



POLITECNICO DI MILANO

Department of Chemistry, Materials, and Chemical Engineering "G. Natta"

Numerical modeling of reacting flows with detailed kinetic mechanisms: challenges and perspectives

Alberto Cuoci

Combustion Fundamentals and New Technologies

*Conservatorio delle Orfane a Terra Murata
Isola di Procida, Napoli, Italy
May 31 – June 5, 2015*





Outline

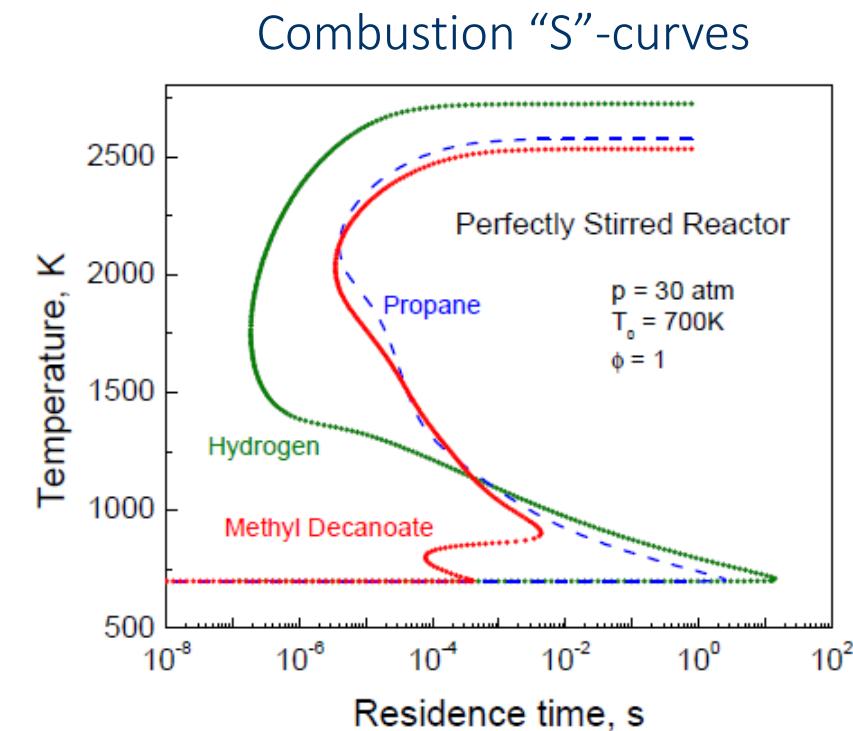
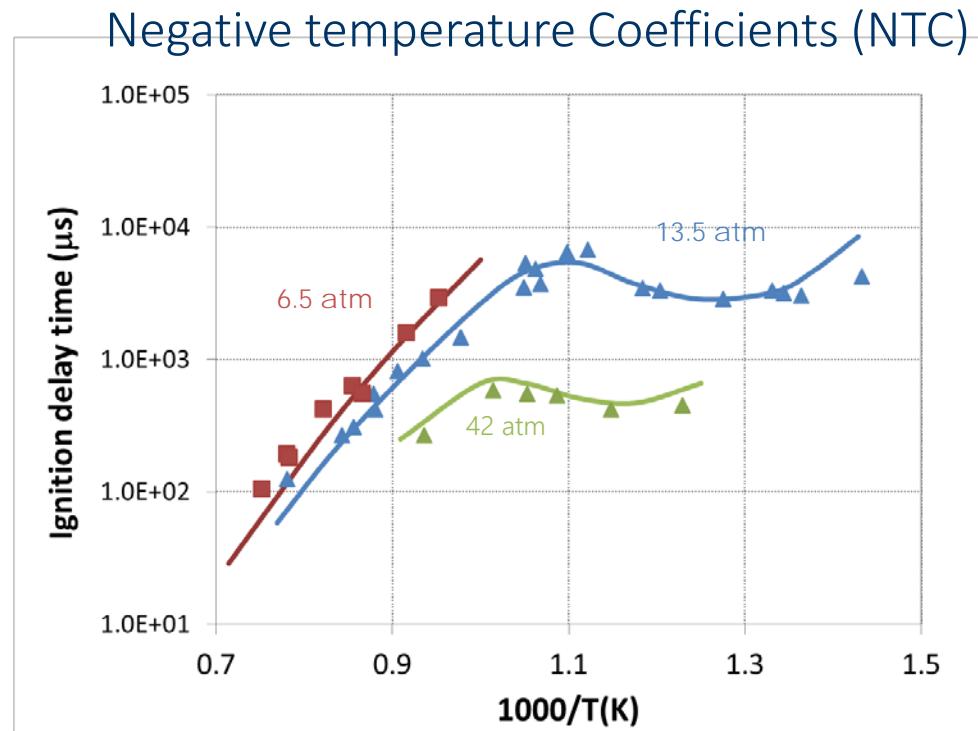
1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks





Need of detailed kinetic mechanisms (I)

Detailed chemistry is important for: ignition, extinction, instabilities ...



Experimental data from:

Ciezki H.K. and Adomeit G., *Shock-tube investigation of self-ignition of n-heptane-air mixtures under engine relevant conditions*, Combustion and Flame 93 p. 421–433 (1993)

Plot from:

Lu T., *Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics*, Princeton-CEFRC Summer School on Combustion (2012)





Need of detailed kinetic mechanisms (II)



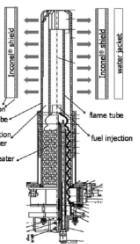
Real fuels and surrogate mixtures

need of modeling synergistic effects between the different components



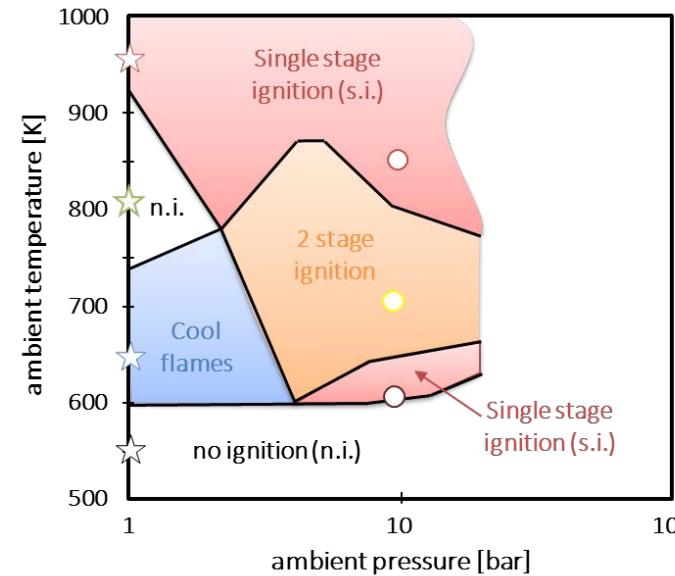
Biofuels

bioalcohols, biodiesel, green diesel, bioethers



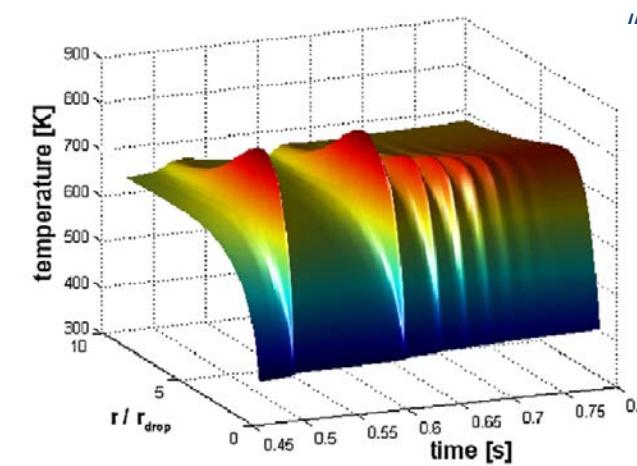
Flameless combustion

(low Damkholer number, slow chemistry)



Auto-ignition regimes of n-alkanes droplets

Tanabe et al., 26th Symposium (International) on Combustion, p. 1637-1643 (1996)

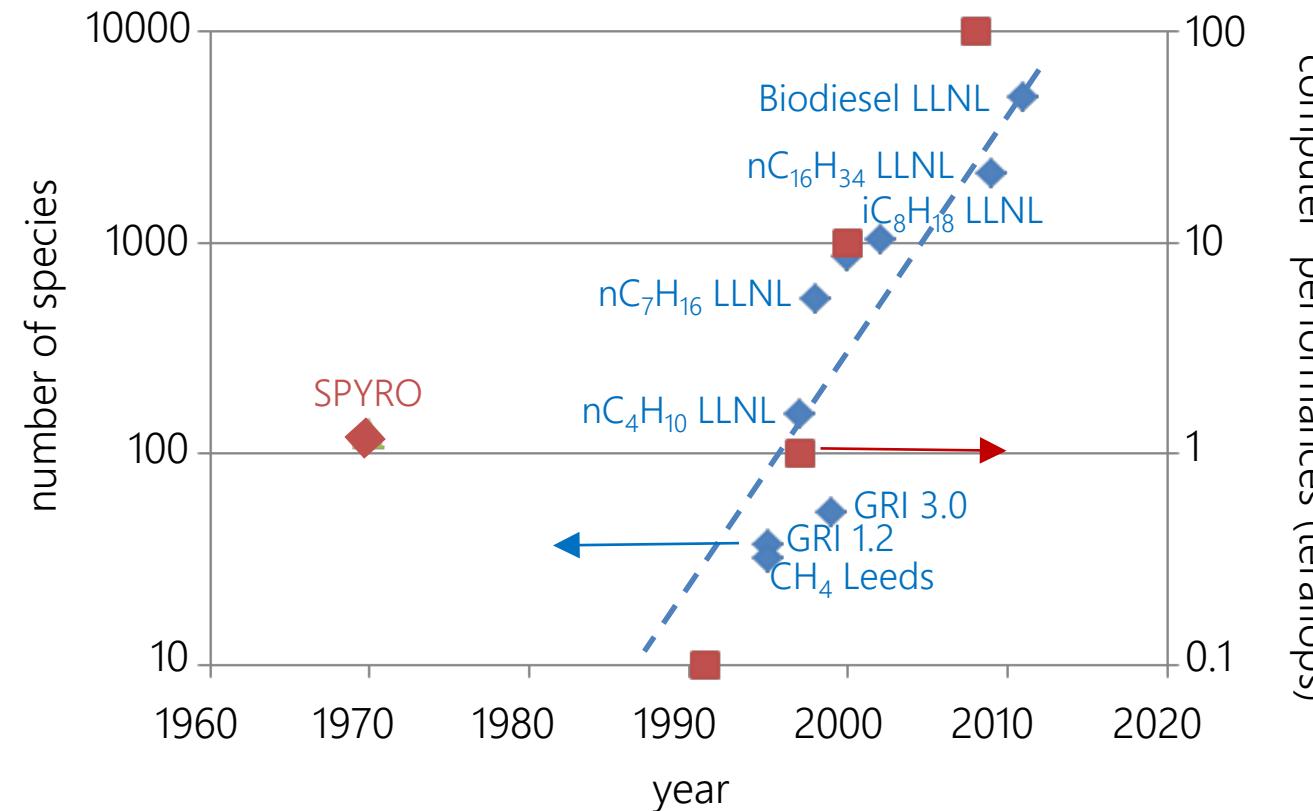


"Anomalous" evaporation of nC10 droplets

Cuoci A. et al., Proceedings of The Combustion Institute, 2015



Mechanisms and computer performances



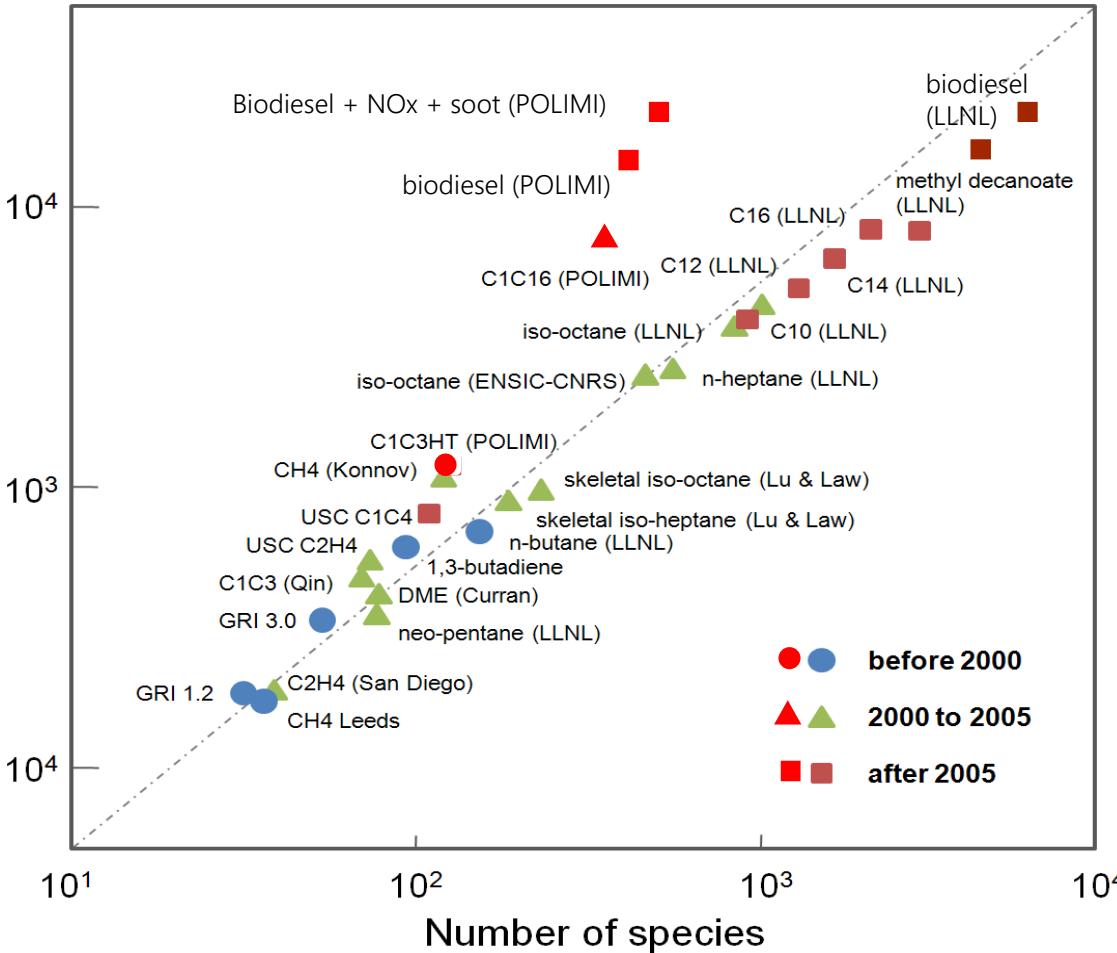
Detailed kinetic mechanisms include many species: the resulting equation systems are very large

Adapted from:

T. Faravelli, *Numerical Modeling of Pollutant Emissions with Detailed Kinetics: from Ideal Reactors to Flames*, Invited Lecture at 14th ICNC 2013, San Antonio (TX)

Mechanism size

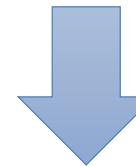
Number of reactions



Adapted from: T.F. Lu, C.K. Law, *Toward accommodating realistic fuel chemistry in large-scale computations*, Progress in Energy and Combustion Science, 35, p. 192–215 (2009)

increasing effort to incorporate more complex reaction mechanisms in simulation of combustion processes and this has led to the development of reaction mechanisms with different levels of detail and comprehensiveness

computational cost associated with such mechanisms is usually very high



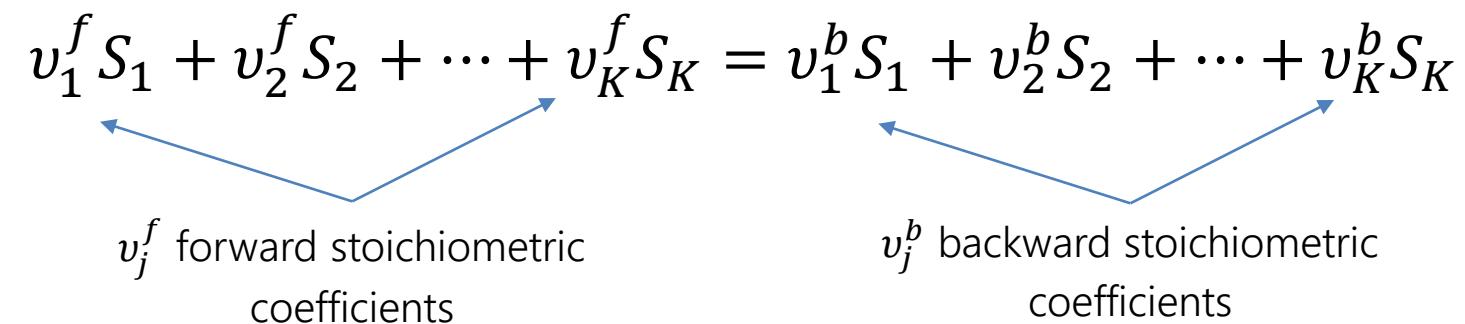
need of **numerical techniques** and **computational tools** to make:

- the use of large kinetic schemes computationally efficient
- easy their integration in new and/or existing numerical codes



Non-linearity and sparsity

A reaction in general form:



The reaction rates are strongly non-linear!

$$\dot{\Omega}_f = k_f(T) \prod_j C_j^{v_j^f}$$

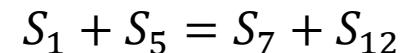
$$k_f(T) = AT^n \exp\left(-\frac{E}{RT}\right)$$

$$\dot{\Omega}_r = k_r(T) \prod_j C_j^{v_j^b}$$

$$k_r(T) = \frac{k_f(T)}{K_{eq}(T)}$$

Detailed chemistry is very sparse!

An elementary reaction only involve a few species
(usually not more than 4)



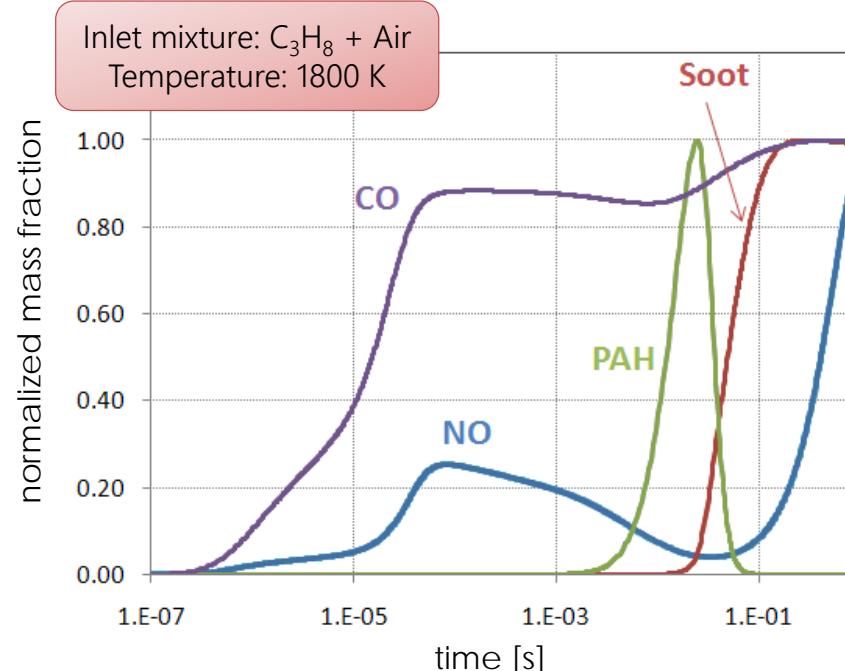
Directly coupled species increases linearly with the mechanism size

Adapted from:

Lu T., *Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics*, Princeton-CEFRC Summer School on Combustion (2012)



Detailed chemistry is stiff

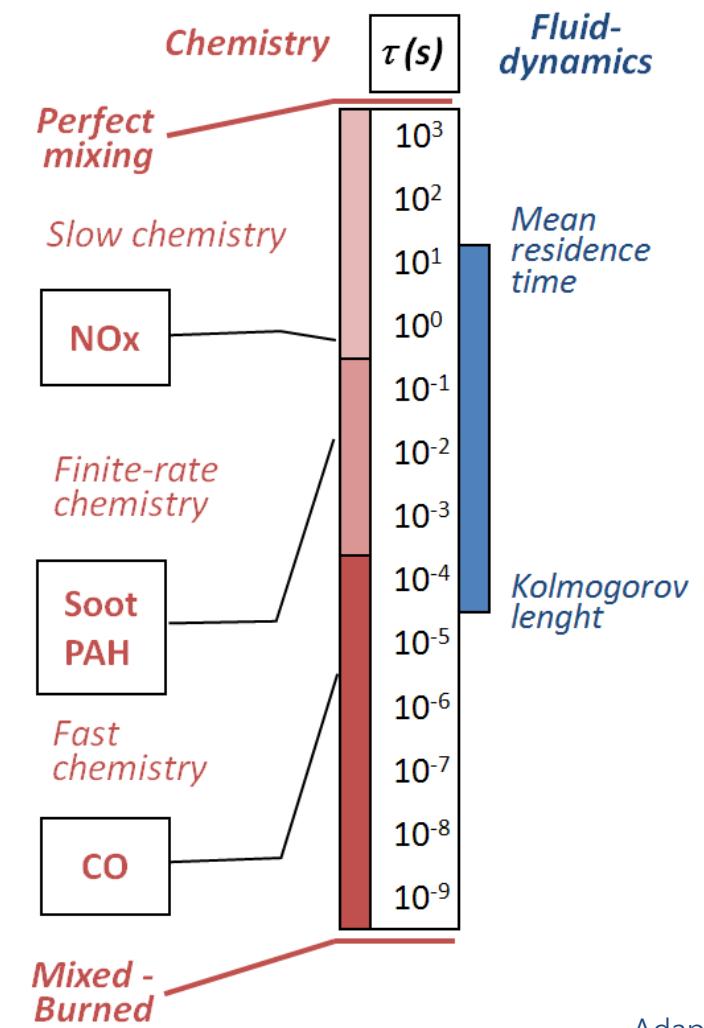


1. Slow modes:

Nox and soot formation
 $\text{CO} \rightarrow \text{CO}_2$ (often rate limiting)

2. Fast modes:

Reactions involving highly reactive radicals ($\text{H}, \text{O}, \text{OH}, \dots$)
 $\text{HCO} \rightarrow \text{CO}$
 $\text{CH}_3\text{O} \rightarrow \text{CH}_2\text{O}$

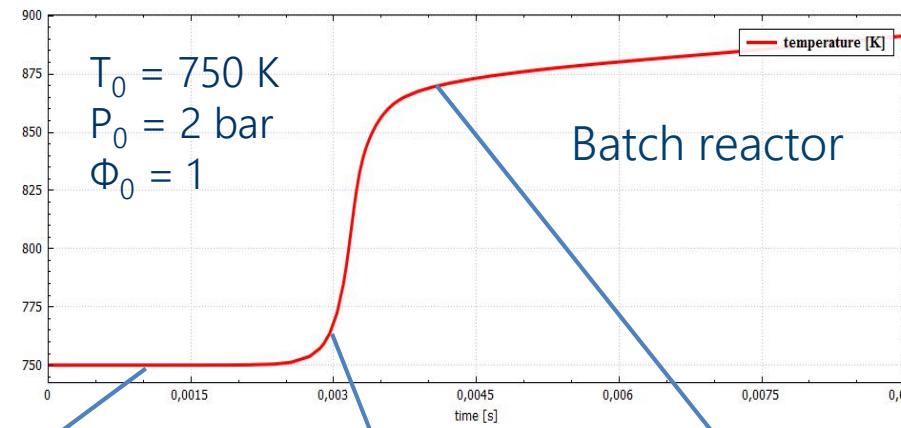


Adapted from:
R. Fox, "Computational models for turbulent reacting flows", Cambridge University Press (2002)

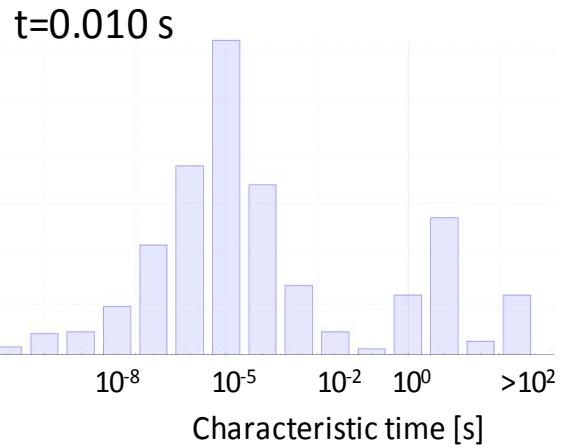
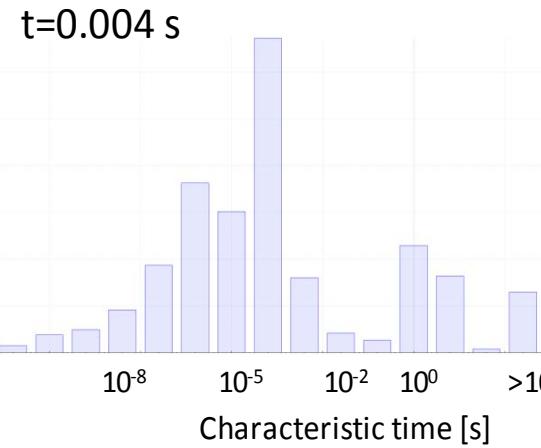
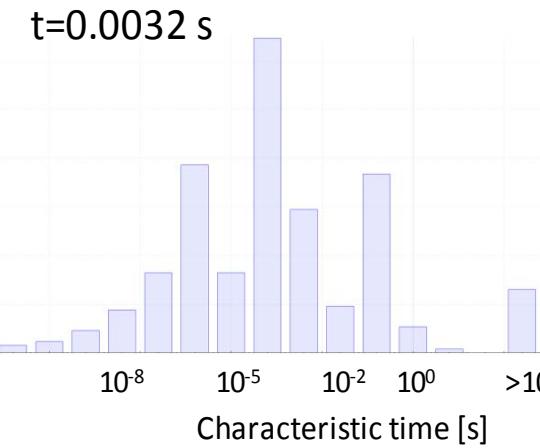
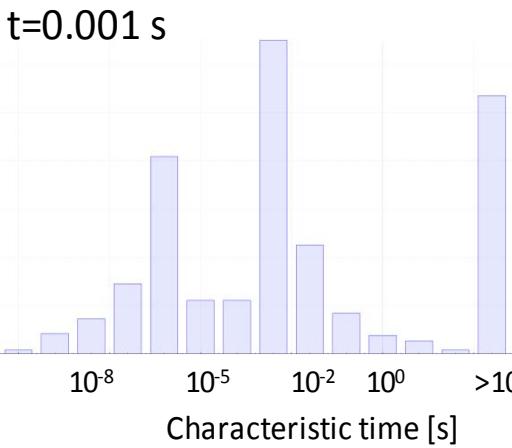


Distribution of characteristic chemical times

LLNL-NC7 Mechanism
Species: 654
Reactions: 2,837

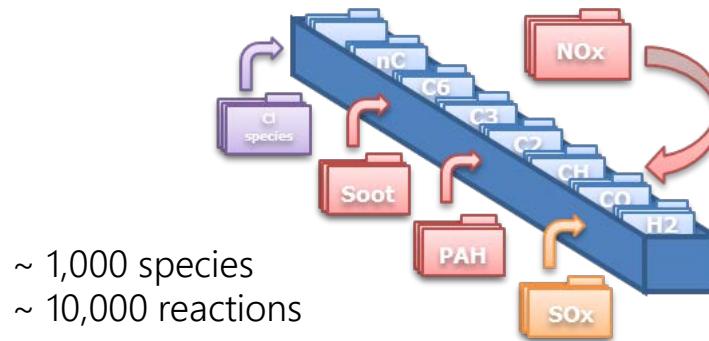


Distribution of characteristic times calculated through the eigenvalue analysis of the Jacobian matrix associated to the system of species and energy equations



Summary

Detailed kinetic schemes



~ 1,000 species
~ 10,000 reactions

1. Number of equations

Since detailed kinetic mechanisms involve hundreds or thousands of species, the number of coupled equations can be very large, especially when multidimensional geometries are simulated

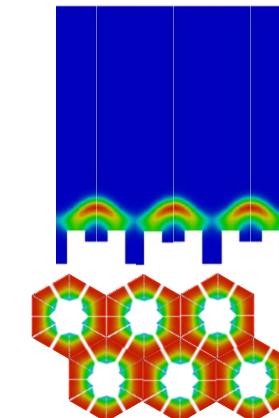
2. Non-linearity

The transport equations of species and energy are very non-linear, because of reaction rates expressions (power-law and exponential)

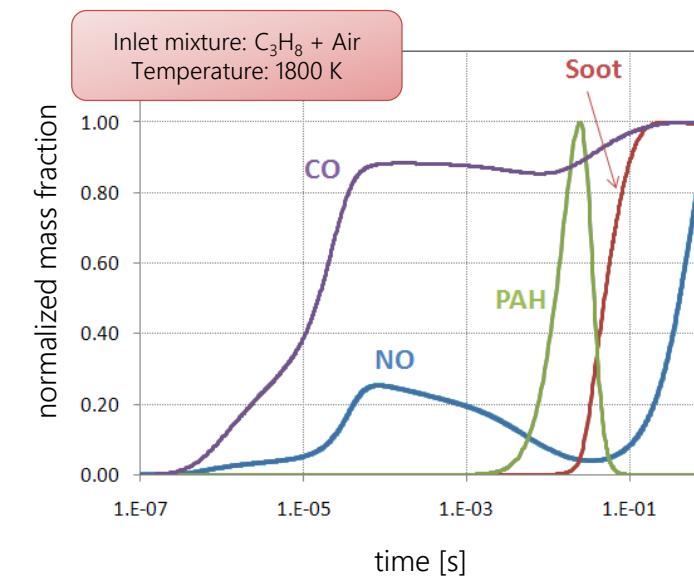
3. Stiffness

The characteristic times of species involved in a kinetic scheme can differ by several orders of magnitudes.

