Applied Modeling & Simulation (AMS) Seminar Series

Development of Mutation^{++}: Multicomponent Thermodynamic and Transport Properties for Ionized Plasmas written in C++

J.B. Scoggins[1,2], Thierry Magin[1,2]

[1]von Karman Institute for Fluid Dynamics, Aerospace and Aeronautics Dept.

[2]Ecole Centrale Paris, Laboratoire EM2C

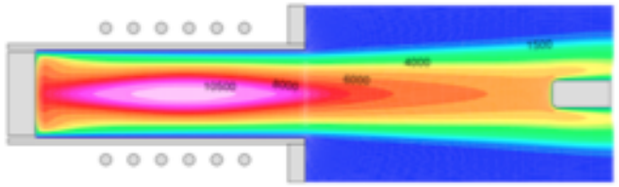
European Research Council

26 June 2014, NASA Ames Research Center

FP7 ERC Starting Grant

AEROSPACEPHYS

"Multiphysics models and simulations for reacting and plasma flows applied to the space exploration program."



computations

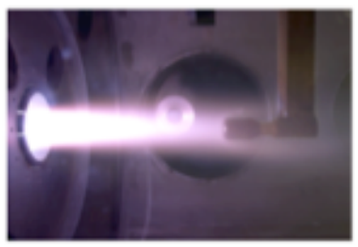
verification validation

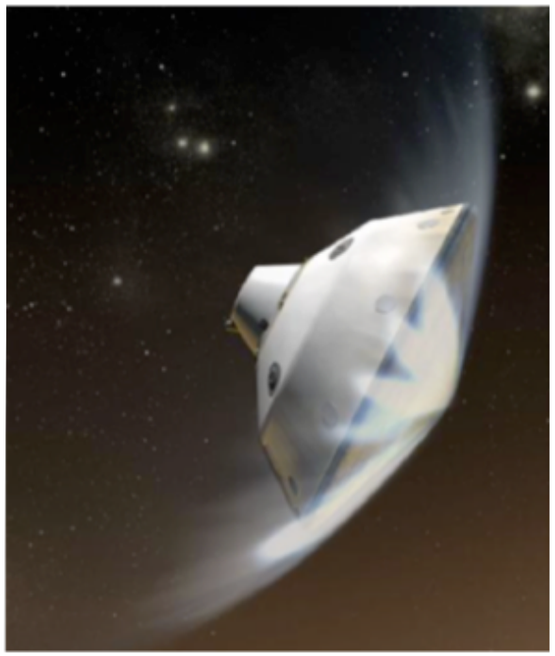
UQ

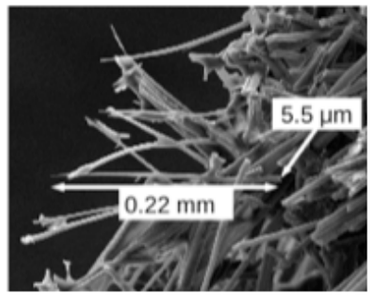
models experiments

$$\sigma_e = \frac{(n_e q_e)^2}{p_e} D_e$$

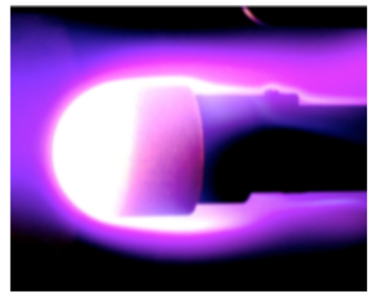
$$D_e = \tfrac{1}{3} p_e T_e M_h [\![\phi_e^{D_e}, \phi_e^{D_e}]\!]_e$$







material



flow



radiation

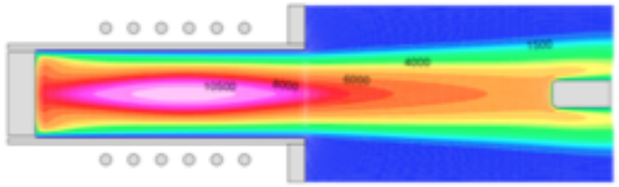
Helber et al., AIAA-2014-2122
Turchi et al., AIAA-2014-2125

European Research Council

FP7 ERC Starting Grant

AEROSPACEPHYS

"Multiphysics models and simulations for reacting and plasma flows applied to the space exploration program."



computations

verification

validation

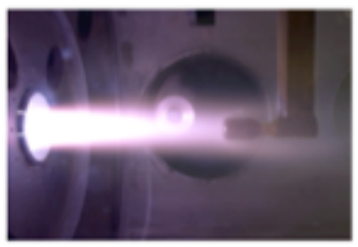
UQ

models

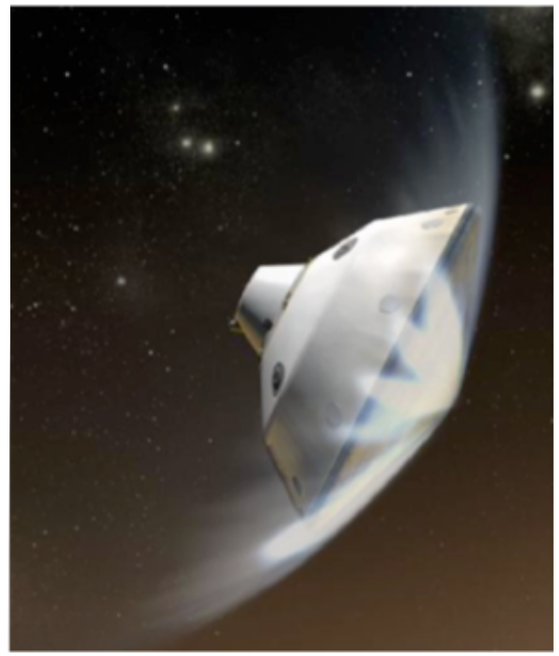
experiments

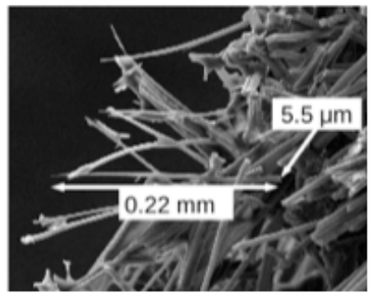
$$\sigma_e = \frac{(n_e q_e)^2}{p_e} D_e$$

$$D_e = \frac{1}{3} p_e T_e M_h [\![\phi_e^{D_e}, \phi_e^{D_e}]\!]_e$$

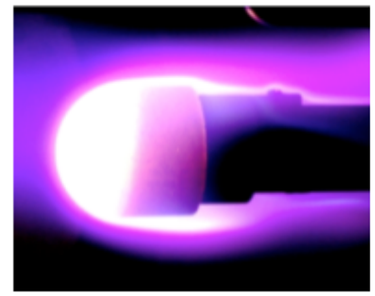


Mutation^{++}





material



flow



radiation

Helber et al., AIAA-2014-2122
Turchi et al., AIAA-2014-2125

European Research Council

Motivation from CFD

Governing equations for typical aerospace flows

N_m mass continuity equations

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$

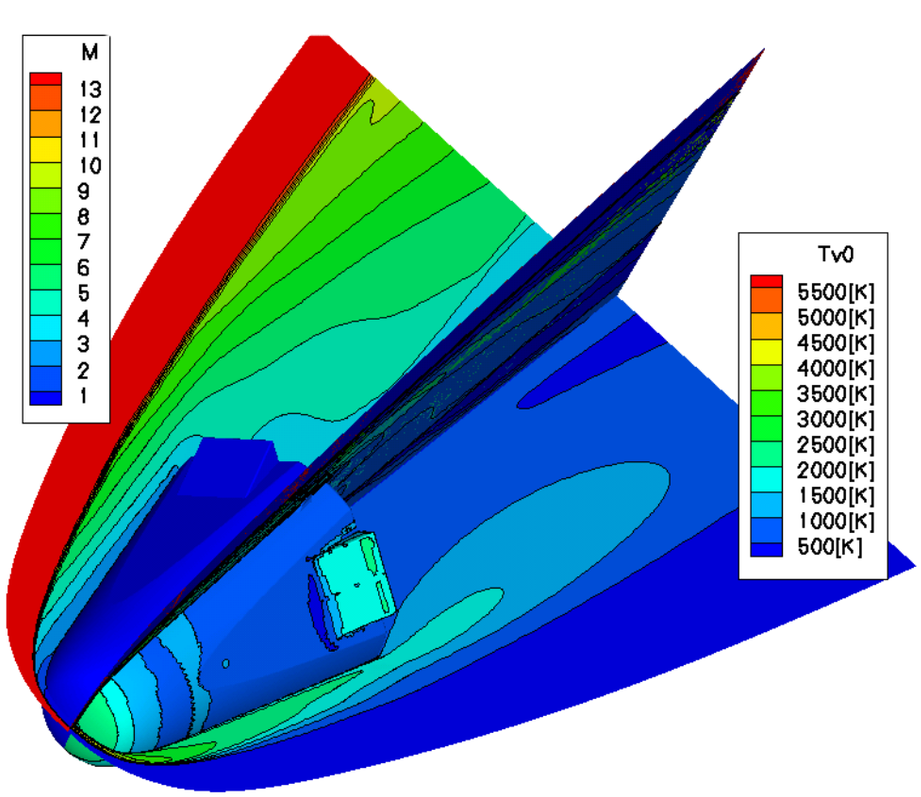
N_d momentum equations

$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$

1 total and N_e internal energy equations

$$\begin{cases} \partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau) \\ \partial_t (\rho e_M) + \nabla_x \cdot (\rho e_M u) = \sum_N \Omega_{MN} + \Omega_{MC} \end{cases}$$

discretization and solution via CFD



Simulation of ESA Expert vehicle with COOLFluid code developed at VKI

- Independent state variables (conserved, primitive, etc…) known everywhere in the CFD domain
- Dependent variables computed using constitutive relations
  - Varying model/algorithm complexities
  - Relying on various datasets which may be model dependent

Motivation from CFD

Governing equations for typical aerospace flows

General model

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$

$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$

$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

$$\partial_t (\rho e_M) + \nabla_x \cdot (\rho e_M u) = \sum_N \Omega_{MN} + \Omega_{MC}$$

Motivation from CFD

Governing equations for typical aerospace flows

General model

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$
$$\partial_t (\rho e_M) + \nabla_x \cdot (\rho e_M u) = \sum_N \Omega_{MN} + \Omega_{MC}$$

Equilibrium or frozen flows

$$\partial_t \rho + \nabla_x \cdot (\rho u) = 0$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

Motivation from CFD

Governing equations for typical aerospace flows

General model

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$
$$\partial_t (\rho e_M) + \nabla_x \cdot (\rho e_M u) = \sum_N \Omega_{MN} + \Omega_{MC}$$

Equilibrium or frozen flows

$$\partial_t \rho + \nabla_x \cdot (\rho u) = 0$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

Multicomponent reacting and STS flows

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

Motivation from CFD

Governing equations for typical aerospace flows

General model

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$
$$\partial_t (\rho e_M) + \nabla_x \cdot (\rho e_M u) = \sum_N \Omega_{MN} + \Omega_{MC}$$

Equilibrium or frozen flows

$$\partial_t \rho + \nabla_x \cdot (\rho u) = 0$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

Multicomponent reacting and STS flows

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

2T thermochemical nonequilibrium flows

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$
$$\partial_t (\rho e_V) + \nabla_x \cdot (\rho e_V u) = \Omega_{VT} + \Omega_{VC}$$

Motivation from CFD

Governing equations for typical aerospace flows

General model

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$
$$\partial_t (\rho e_M) + \nabla_x \cdot (\rho e_M u) = \sum_N \Omega_{MN} + \Omega_{MC}$$

- CFD requires knowledge of number and type of equations to solve
- Can utilize abstraction to develop a general CFD tool capable of solving different equation sets for different flow regimes
- Core library for implementing physical models/alagorithms which is shared amongst several CFD tools

Equilibrium or frozen flows

$$\partial_t \rho + \nabla_x \cdot (\rho u) = 0$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

Multicomponent reacting and STS flows

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$

2T thermochemical nonequilibrium flows

$$\partial_t \rho_i + \nabla_x \cdot (\rho_i u + \rho_i V_i) = \omega_i$$
$$\partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + p\mathcal{I}) = -\nabla_x \cdot \tau$$
$$\partial_t (\rho e) + \nabla_x \cdot (\rho u h) = -\nabla_x \cdot q - \nabla_x (u \cdot \tau)$$
$$\partial_t (\rho e_V) + \nabla_x \cdot (\rho e_V u) = \Omega_{VT} + \Omega_{VC}$$





Objective

Development of a library for high enthalpy plasmas

Quantities relevant to engineering design for hypersonic flows

- Heat flux & shear stress to the surface of the vehicle
- Prediction strongly relies on completeness and accuracy of numerical methods & physico-chemical models

Why a library for physico-chemical models?

