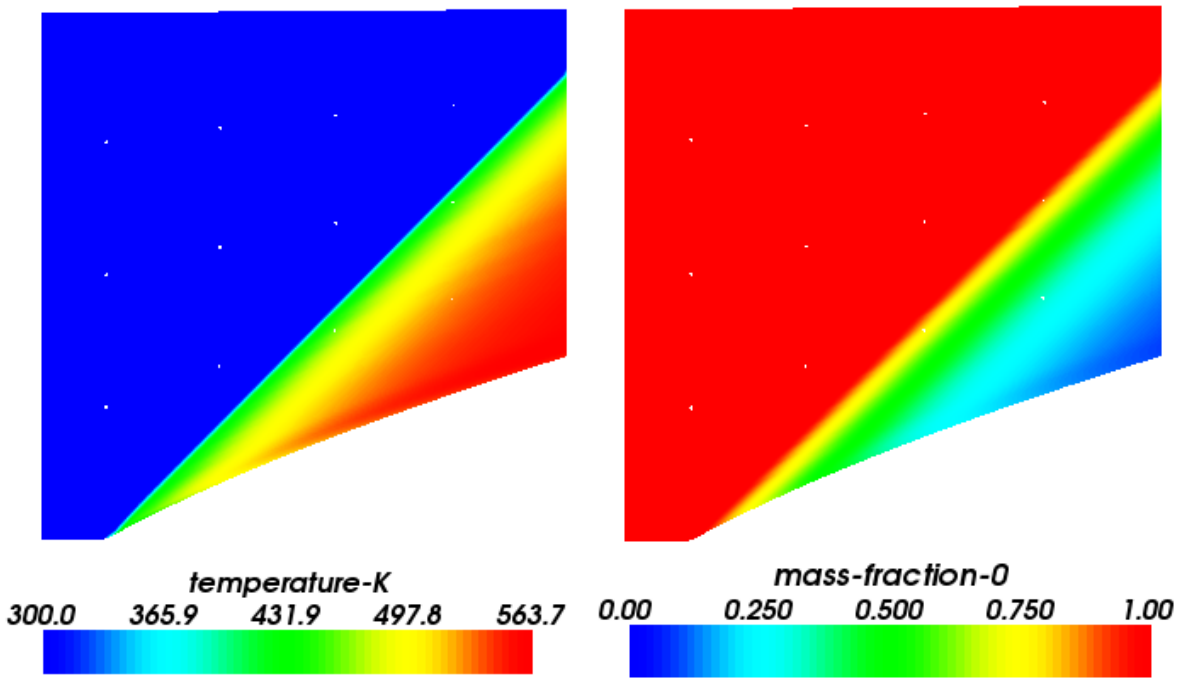






Verification: oblique detonation wave

The flow problem is an oblique detonation wave which is supported by a curved wedge surface. The analytical solution for this problem was first presented by Powers & Stewart (AIAA J. 1992), and first employed for verification by Powers & Aslam (AIAA J. 2006).



In order to make the problem analytically tractable, the reaction mechanism for the detonation is simplified. The reaction is a one-step reaction that proceeds once an ignition temperature is reached.