Low-temperature burner



Unsteady flame





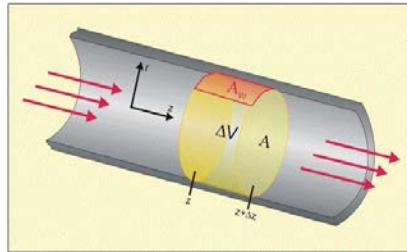
Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



ODE systems

Plug-flow Reactor



$$\begin{cases} \rho v \frac{\partial \omega_i}{\partial z} = \dot{\Omega}_i \\ \omega_i(0) = \omega_i^0 \end{cases}$$

ODE system with initial conditions (IVP)

Continuously Stirred Tank Reactor



$$\begin{cases} \frac{d\omega_i}{dt} = \frac{G_{in}}{\rho V} \omega_{i,in} - \frac{G}{\rho V} \omega_i + \frac{\dot{\Omega}_i}{\rho} - \frac{\omega_i}{\rho} \frac{d\rho}{dt} \\ \omega_i(0) = \omega_i^0 \end{cases}$$

ODE system with initial conditions (IVP)

Batch Reactor



$$\begin{cases} \rho \frac{\partial \omega_i}{\partial t} = \dot{\Omega}_i \\ \omega_i(0) = \omega_i^0 \end{cases}$$

ODE system with initial conditions (IVP)



Need of implicit integrators

Non-linearity

Typically we need Newton's or modified-Newton's methods

Stiffness

Implicit integrators (requiring the Newton's method)



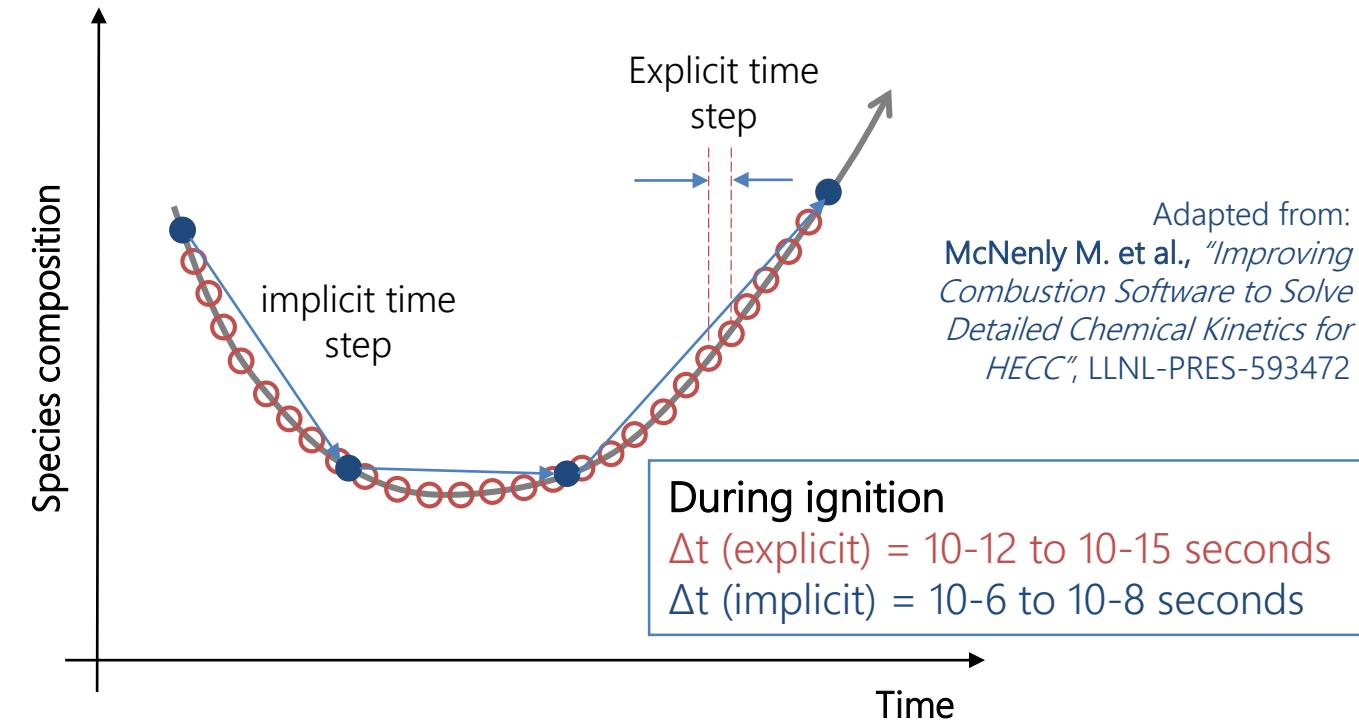
Jacobian evaluation & factorization

$$\frac{\partial \mathbf{y}}{\partial t} = \mathbf{f}$$

ODE system

$$J_{ij} = \frac{\partial f_i}{\partial y_j}$$

Jacobian matrix



Explicit: additions and multiplications
Easy, but many steps

Implicit: solve linear systems
Hard and expensive, but fewer steps





The Jacobian matrix

The Jacobian matrix is required by implicit integrators

For a chemically reacting flow

$$\frac{D\mathbf{y}}{Dt} = \mathbf{f} = \mathbf{r}(\mathbf{y}) + \mathbf{s}(\mathbf{y})$$

Chemical source term Non-chemical terms
(convection, diffusion, etc.)

Evaluation and factorization of Jacobian matrix can be very time consuming!

$$J_{ij} = \frac{\partial f_i}{\partial y_j} = \frac{\partial r_i}{\partial y_j} + \frac{\partial s_i}{\partial y_j}$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \dots & \frac{\partial f_1}{\partial y_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial y_1} & \dots & \frac{\partial f_N}{\partial y_N} \end{bmatrix} = \mathbf{J}_r + \mathbf{J}_s = \begin{bmatrix} \frac{\partial r_1}{\partial y_1} & \dots & \frac{\partial r_1}{\partial y_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_N}{\partial y_1} & \dots & \frac{\partial r_N}{\partial y_N} \end{bmatrix} + \begin{bmatrix} \frac{\partial s_1}{\partial y_1} & \dots & \frac{\partial s_1}{\partial y_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_N}{\partial y_1} & \dots & \frac{\partial s_N}{\partial y_N} \end{bmatrix}$$





Physical meaning of Jacobian matrix

$$\begin{cases} \frac{\partial C_i}{\partial t} = \dot{\Omega}_i \\ C_i(0) = C_i^0 \end{cases}$$

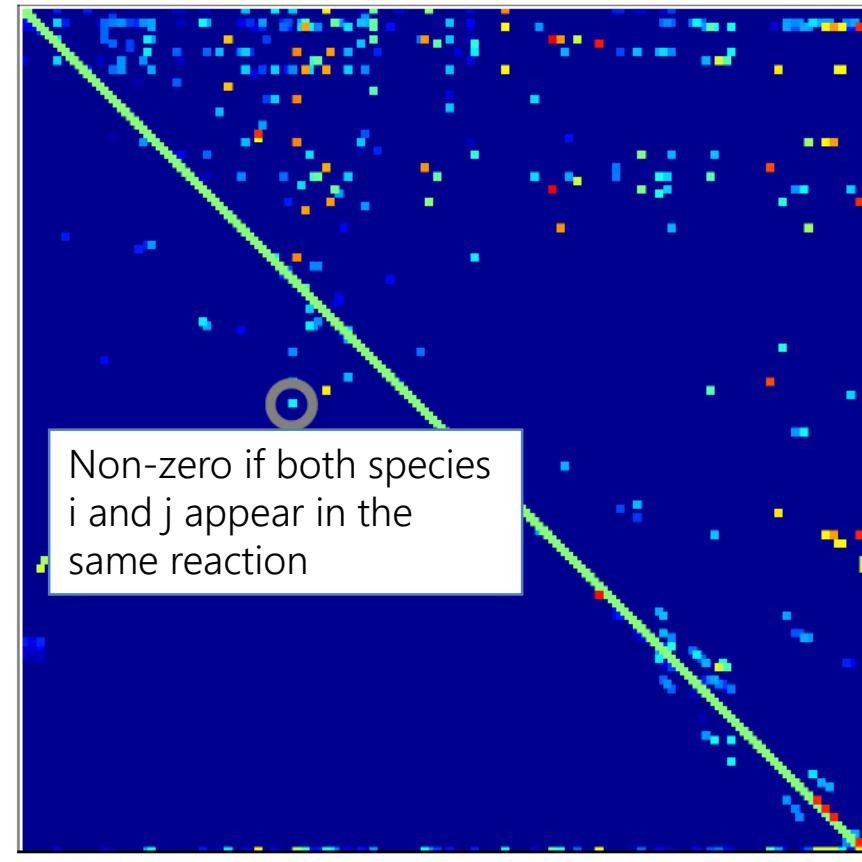
Governing equations for a batch reactor (i.e. closed system)

$$J_{ij} = \frac{\partial \dot{\Omega}_i}{\partial C_j}$$

Magnitude represents the characteristic frequency at which the 2 species (i and j) are coupled

$$\dot{\Omega}_i = \frac{\partial C_i}{\partial t}$$

Jacobian matrix element magnitude



10⁻⁴

10⁰

10⁴

Adapted from:
McNenly M. et al., "Improving Combustion Software to Solve Detailed Chemical Kinetics for HECC", LLNL-PRES-593472



The Jacobian matrix: evaluation

Pseudo-code to numerically calculate the Jacobian matrix

```
Given  $\mathbf{y}$ 
 $\dot{\Omega}' = \text{calculate}(\mathbf{y}')$ 
for j=1:NS
     $y'_j = y_j + \varepsilon_j$ 
     $\dot{\Omega}' = \text{calculate}(\mathbf{y}')$ 
    for i=1:NS
         $J_{i,j} = \frac{\dot{\Omega}'_i - \dot{\Omega}'_j}{\varepsilon_j}$ 
    end
end
```

N_S cycles N_R exponentiations

Cost of numerical Jacobian: $N_S \cdot N_R$ exponentiations

Can be very time consuming (even for 0D systems)

Example

- Auto-ignition of 2-methyl alkanes (LLNL)
- Number of species: $N_S \sim 8000$
- Number of reactions: $N_R \sim 30000$
- Typical Jacobian evaluations ~ 100
- Number of exponentiations $\sim >10^{10}$
- Cost single exponentiation: 20-50 flops
- Total number of flops $\sim 10^{12}$
- Peak speed of this laptop: ~ 10 Gflops
- Time for Jacobian evaluation: ~ 100 s



The Jacobian matrix: factorization

Solution of a linear system: $\tilde{J}\mathbf{x} = \mathbf{b}$

Matrix A is decomposed into the product of upper (U) and lower (L) triangular matrices

$$\tilde{J} = \mathbf{L}\mathbf{U}$$

$$\tilde{J} = \mathbf{L} \quad \mathbf{U}$$

The LU factorization is a relatively expensive process, with the number of operations scaling as **the cube of the size** of the matrix

The solution is accomplished sequentially as a forward substitution followed by a back substitution

$$(\mathbf{L}\mathbf{U})\mathbf{x} = \mathbf{b} \quad \left\{ \begin{array}{l} \mathbf{Ux} = \mathbf{z} \\ \mathbf{Lz} = \mathbf{b} \end{array} \right.$$

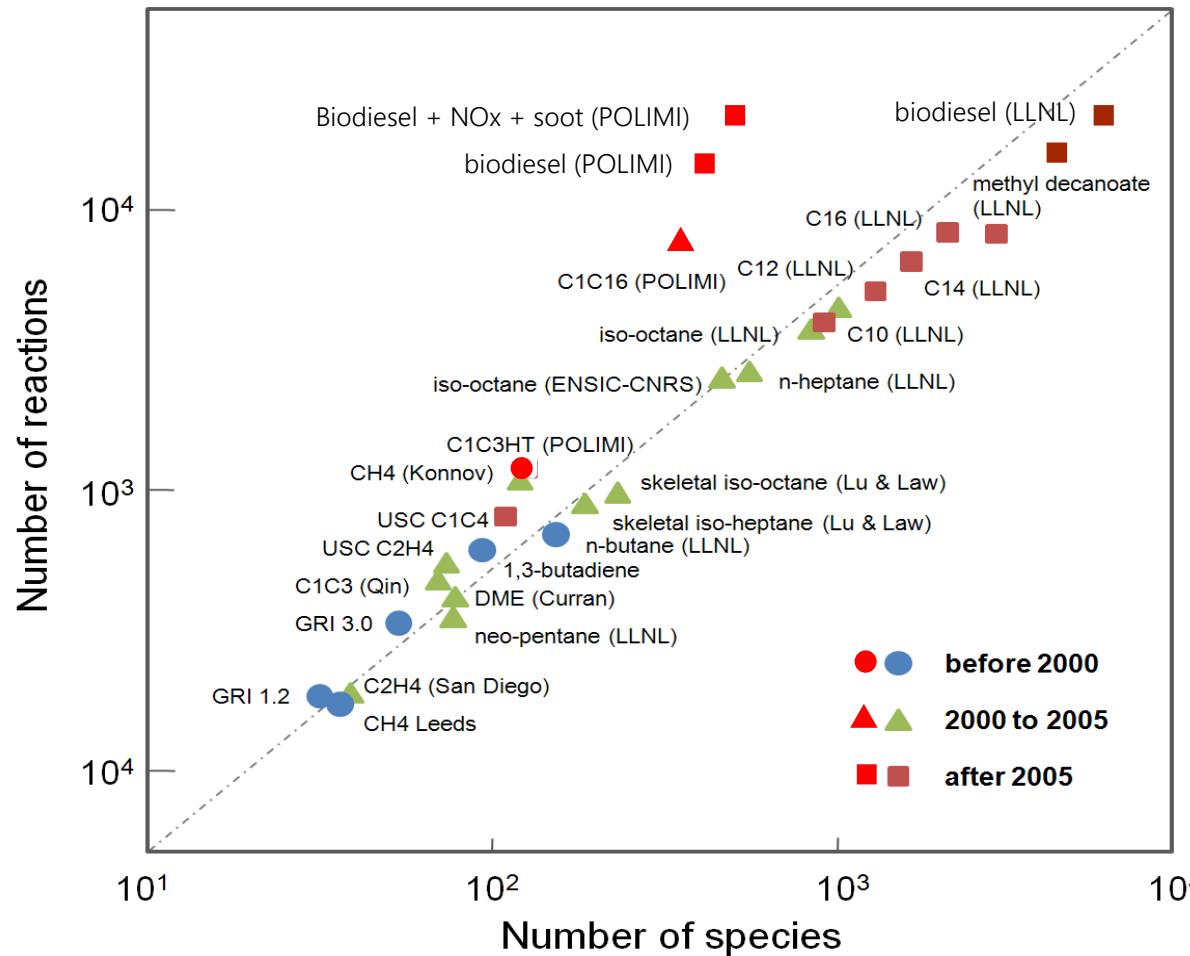
Once decomposed, however, the solution for any b vector scales as **the square of the system size**

Computational cost

$$C \sim \gamma N_s^3$$

Time complexity of Jacobian operations

Typically: $N_R \sim 5 \div 20N_S$



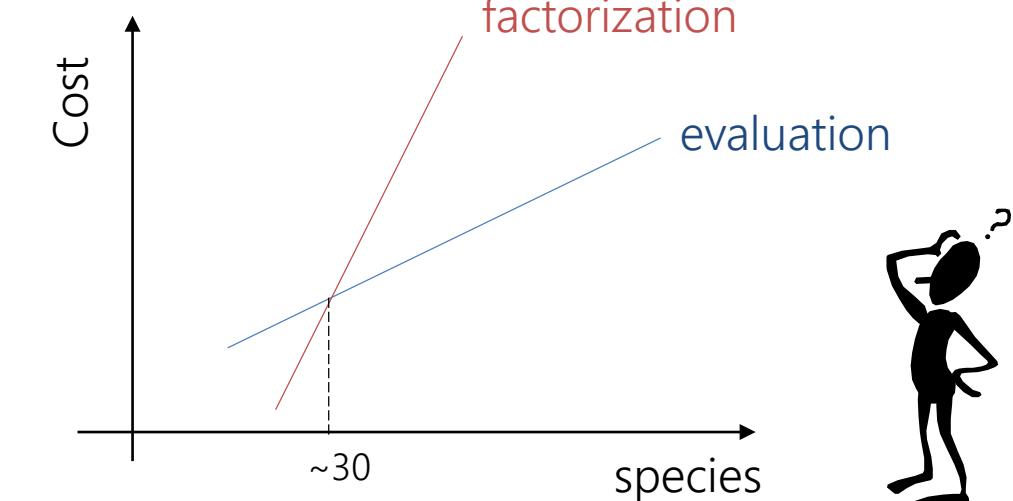
Adapted from: T.F. Lu, C.K. Law, *Toward accommodating realistic fuel chemistry in large-scale computations*, Progress in Energy and Combustion Science, 35, p. 192–215 (2009)

Global computational cost

$$C = \beta N_S^2 + \gamma N_S^3$$

Evaluation of Jacobian (numerically): $\sim 5 \div 20N_S^2$

Factorization of Jacobian (LU): $\sim \frac{2}{3} N_S^3$

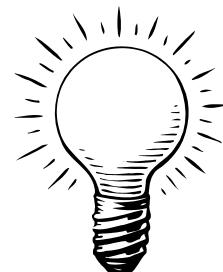


Analytic Jacobian (I)

$$\mathbf{J} = \frac{\partial \dot{\mathbf{Q}}}{\partial \mathbf{y}} = \mathbf{A}\boldsymbol{\Gamma}$$
$$\mathbf{A} = \sum_{i=1}^{NR} \left(\mathbf{v}_i \frac{dr_i}{d\mathbf{C}} \right)$$

It is inexpensive to evaluate analytically, because only the differentiation of each reaction rate r_i with respect to the reactants is needed

Variable transformation matrix (system-dependent). Usually \mathbf{y} represents the mass fractions/ Not time consuming because it does not involve expensive rate evaluations



Cost

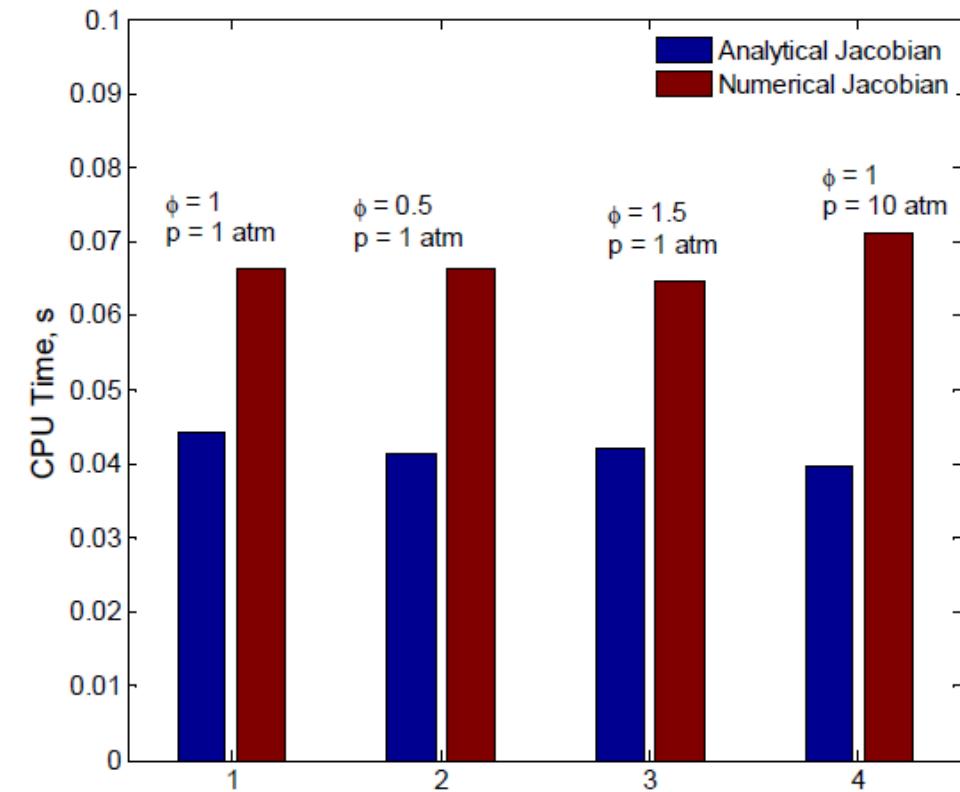
Analytic Jacobian: $\sim N_s$

Numerical Jacobian: $\sim 5 \div 20N_s^2$

Analytic Jacobian is more accurate

Analytic Jacobian should always be used!

Auto-ignition of CH₄/air ($T_0=1000K$)



Plot from:

Lu T., *Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics*, Princeton-CEFRC Summer School on Combustion (2012)





Analytic Jacobian: dynamic vs static (hard-coded)

```
REACTION RATES
printf('Reaction Rates...\n');
progress_bar = statusbar('Reaction Rates - Wait some seconds ...');
for j=1:1:NR
    %FORWARD REACTION
    rd(j) = 1;
    for i=1:1:NC
        if ( nud_matrix(i,j)<0 )
            rd(j) = rd(j)*c(i)^(-nud_matrix(i,j));
        end
    end

    %REVERSE REACTIONS
    if (jEquil(j)==0)
        rinv(j) = 0.;
    else
        rinv(j) = 1.;
        for i=1:1:NC
            if ( nui_matrix(i,j)>0 )
                rinv(j) = rinv(j)*c(i)^nui_matrix(i,j);
            end
        end
        rinv(j) = rinv(j)*uK(j);
    end

    %ASSEMBLING
    rprov(j) = rd(j)-rinv(j);
    rprov_stamp(j) = rprov(j);
    rprov_flat(j) = rprov(j);
end
```

Symbolic preprocessor (MATLAB, Maple, etc.)