- Implementation common to several CFD codes
- Nonequilibrium models are regularly improved
- Basic data such as rate-coefficients, cross-sections, and spectroscopic constants are constantly updated

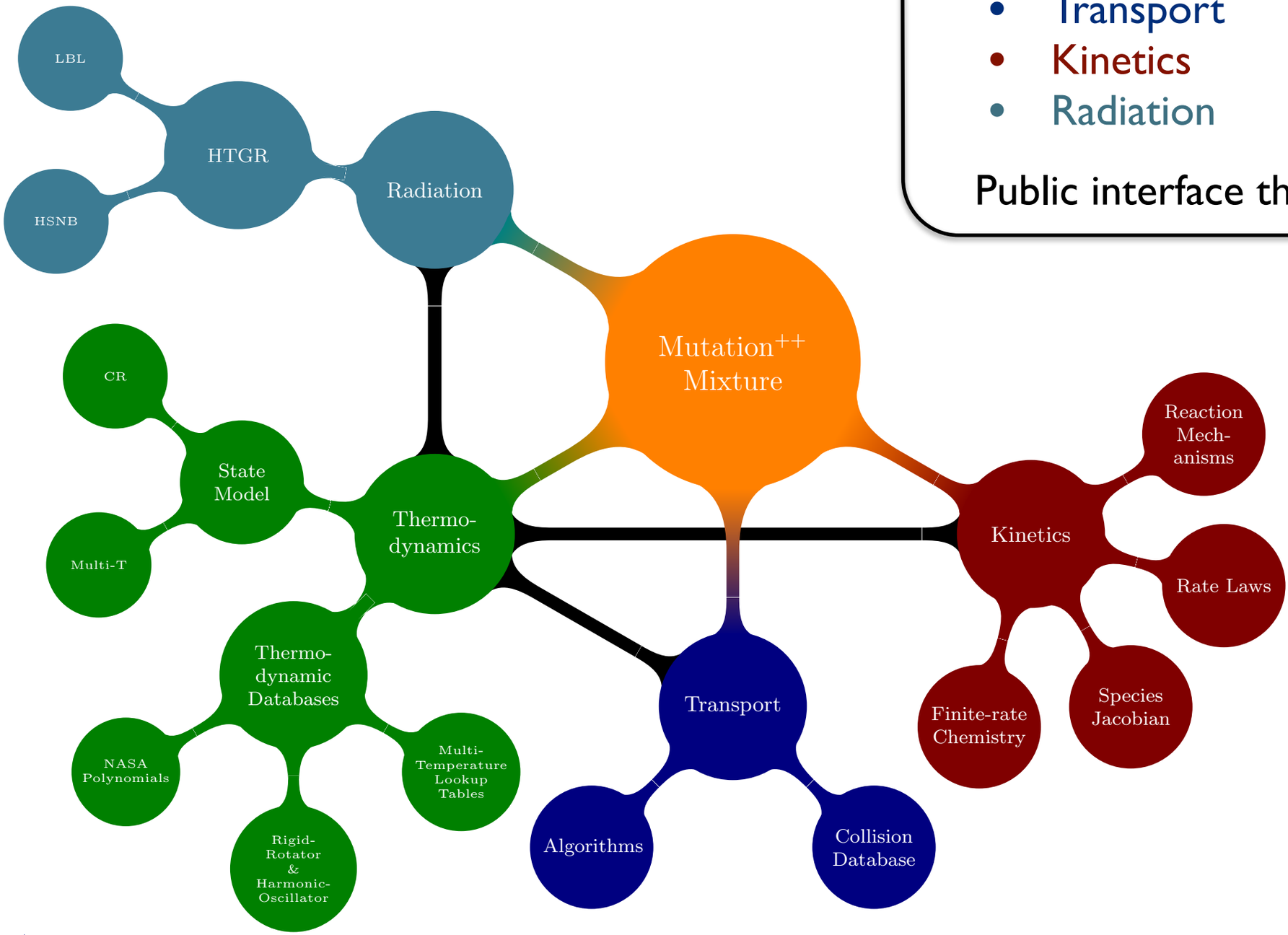
Constraints on library development

- High fidelity of the physical models
  - Laws of thermodynamics satisfied
  - Validation based on experimental results
- Low computational cost
- User-friendly interface



European Research Council

Mutation^{++} Overview



Functionality split into coherent parts

- Thermodynamics
- Transport
- Kinetics
- Radiation

Public interface through a Mixture class

European Research Council

Thermodynamics: Models

Multi-temperature lookup tables for high fidelity models based on Quantum mechanics

$$m_{\mathrm{N_2}} n_{\mathrm{N_2}} e^{\mathrm{rot}}_{\mathrm{N_2}} = \sum_{v=0}^{v_{\max}} \sum_{J=0}^{J_{\max(v)}} n_{i(v,J)} \left(T_{\mathrm{rot}}, T_{\mathrm{vib}}\right) \Delta\epsilon$$

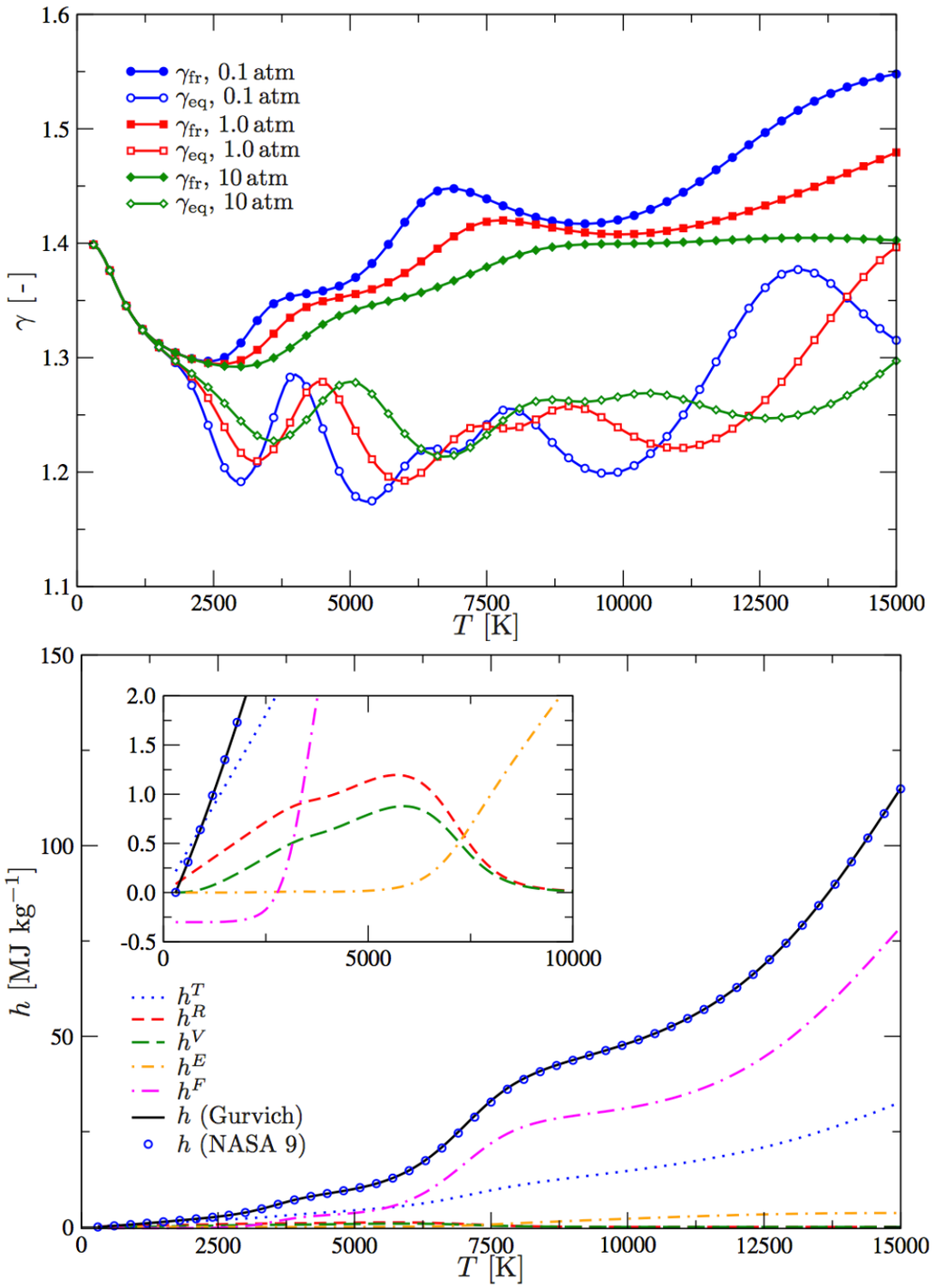
$$m_{\mathrm{N_2}} n_{\mathrm{N_2}} e^{\mathrm{vib}}_{\mathrm{N_2}} = \sum_{v=0}^{v_{\max}} \tilde{E}_v \sum_{J=0}^{J_{\max(v)}} n_{i(v,J)} \left(T_{\mathrm{rot}}, T_{\mathrm{vib}}\right)$$

Simplified model: Rigid-Rotator/Harmonic Oscillator

- Cutoff criterion for maximum number of electronic energy levels

High fidelity curve-fits: NASA polynomial database

- 7 and 9 coefficient databases are supported



Specific heat ratios and enthalpies for equilibrium Air.

European Research Council

Thermodynamics: Databases

Load thermodynamic data from

- Rigid-Rotator / Harmonic-Oscillator data
- NASA 7 or 9 coefficient polynomials

Can provide unique name which aliases NASA DB's

```
<thermodynamics type="NASA-9" dbname="C6H5OH(phenol" />
```

Stoichiometry and element definitions determine molecular weight

```
<!-- Carbon -->
<element name="C"  charge="0">
    <mw units="kg/mol">0.012011</mw>
</element>
```

```
<!-- Tricarbon -->
<species name="C3">
    <stoichiometry>
        C: 3
    </stoichiometry>
    <thermodynamics type="RRHO">
        <linear>
            yes
        </linear>
        <rotational_temperature units="K">
            0.607
        </rotational_temperature>
        <steric_factor>
            2
        </steric_factor>
        <vibrational_temperatures units="K">
            212.582
            212.582
            1723.379
            2990.460
        </vibrational_temperatures>
        <electronic_levels units="1/cm">
            <level degeneracy="1" energy="    0.0" />
            <level degeneracy="6" energy="14000.0" />
            <level degeneracy="6" energy="21500.0" />
            <level degeneracy="3" energy="23800.0" />
            <level degeneracy="2" energy="24675.5" />
            <level degeneracy="6" energy="29100.0" />
            <level degeneracy="3" energy="32800.0" />
            <level degeneracy="1" energy="32900.0" />
            <level degeneracy="2" energy="33700.0" />
            <level degeneracy="2" energy="40500.0" />
        </electronic_levels>
        <formation_enthalpy T="298 K" P="1 atm" units="J/mol">
            823630
        </formation_enthalpy>
    </thermodynamics>
</species>
```

Equilibrium Compositions

Gibbs Function Continuation (GFC) method by Pope **guarantees convergence** for well-posed problems [Pope, 2003]

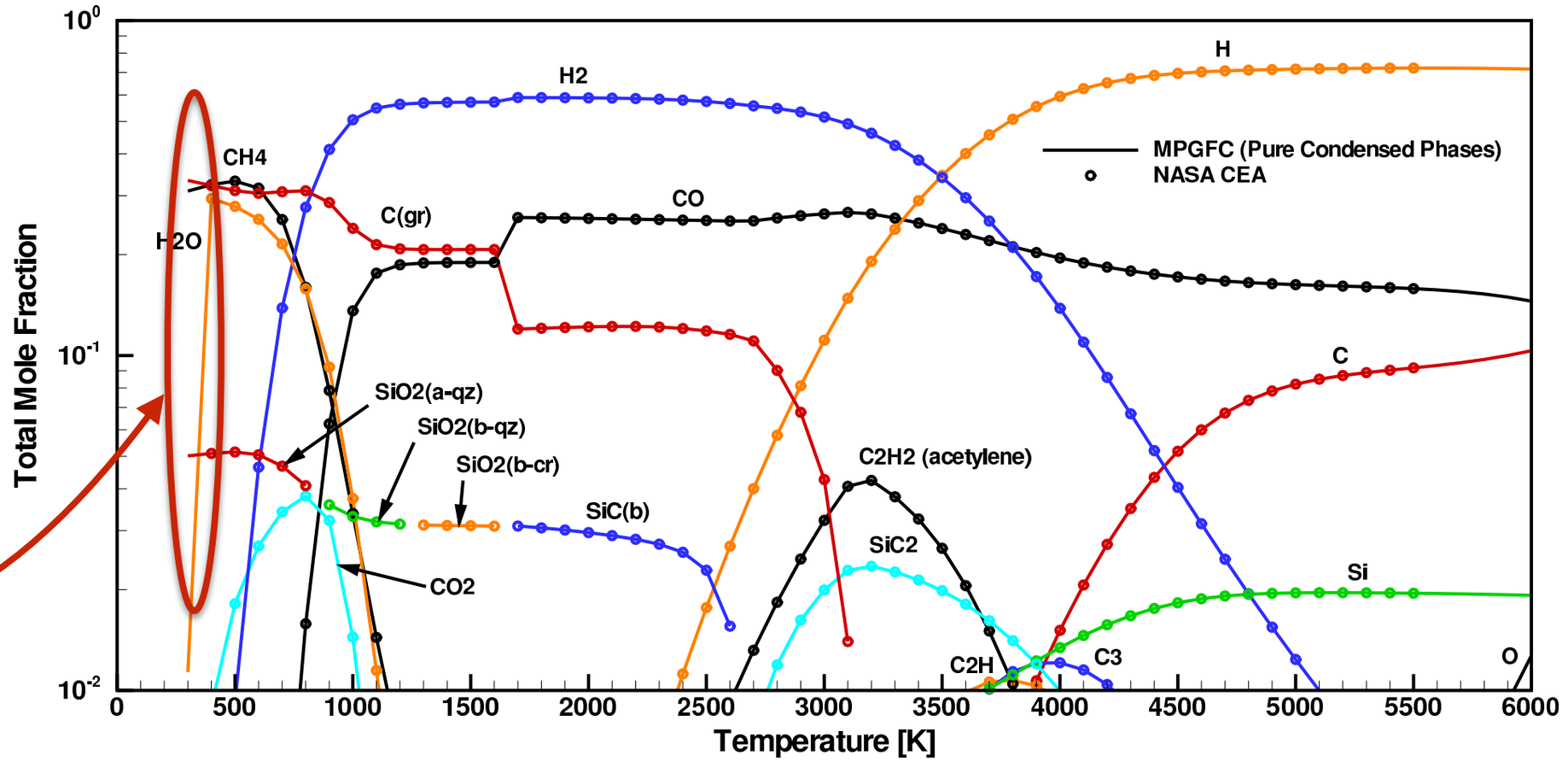
➡ Have developed a new formulation which extends the original GFC method

✓ Multiple phases

✓ General linear constraints (ie: Rate Controlled Chemical Equilibrium)

✓ Surface mass balance (open system) for B' table generation

Conventional Newton methods **can fail** due to numerical stiffness (i.e.: StanJan and CEA) [Bishnu et al., 1997]