The Powers and Aslam gas model

- Two species A, B, with reaction of A to B proceeding at rate

$$\frac{d\rho_B}{dt} = \alpha\, \rho_A\, H(T - T_i)$$

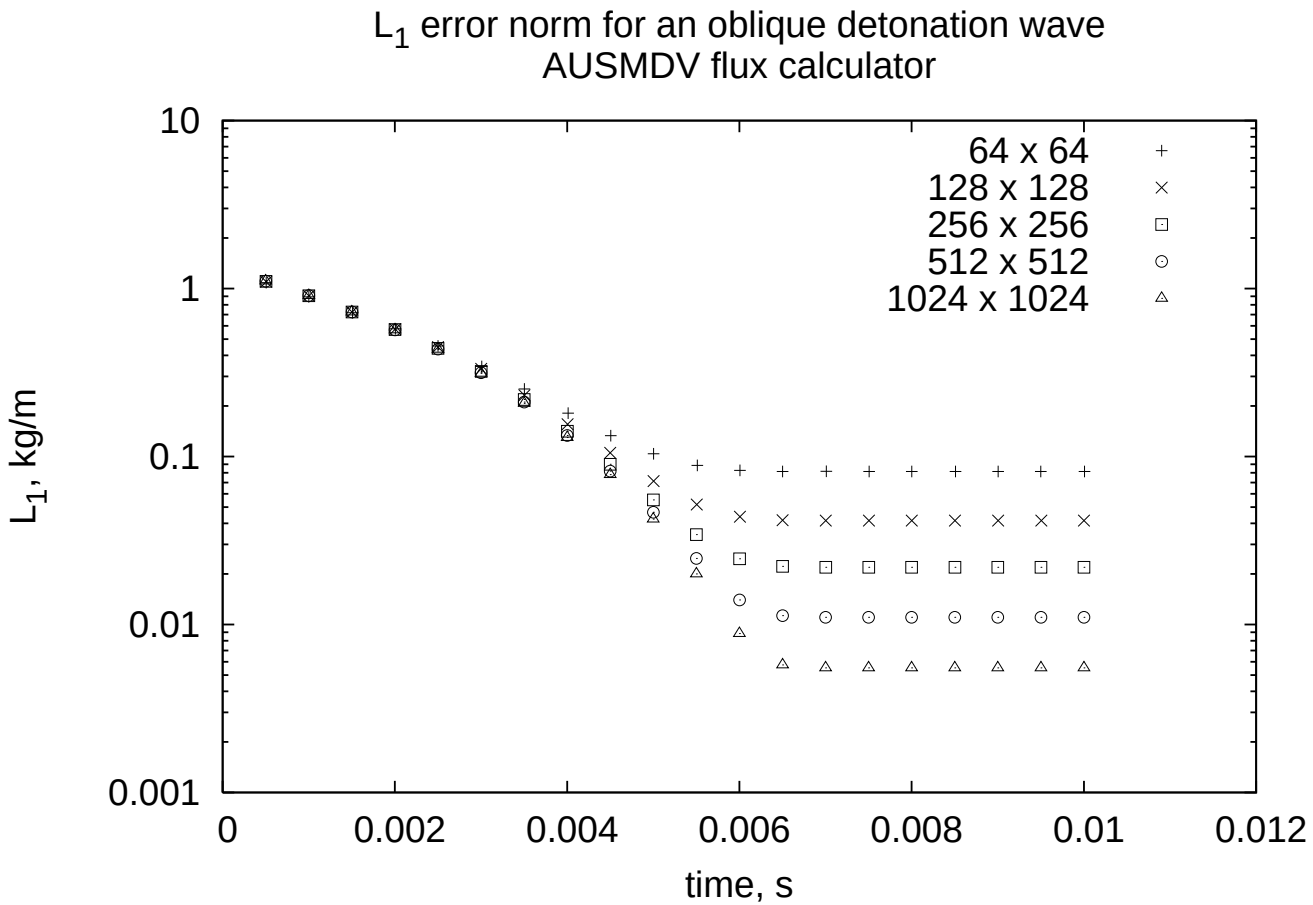
  with rate constant $\alpha = 0.001\,\mathrm{s}^{-1}$
- Reaction progress variable is mass fraction of B: $\lambda = Y_B = \frac{\rho_B}{\rho}$
- $Y_A = 1 - Y_B$
- Equation of state for internal energy:

$$u = \frac{1}{\gamma - 1}\frac{p}{\rho} - \lambda q = C_v T - Y_B q$$

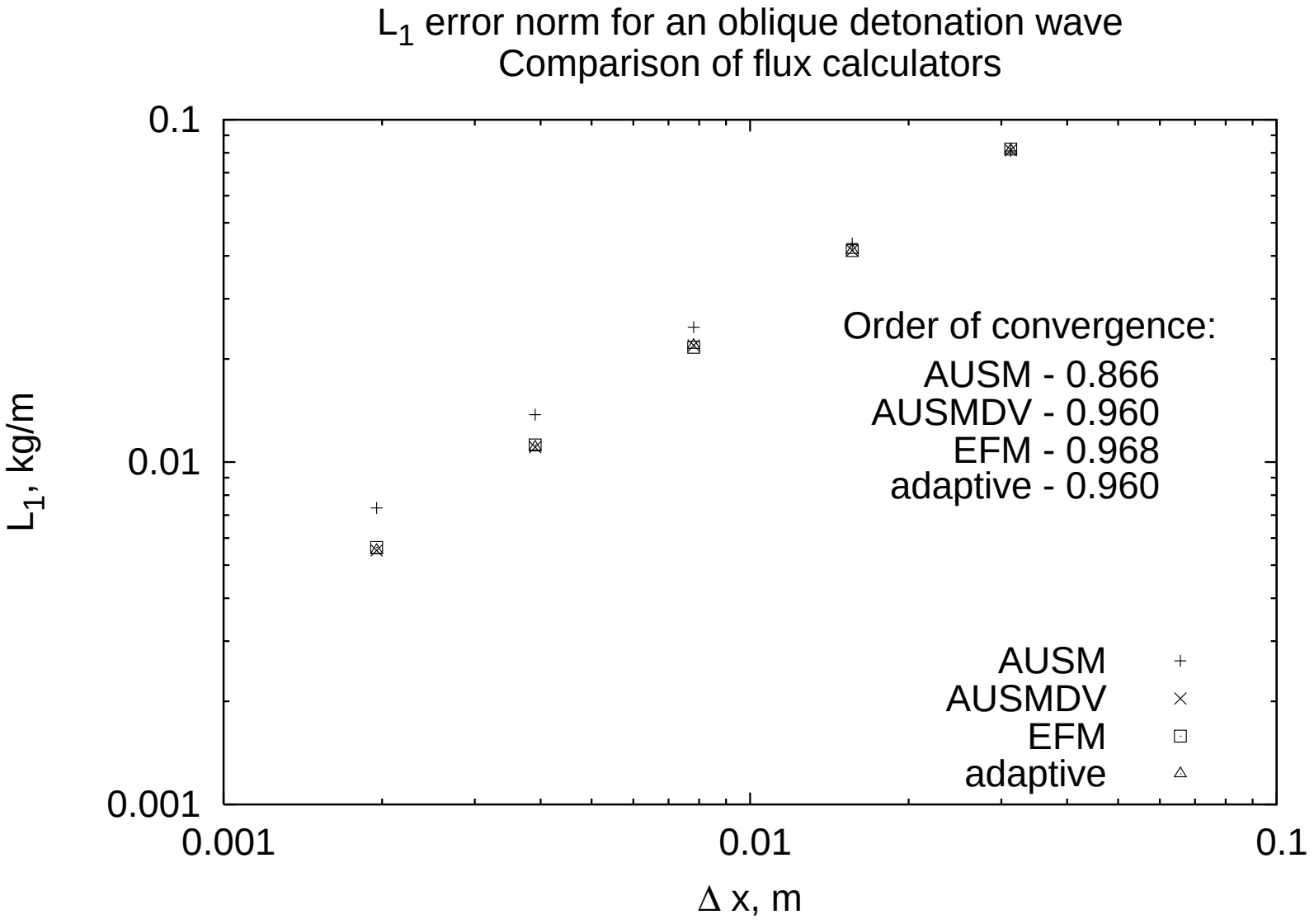
  with heat of reaction $q = 300000\,\mathrm{J/kg}$ and ratio of specific heats $\gamma = 6/5$.
- Pressure: $p = \rho R T$, with gas constant $R = 287\,\mathrm{J/kg.K}$