C++ (or FORTRAN) source code

```
J[76][59] = d706d59+d1190d59-2.0*d1203d59;
J[76][60] = d707d60+d1190d60-2.0*d1203d60;
J[76][61] = d708d61+d1190d61-2.0*d1203d61;
J[76][62] = d709d62+d1190d62-2.0*d1203d62;
J[76][63] = d710d63+d1190d63-2.0*d1203d63;
J[76][64] = d711d64+d1190d64-2.0*d1203d64;
J[76][65] = d1190d65-2.0*d1203d65;
J[76][66] = d1190d66-2.0*d1203d66;
J[76][67] = d712d67+d1190d67-2.0*d1203d67;
J[76][68] = d713d68+d1190d68-2.0*d1203d68;
J[76][69] = d714d69+d1190d69-2.0*d1203d69;
J[76][70] = d715d70+d1190d70-2.0*d1203d70;
J[76][71] = d716d71+d1190d71-2.0*d1203d71;
J[76][72] = d1190d72-2.0*d1203d72;
J[76][73] = d1190d73-2.0*d1203d73;
J[76][74] = d1190d74-2.0*d1203d74;
J[76][75] = d717d75+d1190d75-2.0*d1203d75;
J[76][76] =
-d1249d76-d1272d76+d1279d76+d1284d76-d1288d76+d1314d76-d16d76+
d1344d76-d1350d76-d38d76+d1351d76-d63d76+d1248d76-d1207d76-
d90d76-d115d76-d168d76-d141d76-d220d76-d194d76-d737d76-d691d76-
d663d76-d637d76-d610d76-d582d76-d558d76-d534d76-d507d76-d480d76-
d455d76-d431d76-d405d76-d380d76-d357d76-d330d76-d302d76-d275d76-
d248d76-d1195d76-d1194d76-d1193d76-d1192d76-d1191d76+d1190d76-
d1197d76-d1196d76+d1198d76-d1199d76-2.0*d1203d76-2.0*d1202d76-
2.0*d1201d76-d1200d76+d1232d76+d1229d76+2.0*d1227d76-d1225d76-
d1206d76-d1205d76-d1204d76+d1235d76+d1238d76+d1239d76-d1280d76-d1240d76-
d1190d77-d1191d77-d1193d77-d1196d77-d1200d77-2.0*d1202d77-
2.0*d1203d77+d1232d77+d1239d77+d1248d77+d1279d77+d1351d77;
d1190d78-d1197d78+d1198d78-2.0*d1203d78-d1207d78+d1284d78-d1288d78;
d1190d79-d1199d79-2.0*d1203d79;
```

- ✓ The hard-coded Jacobian is super-fast (static allocation and compiler optimizations)
- ✓ Compilation time is super-long (thousands of lines of codes)
- ✓ Recompilation is needed if the mechanism changes



Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



Example of Sparse Jacobian

Blue pixel (i,j): a possible non-trivial (non-zero) entry in Jacobian matrix

N_S = number of species

N_R = number of reactions, $N_R \sim 5-20 N_S$

Non-zero coefficients per row: ~ 10

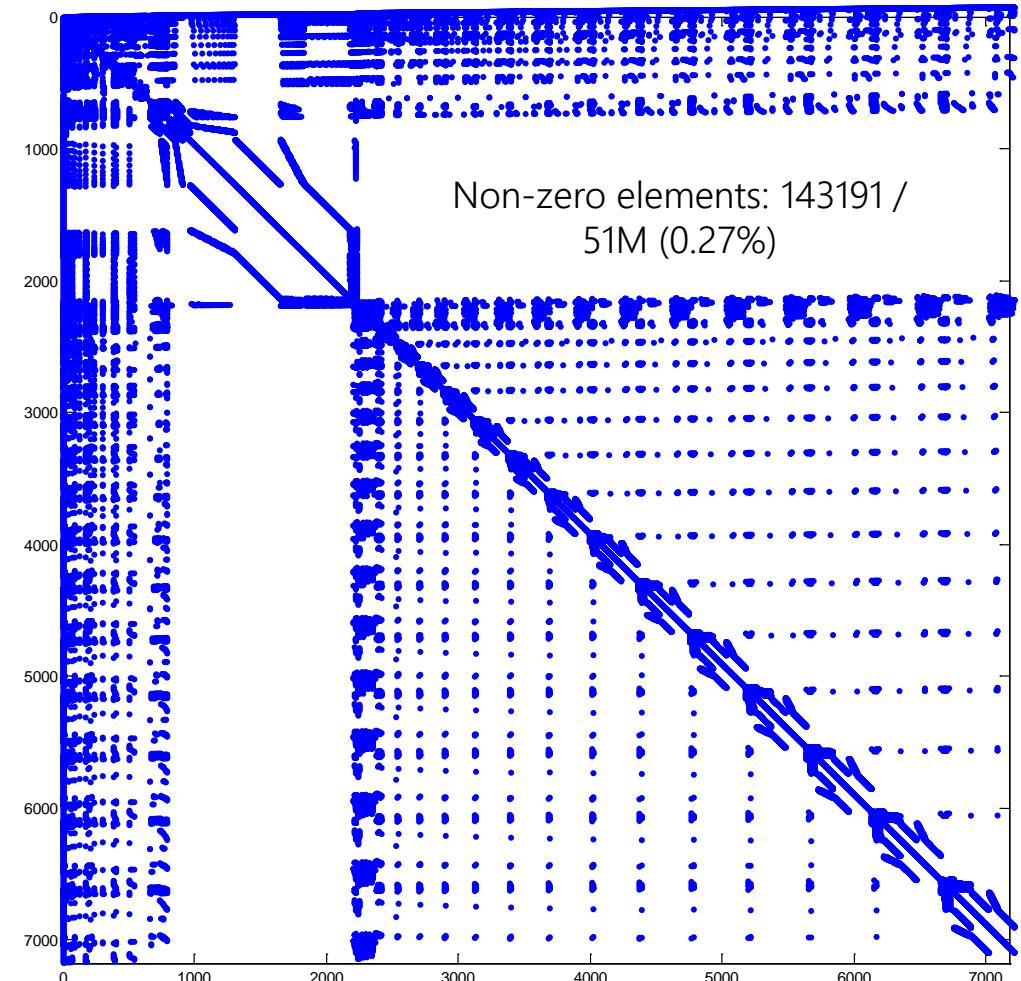
Total number of coefficients: $\sim 10N_S$

Total coefficients in J : $N_S \cdot N_S$

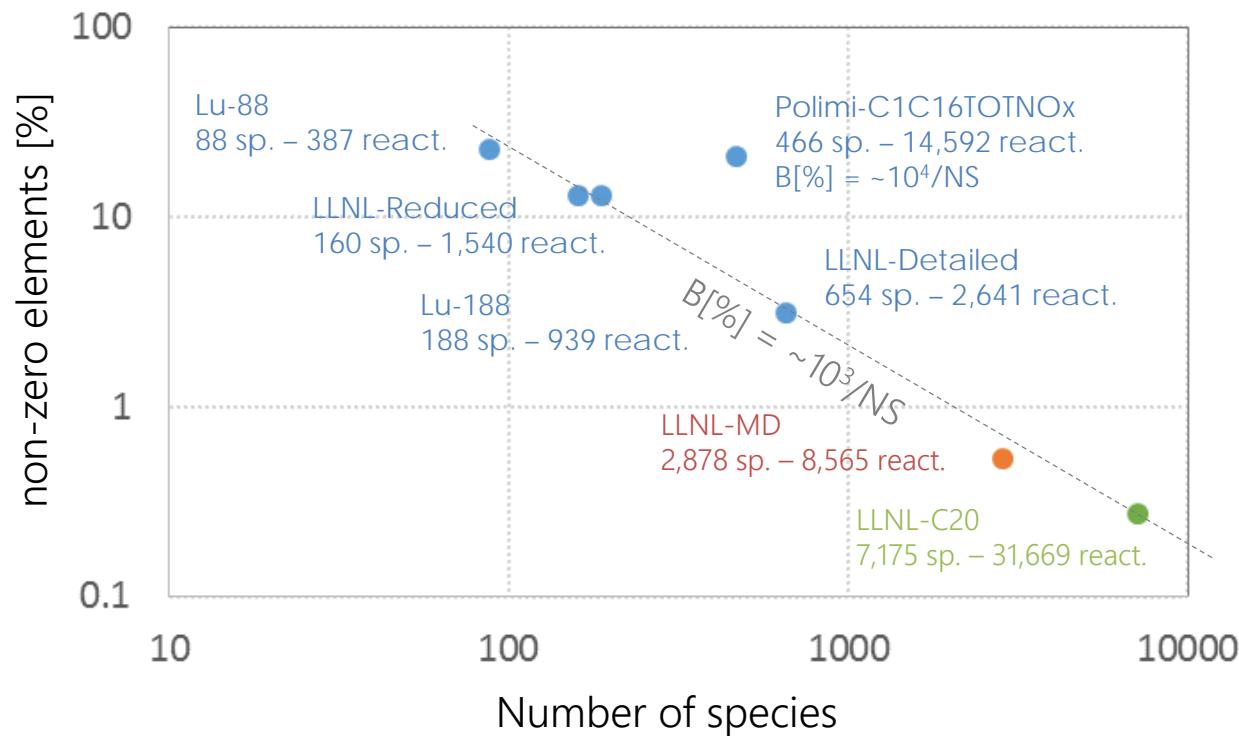
Fraction of non zero coefficients in J : $\sim \frac{10}{N_S}$

Larger mechanisms are sparser!

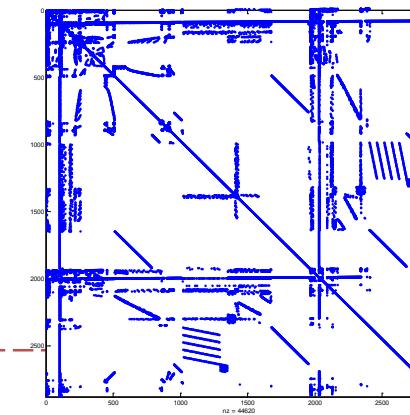
LLNL n-alkanes
Species: 7175, Reactions: 31669



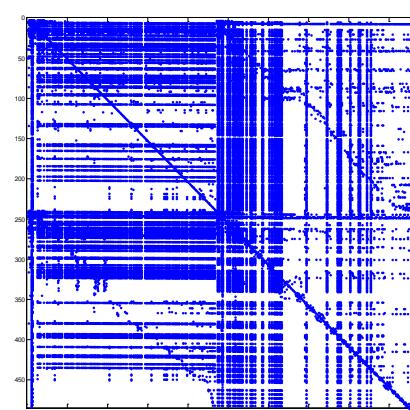
Sparsity increases with size of mechanisms



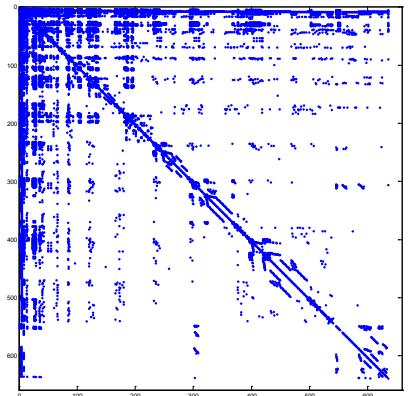
LLNL n-heptane
Species: 658
Reactions: 2827
Non-zeros: 3.03%



LLNL Methyl-decanoate
Species: 2878
Reactions: 8555
Non-zeros: 0.49%



PolimiTOT1412 + NOx
Species: 484
Reactions: 19341
Non-zeros: 16.07%



Increase the sparsity of Jacobian matrix



Species involved in pressure-dependent reactions have dense lines

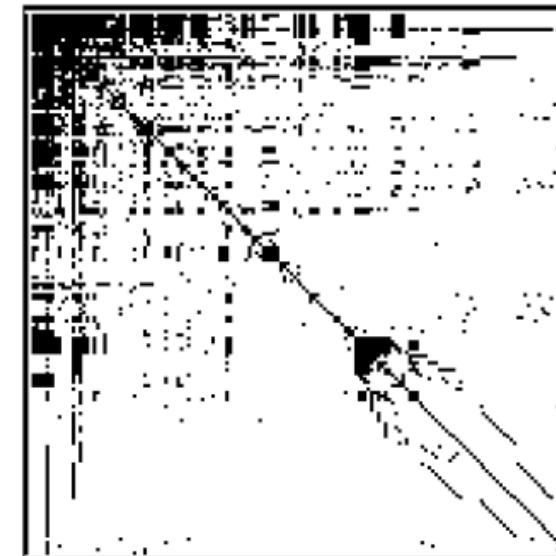


Simplifying assumption: $\frac{\partial c}{\partial Y_j} \approx 0$

full third-body effects



simplified third-body effects



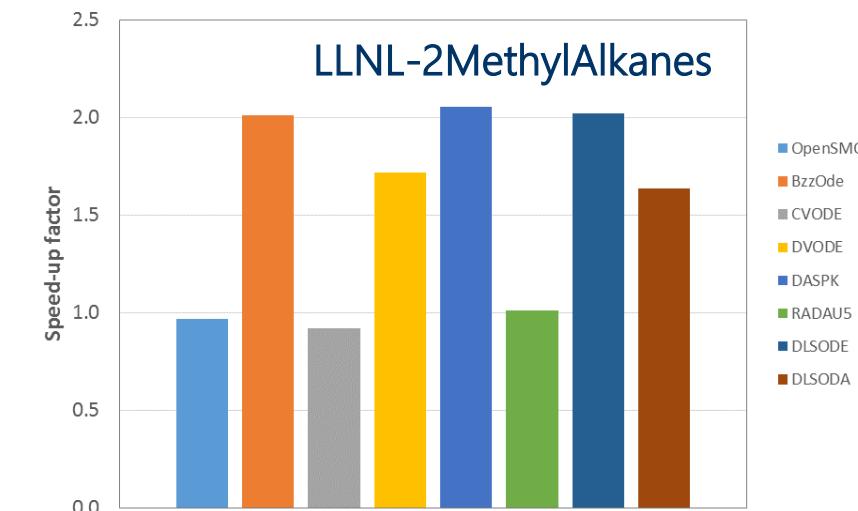
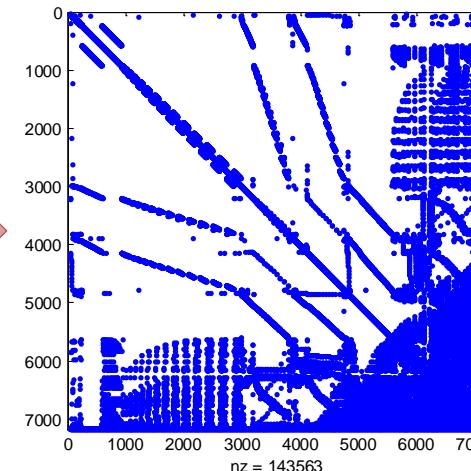
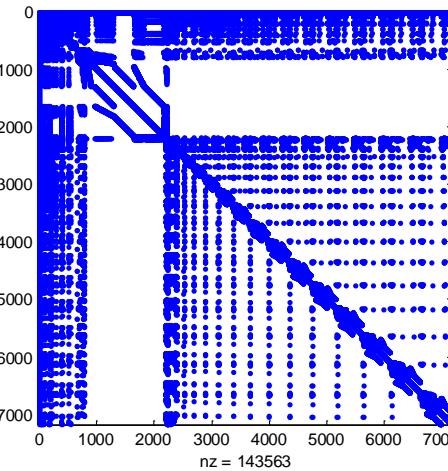
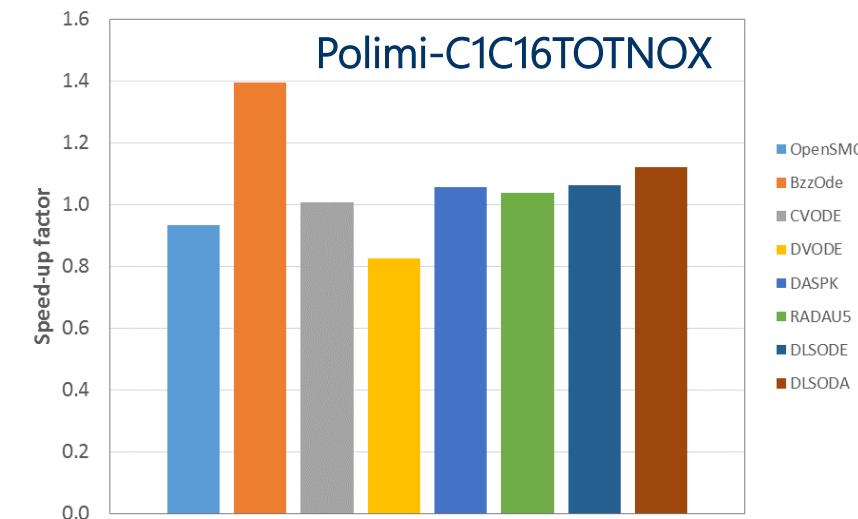
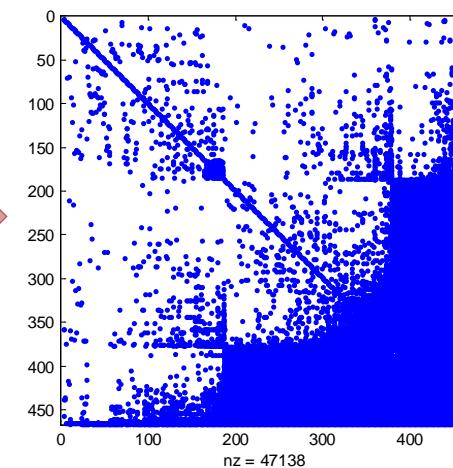
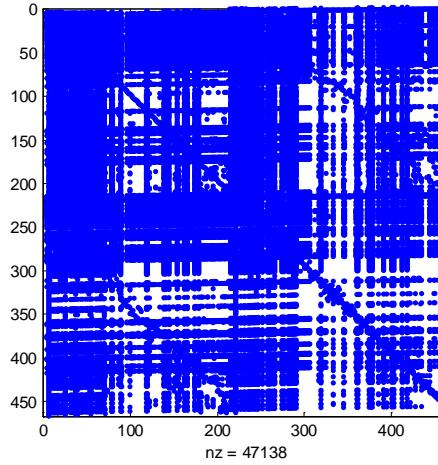
Affected is the Jacobian only, not the problem formulation

Adapted from:

Perini F., *SpeedCHEM, A Sparse Analytical Jacobian Chemistry code for Engine Simulations with Detailed Chemistry*
<http://www.federicoperini.info/publications>



Reordering the species: acceleration for free





Direct vs Iterative solvers (I)

Direct solvers

- Compute a proper decomposition of J
- Can be thought of as variant of LU decomposition that finds triangular factors L and U , so that $J=LU$
- Sparse direct solvers save memory and CPU time by considering the sparsity pattern of J
- Very robust
- Work grows with N^2
- Memory grows with $N^{3/2}$

Do not stuck with some poorly supported packages found somewhere on the internet!
Choose widely used, high-quality software!

Iterative solvers

- Improve the solution in each iteration
- Start with an initial guess
- Continue iterations till a stopping criterion is satisfied (typically that error/residual is less than a specified tolerance)
- Return final guess
- Depending on solver and preconditioner type, work can be only $\sigma(N) \dots$ or much worse!
- Memory increases linearly
- Note: the final guess does not solve $Jx=b$ exactly





Direct vs Iterative solvers (II)

Direct solvers

- Always work, for any invertible matrix
- No need to think about preconditioners
- Faster for problems with < 10k unknowns
- Usually do not scale very well
- Large requests of memory for large problems

Iterative solvers

- Need $\Theta(N)$ memory
- Can solve very large problems (millions of unknowns)
- Often parallelize very well
- Choice of solver/preconditioner depends on the problem

Advantage: only need multiplication with the matrix,
no access to matrix element is required

Disadvantage: Efficiency strongly depends on the
availability of good preconditioners

Preconditioning

Instead of solving the original linear $\mathbf{Ax} = \mathbf{b}$, one may solve the left preconditioned system:

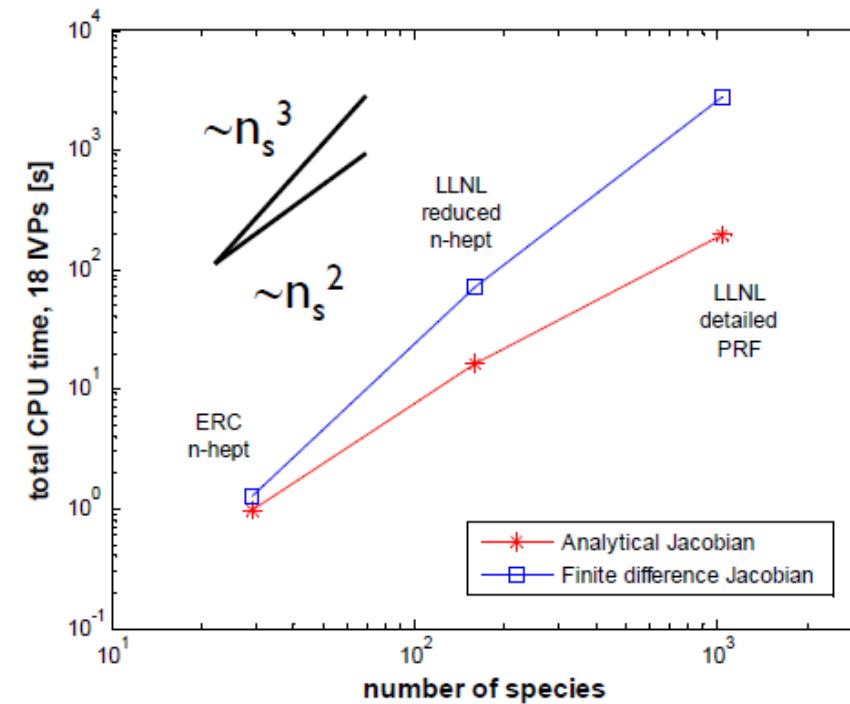
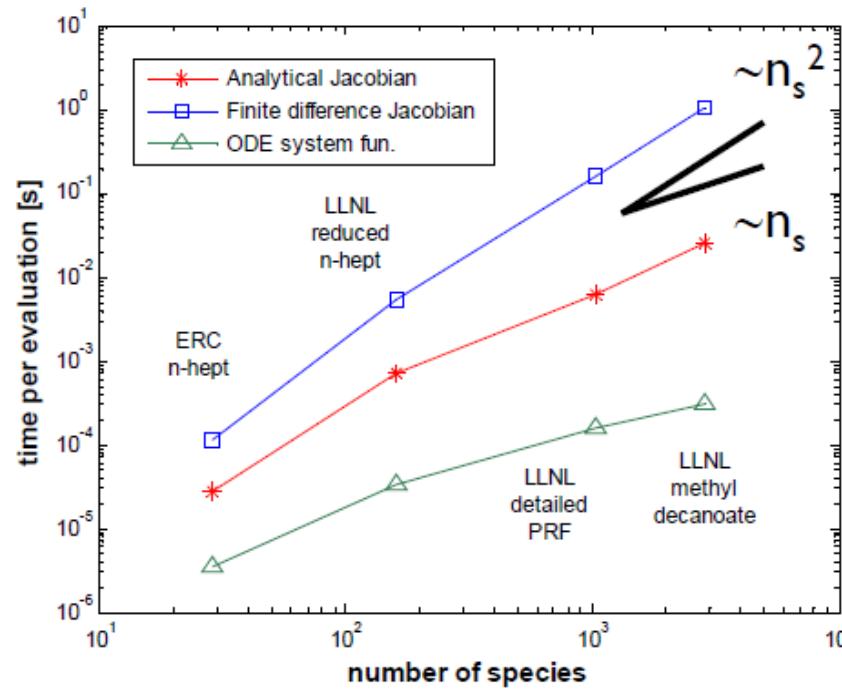
$$\mathbf{P}^{-1}(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}$$

which give the same solution as the original system so long as the preconditioner matrix \mathbf{P} is nonsingular. The right preconditioning is less common.

The goal of this preconditioned system is to reduce the condition number of the left preconditioned system matrix $\mathbf{P}^{-1}\mathbf{A}$. The preconditioned matrix $\mathbf{P}^{-1}\mathbf{A}$ is almost never explicitly formed. Only the action of applying the preconditioner solve operation \mathbf{P}^{-1} to a given vector need to be computed in iterative methods.



Jacobian matrix sparsity effects



Analytical, sparse evaluation of Jacobian: $N_S^2 \Rightarrow N_S$

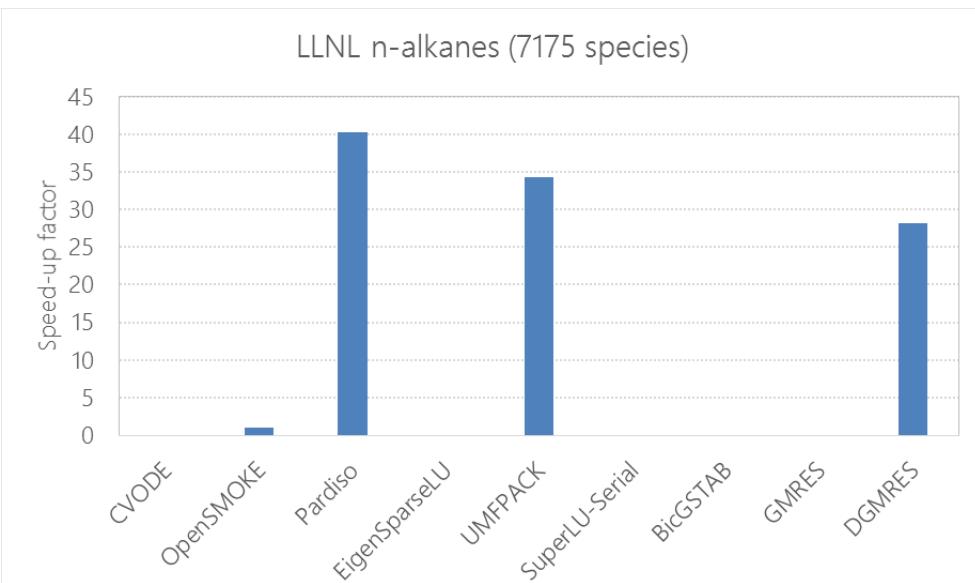
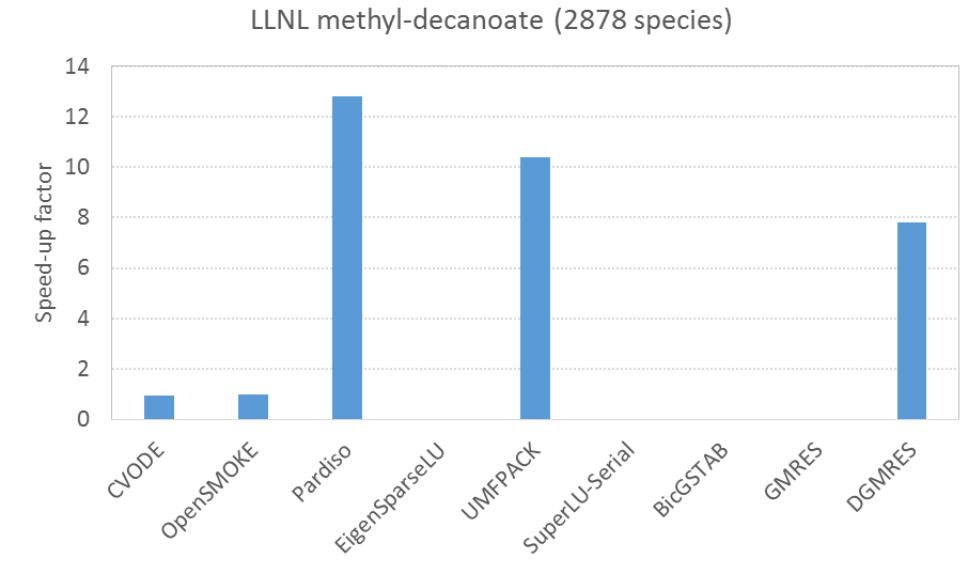
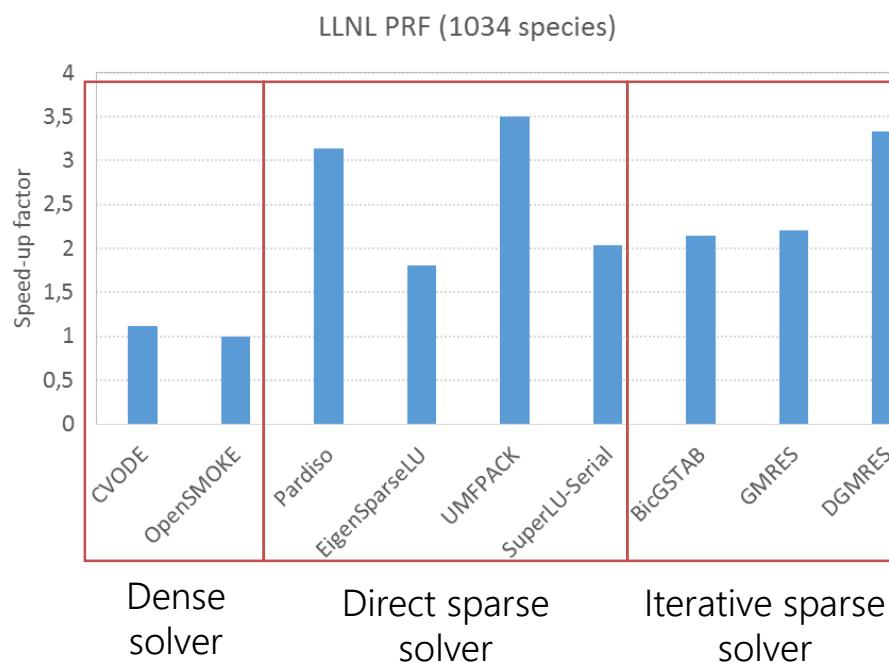
Jacobian matrix storage requirements: $N_S^2 \Rightarrow N_S$

Jacobian matrix factorization: $N_S^2 \Rightarrow N_S$



Plots from:
Perini, F., Galligani, E., Reitz, R. D., *A study of direct and Krylov iterative sparse solver techniques to approach linear scaling of the integration of chemical kinetics with detailed combustion mechanisms*, Combustion and Flame 161 (5), 1180-1195 (2014)

Case study



Same 18 Adiabatic Batch reactors (constant volume)
Contact time: 0.01 s

Same benchmark proposed in:
Perini, F. et al. An analytical Jacobian approach to sparse reaction kinetics for computationally efficient combustion modeling with large reaction mechanisms, Energy and Fuels 26 (8), p. 4804-4822.(2012)



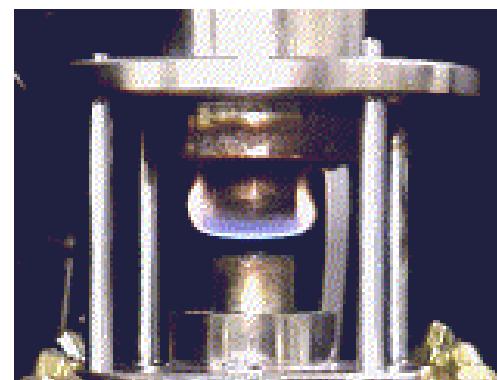
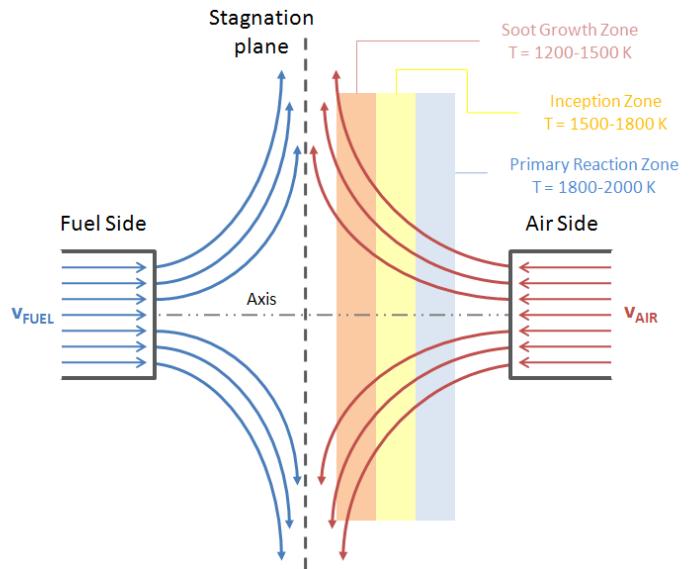
Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks

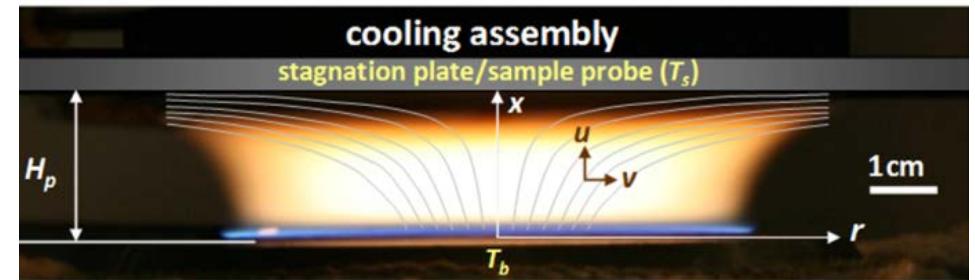


1D flames

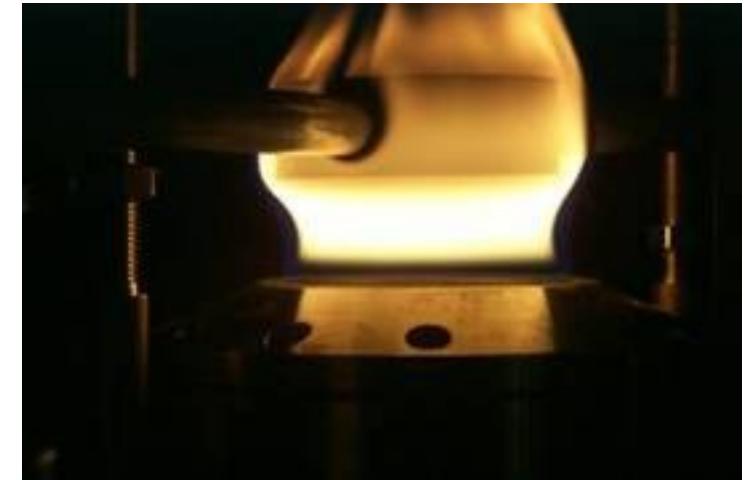
Counterflow diffusion flame



Burner-Stabilized Stagnation Flame



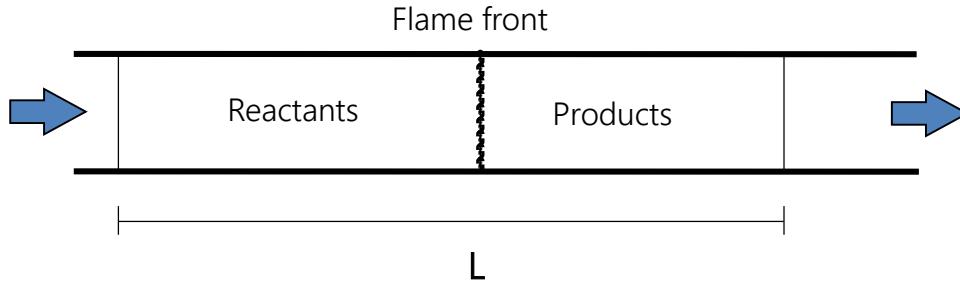
Laminar premixed flat flame



- Twin flames
- Laminar flamelets
- Isolated droplets in microgravity
- ...