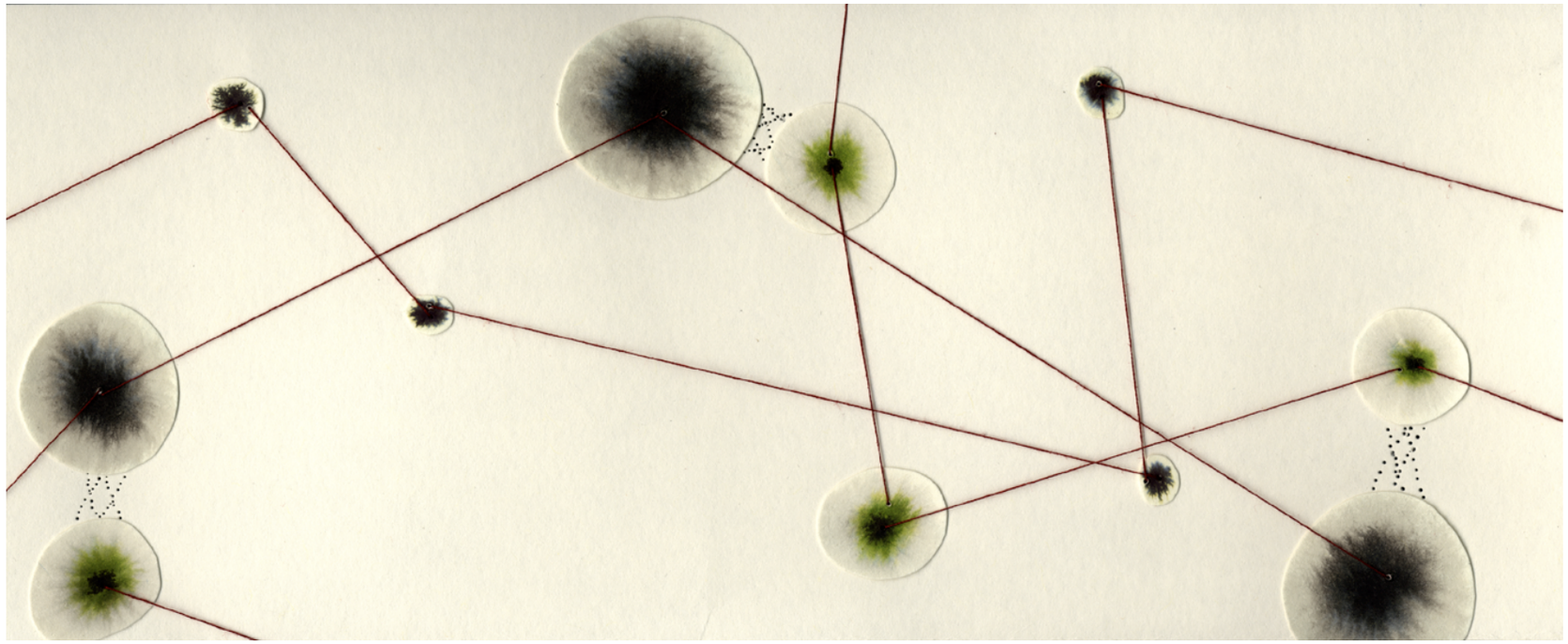


European Research Council

Transport Properties

Closure of the transport fluxes at microscopic scale via solution of the Boltzmann Equation

- Multiscale Chapman-Enskog perturbative solution based on heavy-particle / electron mass disparity [Graille et al., 2009]
- Transport coefficients are then computed using precomputed collision integrals for each collision pair



“Boltzmann Impression”, Losa, Luzern 2004

Transport Properties: Multicomponent

Viscosity or heavy particle thermal conductivity is solution of

$$\begin{pmatrix} G^{\mu} & 0 \\ \beta^{T} & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \mu \end{pmatrix} = \begin{pmatrix} \beta \\ 0 \end{pmatrix}$$

- Conventional solution methods
  - **Determinant** [Hirschfelder, Curtiss & Bird]

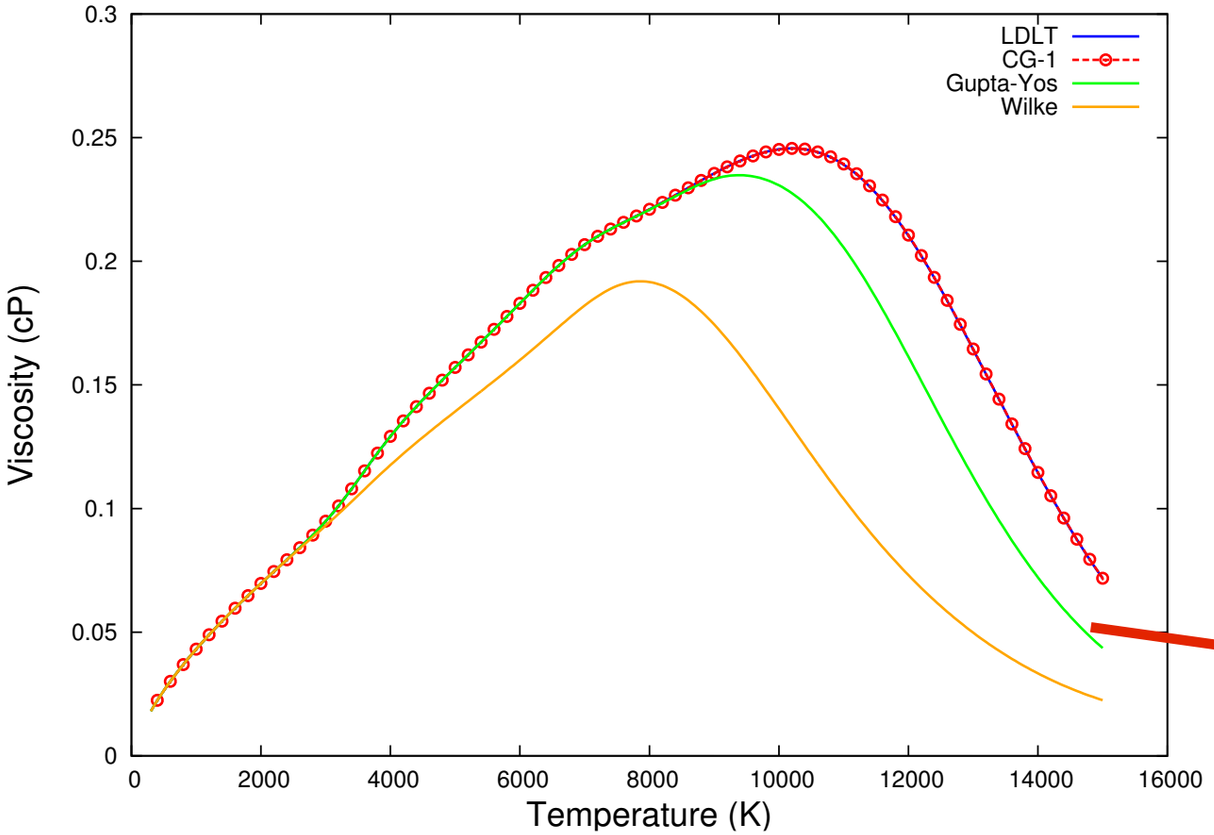
$$\mu = - \begin{vmatrix} G^{\mu} & \beta \\ \beta^{T} & 0 \end{vmatrix} / |G^{\mu}|$$

  - **Mixture rules:** Wilke, Gupta-Yos, Armaly-Sutton, Blottner...

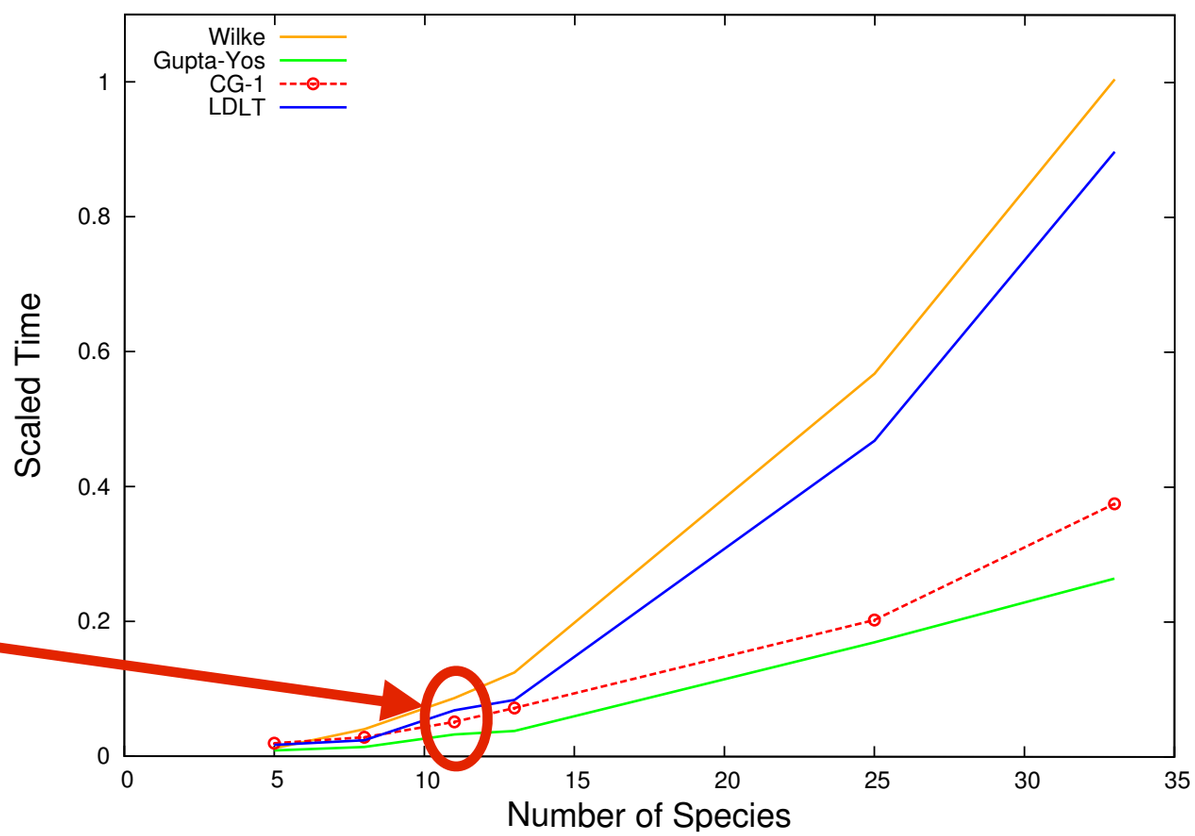
- State-of-the-art methods
  - Symmetric positive definite systems [Ern & Giovangigli]
  - **Direct** solution: LDLT
  - **Iterative** methods: Conjugate-Gradients
  - Iterative methods competitive when iterations less than species / 6



Transport Properties: Accuracy and CPU Cost



Viscosity of equilibrium Air at 1 atm computed using Wilke, Gupta-Yos, CG-1, and LDLT



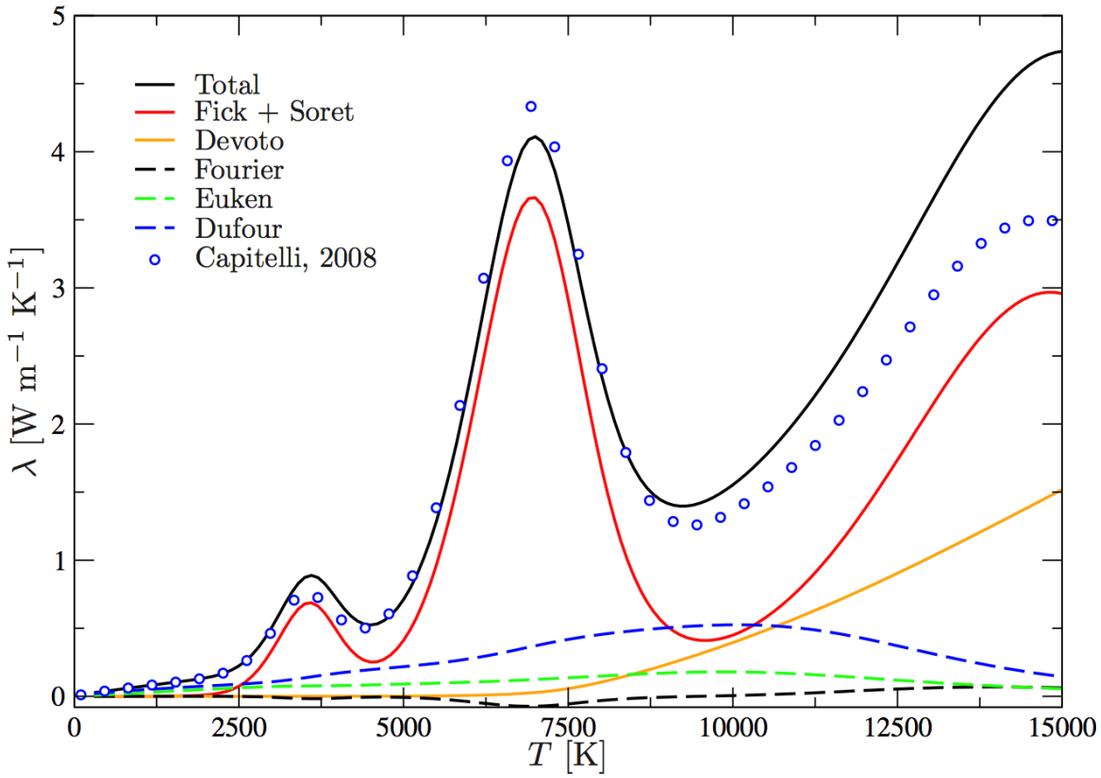
Nondimensionalized CPU time to solve viscosity transport system using Wilke, Gupta-Yos, CG-1, and LDLT

- Gupta-Yos remains accurate below about 9000 K and is fastest overall
- CG-1 is accurate over the whole temperature range, slightly slower than Gupta-Yos

Bottle-neck is still the calculation of transport systems themselves.