Oblique detonation wave results – 1/2

$$L_1 = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \left| \rho_{n,ij} - \rho_{e,ij} \right| \Delta x \Delta y$$



Oblique detonation wave results – 2/2



Flows with chemical nonequilibrium

- Flow can be distinguished (and modelled) in three classifications:
  1. frozen flow (ideal gas model) $-\ t_{chem} \gg t_{flow}$
  2. nonequilbrium flow (finite-rate chemistry) $-\ t_{chem} \approx t_{flow}$
  3. equilibrium flow $-\ t_{chem} \ll t_{flow}$
- Model for rate of chemical change: Law of Mass Action

$$\frac{d[X_i]}{dt} = (\nu_i^{''} - \nu_i^{'}) \left\{ k_f \prod_i [X_i]^{\nu_i^{'}} - k_b \prod_i [X_i]^{\nu_i^{''}} \right\}$$

- Forward and reverse rates are computed from a modified Arrhenius form

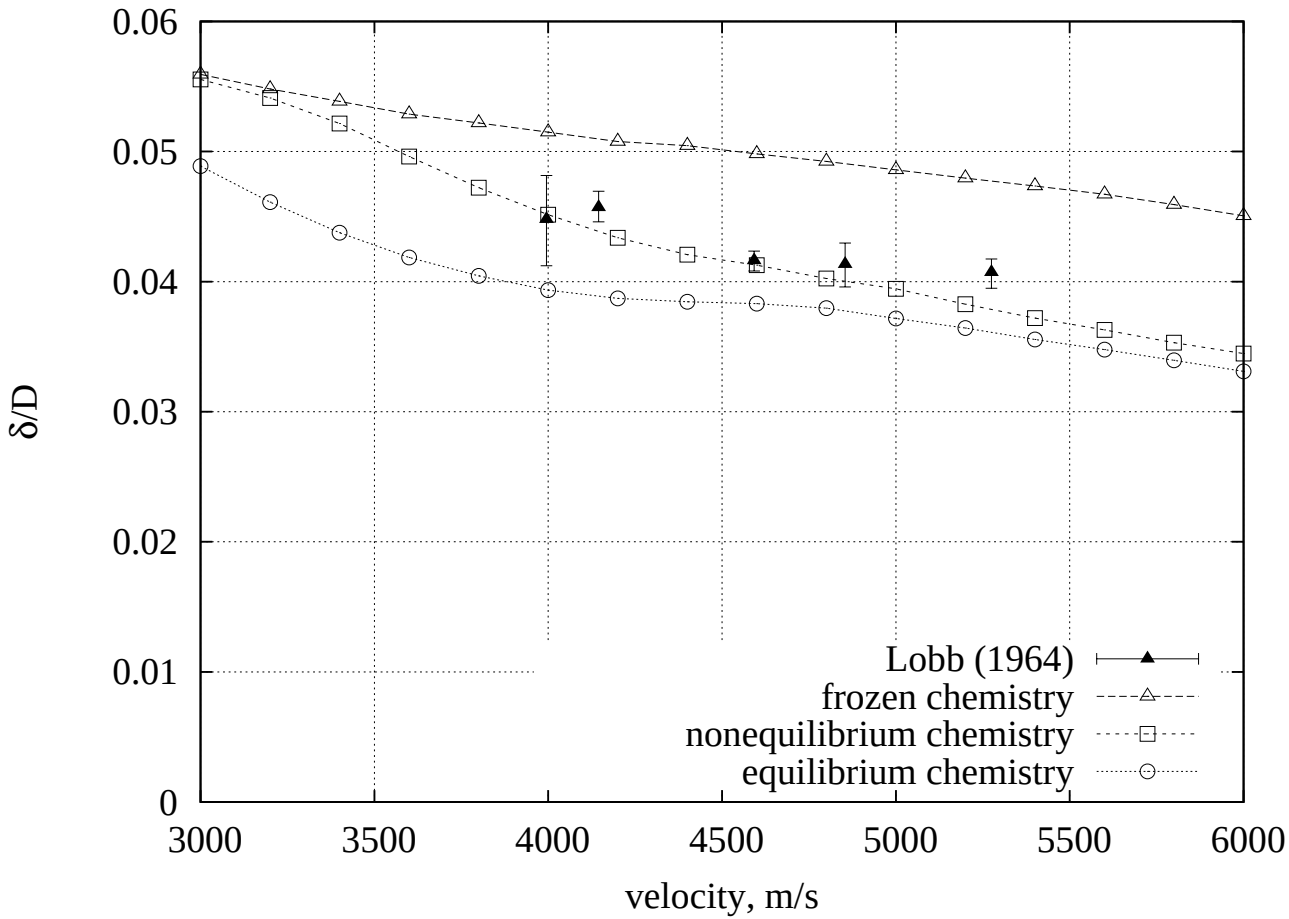
$$k = AT^n e^{-T_a/T}$$