Example: premixed laminar flame

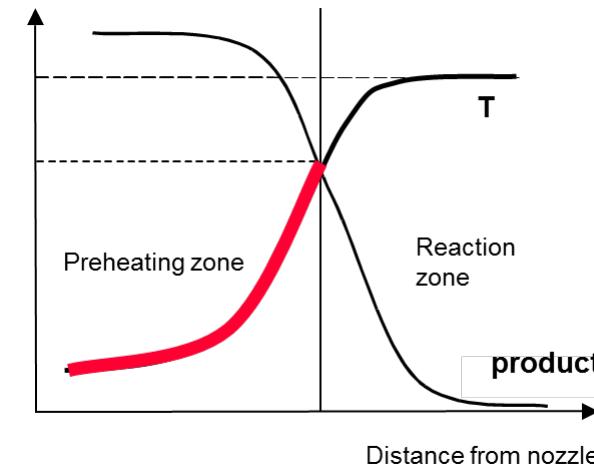


$$\dot{m} = \rho u = \dot{m}_0$$

Mass conservation equation

$$\dot{m} \frac{\partial \omega_k}{\partial x} = - \frac{\partial}{\partial x} (\rho \omega_k V_k) + \dot{\Omega}_k \quad k = 1, \dots, N \quad \text{Species transport equation}$$

$$\dot{m} \frac{\partial T}{\partial x} = \frac{1}{C_P} \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) - \frac{\rho}{C_P} \sum_{k=1}^{N_S} C_{P,k} \omega_k V_k \frac{\partial T}{\partial x} - \frac{1}{C_P} \sum_{k=1}^{N_S} \dot{\Omega}_k \hat{H}_k \quad \text{Energy transport equation}$$

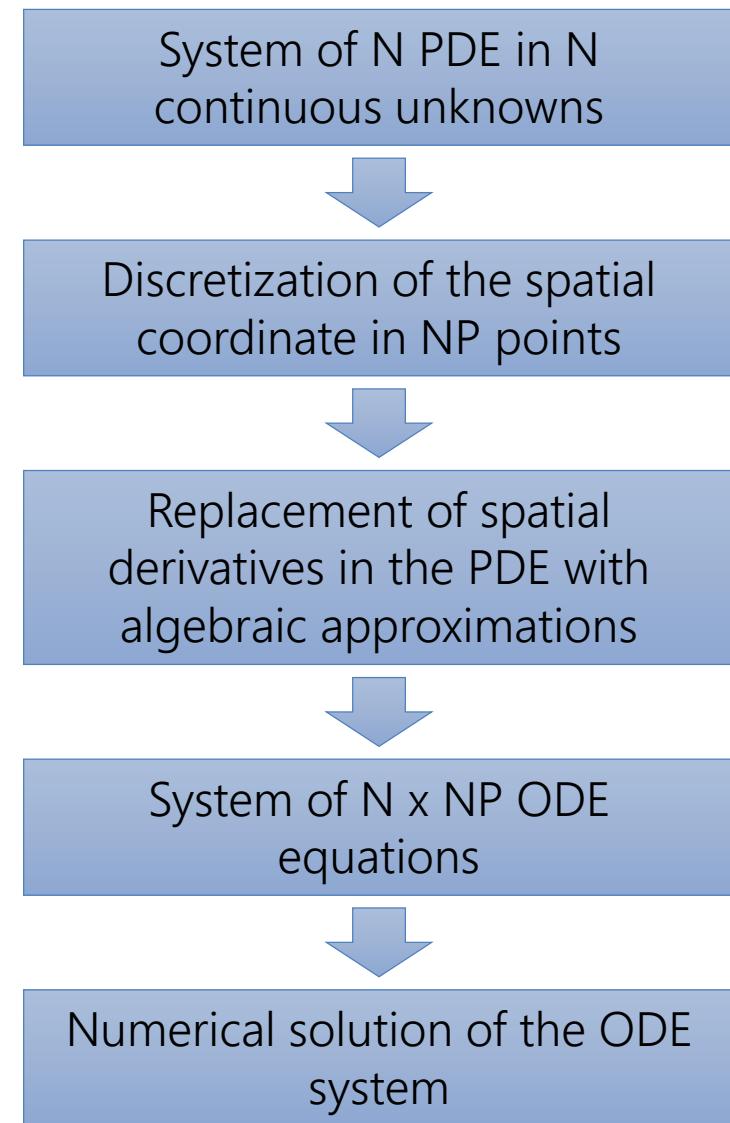


Differential equations with
boundary conditions (BVP)
 $NE = NC + 1$



Method of lines (MOL)

- I. The basic idea of the MOL is to replace the **spatial** (boundary value) derivatives with **algebraic approximations**.
- II. Once this is done, the spatial derivatives are no longer stated explicitly in terms of the spatial independent variables. Thus, only the initial value variable (typically time in a physical problem) remains. In other words, with only one remaining independent variable, we have a **system of ODEs** that approximate the original PDE.
- III. Once this is done, we can apply any integration algorithm for initial value ODEs to compute an approximate numerical solution to the PDE. Thus, one of the salient features of the MOL is the use of existing, and generally well established, numerical methods for ODEs.





Fully-coupled methods

In fully coupled (or global) algorithms all the equations are solved together

$$\frac{\partial \psi_{i,j}}{\partial t} = f_{i,j}$$

System of N x NP
ODES

Strong non linearity of reaction terms
High stiffness

Fully coupled algorithms

- ☺ all the processes and their interactions are considered simultaneously
- ☺ natural way to treat problems with multiple stiff processes
- ☹ the resulting system of equations can be extremely large and the computational cost prohibitive

Example

$$\frac{\partial \Psi}{\partial t} = \mathbf{f} \quad \xrightarrow{\text{Backward Euler}} \quad \frac{\Psi^{(n+1)} - \Psi^{(n)}}{\Delta t} = \mathbf{f}^{(n+1)} \quad \xrightarrow{} \quad \Psi^{(n+1)} = \Psi^{(n)} + \Delta t \mathbf{f}^{(n+1)}$$

$$\mathbf{f}^{(n+1)} = \mathbf{f}^{(n)} + \frac{\partial \mathbf{f}^{(n)}}{\partial \Psi} (\Psi^{(n+1)} - \Psi^{(n)}) \quad \xrightarrow{} \quad \left[\mathbf{I} - \Delta t \frac{\partial \mathbf{f}^{(n)}}{\partial \Psi} \right] (\Psi^{(n+1)} - \Psi^{(n)}) = \Delta t \mathbf{f}^{(n)}$$

Approximation by the first-order
Taylor series expansion



Jacobian sparsity pattern

$$[\mathbf{I} - \Delta t \mathbf{J}^{(n)}] \Delta \boldsymbol{\Psi}^{(n)} = \Delta t \mathbf{f}^{(n)}$$

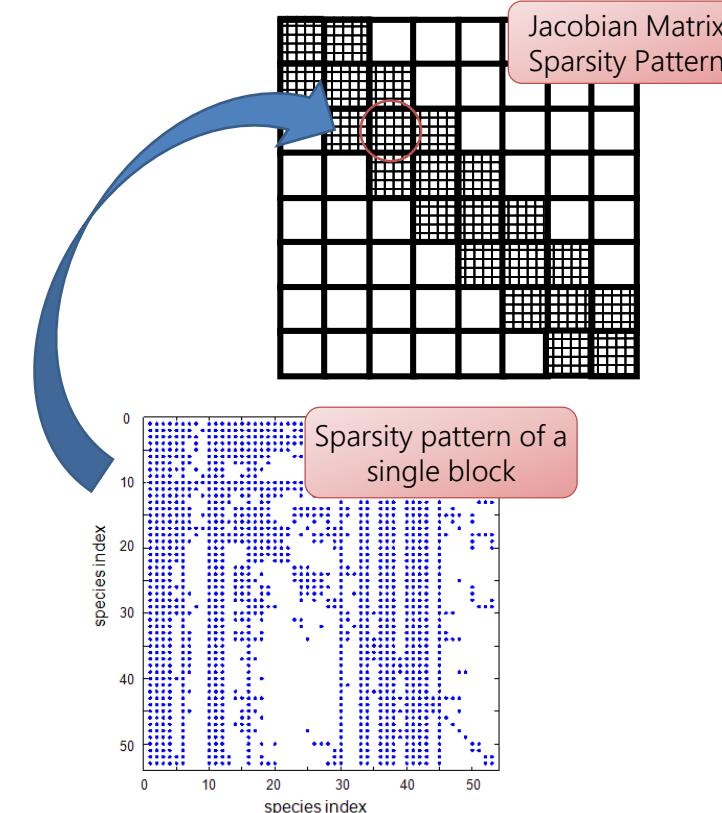
Linear system of
N x NP
equations

1. Sparse Matrix

The matrix is very sparse since in the discretization procedure the finite difference formulas involve only a limited number of points about j (usually the two or the four closest)

2. Tridiagonal Block Structured

The matrix results to be Tridiagonal block structured if the equations are written using a proper ordering: first all the equations referring to point 1, then all the equations belonging to point 2, and so on



Storage and evaluation

The Jacobian matrix must be evaluated by accounting for the sparsity structure, i.e. by avoiding to calculate and store the elements which are equal to zero by definition

Factorization

The Jacobian can be factorized using appropriate and specific algorithms for tridiagonal block matrices, in order to reduce the computational cost



Steady-state problems

For steady-state problems the method of lines can be applied again to the spatial derivatives. Of course the unsteady terms is equal to zero by definition, which means that instead of a ODE system, a system of non linear algebraic equations will be obtained

$$\mathbf{f} = \mathbf{0} \quad \text{System of } N \times NP \text{ non-linear equations (NLS)}$$

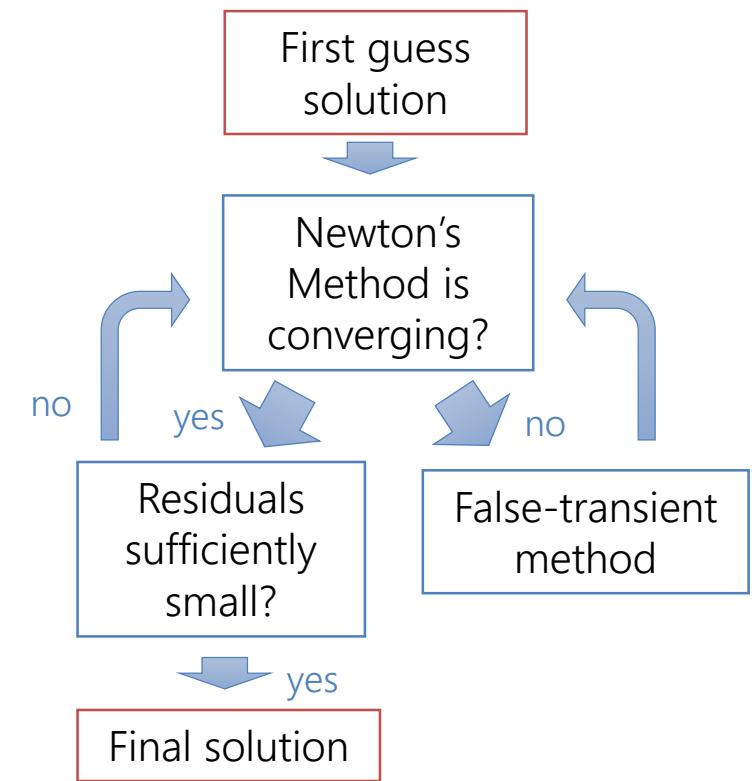
Due to the strong linearity of the reaction terms, the first guess solution does not lie in the convergence domain of the Newtons method.

Pseudo-time stepping methods can be adopted to bring the initial estimate into the convergence domain

$$\frac{\partial \Psi}{\partial t} = \mathbf{f}$$

$$\left[\mathbf{I} - \Delta t \frac{\partial \mathbf{f}^{(m)}}{\partial \Psi} \right] (\Psi^{(m+1)} - \Psi^{(m)}) = \Delta t \mathbf{f}^{(m)}$$

The smaller the time step, more robust (and slower) the method





Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks





Computation Cost Minimization (CCM)

A set of strategies that expedite simulations with little or no accuracy loss through optimization of the computation sequences (Lu & Law, 2009)

code reformulation: many parts of the numerical algorithms are reformulated in a less intuitive way in order to minimize the number of flops needed to perform some calculations

caching: the code is written in order to cache as much as possible, which means storing items for future use in order to avoid retrieving or recalculating them

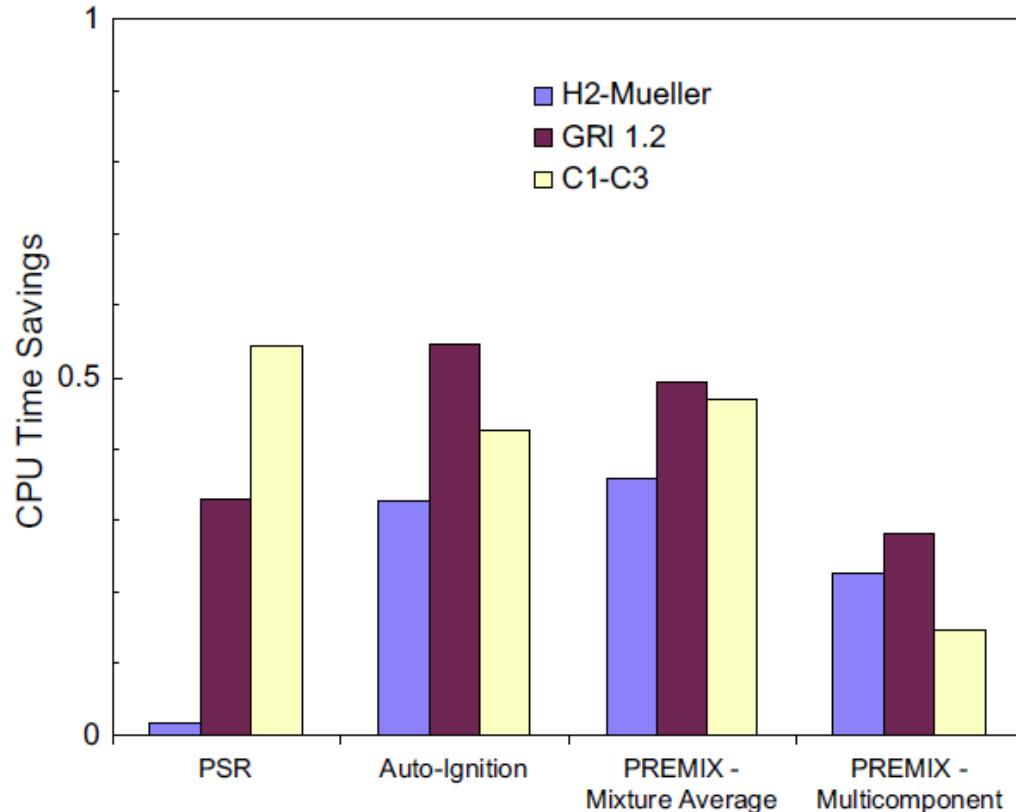
object pools: they are a technique for avoiding the creation and deletion of a large number of objects during the code execution

optimized functions: the numerical algorithms are often reformulated in order to exploit the Intel® MKL Vector Mathematical Functions Library (VML)



An example: code reformulation

Calculated savings in CPU time with CCM normalized by that of detailed mechanisms for H₂, CH₄, and C₂H₄



Plot from: T.F. Lu, C.K. Law, Prog. Energy Comb. Sci., 35 (2009)

Natural implementation

$$k = AT^n \exp\left(-\frac{E}{RT}\right)$$

1 power: ~50 flops
1 exponentiation : ~50 flops
5 multiplications: ~5 flops
Total: ~105 flops

Smart implementation

$$k = \exp(\ln(A) + \alpha \ln(T) - E/RT)$$

1 exponentiation: ~50 flops
3 multiplications: ~5 flops
2 additions: ~2 flops
Total: ~57 flops

The $\ln(T)$ term in the above expression only has to be evaluated once for each call of the rate evaluation subroutine, and the $\ln(A)$ and E/R terms can be pre-evaluated.





Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks





Reduction of detailed kinetic mechanisms

Objective: to eliminate the unimportant species and reactions from a detailed kinetic mechanism

Problem: identification of unimportant species and unimportant reactions

Quantification of the importance of species:

1. Jacobian analysis

The Jacobian matrix coefficients can be arbitrarily large (i.e. difficult to choose a threshold)

2. Relative error on species A induced by elimination of species B

Directed Relation Graph (DRG) (Lu & Law, 2005)

Adapted from:

Lu T., *Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics*, Princeton-CEFRC Summer School on Combustion (2012)



Directed Relation Graph (DRG)

$$r_{AB} = \frac{\sum_i |v_{A,i} \omega_i \delta_{Bi}|}{\sum_i |v_{A,i} \omega_i|}$$

Extent of coupling between species A and B

$$\delta_{Bi} = \begin{cases} 1 & \text{if reaction } i \text{ involves species } B \\ 0 & \text{otherwise} \end{cases}$$

$v_{A,i}$ = stoichiometric coefficient of A in reaction i

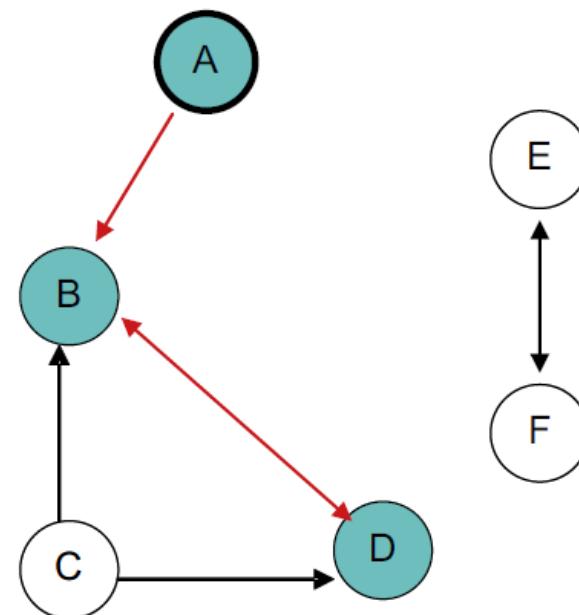
ω_i = net reaction rate of reaction i

B is important to A if (and only if) $r_{AB} > \varepsilon$, a user-specified threshold

Construction of DRG

- Vertex species (A,B,C,...)
- Edges: species dependence, $r_{AB} > \varepsilon$
- Starting vertices: species known to be important (e.g. H, fuel, oxidizer, products, a specific pollutant, ...)

- $A \rightarrow B$ indicates that the elimination of species B will induce a non-negligible error to species A.
- If A is retained in the skeletal mechanism, B should also be retained.



Adapted from: T.F. Lu, C.K. Law, *Toward accommodating realistic fuel chemistry in large-scale computations*, Progress in Energy and Combustion Science, 35, p. 192–215 (2009)

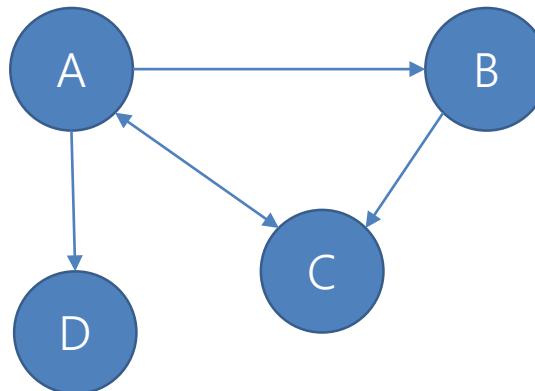


DRG and Sparse Chemical Couplings

An alternative graph representation: adjacency matrix E

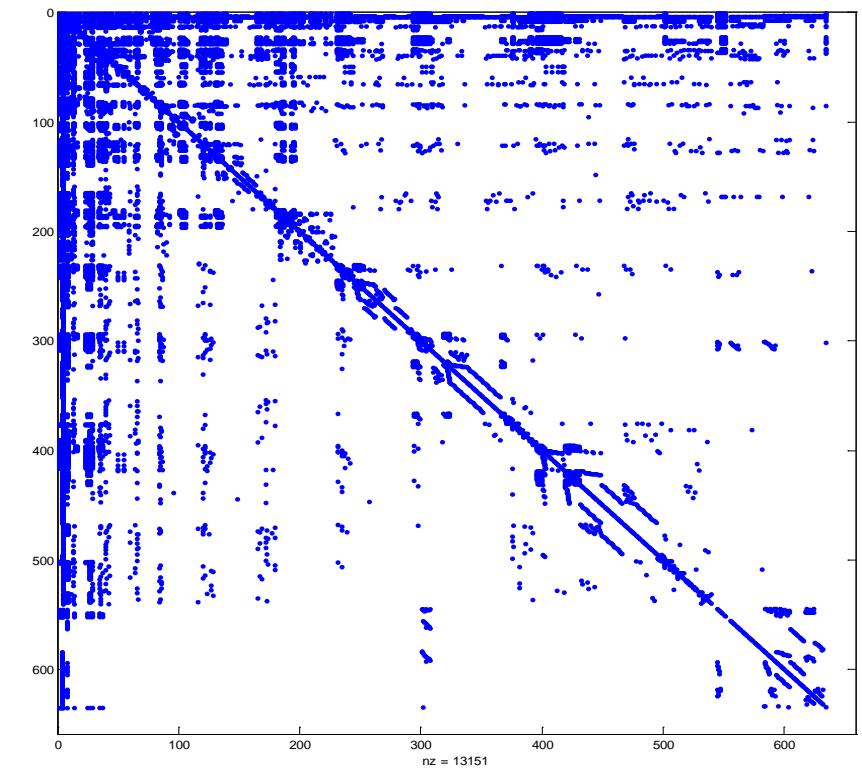
$$E_{i,j} = 1 \quad \text{if } r_{i,j} > \varepsilon$$

- Possible non-zero entries are similar to that in the chemical Jacobian
- DRG is a sparse graph
- Many algorithms in graph theory can take advantage of the sparsity (e.g. depth-first search (DFS), ...)