Example: Equilibrium Heat Flux of Air

Components of heat flux



$$\vec{q}_h = \sum_{j\in\mathcal{H}} \int \frac{1}{2} m_j |\vec{C}_j|^2 \vec{C}_j f_j^0 \phi_j d\vec{C}_j$$

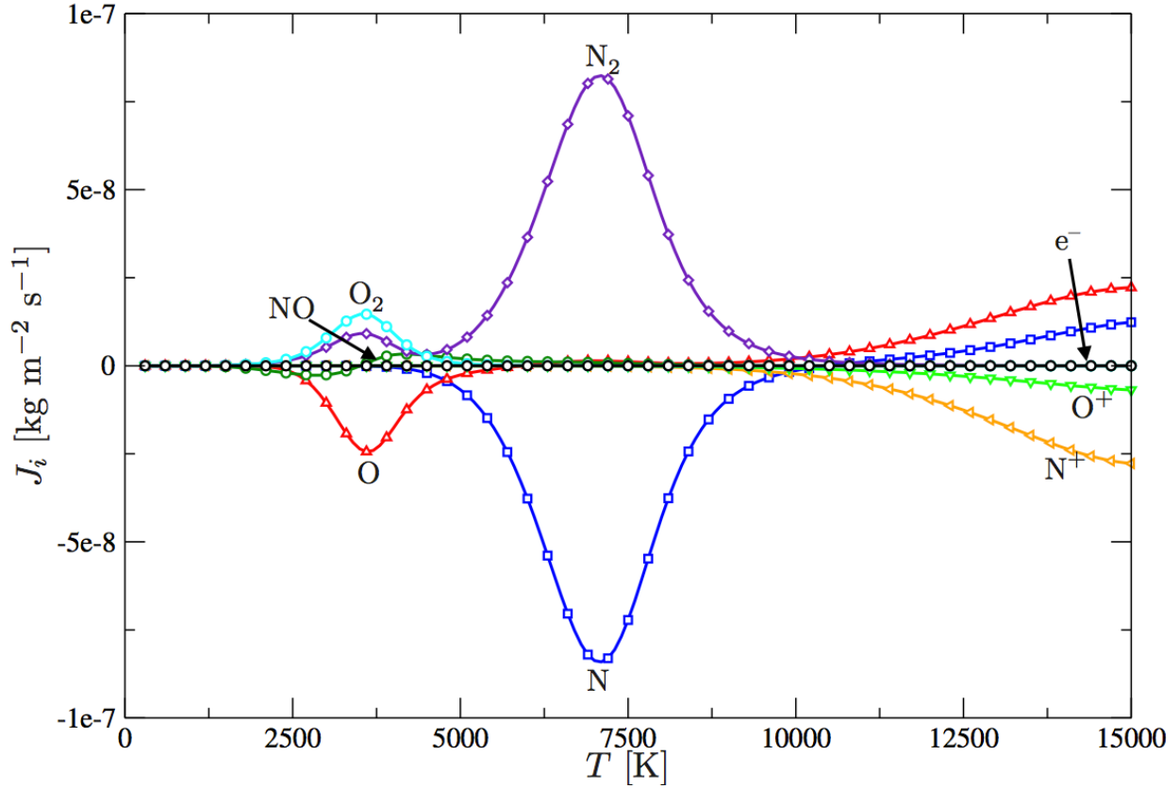
$$= \sum_{j\in\mathcal{H}} \rho_j h_j \vec{V}_j - \lambda_h \nabla T_h + P_h \sum_{j\in\mathcal{H}} \chi_j^h \vec{V}_j$$

$$= -\left(\lambda_D + \lambda_h + \lambda_{TD}\right) \nabla T_h$$

$$\vec{q}_{\text{int}} = -\left(\lambda_R + \lambda_V + \lambda_E\right) \nabla T$$

$$\vec{q}_e = \int \frac{1}{2} m_e |\vec{C}_e|^2 \vec{C}_e f_e \phi_e d\vec{C}_e = -\lambda_e \nabla T_e$$

Diffusion velocities



$$\vec{V}_i = -\sum_{j\in\mathcal{H}} D_{ij} \left(\hat{d}_j + \chi_j^h \nabla \ln T_h\right)$$

$$\sum_{i\in\mathcal{H}} \rho_i \vec{V}_i = 0$$

Thermochemical equilibrium, constant pressure, constant elemental composition

Collision Integral Data

Required Data

Property	Collision Integrals	Number*
η	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(2,2)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
λ_h	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}, \Omega_{ij}^{(1,3)}, \Omega_{ij}^{(2,2)}$	$2n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
λ_{int}	$\Omega_{ij}^{(1,1)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)/2$
$\lambda_e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
$\lambda_e(3)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$a^e(5n^{\mathcal{H}}+3)$
$k_{Ti}^h(2)$	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
$k_{Ti}^e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
$k_{Ti}^e(3)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$a^e(5n^{\mathcal{H}}+3)$
$\vec{V}_i$	$\Omega_{ij}^{(1,1)}, \Omega_{ie}^{(1,1)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)/2 + a^e n^{\mathcal{H}}$
$\sigma_e(1)$	$\Omega_{ie}^{(1,1)}$	$a^e n^{\mathcal{H}}$
$\sigma_e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
all	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}, \Omega_{ij}^{(1,3)}, \Omega_{ij}^{(2,2)}, \Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$2n^{\mathcal{H}}(n^{\mathcal{H}}+1) + a^e(5n^{\mathcal{H}}+3)$

European Research Council

Collision Integral Data

Data Sources

		e-	O+	N+	NO+	N2+	O2+	Ar+	Ar	N	O	NO	N2	O2	C+	CO+	CN+	C	CO	CN	CO2
Earth	e-	-																			
	O+	+	-																		
	N+	+	-	-																	
	NO+	+	-	-	-			300 – 15,000 K													
	N2+	+	-	-	-	-															
	O2+	+	-	-	-	-	-														
	Ar+	+	-	-	-	-	-	-													
	Ar	1	4	4	4	4	4	4	1												
	N	1	3	3	3	3	3	4	1	1											
	O	1	3	3	3	3	3	4	1	1	1										
	NO	1	3	3	3	3	3	4	1	1	1	1									
	N2	1	3	3	3	3	3	4	1	1	1	1	1			300 – 20,000 K					
	O2	1	3	3	3	3	3	4	1	1	1	1	1	1							
+ Mars / Venus	C+	+	-	-	-	-	-	-	L	2	2	2	2	2	-						
	CO+	+	-	-	-	-	-	-	L	2	2	2	2	2	-	-					
	CN+	+	-	-	-	-	-	-	L	2	2	2	2	2	-	-	-				
	C	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2			
	CO	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2	2		
	CN	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2	2	2	
	CO2	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2	2	2	2

Est. Error
< 5%
< 10%
< 15%
< 20%
< 25%
< 30%
< 35%
> 35%
Unknown

+/-	Attractive/Repulsive Shielded Coulomb Potential
1	Wright, 2005, AIAA
2	Wright, 2007, AIAA
3	Levin, 2004, JTHT
4	Wright, 2005, JTHT
LJ	Neutral/Neutral – Leonard-Jones Potential
L	Ion/Neutral – Polarization (Langevin) Potential

Required Data

Property	Collision Integrals	Number*
η	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(2,2)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
λ_h	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}, \Omega_{ij}^{(1,3)}, \Omega_{ij}^{(2,2)}$	$2n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
λ_{int}	$\Omega_{ij}^{(1,1)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)/2$
$\lambda_e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
$\lambda_e(3)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$a^e(5n^{\mathcal{H}}+3)$
$k_{Ti}^h(2)$	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
$k_{Ti}^e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
$k_{Ti}^e(3)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$a^e(5n^{\mathcal{H}}+3)$
$\vec{V}_i$	$\Omega_{ij}^{(1,1)}, \Omega_{ie}^{(1,1)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)/2 + a^e n^{\mathcal{H}}$
$\sigma_e(1)$	$\Omega_{ie}^{(1,1)}$	$a^e n^{\mathcal{H}}$
$\sigma_e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
all	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}, \Omega_{ij}^{(1,3)}, \Omega_{ij}^{(2,2)}, \Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$2n^{\mathcal{H}}(n^{\mathcal{H}}+1) + a^e(5n^{\mathcal{H}}+3)$

European Research Council

Collision Integral Data

Data Sources

		e-	O+	N+	NO+	N2+	O2+	Ar+	Ar	N	O	NO	N2	O2	C+	CO+	CN+	C	CO	CN	CO2
Earth	e-	-																			
	O+	+	-																		
	N+	+	-	-																	
	NO+	+	-	-	-			300 – 15,000 K													
	N2+	+	-	-	-	-															
	O2+	+	-	-	-	-	-														
	Ar+	+	-	-	-	-	-	-													
	Ar	1	4	4	4	4	4	4	1												
	N	1	3	3	3	3	3	4	1	1											
	O	1	3	3	3	3	3	4	1	1	1										
	NO	1	3	3	3	3	3	4	1	1	1	1									
	N2	1	3	3	3	3	3	4	1	1	1	1	1			300 – 20,000 K					
	O2	1	3	3	3	3	3	4	1	1	1	1	1	1							
+ Mars / Venus	C+	+	-	-	-	-	-	-	L	2	2	2	2	2	-						
	CO+	+	-	-	-	-	-	-	L	2	2	2	2	2	-	-					
	CN+	+	-	-	-	-	-	-	L	2	2	2	2	2	-	-	-				
	C	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2			
	CO	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2	2		
	CN	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2	2	2	
	CO2	2	2	2	2	2	2	L	LJ	2	2	2	2	2	2	2	2	2	2	2	2

Est. Error
< 5%
< 10%
< 15%
< 20%
< 25%
< 30%
< 35%
> 35%
Unknown

+/-	Attractive/Repulsive Shielded Coulomb Potential
1	Wright, 2005, AIAA
2	Wright, 2007, AIAA
3	Levin, 2004, JTHT
4	Wright, 2005, JTHT
LJ	Neutral/Neutral – Leonard-Jones Potential
L	Ion/Neutral – Polarization (Langevin) Potential

Database Generation

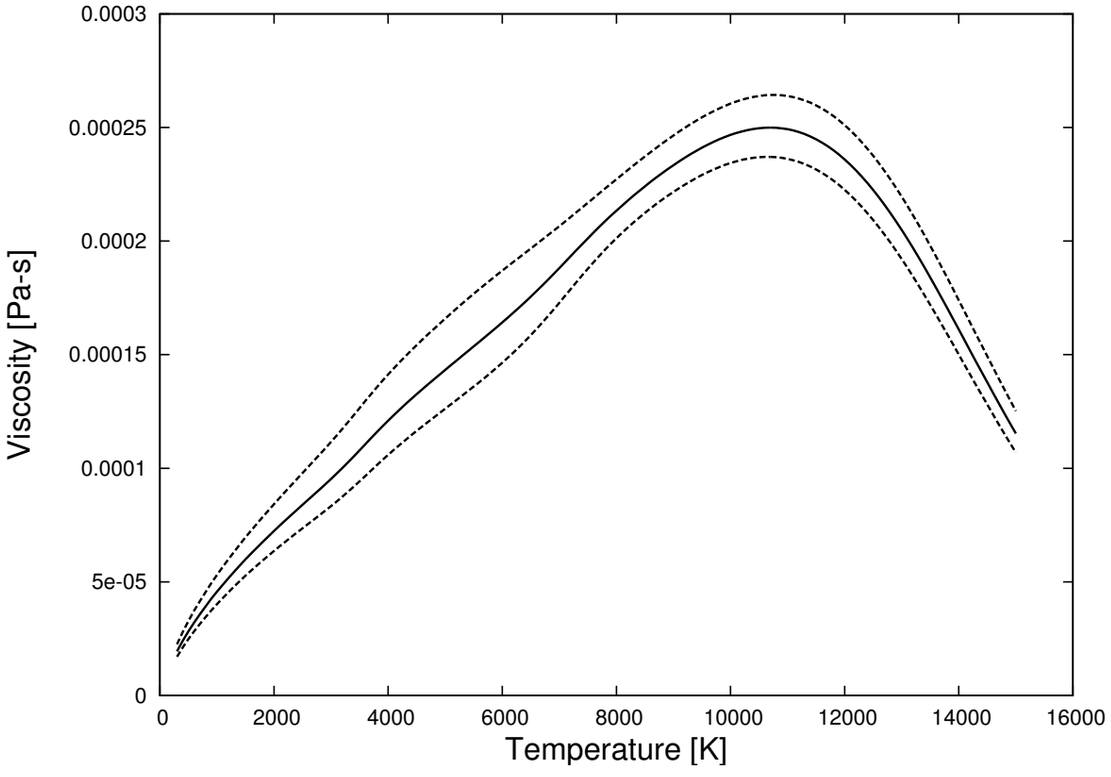
```
<pair sp1="H+" sp2="H">
   <Q11 A="-0.016721" B="0.30197" C="-2.0881" D="8.3358"/>
   <Q12 A="-0.01634" B="0.27935" C="-1.8718" D="7.6733"/>
   <Q13 A="-0.015136" B="0.24217" C="-1.5718" D="6.8795"/>
   <Q14 A="-0.013595" B="0.20051" C="-1.2553" D="6.0894"/>
   <Q15 A="-0.011967" B="0.1592" C="-0.95267" D="5.3594"/>
   <Q22 A="-0.01966" B="0.37756" C="-2.6459" D="9.5171"/>
   <Q23 A="-0.020057" B="0.37523" C="-2.5756" D="9.1975"/>
   <Q24 A="-0.019825" B="0.36032" C="-2.4252" D="8.7286"/>
   <Q33 A="-0.013939" B="0.23967" C="-1.6551" D="7.3167"/>
   <Bstar A="-0.0036323" B="0.082192" C="-0.55691" D="1.3265"/>
</pair>
<pair sp1="H+" sp2="H2">
   <Q11 A="-0.022201" B="0.40038" C="-2.6252" D="8.9468"/>
   <Q12 A="-0.028092" B="0.50051" C="-3.1936" D="9.9347"/>
   <Q13 A="-0.032868" B="0.57751" C="-3.606" D="10.6005"/>
   <Q14 A="-0.037076" B="0.64316" C="-3.9444" D="11.1221"/>
   <Q15 A="-0.04086" B="0.70066" C="-4.2322" D="11.5506"/>
   <Q22 A="-0.024738" B="0.47817" C="-3.2958" D="10.4823"/>
   <Q23 A="-0.031264" B="0.59941" C="-4.0315" D="11.8592"/>
   <Q24 A="-0.034729" B="0.65776" C="-4.3526" D="12.3765"/>
   <Q33 A="-0.019124" B="0.34409" C="-2.3298" D="8.407"/>
   <Bstar A="-0.011798" B="0.27196" C="-1.9614" D="4.612"/>
</pair>
```