  alternatively, reverse rates may come from computing the equilibrium constant

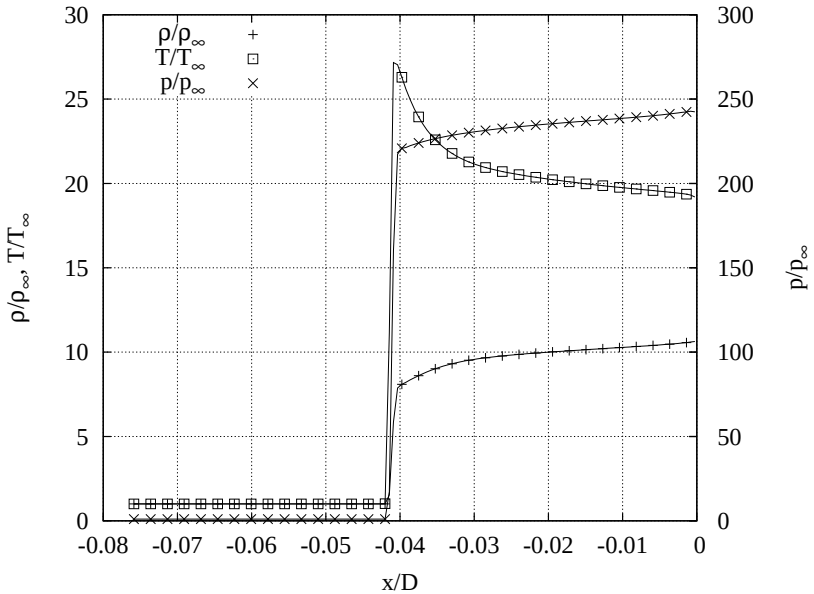
- This ODE integrators available are:
  - Mott's α Quasi-Steady-State approach (for stiff systems, eg. combustion)
  - Runge-Kutta-Fehlberg (efficient for non-stiff systems, eg. endothermic chemistry)

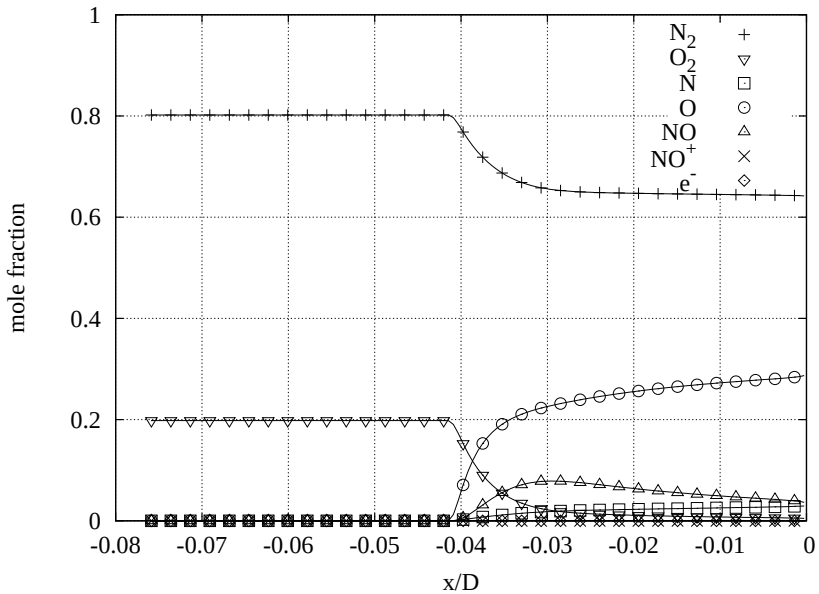
Validation of reacting flow model for air

- Lobb (1964) performed a series of experiments to measure shock detachment distance on spheres fired into air at hypervelocities
- A 0.5 in nylon sphere was fired in a ballistic range into air at various speeds and pressures



Stagnation streamline properties for reacting air over sphere





Flows with thermal nonequilibrium

molecules:	atoms:
translation	translation
rotation	
vibration	
electronic	electronic

Why isn't the vibrational energy in equilibrium?