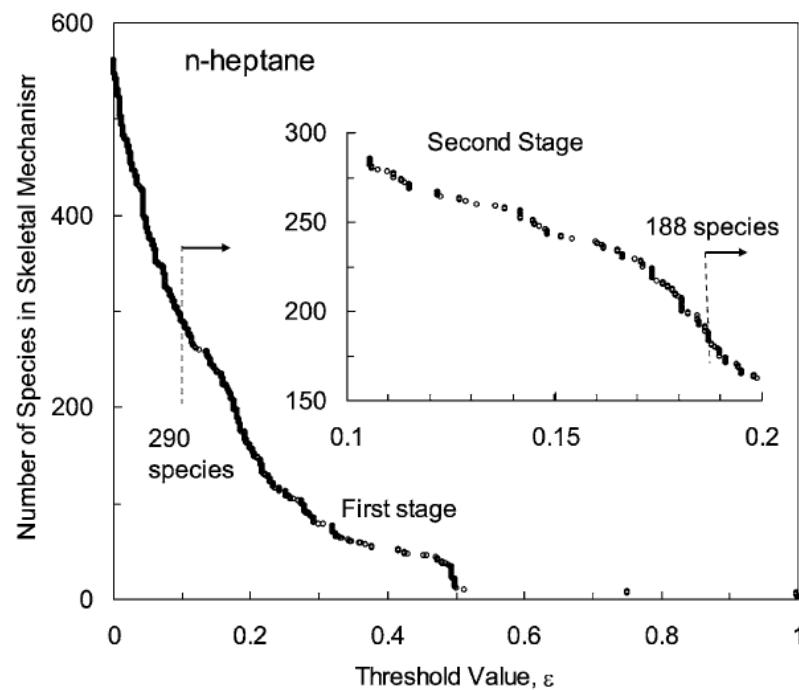
	A	B	C	D
A	1	1	1	1
B	0	1	1	0
C	1	0	1	0
D	0	0	0	1

LLNL n-heptane
Species: 658
Jacobian matrix sparsity pattern

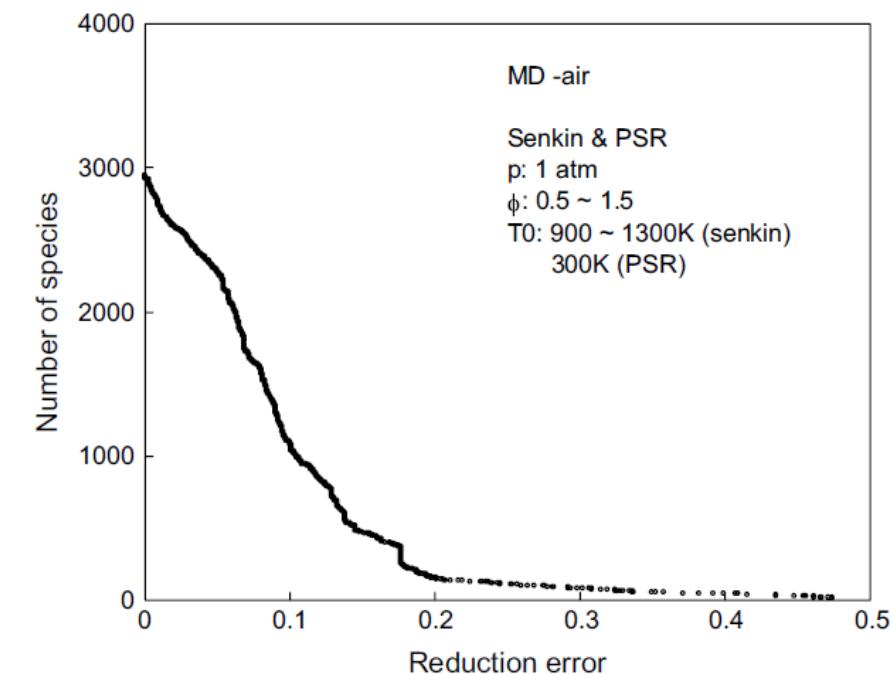


Reduction Curves of DRG

A two-stage DRG reduction can produce a skeletal mechanism smaller than that from a single-stage DRG reduction. The 1st stage of DRG reduction is the major reduction (a large number of species is eliminated), and the 2nd stage is a minor stage.



Plot from: T.F. Lu, C.K. Law, Linear time reduction of large kinetic mechanisms with directed relation graph: n-Heptane and iso-octane, Combustion and Flame 144, p. 24-36 (2006)



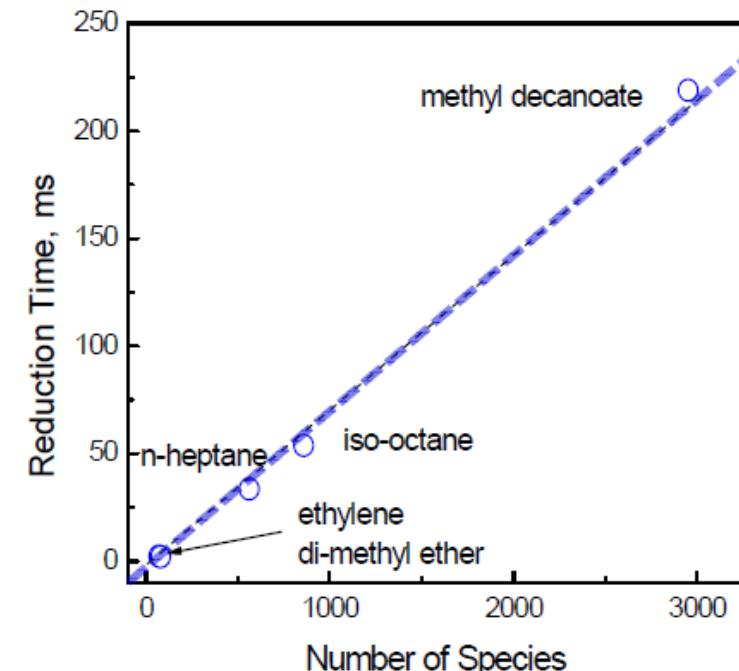
Plot from: T.F. Lu, C.K. Law, *Toward accommodating realistic fuel chemistry in large-scale computations*, Progress in Energy and Combustion Science, 35, p. 192-215 (2009)



More about DRG

DRG is very suitable for large detailed mechanisms

- Linear reduction time: cost **linear** with number of species
- Error control at reduction time
- Fully automated



Other graph-based reduction methods

- DRG aided sensitivity analysis (**DRGASA**) [Zheng et al., 2007; Sankaran et al. 2007]
- DRG with error propagation (**DRGEP**) [Pepiot-Desjardins and Pitsch 2008; Liang et al. 2009, Shi et al. 2010]
- Path flux analysis (**PFA**) [Sun et al., 2009]
- DRGEP with sensitivity analysis (**DRGEPSA**) [Niemeyer et al. 2010]
- Transport flux based DRG (on-the-fly reduction) [Tosatto et al. 2011]
- DRG with expert knowledge (**DRGX**) [Lu et al. 2011]

Adapted from:
Lu T., *Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics*, Princeton-CEFRC Summer School on Combustion (2012)





A multi-step approach to skeletal reduction (I)

1. Reacting Flux Analysis

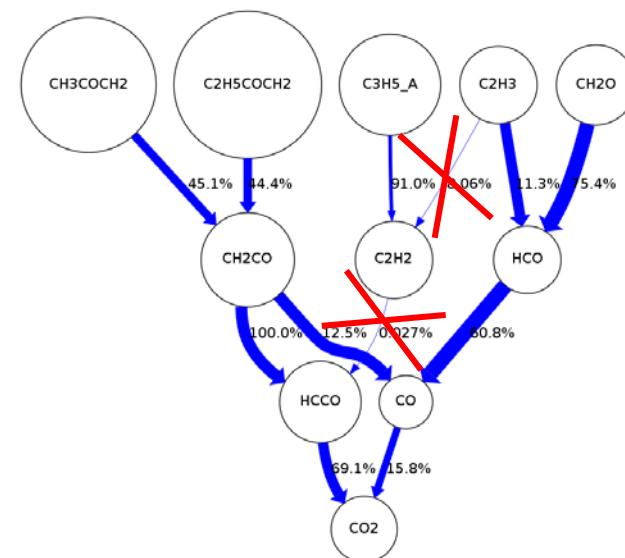
Ranking species according to production and consumption rate history in ideal reactors.

Absolute Net reaction rate

$$R_i(t) = \sum_{j=1}^{NR} |v_{i,j} r_j|$$

Reacting flux

$$F_i = MW_i \int_0^{\tau_{ign}} R_i dt$$



Stagni A., Cuoci A., Frassoldati A., Faravelli T., Ranzi E., *Lumping and reduction of detailed kinetic schemes: an effective coupling*, Industrial & Engineering Chemistry Research, 53(22), p. 9004-9016 (2014)

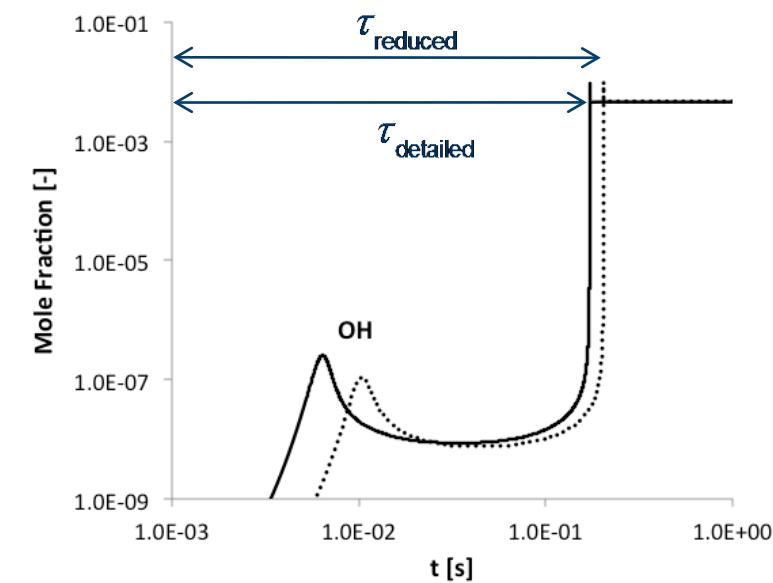
2. Species Sensitivity Analysis

Ranking critical species according to induced error in the reduced mechanism

Induced error

$$\epsilon = \frac{|\tau_{\text{reduced}} - \tau_{\text{detailed}}|}{\tau_{\text{detailed}}}$$

Monitored property is ignition delay time



A multi-step approach to skeletal reduction (II)

3. Reactions Sensitivity Analysis

Understanding the governing parameters (reaction rates) of dynamic systems

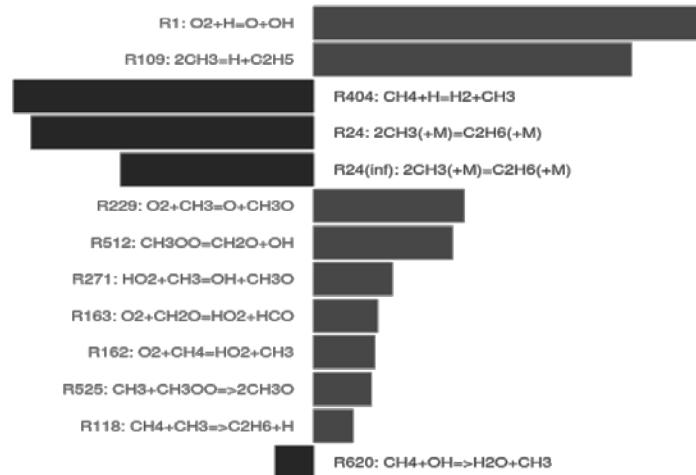
Reactor model

$$\begin{cases} \frac{dy}{dt} = f(y, t, \alpha) \\ y(t_0) = y_0 \end{cases}$$

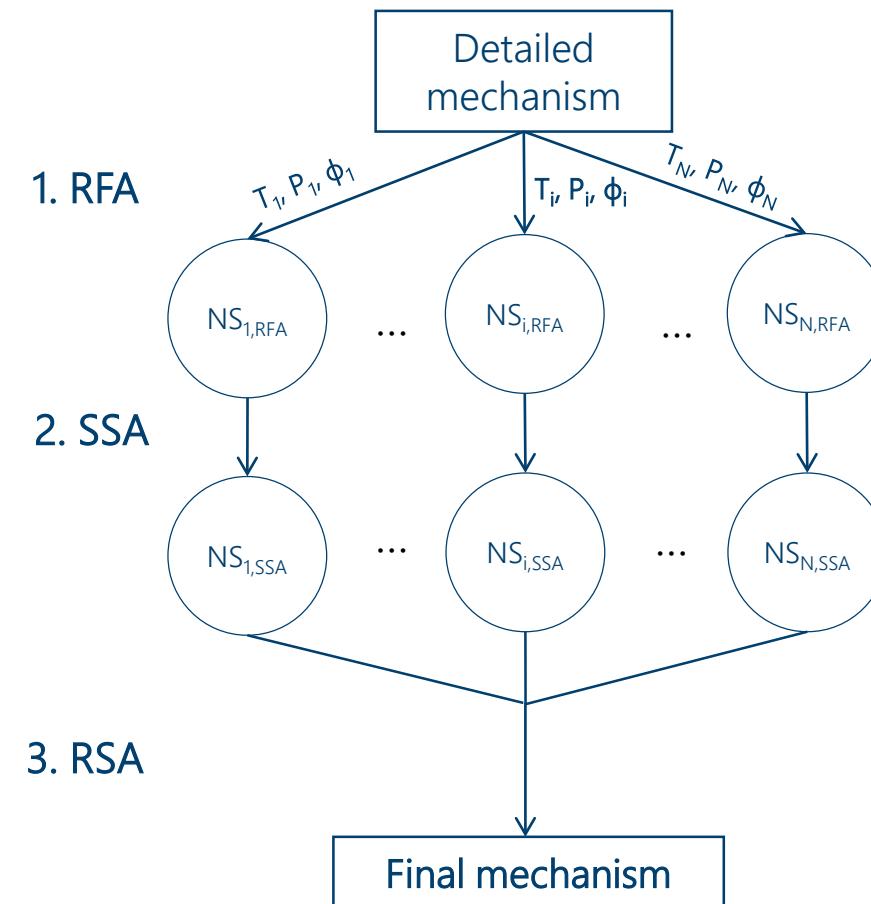
Normalized sensitivity coefficients

$$\tilde{s}_{ij}(t) = \frac{\partial \ln(y_i)}{\partial \ln(\alpha_j)} = \frac{\partial y_i(t)}{\partial \alpha_j} \cdot \frac{\alpha_j}{y_i(t)}$$

Sensitivity Analysis - OH



Courtesy of Alessandro Stagni
(Politecnico di Milano)



DoctorSMOKE v0.1



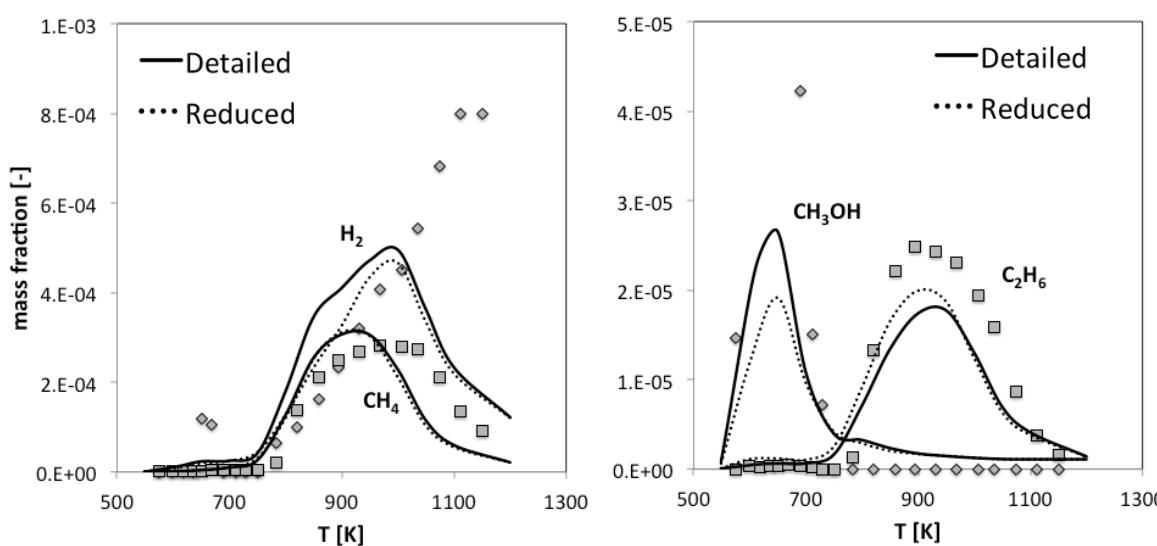
Validation: n-heptane

Target: LLNL (1998) n-heptane model: 561 species – 2539 reactions

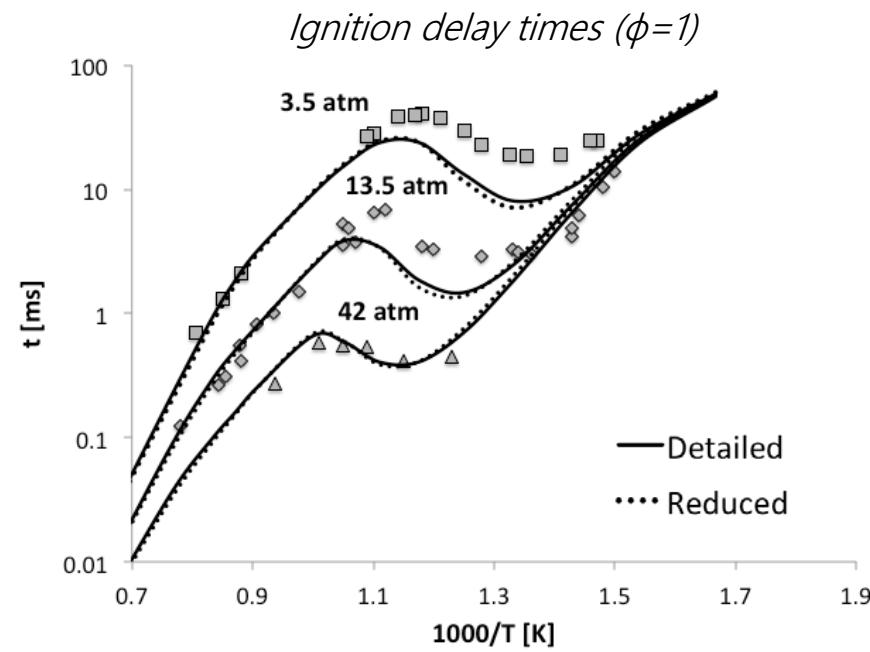
	Species	Reactions	Max error (%)
Detailed	561	2539	-
DoctorSMOKE	126	560	19.6
* DRGEP	149	714	19.5

* Comparison with **state-of-the-art** approaches to mechanism reduction (Niemeyer, 2010)

Niemeyer et al, *Skeletal mechanism generation for surrogate fuels using directed relation graph with error propagation and sensitivity analysis*, Combustion and Flame 157, p. 1760-1770 (2010)



Speciation in isothermal Jet Stirred Reactors



Observations

Strengths

- The performance of the reduced mechanism is similar to that of the detailed one at a reduced computational cost
- No need to dynamically change the mechanism's dimensions allows for a tailored approach to the ODE solution

Weakness

- The operating and initial conditions chosen as reference are defined by the user and are problem dependent
- The possibility of considering transport-driven cases (e.g. 1D laminar flames) is limited by their computational demand (unviable for genetic optimization)

Adapted from:

Perini F., *SpeedCHEM, A Sparse Analytical Jacobian Chemistry code for Engine Simulations with Detailed Chemistry*, <http://www.federicoperini.info/publications>





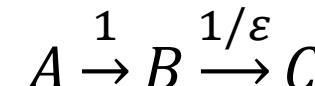
Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



Quasi Steady State Approximations (QSSA)

B is a QSS species: consumption much faster than formation



The net production rate of the QSS species is therefore negligible compared with both the creation and the destruction rates

Its concentration is described by a non-linear, algebraic equation

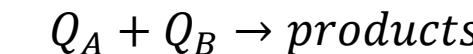
Lu T. and Law C.K., *A criterion based on computational singular perturbation for the identification of quasi steady state species: A reduced mechanism for methane oxidation with NO chemistry*, Combustion and Flame, 154 p. 761–774 (2008)

$$\frac{dB}{dt} = A - \frac{B}{\varepsilon} \approx 0$$

$$B \approx A\varepsilon = \sigma(\varepsilon)$$

Hypotheses

1. QSS species stay in low concentrations
2. Reactions involving two QSS reactors are likely unimportant



$$\omega = k_f [Q_A][Q_B] = \sigma(\varepsilon^2)$$



QSSA are intrinsically a linear problem

Lu T. and Law C.K., *Systematic Approach To Obtain Analytic Solutions of Quasi Steady State Species in Reduced Mechanisms*, Journal of Physical Chemistry A, 110, p. 13202-13208 (2006)



Solving linear-QSSA Equations

QSS species: governing (algebraic) equations

$$\frac{d\mathbf{y}_{QSS}}{dt} = \mathbf{g}(\mathbf{y}_{QSS}; \mathbf{y}_{major}, p, T) \approx \mathbf{0}$$

Traditional approach: algebraic iterations

- Slow convergence (inefficiency)
- Possible divergence (crashes, ...)

New approach: linear QSSA (analytic solution)

- Higher accuracy
- Higher efficiency
- Higher robustness

$$D_i x_i = \sum_{k \neq i} C_{ik} x_k + C_{i0}$$

System of linear equations

Formation rate involving major species

Formation rate involving other QSS species

Destruction rate

1. the LQSSA equations can be analytically solved with a directed graph, (QSSG), which is abstracted from the interdependence of QSS species.
2. To obtain analytic solutions, the groups of strongly connected QSS species are first identified in the QSSG.
3. The inter group couplings are then resolved by a topological sort, and the inner group couplings are solved with variable elimination by substitution.



Dynamic Chemical Stiffness Removal (DCSR)

Problem: Mechanisms are still stiff, even after skeletal reduction and QSSA

Consequences: Implicit solvers needed for stiff chemistry

- Evaluation of Jacobian: $\sim\sigma(N_S)$ or $\sim\sigma(N_S^2)$
- Factorization of Jacobian: $\sim\sigma(N_S^3)$

Adapted from:

Lu T., *Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics*, Princeton-CEFRC Summer School on Combustion (2012)

Idea of Dynamic Chemical Stiffness Removal

- Chemical stiffness is induced by fast reactions
- Fast reactions results in either QSSA or PEA
- Classified a priori
- Analytically solved on the fly

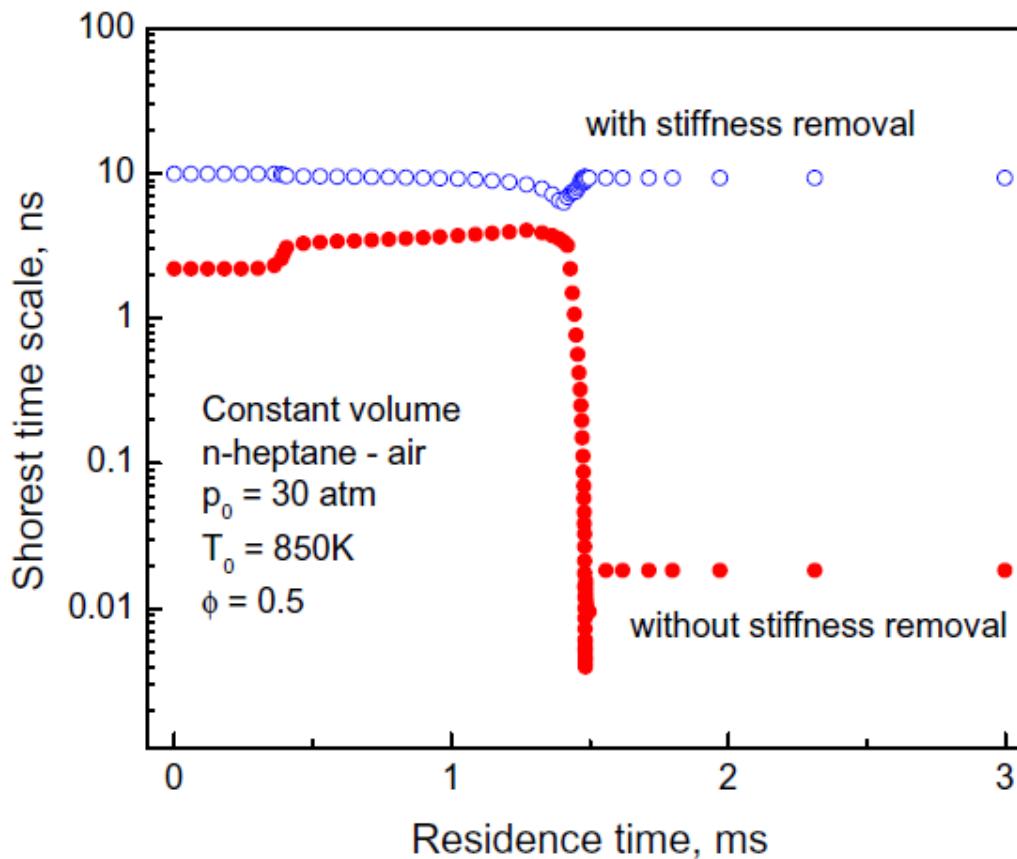
Explicit solver can be used after DCSR

- Time-step limited by CFL condition
- Cost of DNS: $\sim\sigma(N_S)$

The classification of QSS species and PE reactions is usually time-consuming. Thus, it must be performed *a priori* through tabulation or regression networks.



Dynamic Chemical Stiffness Removal (DCSR)



Plot from: Lu et al., *Dynamic stiffness removal for direct numerical simulations*, Combustion and Flame 156, p. 1542-1551 (2009)

Shortest time-scale associated with the eigen-modes of the Jacobian, evaluated with and without the stiffness removal procedure, respectively, for constant-volume homogeneous auto-ignition for an initial pressure of 30 atm, equivalence ratio of 0.5, and initial temperature of 850 K.

Notes

The DCSR is particularly suitable for DNS, because of the small integration time-steps imposed by the CFL condition.

The relative reduction error is consequently small due to the small time-step. The stiffness removal procedure should be carefully validated if larger integration steps are assumed, as for example, in large-eddy simulations.

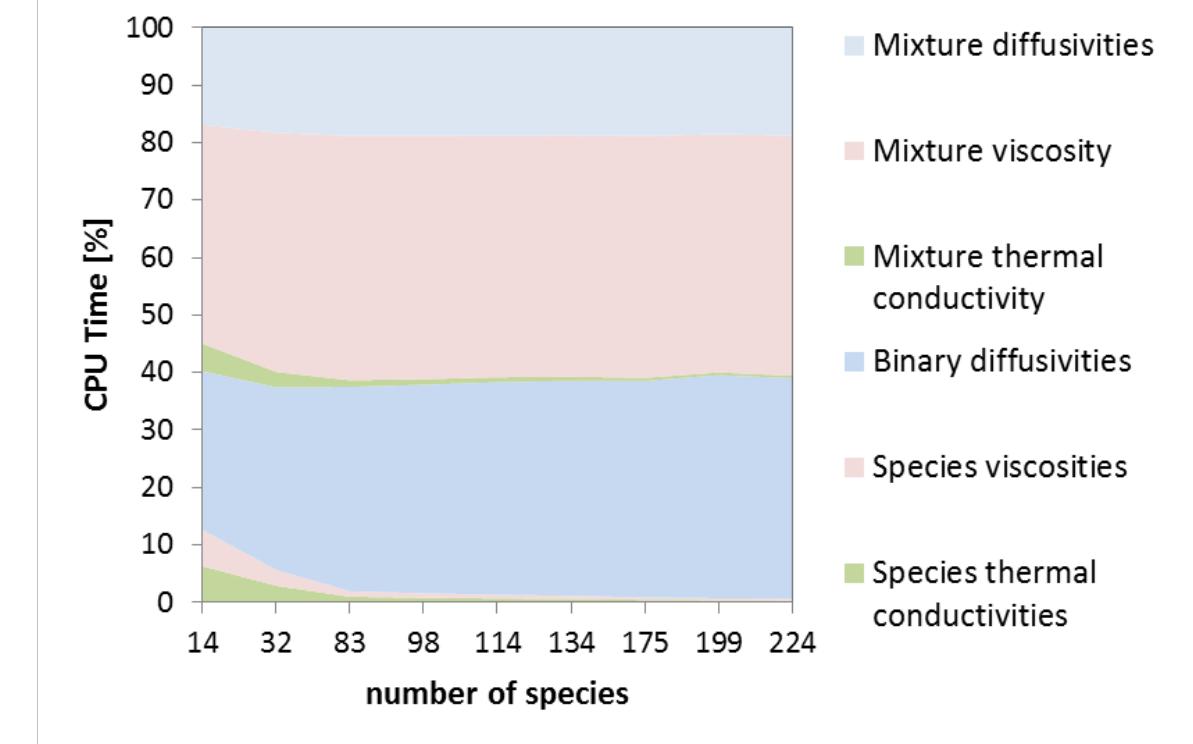
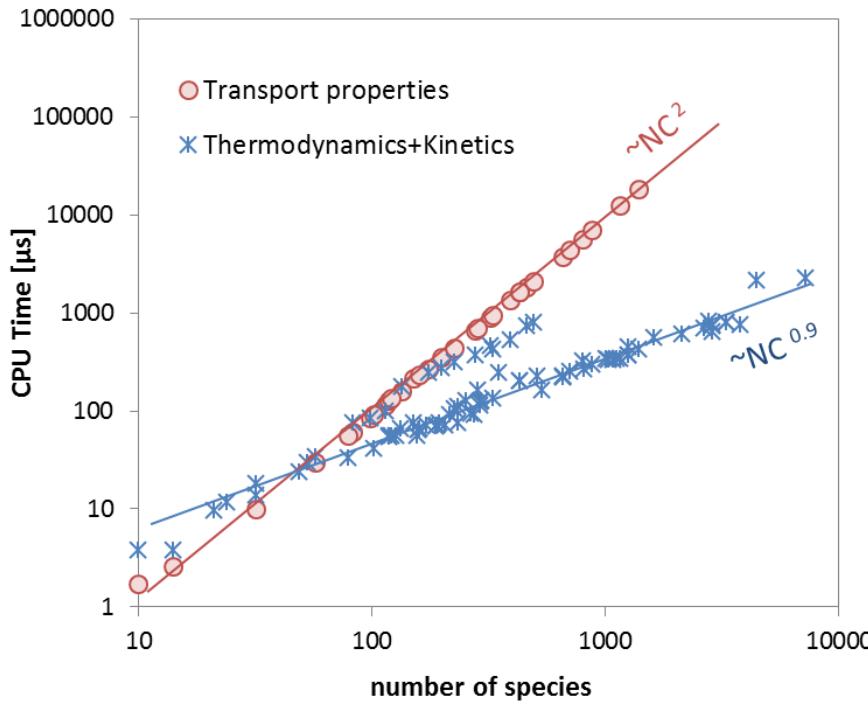


Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



What about transport properties?