Required Data

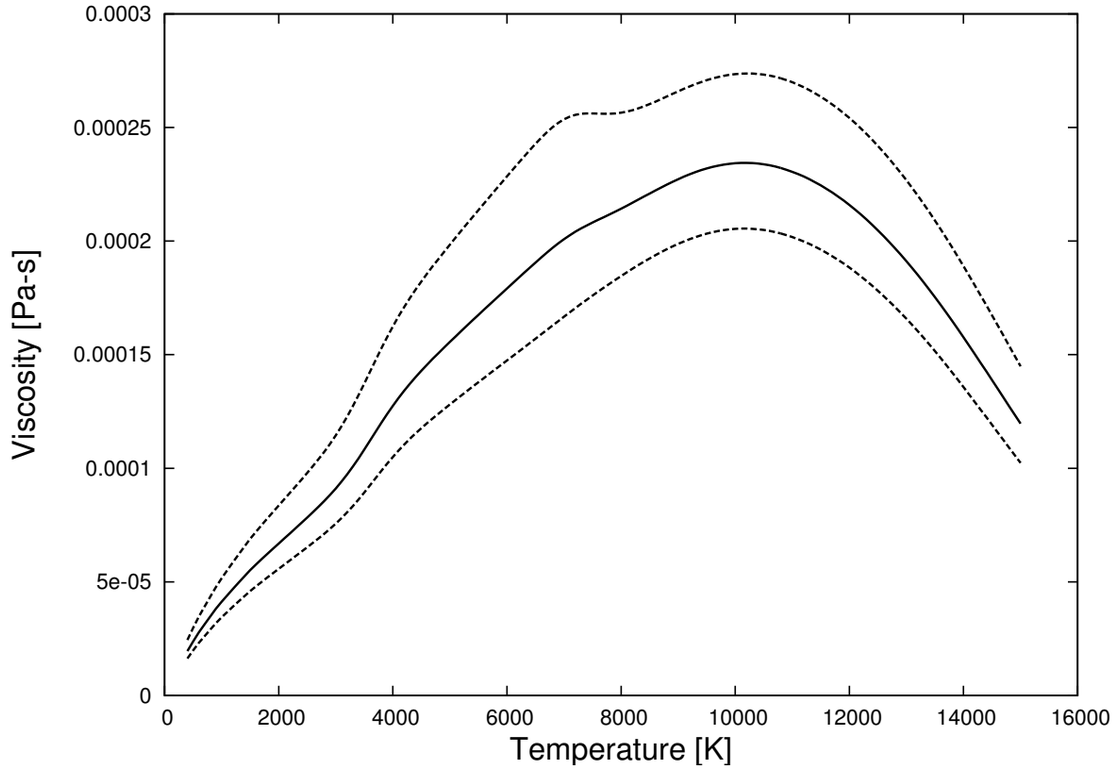
Property	Collision Integrals	Number*
η	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(2,2)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
λ_h	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}, \Omega_{ij}^{(1,3)}, \Omega_{ij}^{(2,2)}$	$2n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
λ_{int}	$\Omega_{ij}^{(1,1)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)/2$
$\lambda_e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
$\lambda_e(3)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$a^e(5n^{\mathcal{H}}+3)$
$k_{Ti}^h(2)$	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)$
$k_{Ti}^e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
$k_{Ti}^e(3)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$a^e(5n^{\mathcal{H}}+3)$
$\vec{V}_i$	$\Omega_{ij}^{(1,1)}, \Omega_{ie}^{(1,1)}$	$n^{\mathcal{H}}(n^{\mathcal{H}}+1)/2 + a^e n^{\mathcal{H}}$
$\sigma_e(1)$	$\Omega_{ie}^{(1,1)}$	$a^e n^{\mathcal{H}}$
$\sigma_e(2)$	$\Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ee}^{(2,2)}$	$a^e(3n^{\mathcal{H}}+1)$
all	$\Omega_{ij}^{(1,1)}, \Omega_{ij}^{(1,2)}, \Omega_{ij}^{(1,3)}, \Omega_{ij}^{(2,2)}, \Omega_{ie}^{(1,1)}, \Omega_{ie}^{(1,2)}, \Omega_{ie}^{(1,3)}, \Omega_{ie}^{(1,4)}, \Omega_{ie}^{(1,5)}, \Omega_{ee}^{(2,2)}, \Omega_{ee}^{(2,3)}, \Omega_{ee}^{(2,4)}$	$2n^{\mathcal{H}}(n^{\mathcal{H}}+1) + a^e(5n^{\mathcal{H}}+3)$