- it typically takes 5-10 collisions for the rotational mode to equilibriate at the translational temperature
- however, it can take on the order of 10^2-10^3 collisions for vibrational modes to equilibriate

A specific example...
For a blunt body of nose radius 1 m with a velocity of 10 km/s at an altitude of 75 km:

$$\tau_{flow} \approx 10^{-4} s, \tau_{chem} \approx 10^{-3} s, \tau_{vib} \approx 10^{-5}$$

Thivet, Perrin and Candel (1991)

Engineering model for flows with thermal nonequilibrium

$$\frac{\partial}{\partial t}\rho_i e_{v,i} + \nabla \cdot \rho_i e_{v,i}\mathbf{u} = \nabla \cdot [k_{v,i}\nabla T_{v,i}] - \nabla \cdot e_{v,i}\mathbf{J}_i + Q_{T-V_i} + Q_{V-V_i} + Q_{\text{Chem}-V_i} - Q_{\text{rad}_i}$$

- Internal energy modes are separable (Born-Oppenheimer approximation)
- Each separated mode is assumed to populate a Boltzmann distribution at a particular temperature
- Nonequilibrium occurs when these governing temperatures differ for different energy modes
- Depending on extent of nonequilbrium, each molecular species might have its own describing vibrational temperature
- Example: four-temperature air model has one temperature to describe tranlsational/rotational energy and one temperature each for vibrational energy of N_2, O_2 and NO

Thermal nonequilibrium: energy exchange

$$\frac{\partial}{\partial t}\rho_i e_{v,i} + \nabla \cdot \rho_i e_{v,i}\mathbf{u} = \nabla \cdot [k_{v,i}\nabla T_{v,i}] - \nabla \cdot e_{v,i}\mathbf{J}_i + Q_{T-V_i} + Q_{V-V_i} + Q_{\mathrm{Chem}-V_i} - Q_{\mathrm{rad}_i}$$

Translational-vibrational energy exchange

$$Q_{T-V_i} = \sum_{c=1}^{n_s} x_c \frac{e^*_{v,p} - e_{v,p}}{(\tau_v^{p-c})_{V-T}}$$

Vibrational-vibrational energy exchange

$$Q_{V-V_i} = \sum_{q=1,q\neq p}^{n_v} \frac{x_q}{(\tau_v^{p-q})_{V-V}} \left(\frac{1-\exp(-\Theta_{v,p}/T)}{1-\exp(-\Theta_{v,q}/T)} \frac{e_{v,q}}{\hat{e}_{v,q}} (\bar{e}_{v,p} - e_{v,p}) - \frac{e_{v,p}}{\hat{e}_{v,q}} (\bar{e}_{v,q} - e_{v,q}) \right)$$

Thermal nonequilibrium: chemistry-energy coupling

$$\frac{\partial}{\partial t}\rho_i e_{v,i} + \nabla \cdot \rho_i e_{v,i}\mathbf{u} = \nabla \cdot [k_{v,i}\nabla T_{v,i}] - \nabla \cdot e_{v,i}\mathbf{J}_i + Q_{T-V_i} + Q_{V-V_i} + Q_{\text{Chem}-V_i} - Q_{\text{rad}_i}$$

Chemistry-vibration coupling

- If the gas is in nonequilibrium, how are the reaction rates affected?

$$k = k^{EQ}(T)\Psi(T, T_{v,i})$$

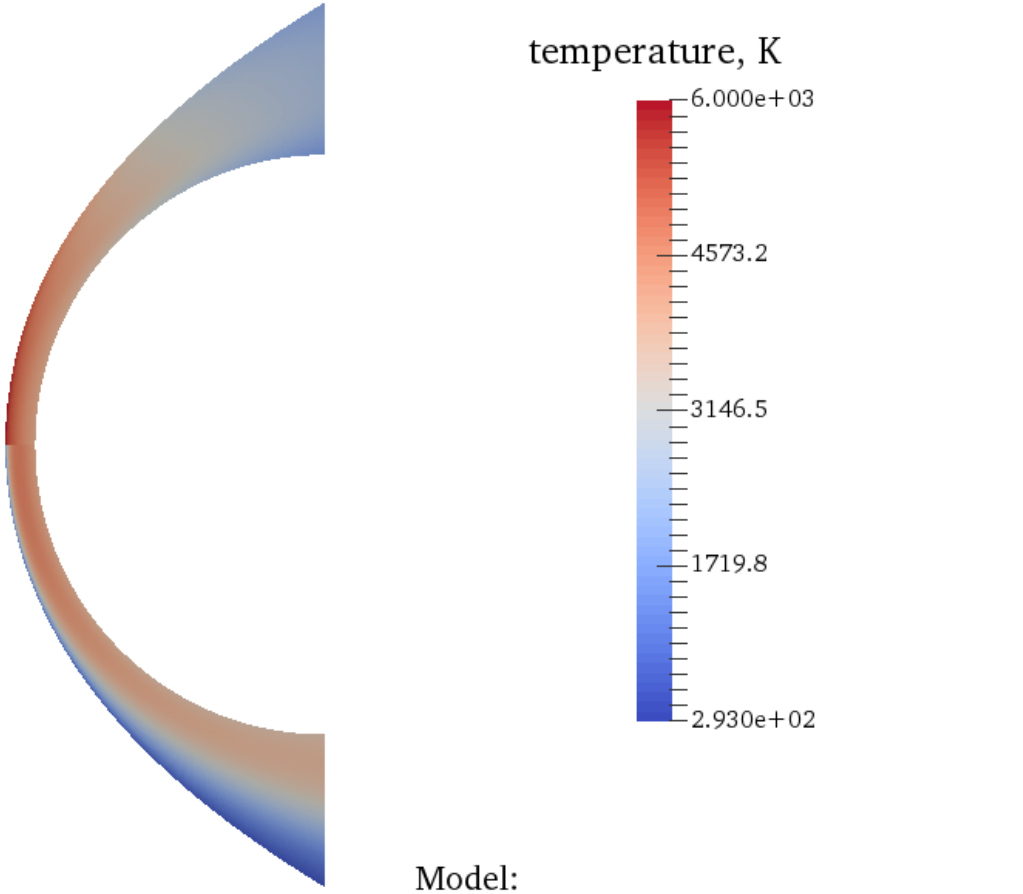
- When chemical reactions take place, how does this effect the average vibrational energy?
- A preferential model is implemented in Eilmer by Knab et al. This is a generalised and extended version of Treanor & Marrone's preferential dissociation model.