- Mixture-averaged models are usually adopted
- The cost of evaluation of transport properties (in particular diffusion) increases quadratically with the number of species
- For large mechanisms (>100 species) the computational cost of transport properties is not negligible
- In fully-coupled methods proper techniques must be applied to reduce the computational cost of transport properties (they can be the bottleneck in evaluation of Jacobian matrix)





Species bundling for diffusion coefficient reduction

Global computational cost

$$C = C_0 + \alpha N_S + \beta N_S^2 + \gamma N_S^3$$

Jacobian evaluation

Computation over-head

Jacobian factorization (with implicit solvers)

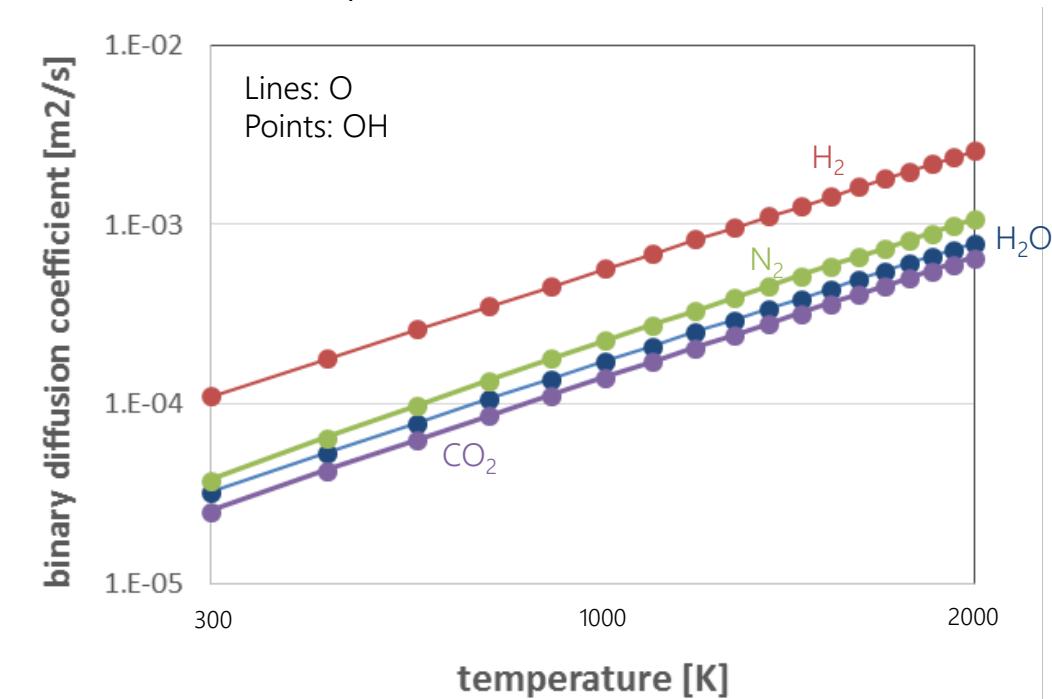
Diffusion

Many species possess similar diffusivities because of similar molecular properties (molecular weight, structure, collision cross section, etc.)

Such species are expected to behave similarly in terms of diffusive transport

Species with similar diffusivities can be bundled in a same group with a representative species

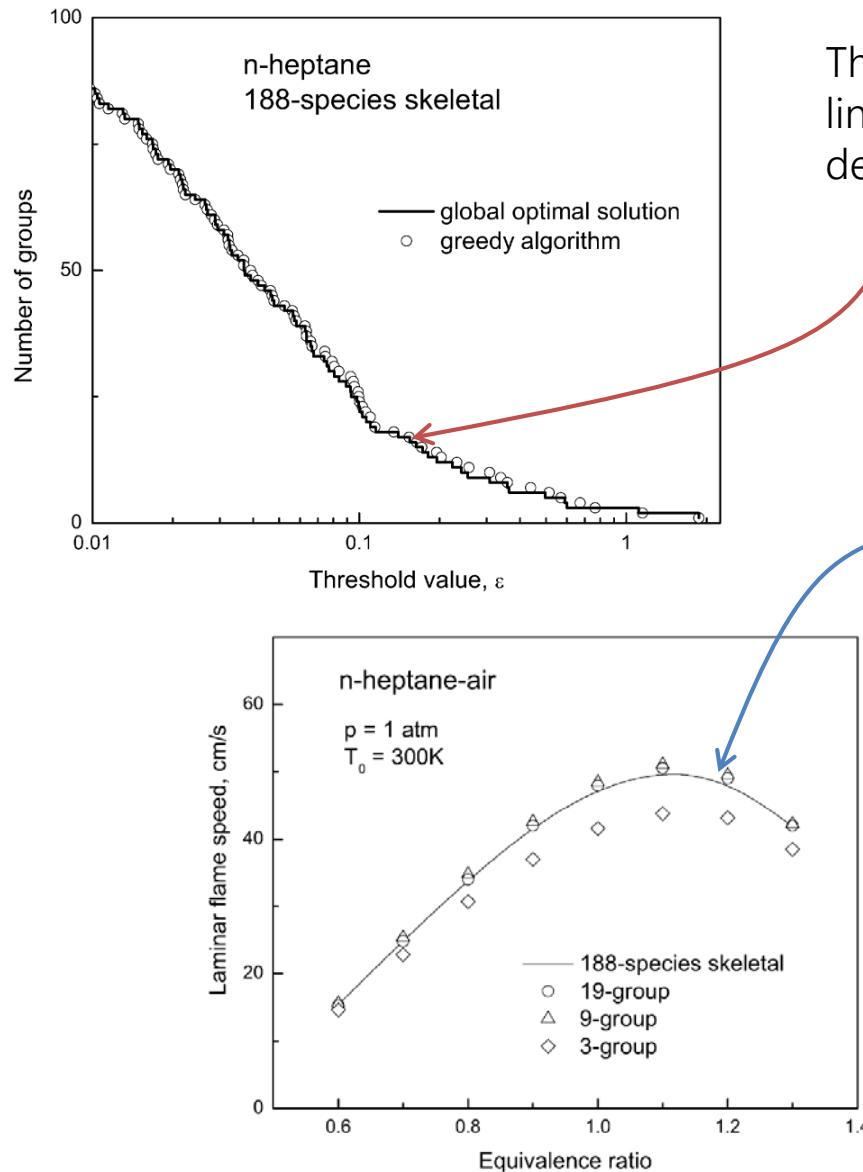
The diffusivities of O and OH with other species are almost identical



Lu, Law, *Diffusion coefficient reduction through species bundling*, Combustion and Flame, 148, p. 117-126 (2007)



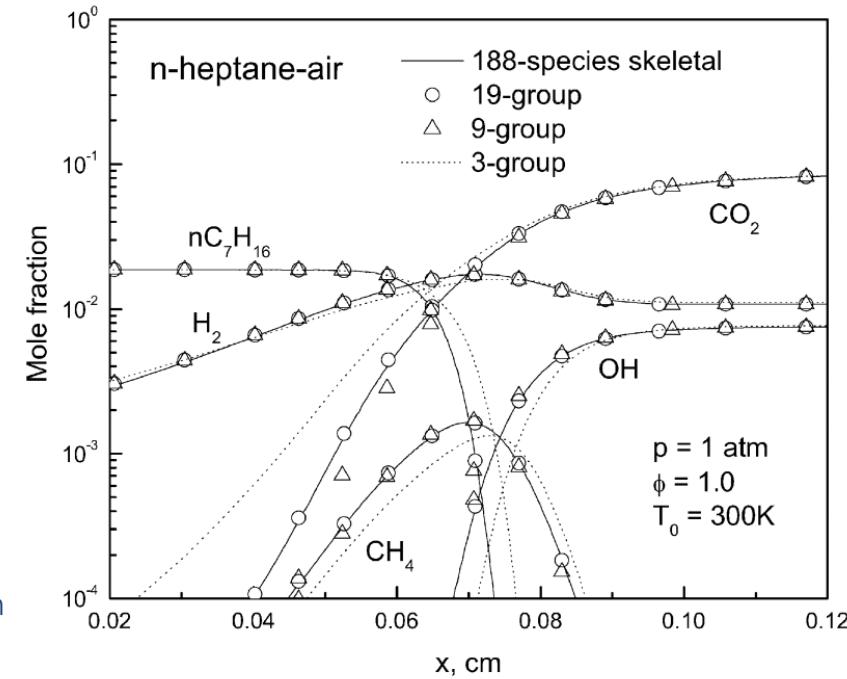
Performances



The number of species groups decreases dramatically as the ϵ value increases. A roughly linear trend is observed in the reduction curve for $\epsilon < 0.1$ in the log plot, showing the rapid decrease in the number of groups for even a slight increase in the reduction error.

The worst error in the flame speed is about 1 cm/s for 19 and 9 groups, a value which is smaller than typical uncertainties in experimental measurements

For a premixed, flat laminar flame, the 19-group model agrees very well with the original model and there is no almost visible error



Plots from: Lu, Law, *Diffusion coefficient reduction through species bundling*, Combustion and Flame, 148, p. 117-126 (2007)

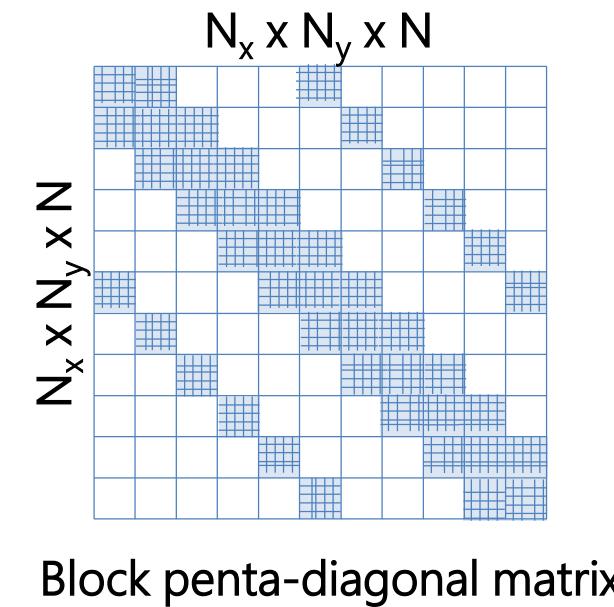
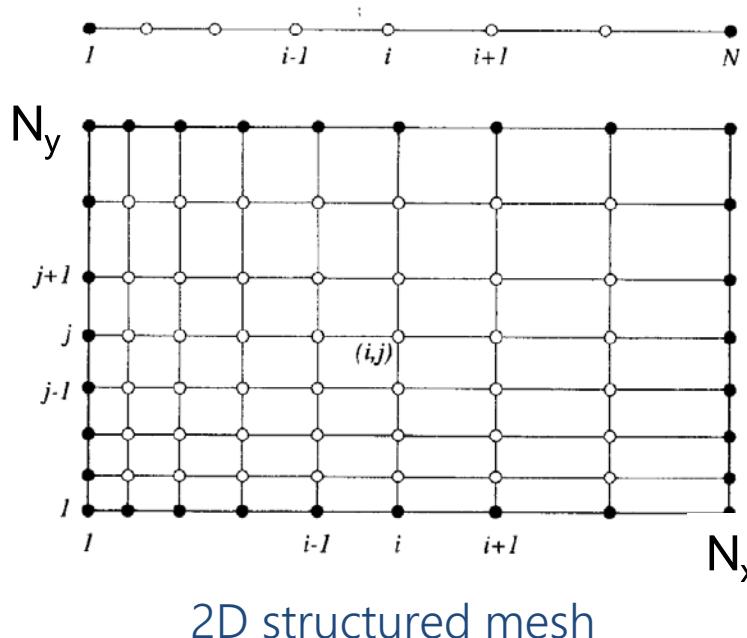


Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



Multidimensional simulations



Fully-coupled methods are unsuitable for 3D (and in most cases also for 2D) when detailed kinetic mechanisms are employed

Example: 2D case

100 x-points x 100 y-points x 500 species $\sim 5,000,000$ eqs

Single block dimension $\sim 500 \times 500 = 250,000$

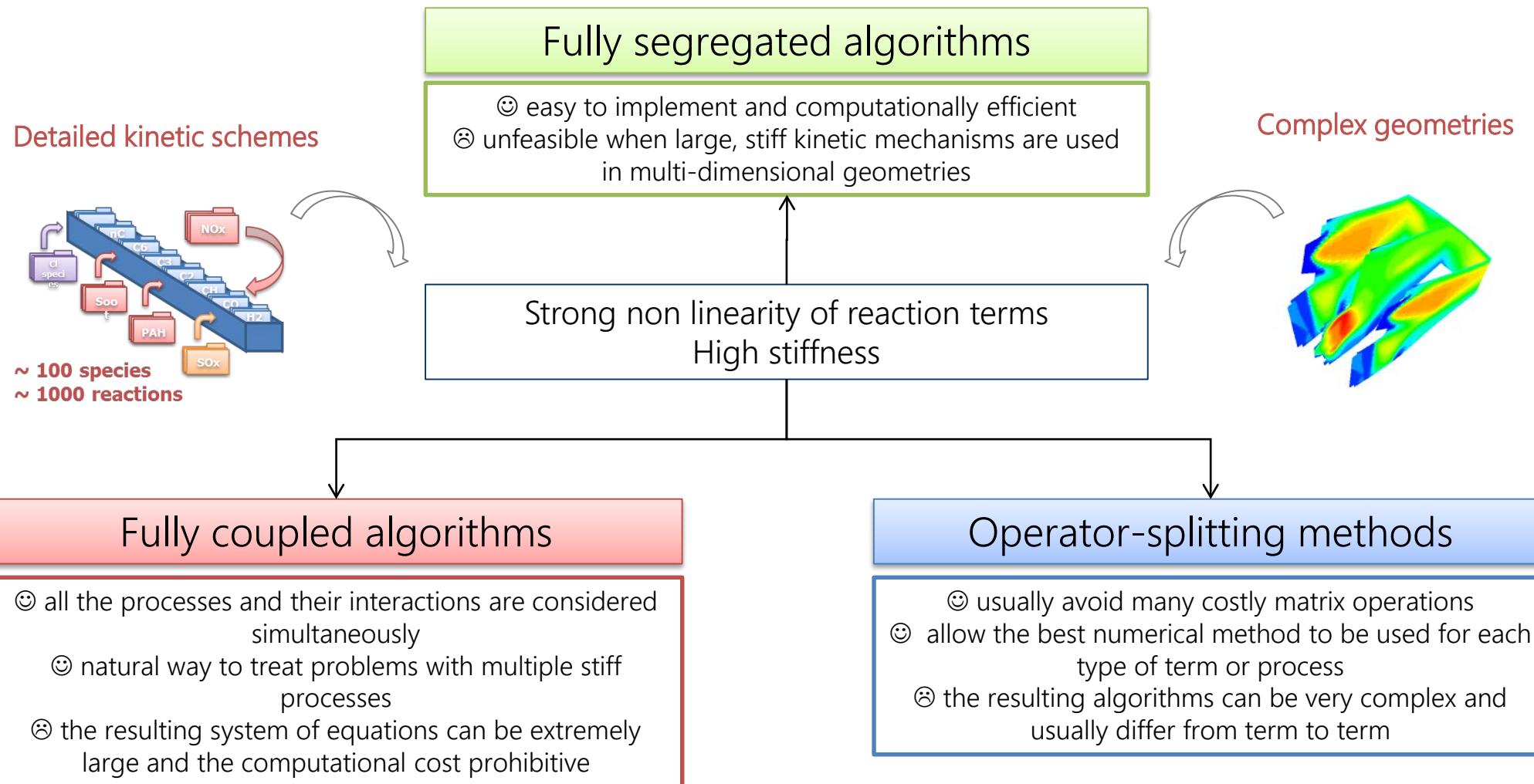
Number of blocks $\sim 5 \times 10,000 = 50,000$

Example: 3D case

100 x-points x 100 y-points x 100 z-points x 500 species $\sim 500,000,000$ eqs



Fully-coupled vs Operator-Splitting Methods





Operator-splitting methods (I)

$$\frac{\partial \Psi}{\partial t} = \mathbf{f} \quad \longrightarrow \quad \frac{\partial \Psi}{\partial t} = \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_3 + \cdots + \mathbf{f}_M$$

Oran, Boris, *Numerical simulation of reactive flows*, Cambridge University Press (2001)

The right hand side function \mathbf{f} has been broken into M different (constituent) processes

$$\begin{cases} \Delta \Psi_1^n = \Delta t \mathbf{f}_1^n \\ \Delta \Psi_2^n = \Delta t \mathbf{f}_2^n \\ \Delta \Psi_3^n = \Delta t \mathbf{f}_3^n \\ \dots \\ \Delta \Psi_M^n = \Delta t \mathbf{f}_M^n \end{cases}$$

$$\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_3 + \cdots + \mathbf{f}_M$$

convection diffusion reaction

Each function in the set \mathbf{f} contributes some amount to the overall change in Ψ during the numerical time step

$$\Psi^{n+1} = \Psi^n + \sum_{m=1}^M \Delta \Psi_m^n$$

The solution for the new values Ψ^{n+1} is found by summing all of the partial contributions

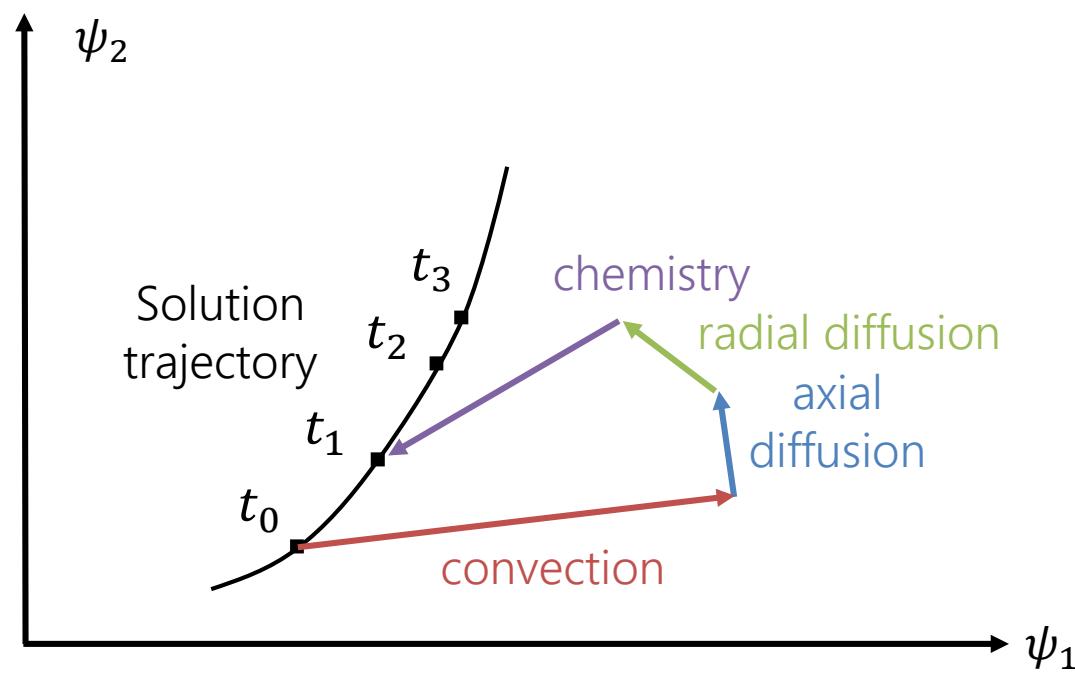




Operator-splitting methods (II)

Each process is simulated individually and the results are combined together

- ✓ allow the best numerical method to be used for each type of term or process (modular simulation paradigm)
- ✓ Parallel architectures can be better exploited
- ✓ the time step must be chosen carefully, in order to avoid instabilities (stability is not guaranteed)
- ✗ Less strain on computational resources, but more efforts on the part of the modeler



Adapted from: Oran, Boris, *Numerical simulation of reactive flows*, Cambridge University Press (2001)

In steady-state conditions, each loop closes perfectly since the solution does not change

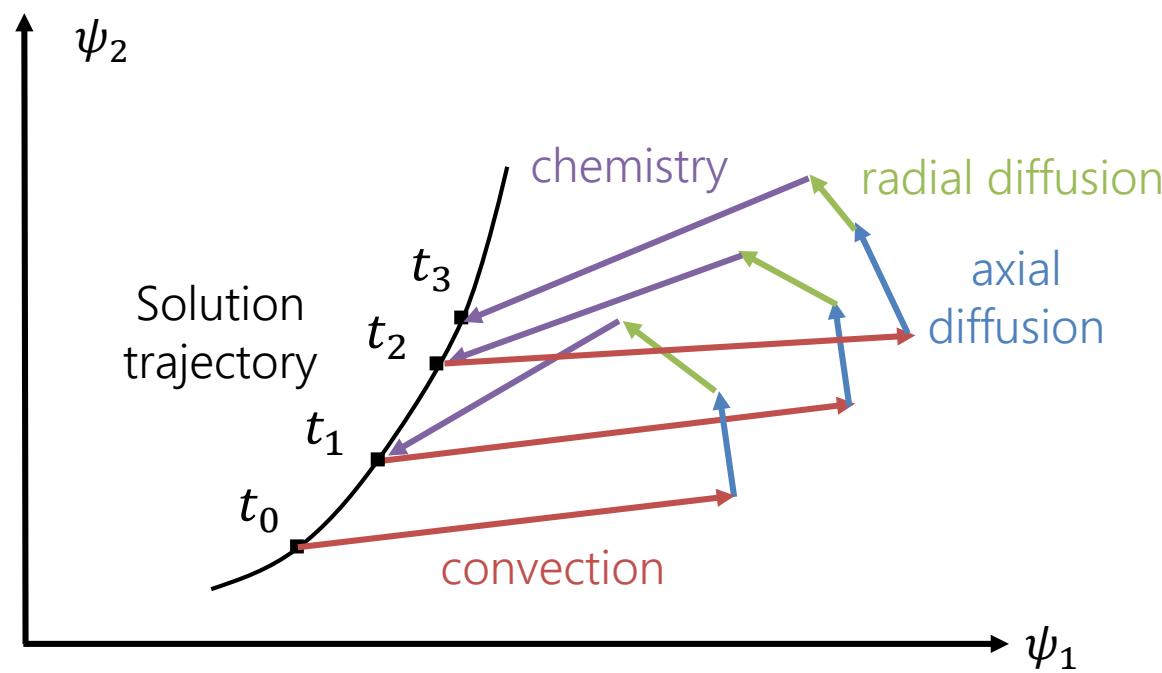




Operator-splitting methods (II)

Each process is simulated individually and the results are combined together

- ✓ allow the best numerical method to be used for each type of term or process (modular simulation paradigm)
- ✓ Parallel architectures can be better exploited
- ✓ the time step must be chosen carefully, in order to avoid instabilities (stability is not guaranteed)
- ✗ Less strain on computational resources, but more efforts on the part of the modeler



Adapted from: Oran, Boris, *Numerical simulation of reactive flows*, Cambridge University Press (2001)

In steady-state conditions, each loop closes perfectly since the solution does not change



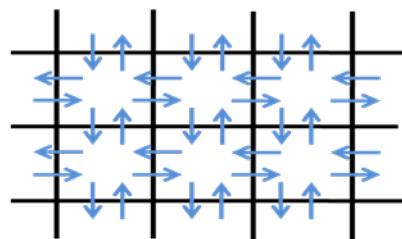
Strang-splitting algorithm (I)

$$\frac{\partial \Psi}{\partial t} = \mathbf{M} + \mathbf{S}$$

→ *Transport
(convection + diffusion)*
→ *Chemistry
(reactions)*

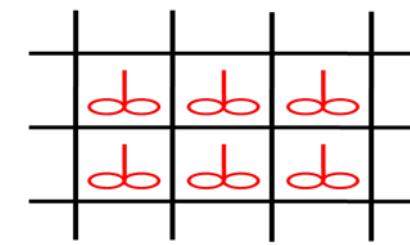
$$\begin{cases} \Delta \Psi_1^n = \frac{\Delta t}{2} \mathbf{S}^n \\ \Delta \Psi_2^n = \Delta t \mathbf{M}^n \\ \Delta \Psi_3^n = \frac{\Delta t}{2} \mathbf{S}^n \end{cases}$$

Transport step (Δt)



$$\frac{\partial \Psi}{\partial t} = \mathbf{M}$$

Chemical step (Δt)



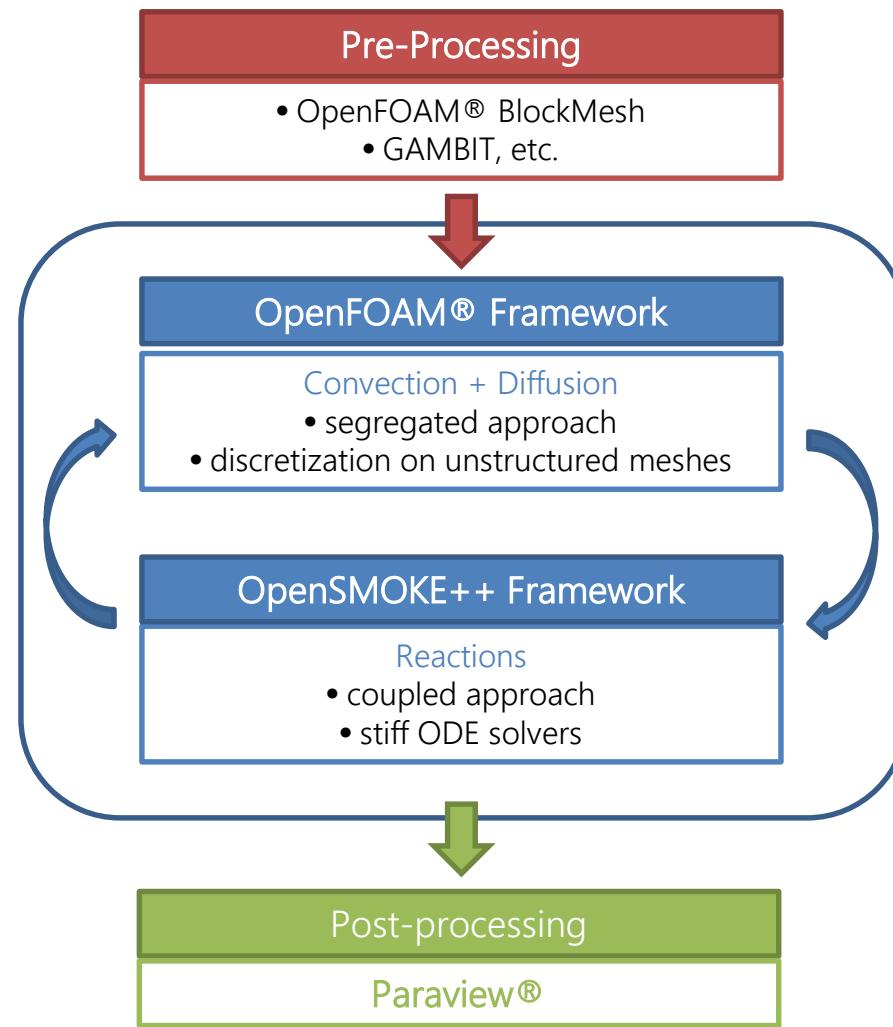
$$\frac{\partial \Psi}{\partial t} = \mathbf{S}$$