European Research Council

Error Propagation in Transport Properties



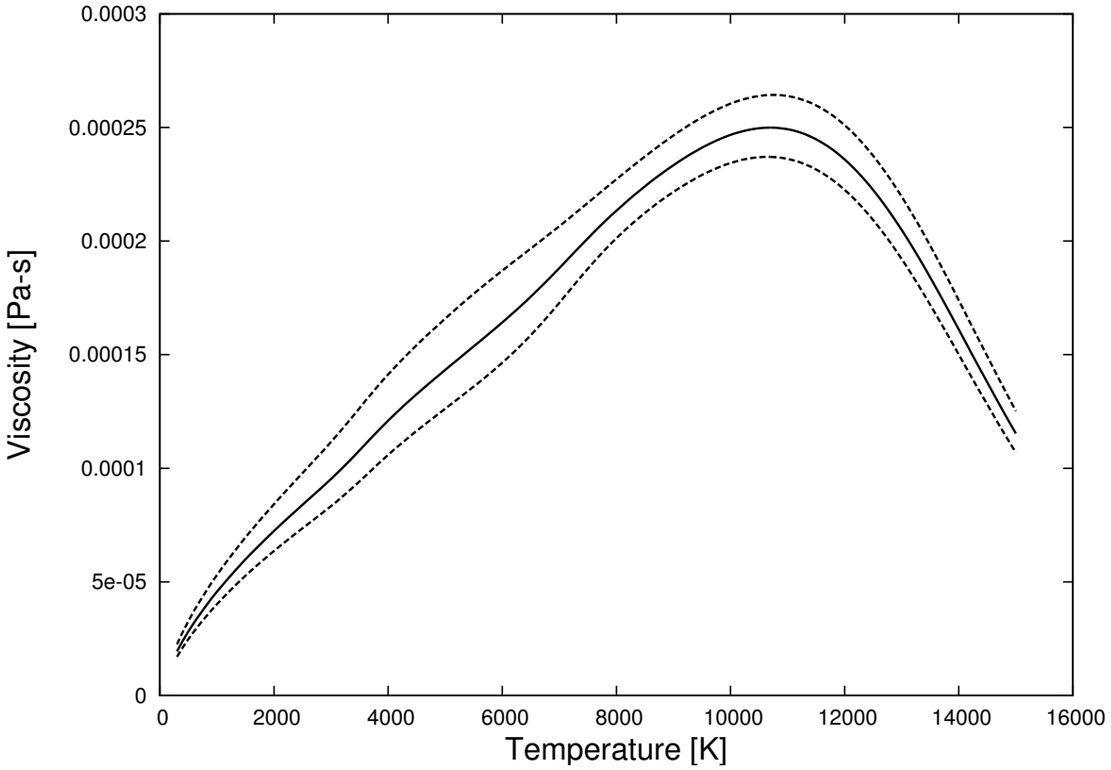
Equilibrium Air

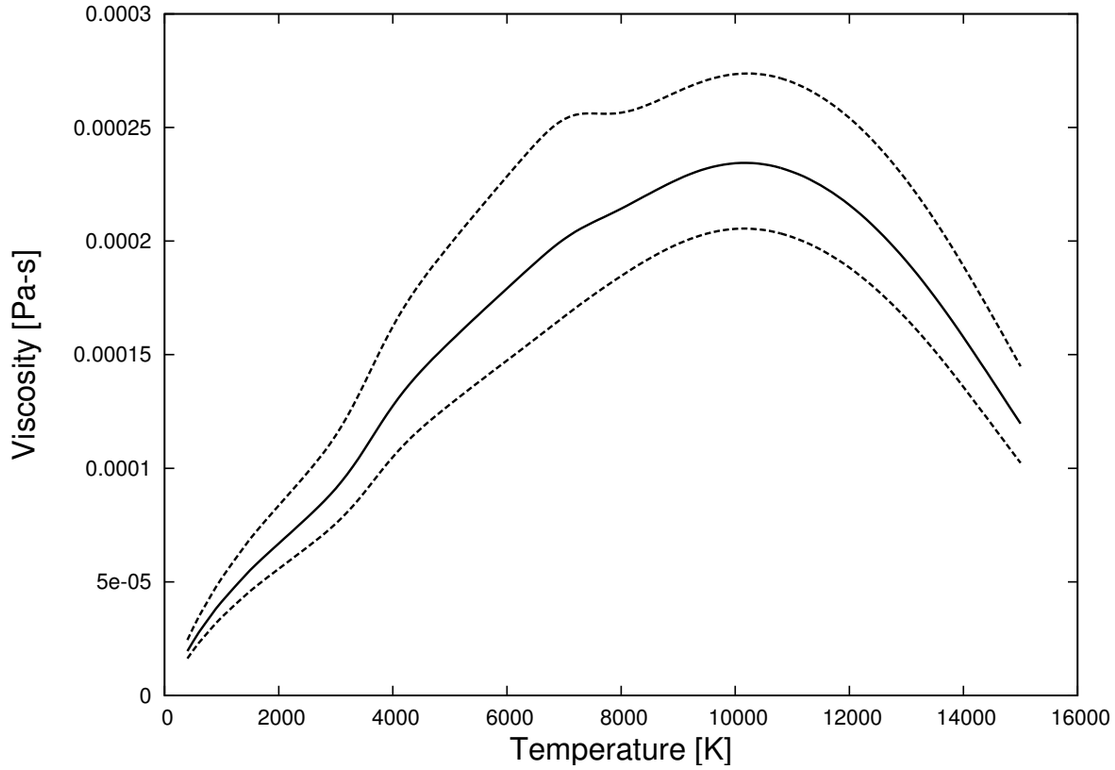


Equilibrium CO_2

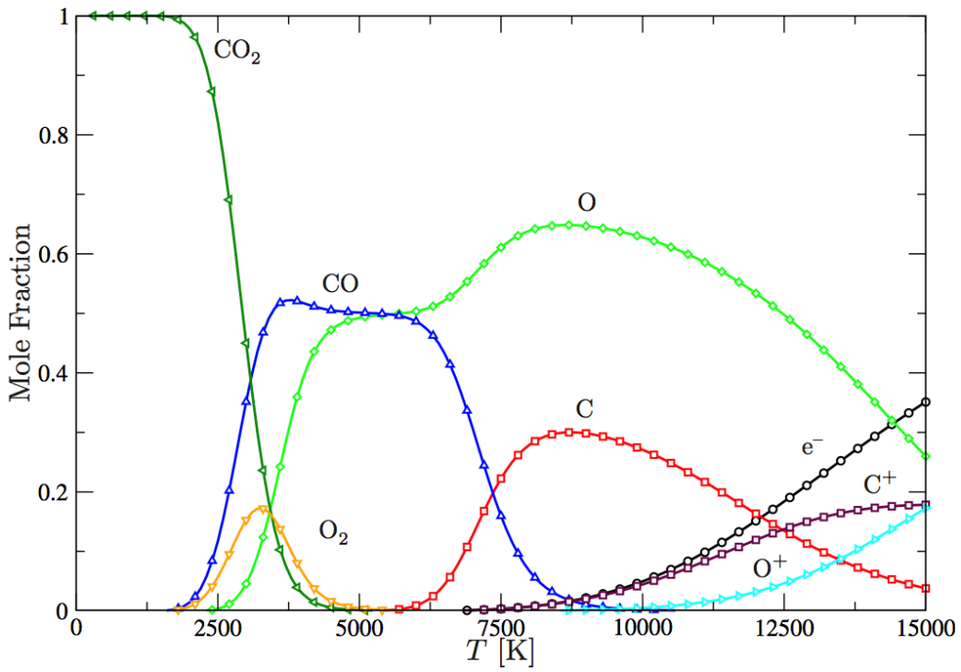
European Research Council

Error Propagation in Transport Properties



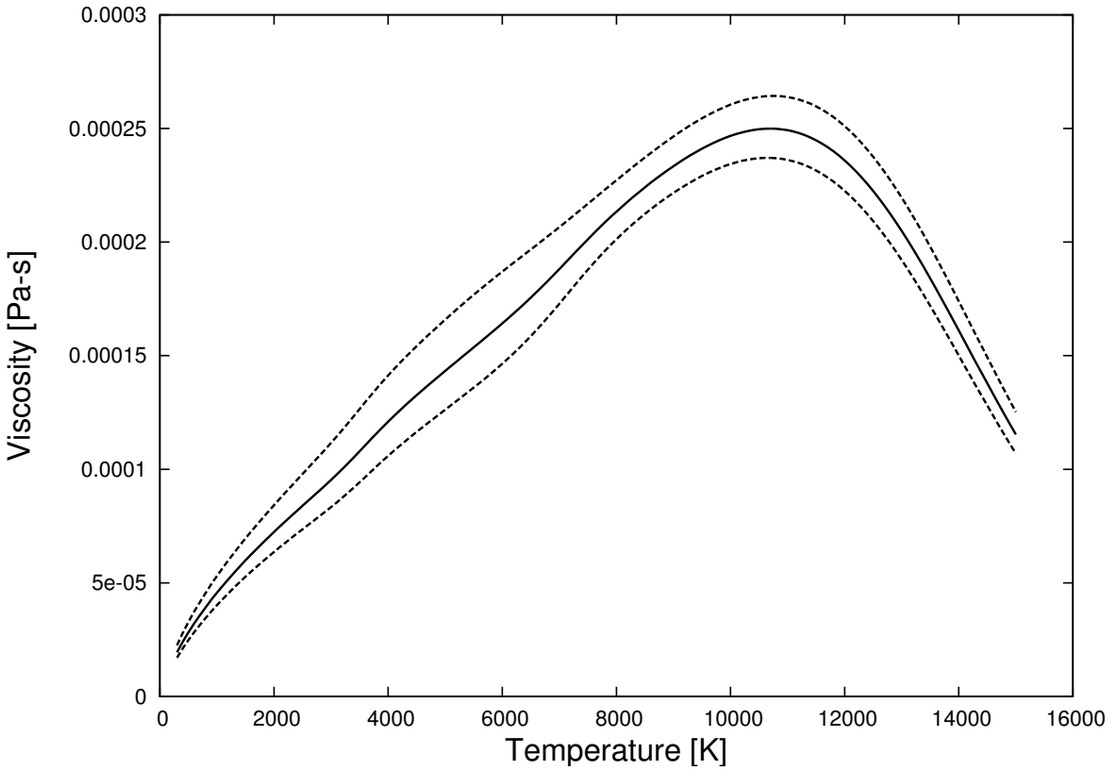


Equilibrium Air

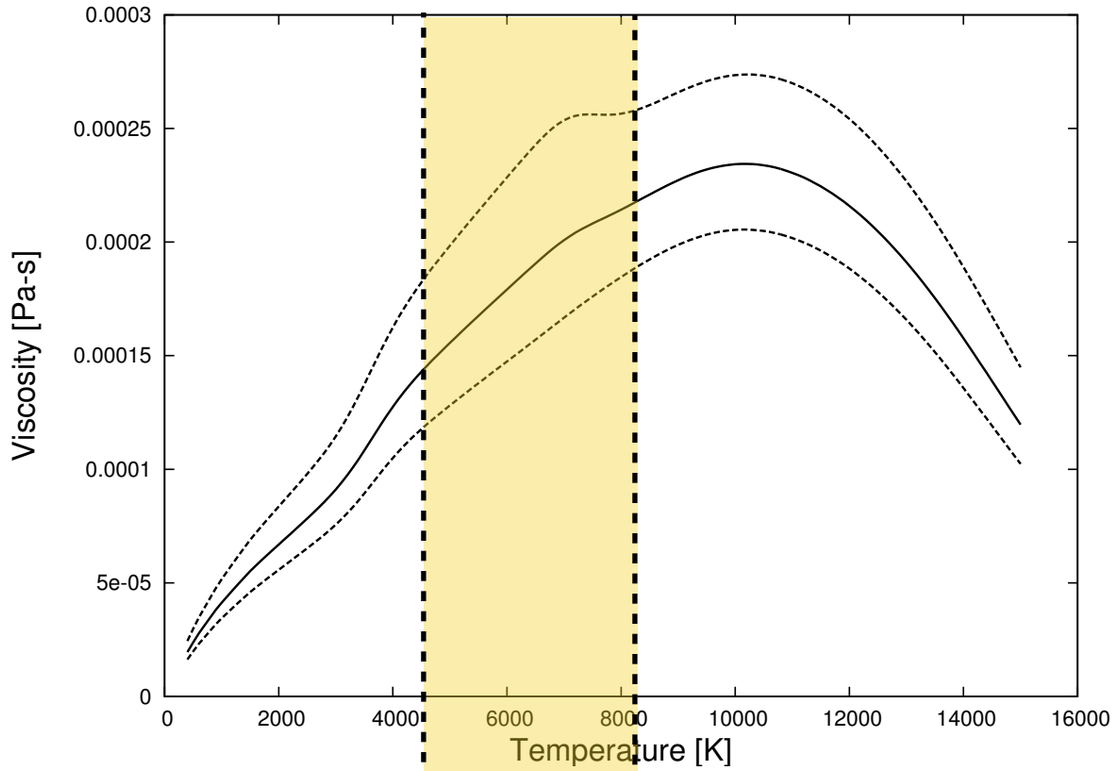


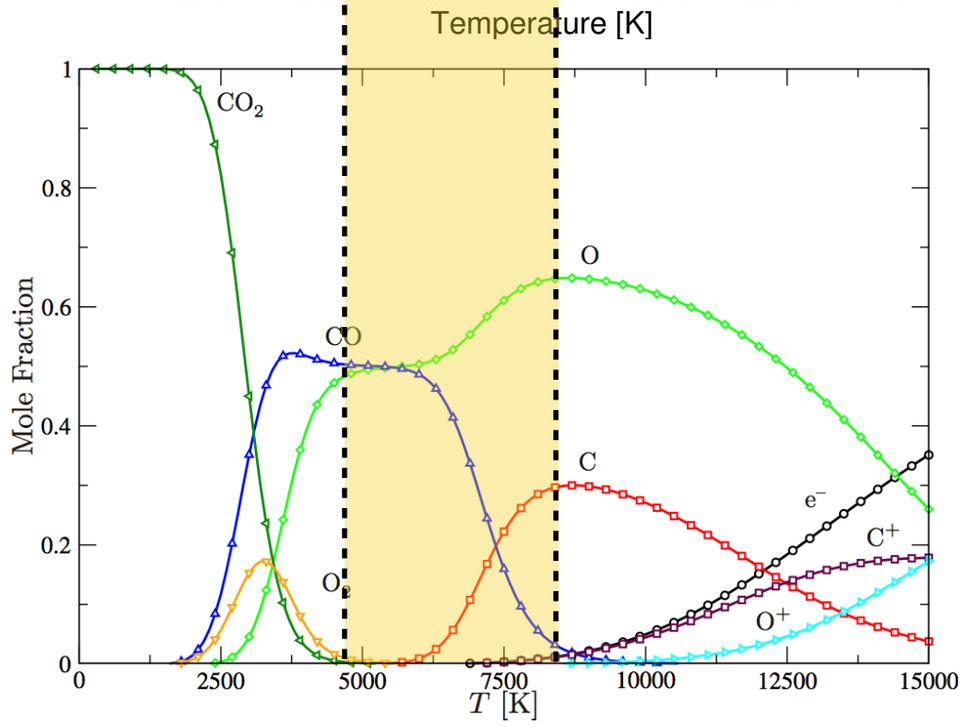
European Research Council

Error Propagation in Transport Properties



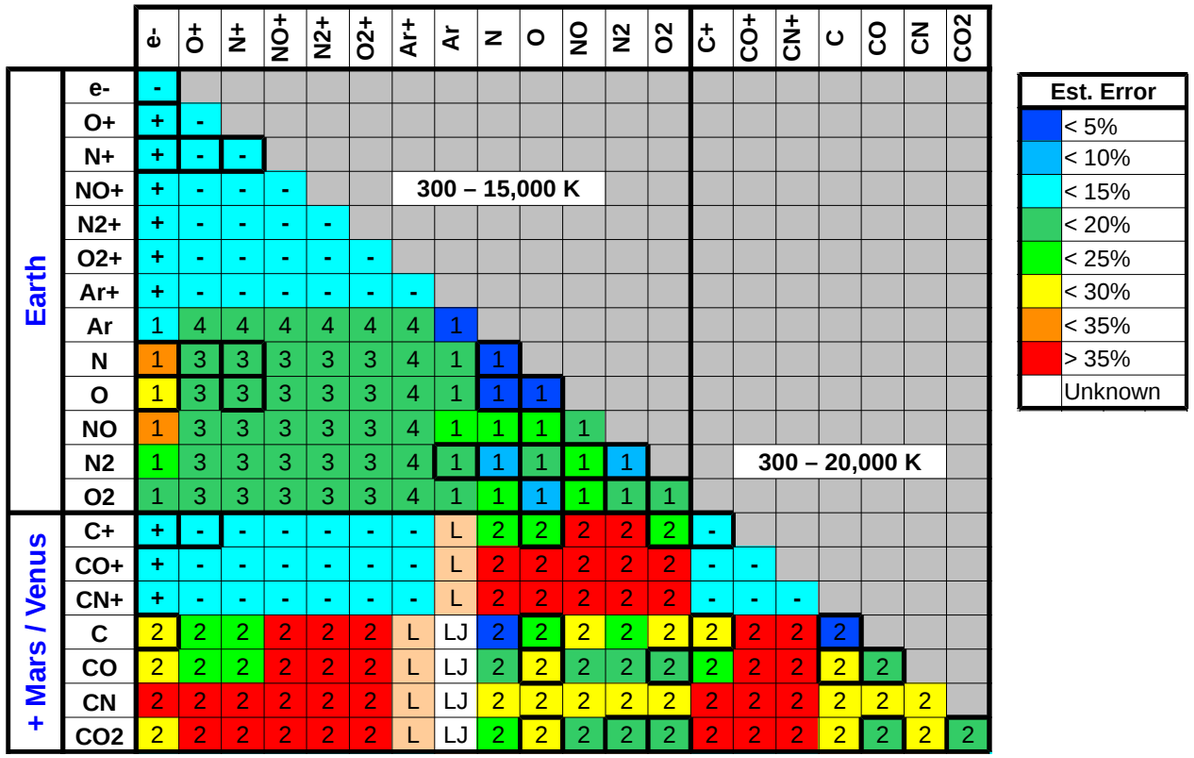
Equilibrium Air





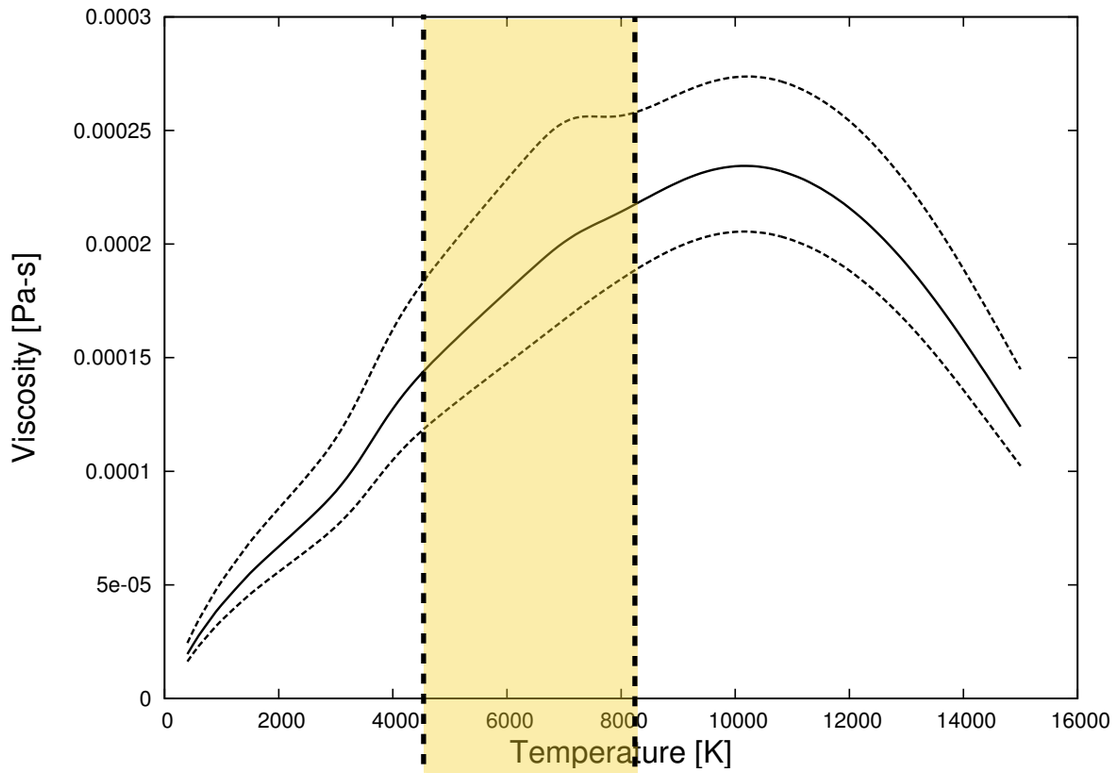
European Research Council

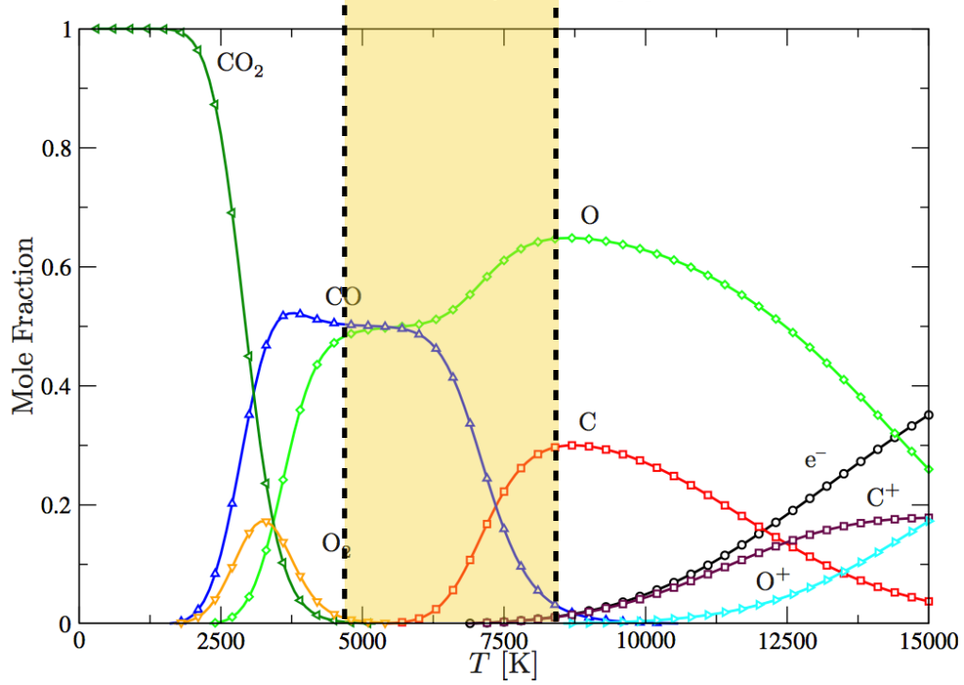
Error Propagation in Transport Properties



Est. Error
< 5%
< 10%
< 15%
< 20%
< 25%
< 30%
< 35%
> 35%
Unknown

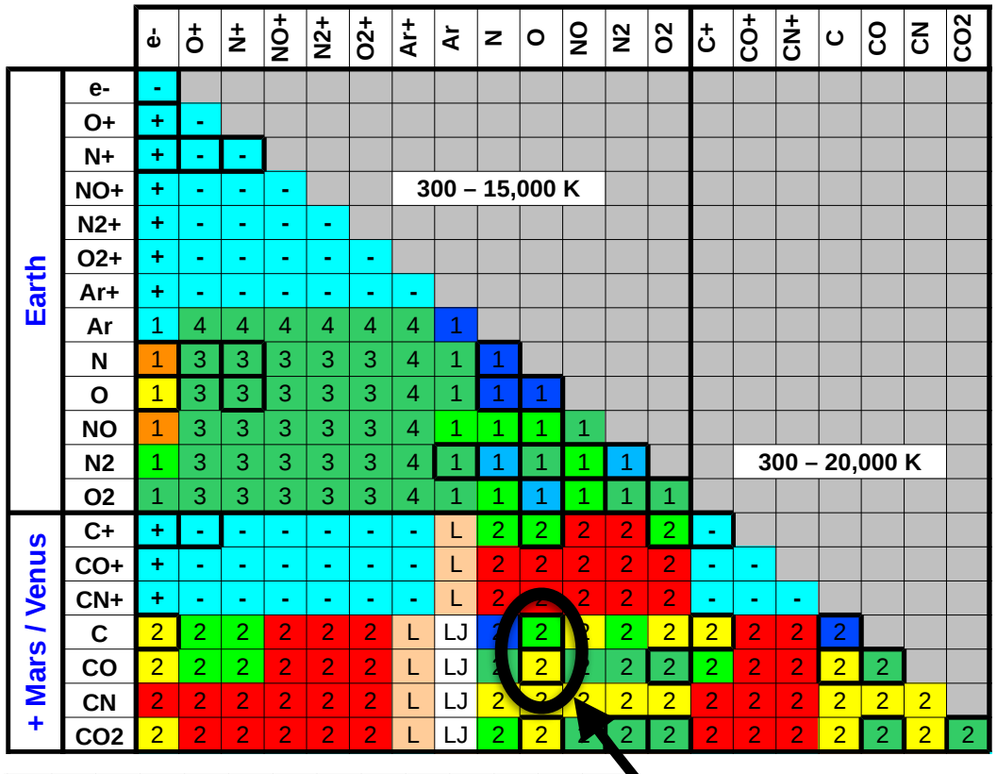
+/-	Attractive/Repulsive Shielded Coulomb Potential
1	Wright, 2005, AIAA
2	Wright, 2007, AIAA
3	Levin, 2004, JTHT
4	Wright, 2005, JTHT
LJ	Neutral/Neutral – Leonard-Jones Potential
L	Ion/Neutral – Polarization (Langevin) Potential