$$Q_{\text{Chem}-V_i} = \sum_{j=1}^{Nr} \left[-G_{va,ij} \frac{1}{f_i} \left(\frac{df_i}{dt} \right)_f + G_{app,ij} \frac{1}{f_i} \left(\frac{df_i}{dt} \right)_r \right]$$

Validation: 2-temperature reacting air flow over a sphere

Nonaka experiment:
vel.x = 3490 m/s
p_inf = 4850 Pa
T_inf = 293 K
f_N2 = 0.767
f_O2 = 0.233

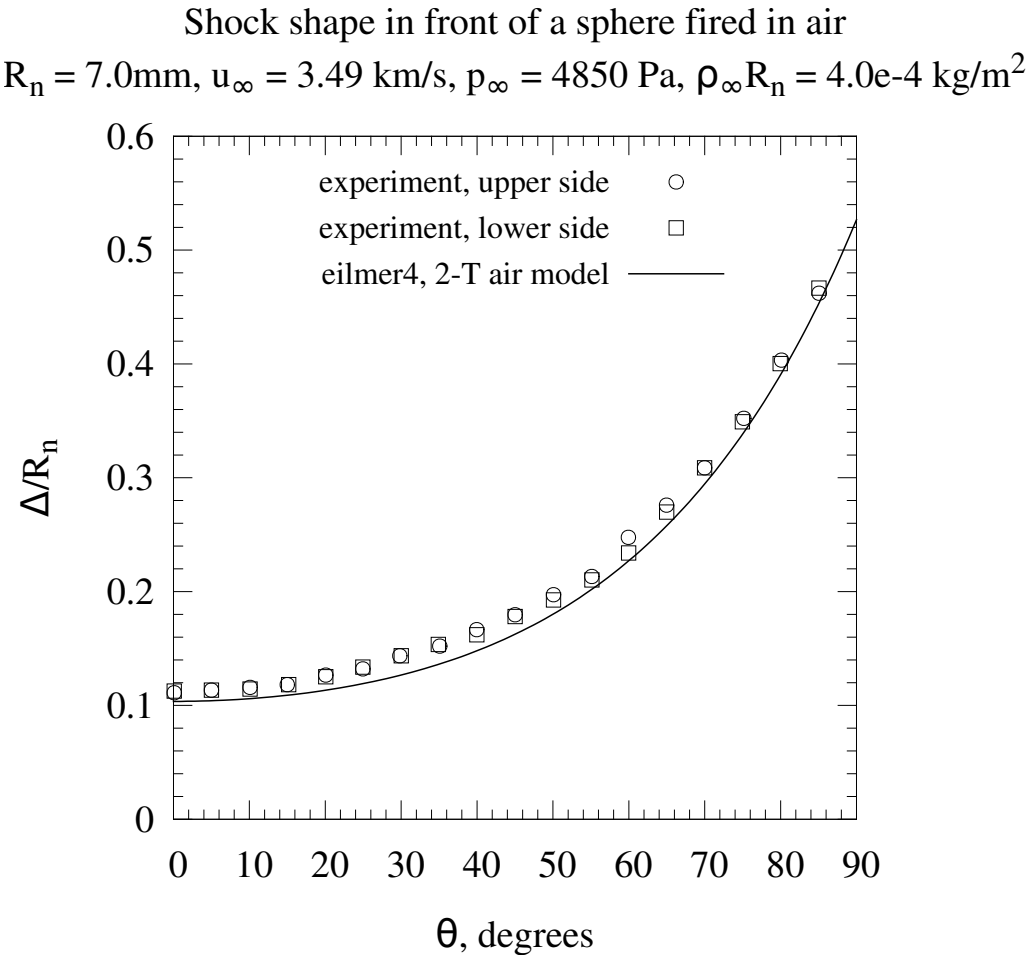
Top half: transrotaional temperature
Bottom half: vibroelectronic temperature



t = 60.172 micro s

Model:
5-species, 2-temperature air
Gupta et al reaction rates
Park 2-T modifications for rate constants

Validation: 2-temperature reacting air flow over a sphere



Recent Development Activity

Recent Development Activity

Advanced and experimental features:

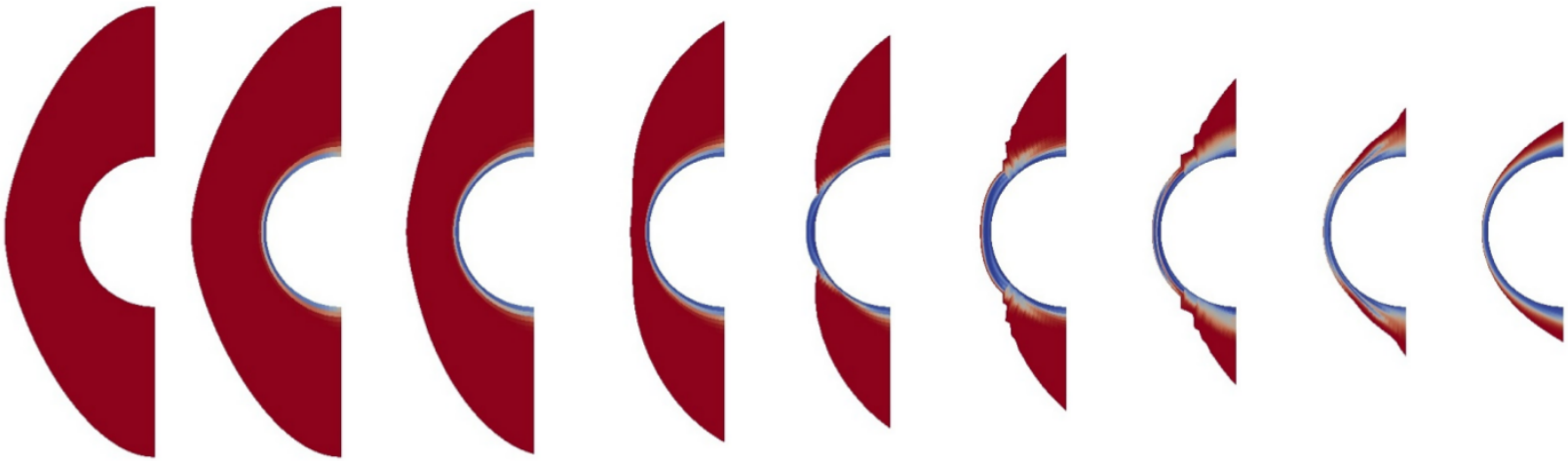
- user customisation/control of boundary conditions, source terms and grid motion via Lua scripts
- moving grids
- shock-fitting boundaries
- wall models for RANS simulations
- GPU-acceleration for reacting flows
- coupled fluid/solid domains for conjugate heat transfer problems
- steady-solver using Newton-Krylov method
- adjoint solver for CFD-in-the-loop optimisation

Moving grid capability

- Moving grids and moving boundaries are also controlled using small Lua programs built by the user.
- It is hard to predict all the types of motion that users would want to simulate, hence the user-built programs to control the motion.
- The user needs to set the velocity of the grid points and the simulation code takes care of updating to the new positions.
- With enough creativity, this can be used to simulate fluid-structure interation problems.

Shock-fitting boundaries

- For blunt body flows, it is critical to have nicely aligned grids with the bow shock.
- One solution is to do direct shock-fitting at the boundaries.
- Eilmer uses moving grid capability and a specialised shock-fitting boundary condition to allow shock-fitting at inflow boundaries.



GPU acceleration for chemically reacting flows

Graphics Processing Units for High Performance Computing

CPUs:

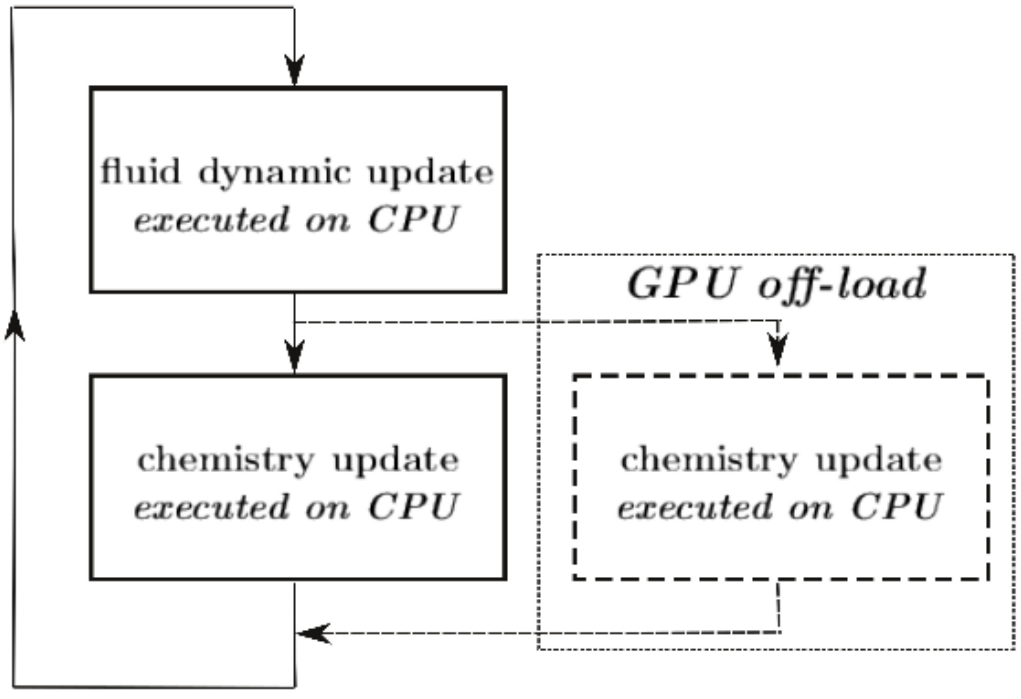
+ excellent serial thread performance
+ modern branch prediction and efficient memory access
+ handle complex algorithms well without too much developer intervention