Strang, *On the construction and comparison of difference schemes*.
SIAM Journal of Numerical Analysis, 5, p. 506-517 (1968)

Ren, Pope, *Second-order splitting schemes for a class of reactive systems*. Journal of Computational Physics, 227 p. 8165-8176 (2008)



The laminarSMOKE code



- ✓ Multidimensional, finite-volume code
- ✓ Structured and unstructured meshes
- ✓ Unsteady solutions
- ✓ Kinetics in CHEMKIN format
- ✓ Several ODE solvers (BzzMath, CVODE, DVODE, RADAU, LSODE, LSODA)
- ✓ Freely available (open-source)

The code is freely available for OpenFOAM 2.2 (and higher) at the following address:

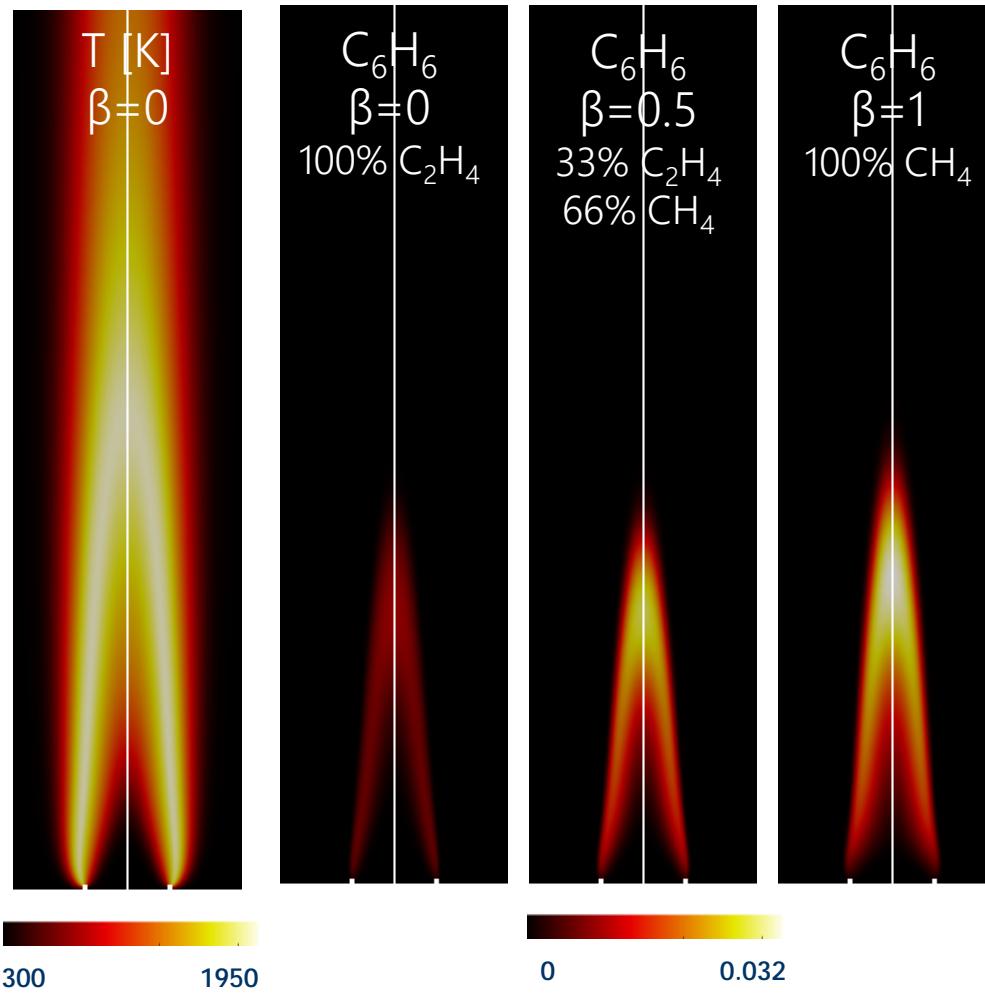
www.opensmoke.polimi.it

Cuoci A., Frassoldati A., Faravelli T., Ranzi E., *Numerical modeling of laminar flames with detailed kinetics based on the operator-splitting method*, Energy & Fuels 27(12), p. 7730-7753 (2013)





C₂H₄/CH₄/N₂ coflow flames (I)



Flame details

Fuel: CH₄/C₂H₄
Air: O₂/N₂ (23.2%, 76.8% mass)

V_{fuel}: 12.52 cm/s

V_{air}: 10.50 cm/s

Fuel nozzle diameter: 11.1 mm
Chamber diameter: 110 mm

Computational details

Domain: 2D axisymmetric (55 x 200 mm)

Computational grid: ~25,000 cells

Discretization: second order centered

Kinetic scheme

POLIMI-HT1212:

198 species, 6307 reactions

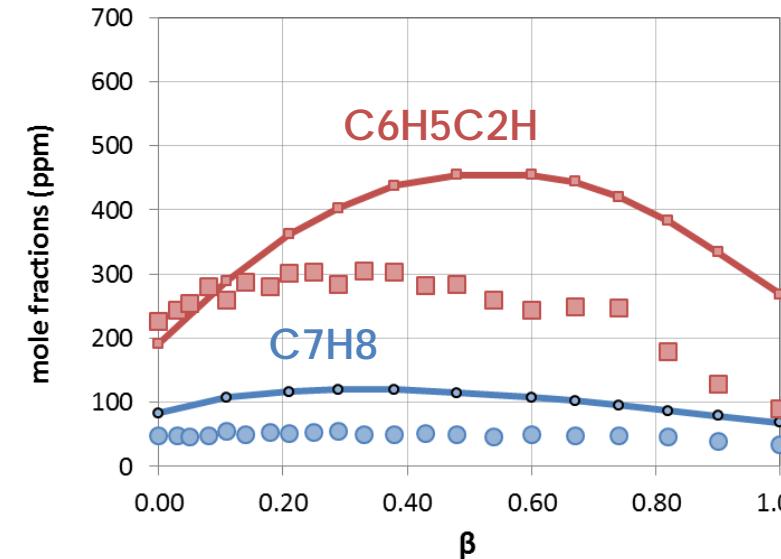
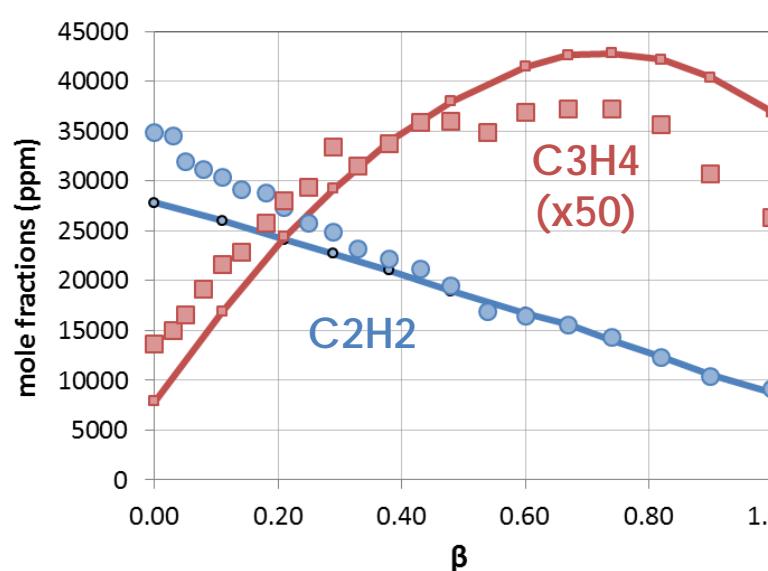
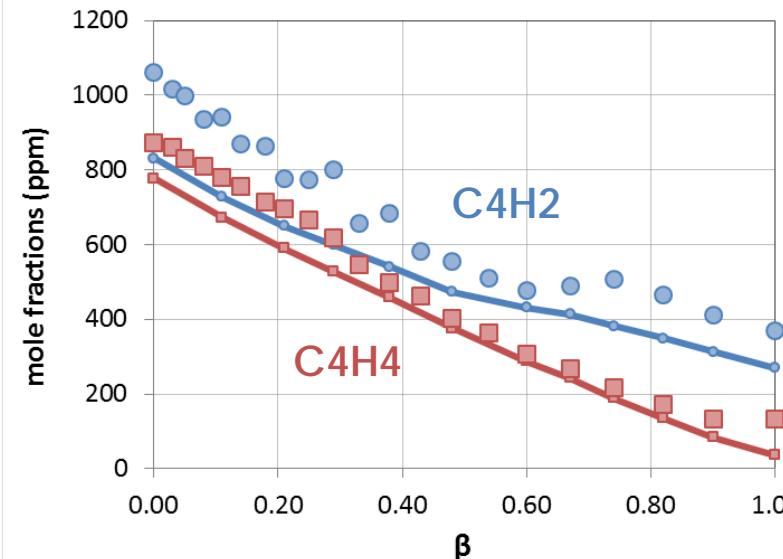
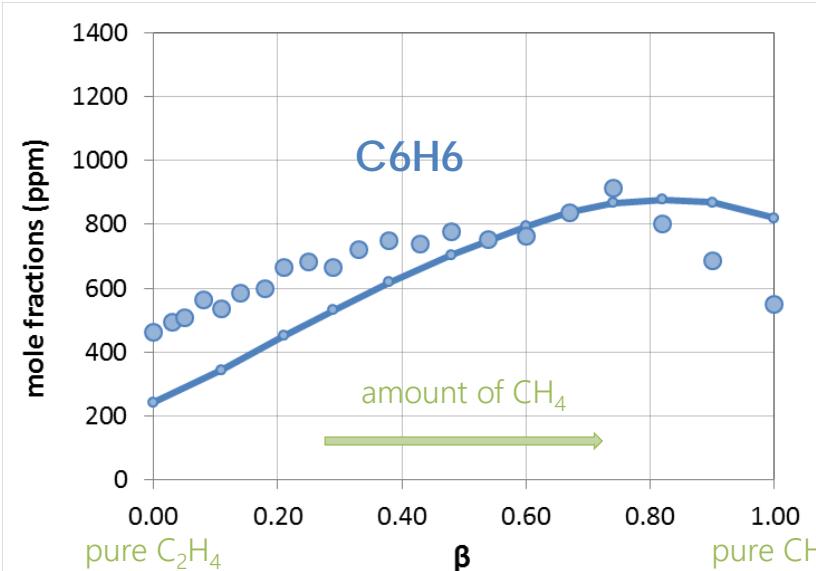
The concentrations of C₂H₄ and CH₄ are identified by the mixture parameter β :

$$\beta = \frac{X_{CH_4}}{X_{CH_4} + 2X_{C_2H_4}}$$

Cuoci A., Frassoldati A., Faravelli T., Ranzi E., *A computational tool for the detailed kinetic modeling of laminar flames: application to C₂H₄/CH₄ coflow flames*. Combustion and Flame, 160(5), p. 870-886 (2013)



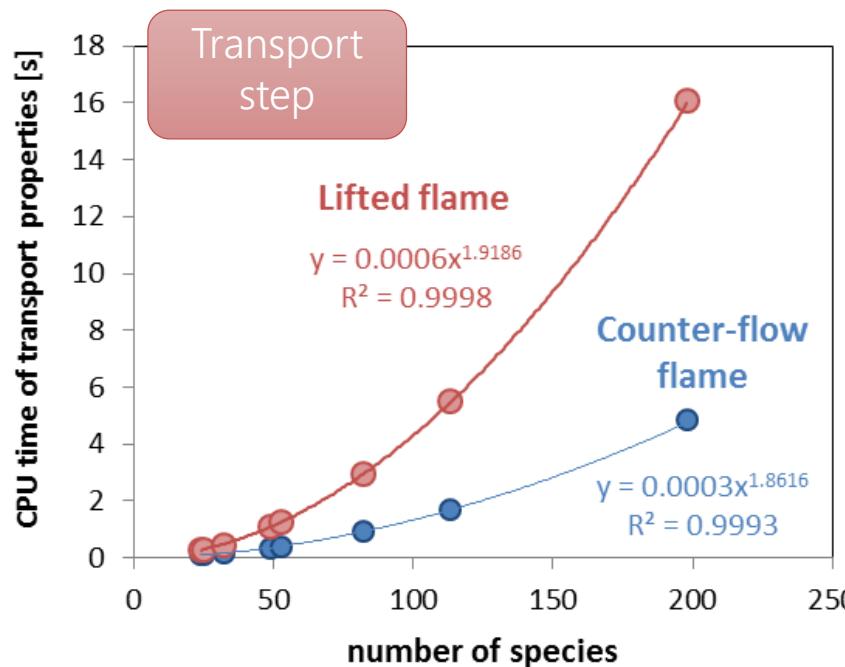
$\text{C}_2\text{H}_4/\text{CH}_4/\text{N}_2$ coflow flames (II)



The peak values (along the center-line) of mole fractions of several species are reported versus the the β parameter and compared with the experimental measurements (points).

Experimental data from:
J.F. Roesler, M. Martinot, C.S. McEnally, L.D. Pfefferle, J.L. Delfau, C. Vovelle, *Combustion and Flame*, 134 (2003) 249-260.

Computational cost

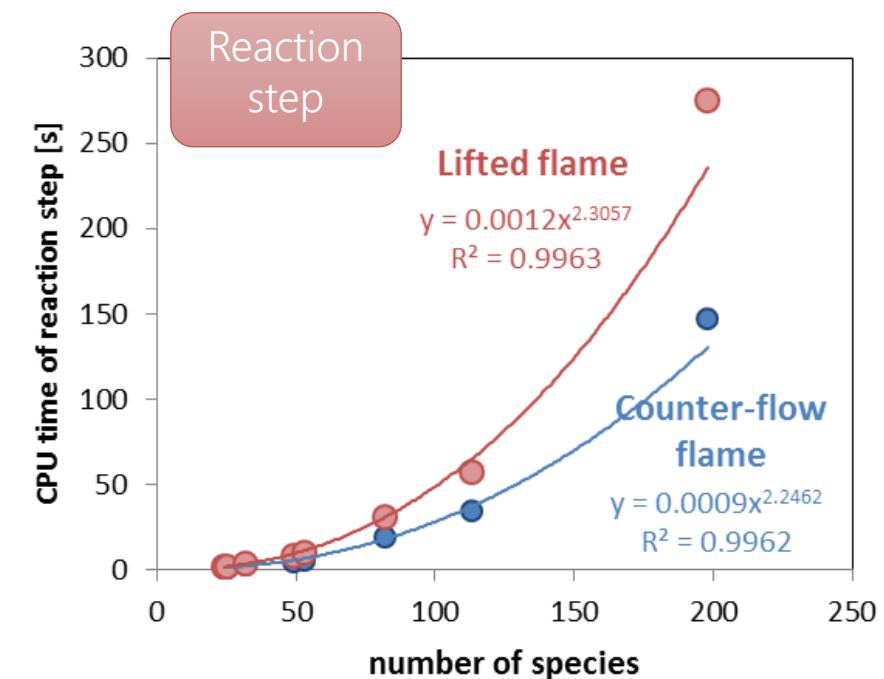


The reaction step results to be the most consuming part of the code, requiring more than 80-85% of the total computational time.

The evaluation of the transport properties and the transport step cover the 5-7% and the 10% of the total time, respectively.

The CPU time of the reaction steps increases more than quadratically (~ 2.3) with the number of species, while the transport properties with a power of ~ 1.8

Increasing the number of species, the relative weight of the reaction step increases.



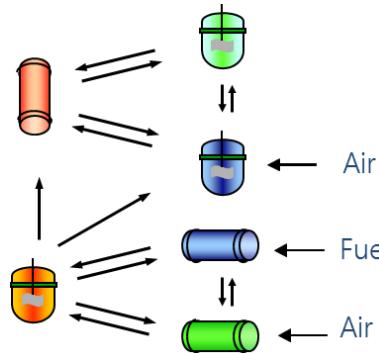
Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



Reducing the number of reacting environments

(Static) Reactor Network Analysis

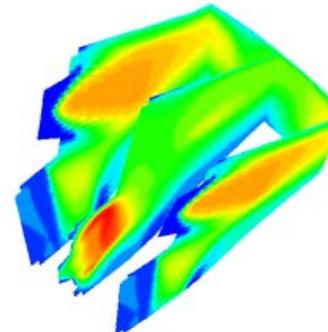


A network of ideal reactors (PSR and PFR) is built on the basis of CFD velocity and temperature field

A detailed kinetic mechanism is adopted to solve the network

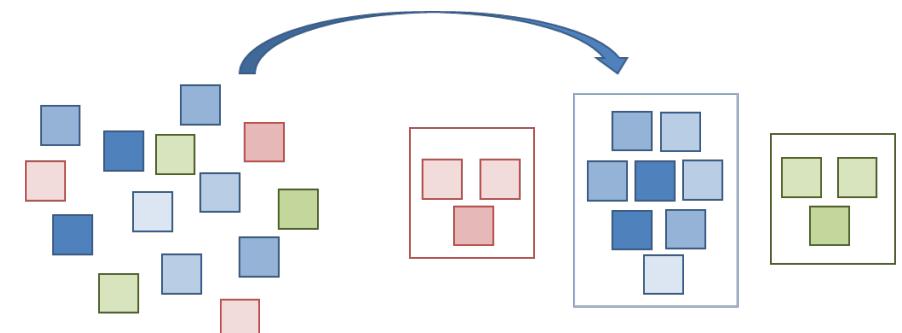
Particularly useful as a post-processing tool

CFD simulation



Dynamic Cell Clustering

Cells having similar thermochemical conditions (not necessarily closed in physical space) are grouped together. This reduces the number of detailed chemistry calculations executed at every time step, as calculations are now executed for a group of cells (i.e. the cluster), and not for each and every cell.

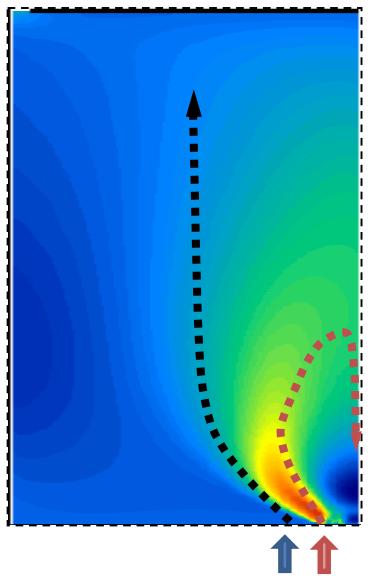


Particularly useful to model internal combustion engines

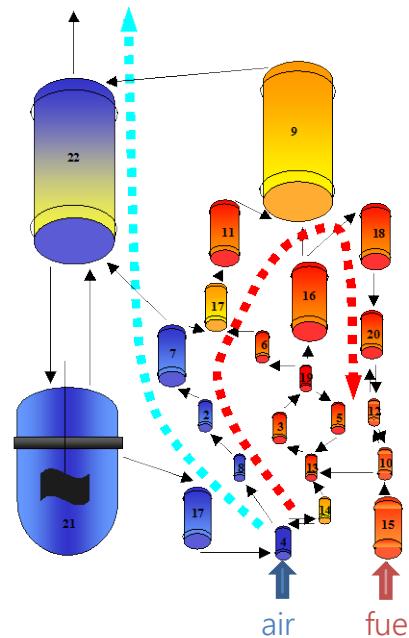


Reaction Network Analysis (RNA) (I)

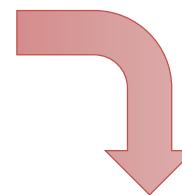
1. CFD Simulation



2. Clustering and network construction



- ✓ The clustering reduces the overall dimensions of the problem
- ✓ According to the clustering, a complex reactor network is constructed



3. Network solution

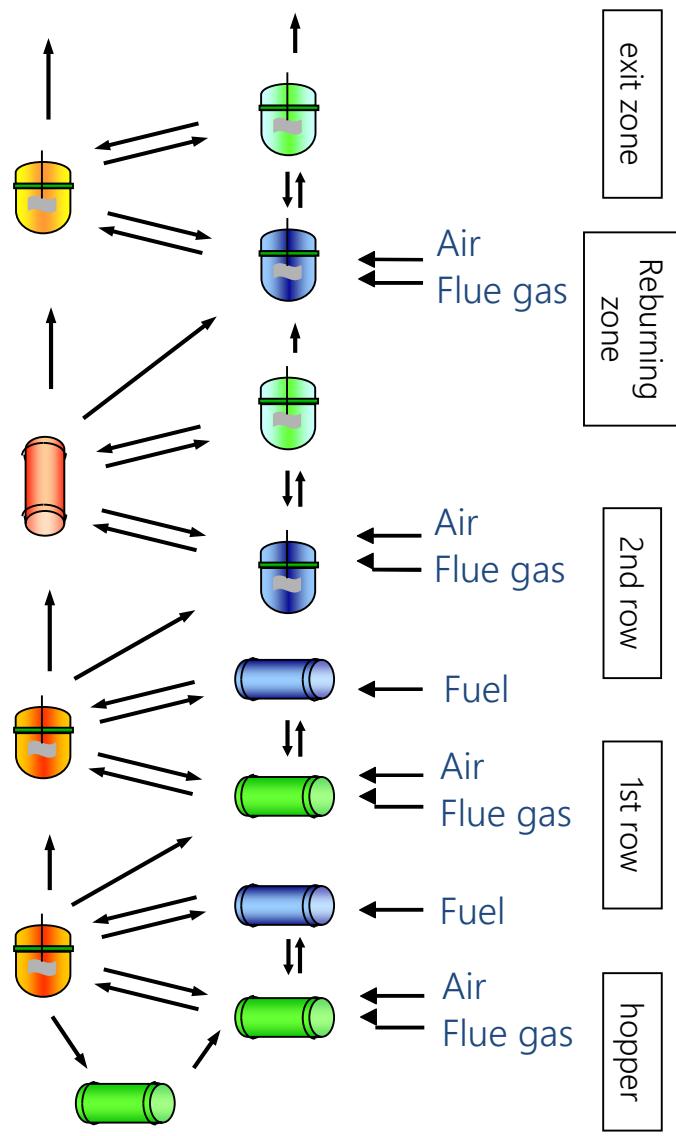
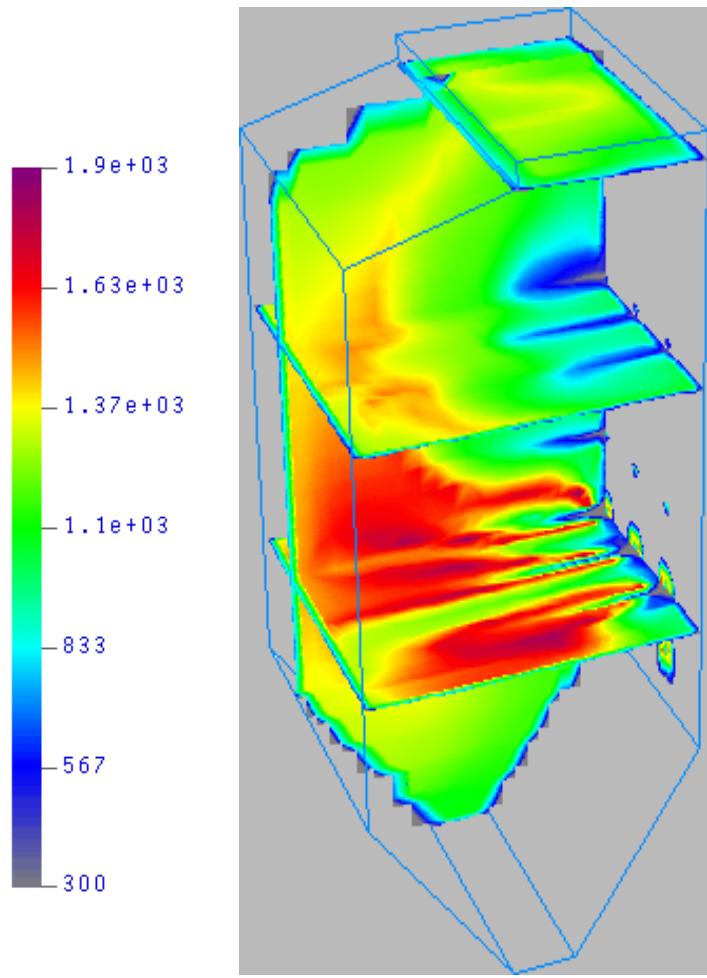
- ✓ A very detailed kinetic scheme is used
- ✓ High number of non linear equations
example: 500 species x 5,000 reactors = 2,500,000 eqs
- ✓ Specifically conceived numerical method

Ehrhardt K. et al., *Modeling of NO_x reburning in a pilot scale furnace using detailed reaction kinetics*. Combust. Sci. Technol. 1998, 131 (1–6), 131–146

Falcitelli M. et al., *CFD + reactor network analysis: An integrated methodology for the modeling and optimization of industrial systems for energy saving and pollution reduction*, Appl. Therm. Eng. 2002, 22 (8), 971–979

Reaction Network Analysis (RNA) (II)

Industrial furnace (Cassano d'Adda, 75 MW)



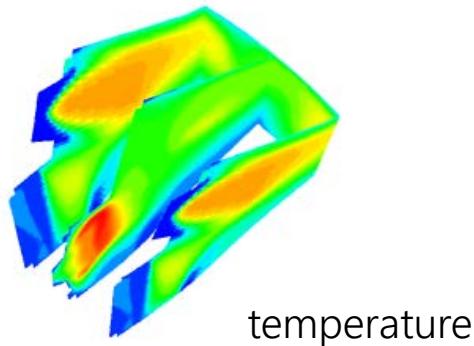
On the basis of the computed three-dimensional (3D) results for flow, temperature, and stoichiometric fields, the volume of the combustion device is reduced to a simplified network of ideal PSRs or plug flow reactors.

Then, within each reactor, a detailed kinetic model is used to predict the concentrations of additional species (especially pollutants like NOx).