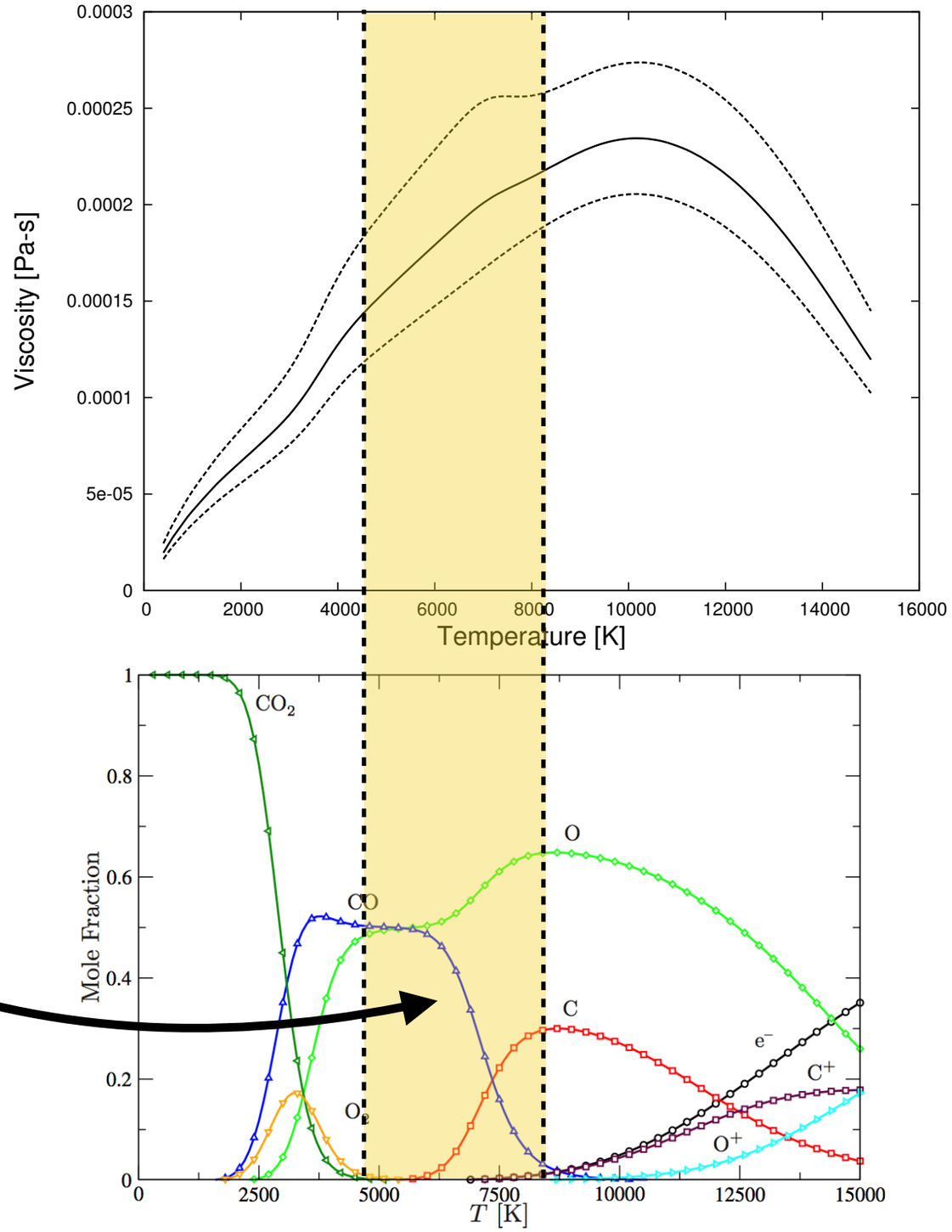
European Research Council

Error Propagation in Transport Properties



Est. Error
< 5%
< 10%
< 15%
< 20%
< 25%
< 30%
< 35%
> 35%
Unknown

+/-	Attractive/Repulsive Shielded Coulomb Potential
1	Wright, 2005, AIAA
2	Wright, 2007, AIAA
3	Levin, 2004, JTHT
4	Wright, 2005, JTHT
LJ	Neutral/Neutral – Leonard-Jones Potential
L	Ion/Neutral – Polarization (Langevin) Potential



Uncertainty in CO-O and C-O integrals account for largest uncertainty in viscosity

European Research Council

Finite-rate Chemistry

Species production rates depend on evaluation of

Law of Mass Action

$$\dot{\omega}_s = \mathrm{M}_{w,s} \sum_{j \in \mathcal{R}} \left(\nu''_{sj} - \nu'_{sj} \right) \left[k_{f,j} \prod_{i \in \mathcal{S}} \tilde{\rho}_i^{\nu'_{ij}} - k_{b,j} \prod_{i \in \mathcal{S}} \tilde{\rho}_i^{\nu''_{ij}} \right] \Theta_j$$

- Reverse reaction rate satisfies equilibrium (second law of thermo)
- Forward-rate and equilibrium constants based on (multiple) temperatures(s) given a rate-law
- Sums and products are evaluated in a sparse matrix form
- Thirdbody reactions treated in compact notation through efficiency factors

Species Jacobians provided automatically

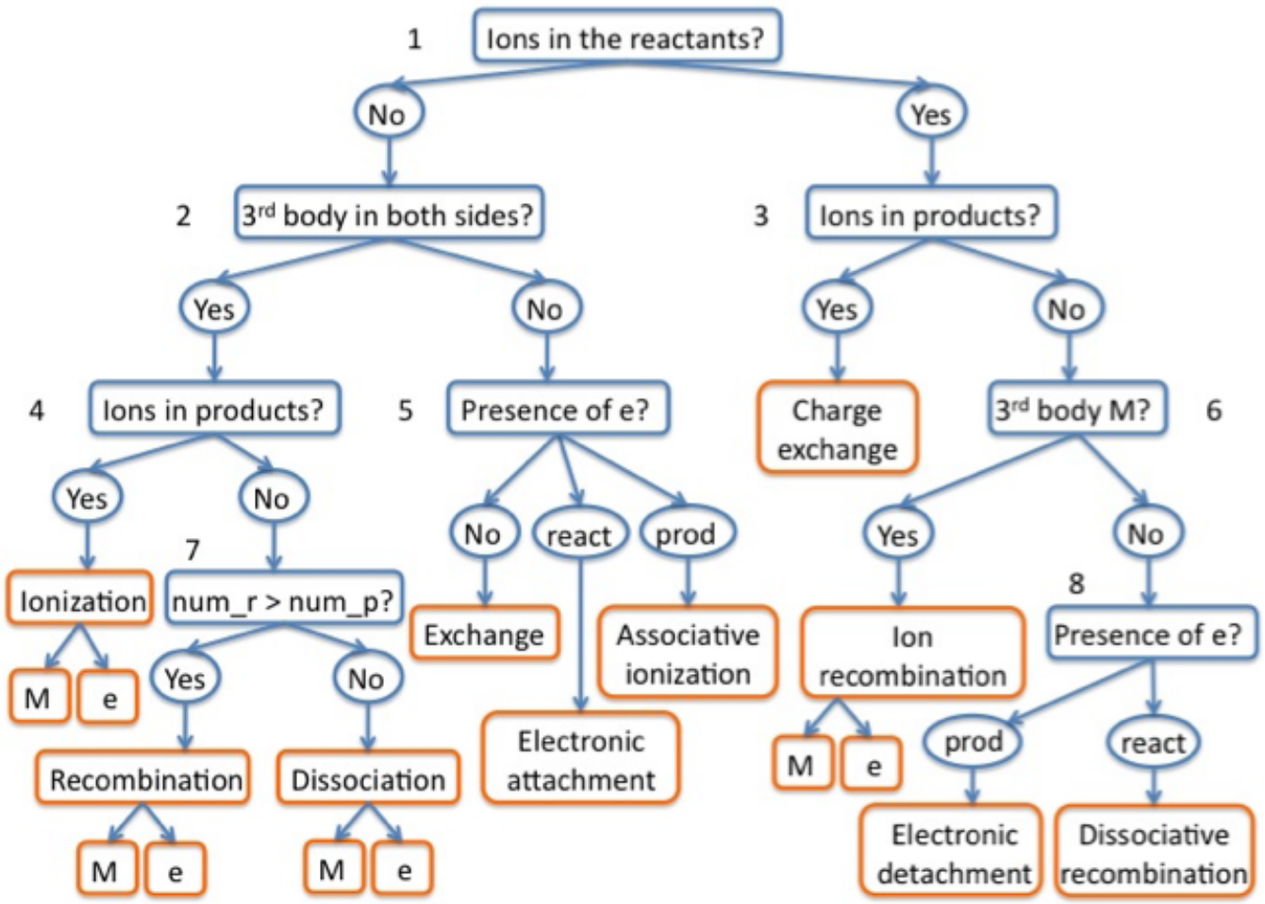
$$\frac{\partial \dot{\omega}_s}{\partial \rho_i}, \frac{\partial \dot{\omega}_s}{\partial T_t} \longrightarrow$$

Requires transformation to user's variables



Classification of Reaction Types

Automatic classification of reactions for selecting temperature dependence on reaction rates



Example logic tree for determining reaction types.
(Credit: Gabriele Lucherini)

Dissociation / Recombination

$$k_f = k_f(\sqrt{TT_v}), k_b = k_b(T)$$
$$k_f = k_f(T_e), k_b = k_b(T_e)$$

Ionization / Ion Recombination

$$k_f = k_f(T), k_b = k_b(T, T_e)$$
$$k_f = k_f(T_e), k_b = k_b(T_e)$$

Exchange (Zeldovich)

$$k_f = k_f(T), k_b = k_b(T)$$

Associative Ionization / Dissociative Recombination

$$k_f = k_f(T), k_b = k_b(T_e)$$

Charge Exchange

$$k_f = k_f(T), k_b = k_b(T)$$

Electron Attachment / Detachment

$$k_f = k_f(T_e), k_b = k_b(T)$$

European Research Council

Reaction Mechanisms

Mixtures given unique names which can be accessed in code

```
<mixture mechanism="air5" thermo_db="RRHO">

    <default_element_fractions>
        N  : 0.79,
        O  : 0.21
    </default_element_fractions>

    <species>
        N O NO N2 O2
    </species>

</mixture>
```

Reaction mechanisms assigned to mixtures

```
<!-- Park Air-5 Reaction Mechanism -->
<mechanism name="air5">

    <arrhenius_units A="mol,cm,s,K" E="kcal,mol,K" />

    <!-- 1 -->
    <reaction formula="N2+M=2N+M">
        <arrhenius A="3.0E+22" n="-1.6" T="113200.0" />
        <M>N2:0.2333, NO:0.2333, O2:0.2333</M>
    </reaction>

    <!-- 2 -->
    <reaction formula="O2+M=2O+M">
        <arrhenius A="1.0E+22" n="-1.5" T="59360.0" />
        <M>N2:0.5, NO:0.5, O2:0.5</M>
    </reaction>

    <!-- 3 -->
    <reaction formula="NO+M=N+O+M">
        <arrhenius A="5.0E15" n="+0.0" T="75500.0" />
        <M>NO:20.0, N:20.0, O:20.0</M>
    </reaction>

    <!-- 4 -->
    <reaction formula="N2+O=NO+N">
        <arrhenius A="5.69E+12" n="+0.42" T="42938.0" />
    </reaction>

    <!-- 5 -->
    <reaction formula="O2+N=NO+O">
        <arrhenius A="2.49E+09" n="+1.18" T="4005.5" />
    </reaction>

</mechanism>
```

Reactions validated at runtime

- ✓ No duplicate reactions
- ✓ Element balance
- ✓ Charge balance
- ✓ Species present in mixture

Lookup Tables

Some model functions are expensive to calculate on the fly yet depend on a single input parameter

Arrhenius Rate Law

$$k_{f,i}(T) = AT^{\beta} \exp\left(\frac{-E_a}{RT}\right)$$

Transport Systems

$$G_{ij}^{\alpha} = x_i x_j \hat{G}_{ij}^{\alpha}(T), \quad i,j \in \mathcal{H}$$

Equilibrium Constant

$$k_{C,i}(T) = \left(\frac{P_{\text{atm}}}{RT}\right)^{-\Delta\nu} \exp\left(-\frac{\Delta G}{RT}\right)$$

Mutation^{++} uses efficient lookup tables to drastically reduce computational cost of expensive functions

- Can load predefined tables, or **generate at run-time** using error constraints
- Custom or predefined interpolation methods (nearest index, linear, exponential)
- Requires careful attention to derivatives when using finite-difference