GPUs:

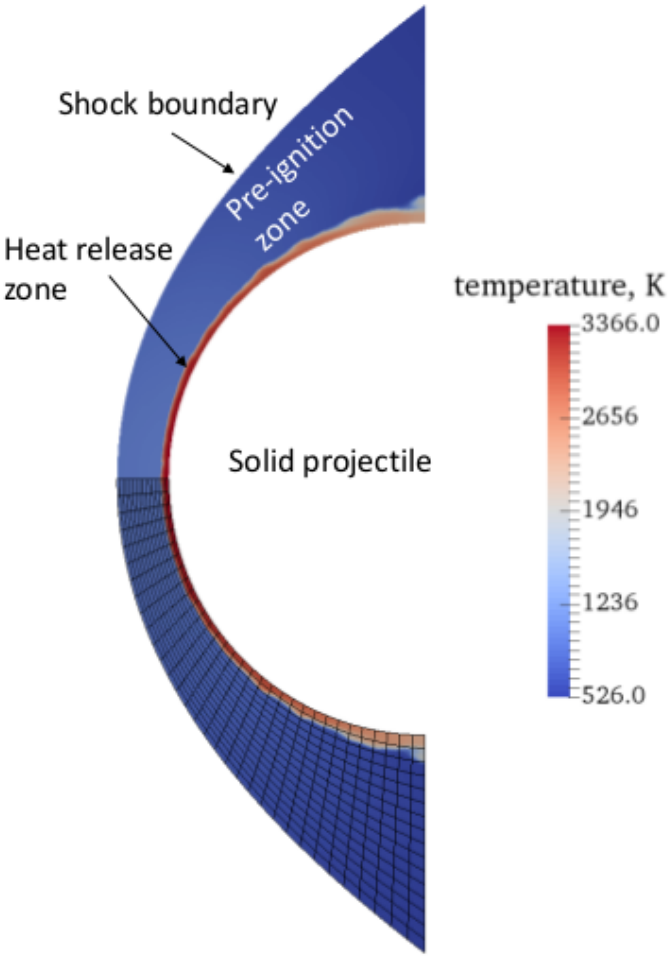
+ excel at floating point operations
+ high performance through thread parallelism
+ parallelism is Single-Instruction, Multiple-Data (SIMD) type
+ hardware abstraction is low: requires careful effort on part of developer

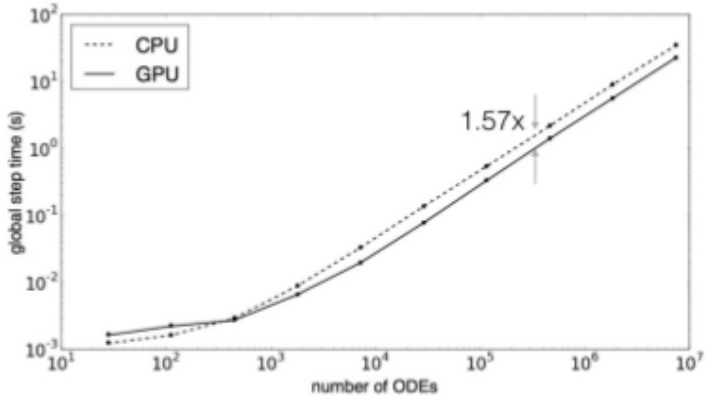
GPU acceleration: Eilmer implementation

+ operator-splitting algorithm

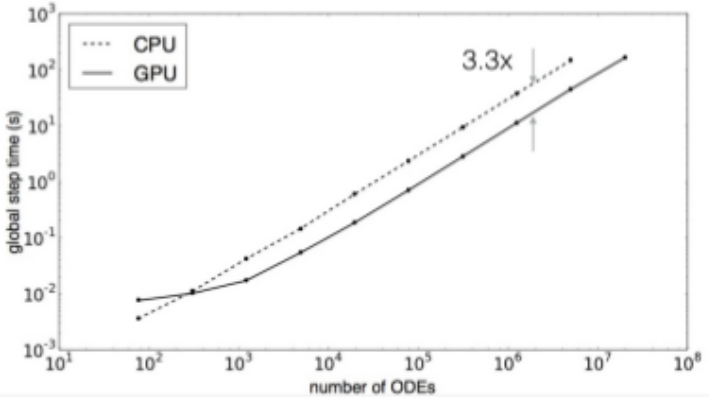


GPU acceleration: proof-of-concept, early results





Hydrogen combustion



Methane combustion

Eilmer – find out more

Website:

`cfcfd.mechmining.uq.edu.au/eilmer`

Source code repository:

`bitbucket.org/cfcfd/dgd`

Documentation in the Eilmer 4.0 guides:

- *Guide to the transient flow solver*
- *Guide to the basic gas models package*
- *Guide to the geometry package*
- *Formulation of the transient flow solver*
- *Reacting gas thermochemistry*