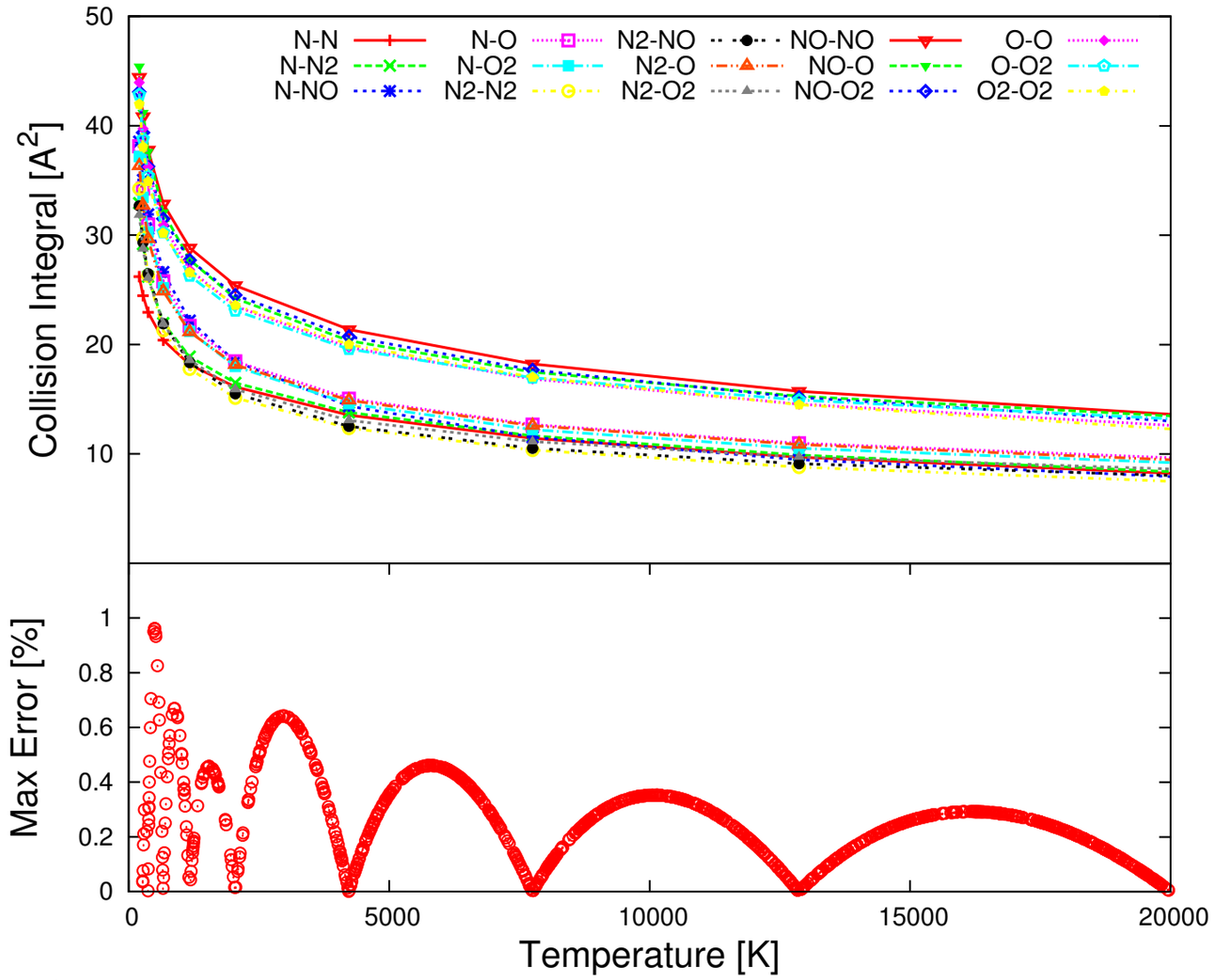




Example Table Creation

Automatic tabulation of binary collision integrals for {N, O, NO, N2, O2} mixture

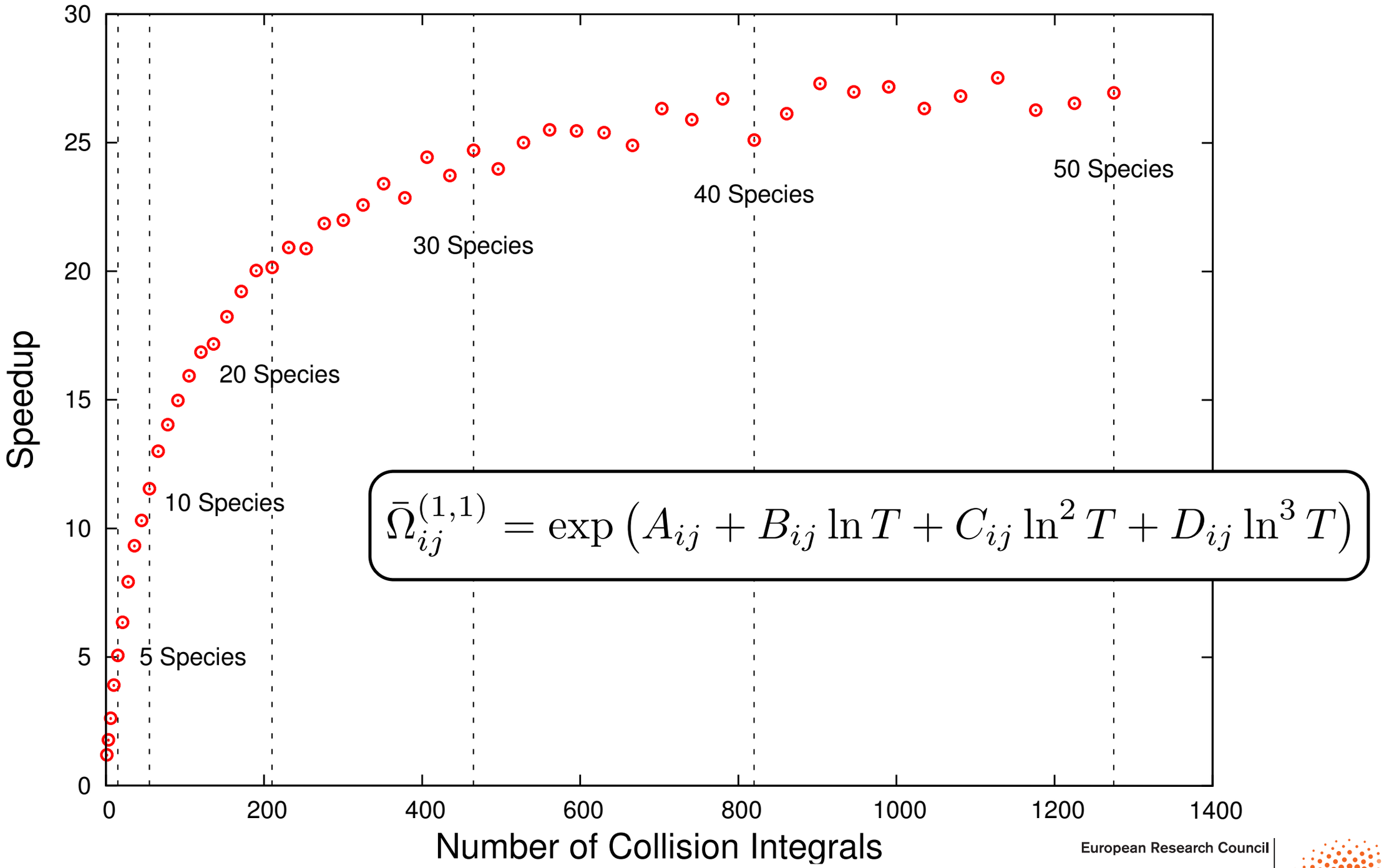
$$\bar{\Omega}_{ij}^{(1,1)} = \exp\left(A_{ij} + B_{ij}\ln T + C_{ij}\ln^2 T + D_{ij}\ln^3 T\right)$$



- Based on maximum error of 1% with linear interpolation
- Points located to maintain error tolerance
- Retains small table size (10 points)
- 5X speed up over optimal direct calculation
- Efficiency increases with number of functions

European Research Council

Lookup Tables: Performance



European Research Council

Doxygen provides automatic creation of reference manual

Latex equations, figures & tables
Class dependency graphs

Example code snippets
Todo lists and internal notes for developers

```
/**
 * Interpolates table data between two table entries nearest a search index.
 *
 * The two nearest data points to the index value in the table are first
 * found using one of two methods. If the change in index values between
 * each neighboring row is constant for the entire table, then a simple hash
 * function is used to compute the rows nearest the lookup index directly.
 * If however the change in index values between data entries varies, a
 * binary search algorithm is employed to find the nearest rows in O(log(n))
 * time where n is the number of rows in the table.
 *
 * Once the two nearest rows are determined, an interpolation scheme is
 * employed to compute the table values at the lookup index. Currently,
 * there are three schemes implemented.
 *
 * - Nearest Index: Chooses the table entry closest to the lookup index.
 * - Linear:
 *    \f[ y = y_1 + \frac{x - x_1}{x_2 - x_1} (y_2 - y_1) \f]
 * - Exponential:
 *    \f[
 *       y = \exp(\ln y_1 + \frac{x - x_1}{x_2 - x_1} (\ln y_2 - \ln y_1))
 *    \f]
 *
 * @param index   the independent value index.
 * @param p_values pointer to T array that will hold function values on
 *                 return.
 * @param scheme  the interpolation scheme to use between table values.
 *
 * @see InterpolationScheme
 */
void lookup(const IndexType &index, DataType *const p_values,
            const InterpolationScheme scheme = NEAREST_INDEX) const;
```



mutation++ 1.0

Main Page | Related Pages | Namespaces | Classes | Files | Search

Class List | Class Hierarchy | Class Members

Public Member Functions | Private Member Functions | Private Attributes

LookupTable< IndexType, DataType > Class Template Reference

```
template<typename IndexType, typename DataType>
void LookupTable< IndexType, DataType >::lookup ( const IndexType &      index,
                                                  DataType *const        p_values,
                                                  const InterpolationScheme scheme = NEAREST_INDEX
                                                ) const
```

Interpolates table data between two table entries nearest a search index.

The two nearest data points to the index value in the table are first found using one of two methods. If the change in index values between each neighboring row is constant for the entire table, then a simple hash function is used to compute the rows nearest the lookup index directly. If however the change in index values between data entries varies, a binary search algorithm is employed to find the nearest rows in O(log(n)) time where n is the number of rows in the table.

Once the two nearest rows are determined, an interpolation scheme is employeed to compute the table values at the lookup index. Currently, there are three schemes implemented.

- Nearest Index: Chooses the table entry closest to the lookup index.
- Linear:

$$y = y_1 + \frac{x - x_1}{x_2 - x_1}(y_2 - y_1)$$

- Exponential:

$$y = \exp(\ln y_1 + \frac{x - x_1}{x_2 - x_1}(\ln y_2 - \ln y_1))$$

Parameters:

index	the independent value index.
p_values	pointer to T array that will hold function values on return.
scheme	the interpolation scheme to use between table values.

See also:
InterpolationScheme

Concluding Remarks

- Mutation++ is being developed to abstract away physico-chemical modeling from user CFD codes
- Provides a unique framework for model development and testing in hypersonics similar to tools used in the combustion community such as Cantera and Chemkin
- Simple to maintain physics package allows multiple CFD codes to quickly inherit changes in model implementation & physical constants and minimizes code duplication and effort when adding new models
- Acts as a user multiplier which increases testing of core modeling
- Source code and published data will be accessible for the first public release around October
- Fortran interface is available as well

Future Work

- Transport properties for magnetized plasmas where the magnetic field induces anisotropic fluxes due to electron gyration around magnetic lines
- Incorporate CTest/CDash framework which supplies regression testing
- Surface catalysis/ablation module
- Extension of B' solver for arbitrary ideal solid phases (only C is available now)
- Radiation module utilizing the HTGR database from EM2C, Ecole Central Paris will provide atomic spectra and Statistical Narrow Band parameters for molecular systems

Applied Modeling & Simulation (AMS) Seminar Series

Development of Mutation^{++}: Multicomponent Thermodynamic and Transport Properties for Ionized Plasmas written in C++

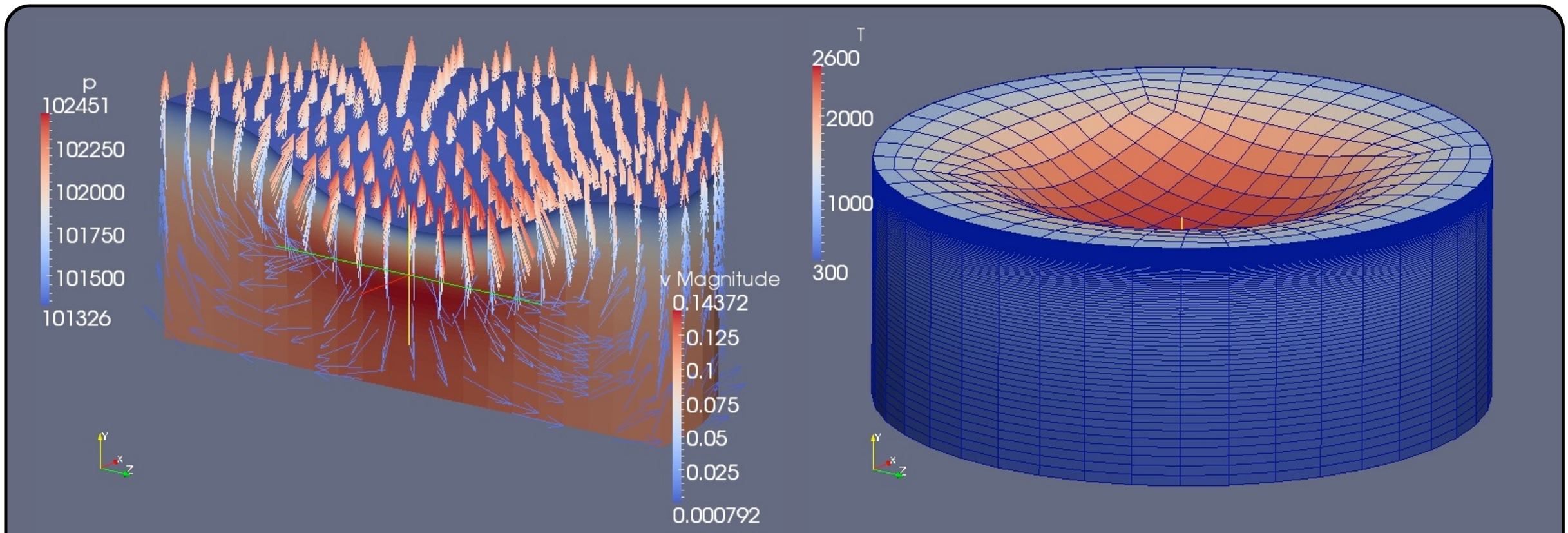
J.B. Scoggins[1,2], Thierry Magin[1,2]

[1]von Karman Institute for Fluid Dynamics, Aerospace and Aeronautics Dept.

[2]Ecole Centrale Paris, Laboratoire EM2C

European Research Council

26 June 2014, NASA Ames Research Center



Results obtained by NASA ARC ablation code PATO coupled with Mutation^{++} for a Nylon cylinder with an impinging jet.
(Credit: Jean Lachaud, University of California, Santa Cruz)