WARNING!
it is possible to obtain fairly network-independent results only for proper construction of reactor network

Kinetic Post-Processing

CFD simulation

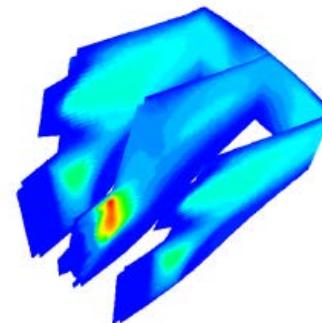


temperature

a steady state CFD simulation of the combustion device is performed using a **global kinetic mechanism**, which allows for the correct prediction of **thermal and flow fields**, but which cannot give us information on pollutant species

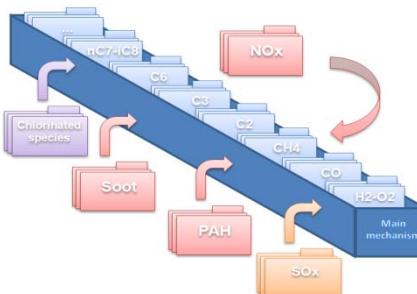
pollutant species usually affect only marginally the main combustion process and consequently do not influence the overall temperature and flow field

Post-Processing



nitrogen oxides
"slow" pollutant species

Detailed chemistry

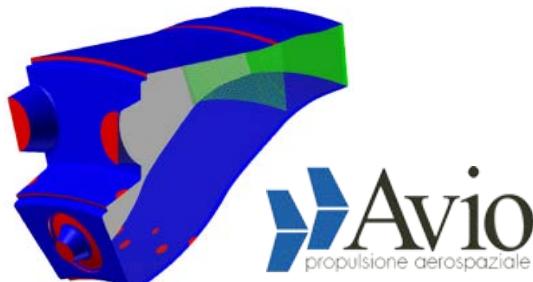


Faravelli T. et al., *A new procedure for predicting NO_x emissions from furnaces*, Comput. Chem. Eng. 2001, 25 (4–6), 613–618

Skjøth-Rasmussen M. et al., *Post-processing of detailed chemical kinetic mechanisms onto CFD simulations*, Comput. Chem. Eng. 2004, 28 (11), 2351–2361

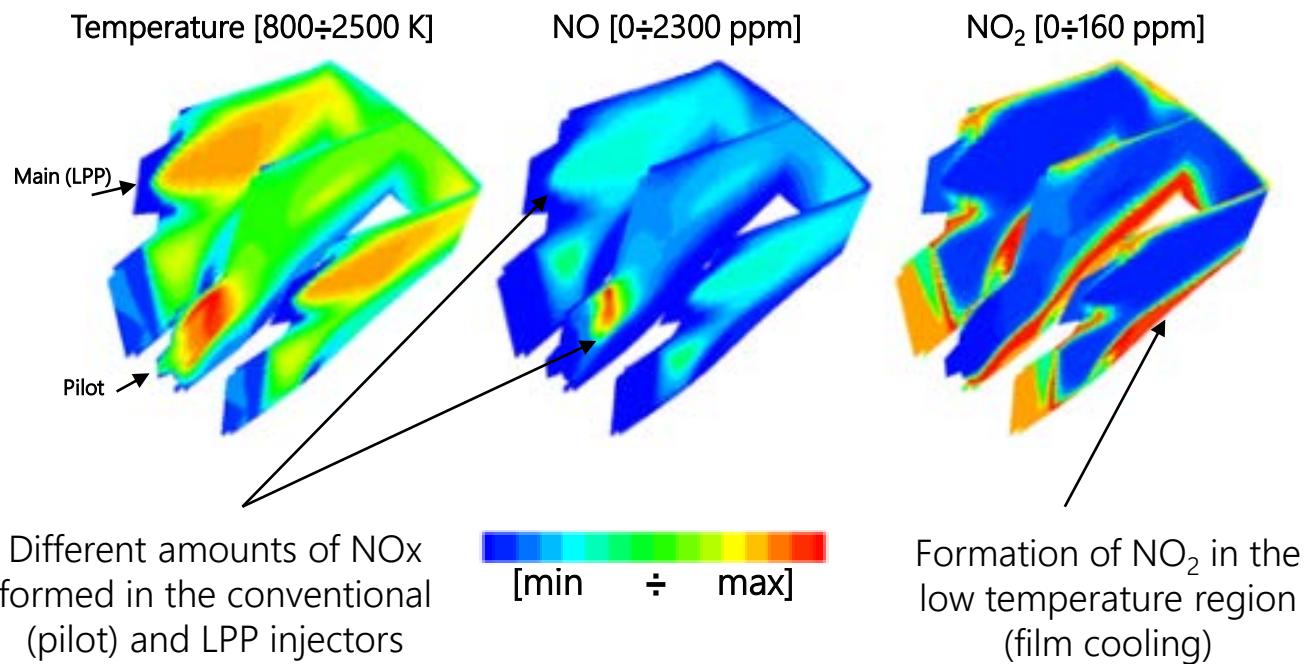
Fichet, V. et al., *A reactor network model for predicting NO_x emissions in gas turbines*, Fuel 2010, 89 (9), 2202–2210.

Low-NO_x combustor for aero-engine turbofan

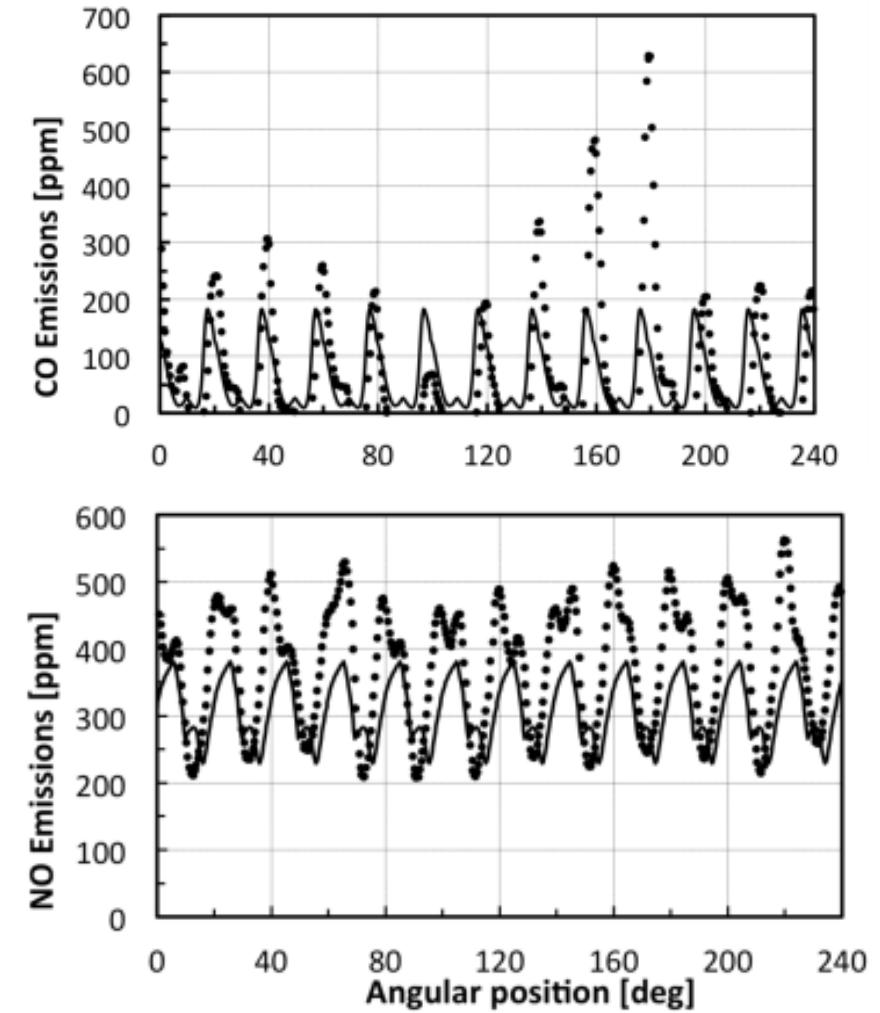


Stagni et al., *A fully coupled, parallel approach for the post processing of CFD data through reactor network analysis*, Computers and Chemical Engineering, 60, p. 197-212 (2014).

Frassoldati et al., *Experimental and modeling study of a low NO_x combustor for aero-engine turbofan*, Combustion Science and Technology 181, p. 483-495 (2009)

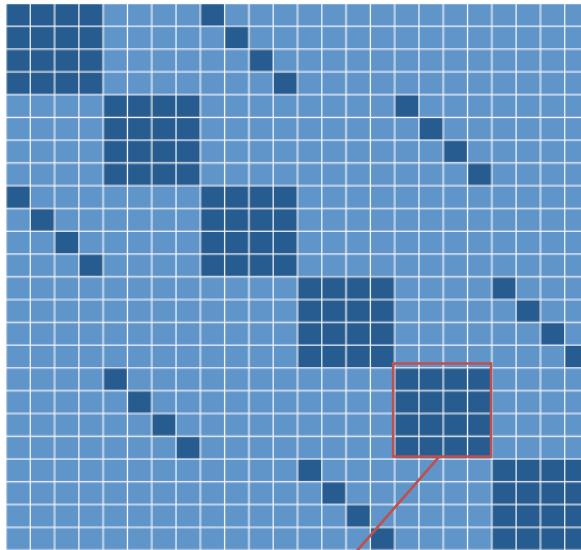


500k cells x 100 species = 50 M equations
Fully-coupled solution!



Non Linear System (NLS) of equations

Jacobian sparsity pattern



Single reactor

- Jacobian is sparse and block-unstructured
- High degree of accuracy is sought

Convection

$$\left(\sum_{k=1}^{N_{AD}} \dot{m}_{k,j} \omega_{k,j,i} - \dot{m}_j \omega_{j,i} \right) + \sum_{k=1}^{N_{AD}} \left(J_{j,k,i} \cdot S_{j,k} \right) + V_j^* \tilde{\Omega}_{j,i} + \Pi_{j,i} = 0$$



Linear

Diffusion

$$J_i = -\frac{\mu_t}{Sc_t} \cdot \nabla \omega_i$$

Linear

Reaction Evaporation

Non Linear
(Power Law)

A fully coupled resolution is implemented

$$C(\boldsymbol{\omega}) + R(\boldsymbol{\omega}) + \mathbf{f} = \mathbf{0}$$

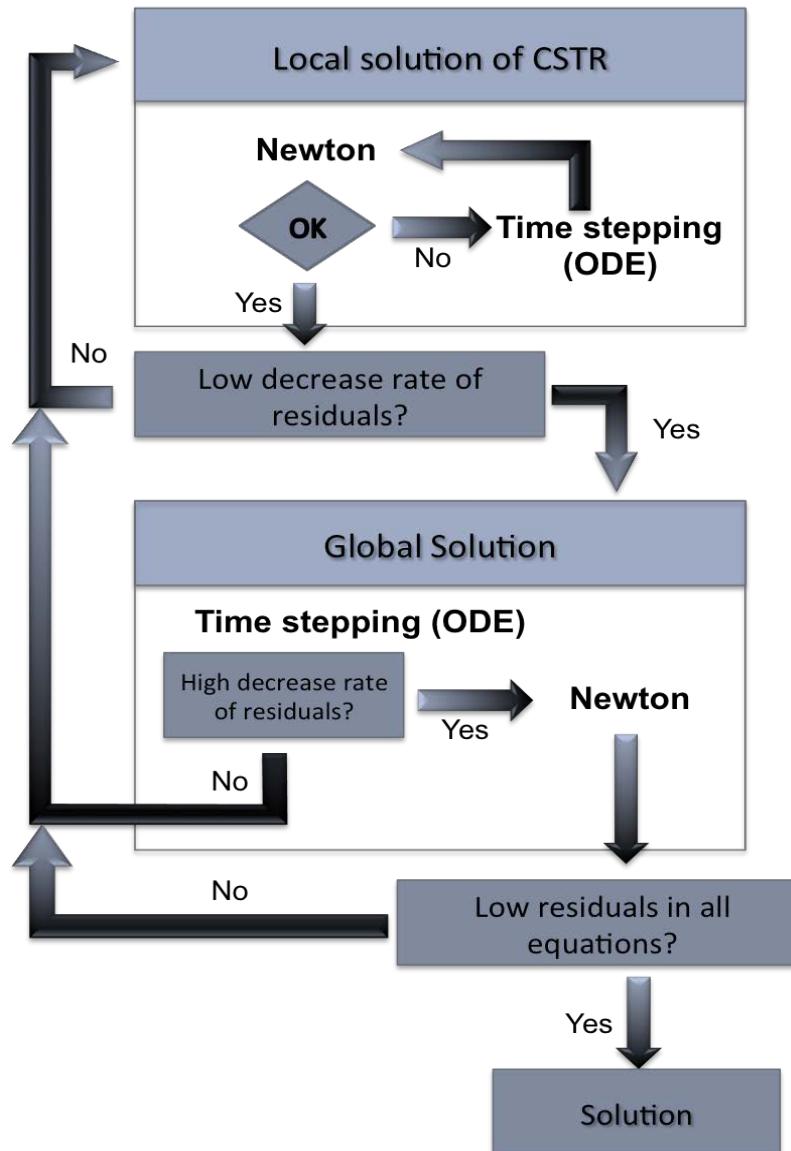
Linear

Non-Linear

External feeds



Numerical procedure



the numerical procedure combines different techniques to obtain the final solution, because the global Newton's method can be successfully applied only if the first-guess solution is close to the real solution.

1. Global Newton's Method
2. Global ODE (Backward Euler)
3. Direct Substitutions (Local solution)
 - a. Local Newton's Method
 - b. Local ODE system (stiff solver)

Cuoci, A., Frassoldati, A., Stagni, A., Faravelli, T., Ranzi, E., Buzzi-Ferraris, G., Numerical modeling of NO_x formation in turbulent flames using a kinetic post-processing technique (2013) Energy and Fuels, 27 (2), pp. 1104-1122, DOI: 10.1021/ef3016987

Local solution

The individual reactors are solved sequentially to take the whole system closer to the solution. This means that each reactor is solved using a local Newton's method.

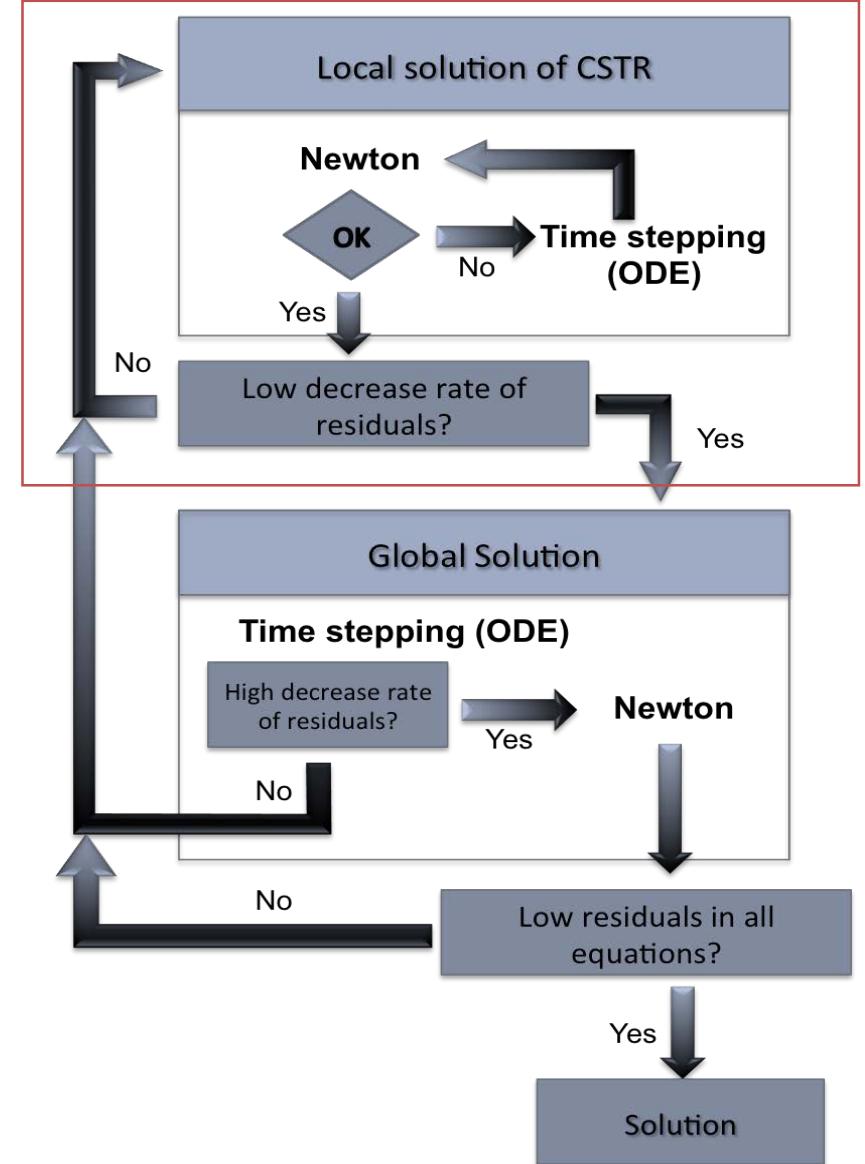
$$[\mathbf{C}_{in}(\boldsymbol{\omega}) + \mathbf{f}]_{old} + \mathbf{C}_{out}(\boldsymbol{\omega}) + \mathbf{R}(\boldsymbol{\omega}) = \mathbf{0}$$

To improve the robustness, especially in the first iteration, a false transient method is used to solve the single reactors. The NLS is transformed into a ODE system by adding the unsteady term

$$\mathbf{m} \frac{d\boldsymbol{\omega}}{dt} = [\mathbf{C}_{in}(\boldsymbol{\omega}) + \mathbf{f}]_{old} + \mathbf{C}_{out}(\boldsymbol{\omega}) + \mathbf{R}(\boldsymbol{\omega})$$

Stiff ODE solvers

CVODE, DVODE, LSODE, RADAU5, BzzMath



Global solution

the global Newton's method, to ensure the accuracy needed to correctly predict chemical species present in very small amounts (ppm or smaller)

$$\mathbf{C}(\boldsymbol{\omega}) + \mathbf{R}(\boldsymbol{\omega}) + \mathbf{f} = \mathbf{0}$$

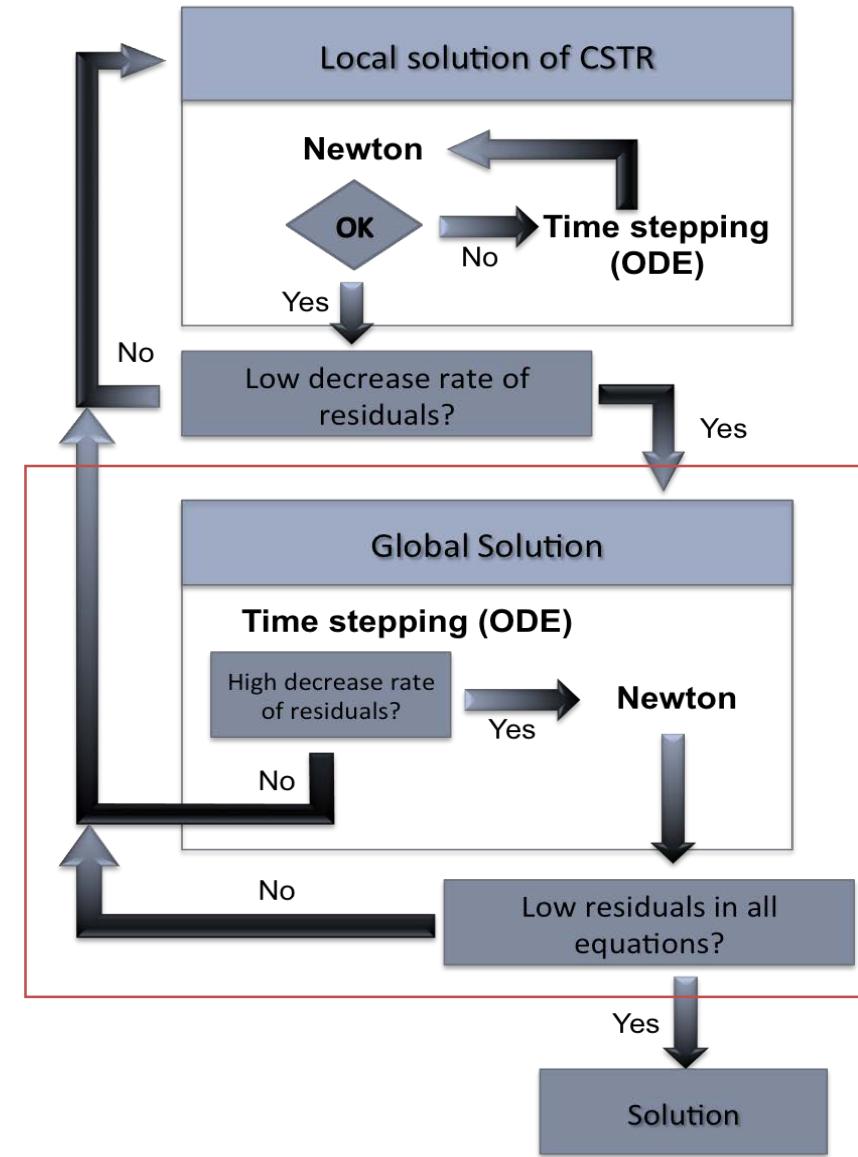
When complex flows are investigated, the sequential approach (i.e., direct substitutions) could not be enough to reduce the residuals of equations to sufficiently small values to successfully apply the global Newton's method. In such a case, a global time-stepping procedure must be taken into account.

$$\mathbf{m}_{tot} \frac{\boldsymbol{\omega}^{n+1} - \boldsymbol{\omega}^n}{\Delta t} = \mathbf{C}(\boldsymbol{\omega}^{n+1}) + \mathbf{R}(\boldsymbol{\omega}^{n+1}) + \mathbf{f}$$

Linear System solvers

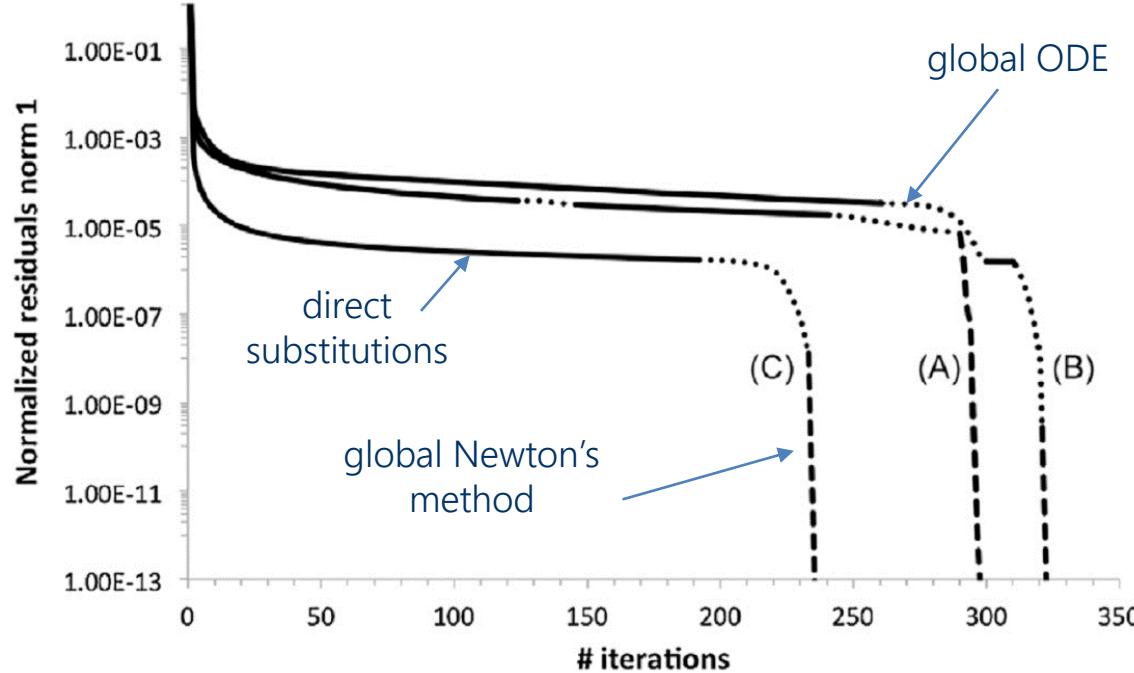
MUMPS 4.10 (Direct Solver)

LIS 1.24 (Iterative Solver)

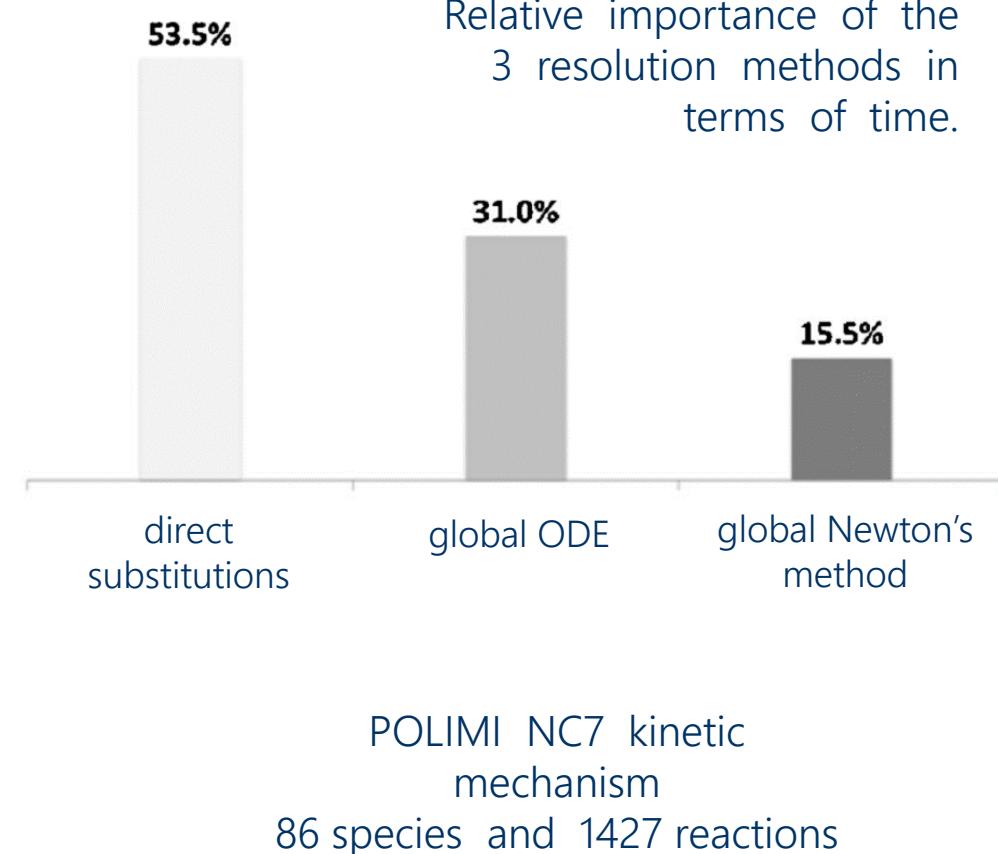


Numerical performances

- A: a tubular combustor (56,150 reactors, 4.8M eqs)
- B: an aircraft combustor (252,885 reactors, 22M eqs)
- C: an aircraft combustor (290,764 reactors, 25M eqs)



Residuals norm 1 trends, normalized with respect to their initial value (set equal to 1)

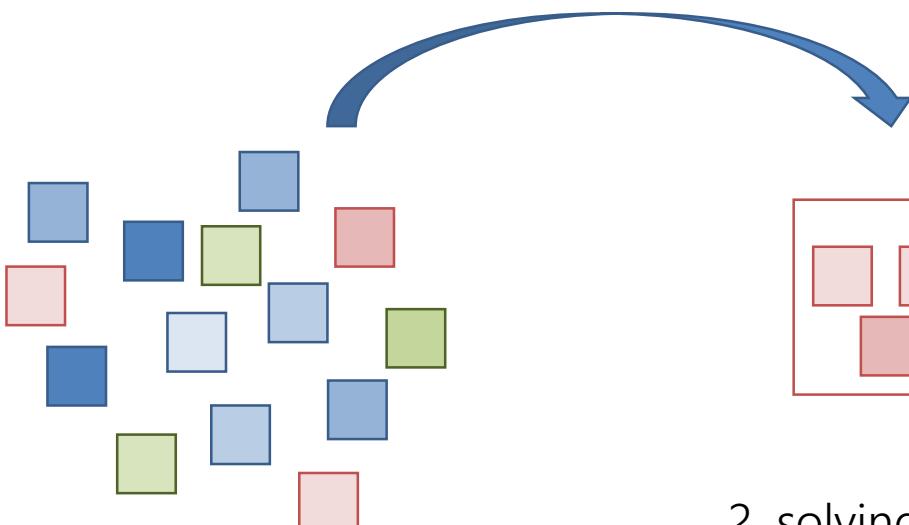


Plots from: Stagni et al., *A fully coupled, parallel approach for the post processing of CFD data through reactor network analysis* Computers and Chemical Engineering, 60, p. 197-212 (2014).

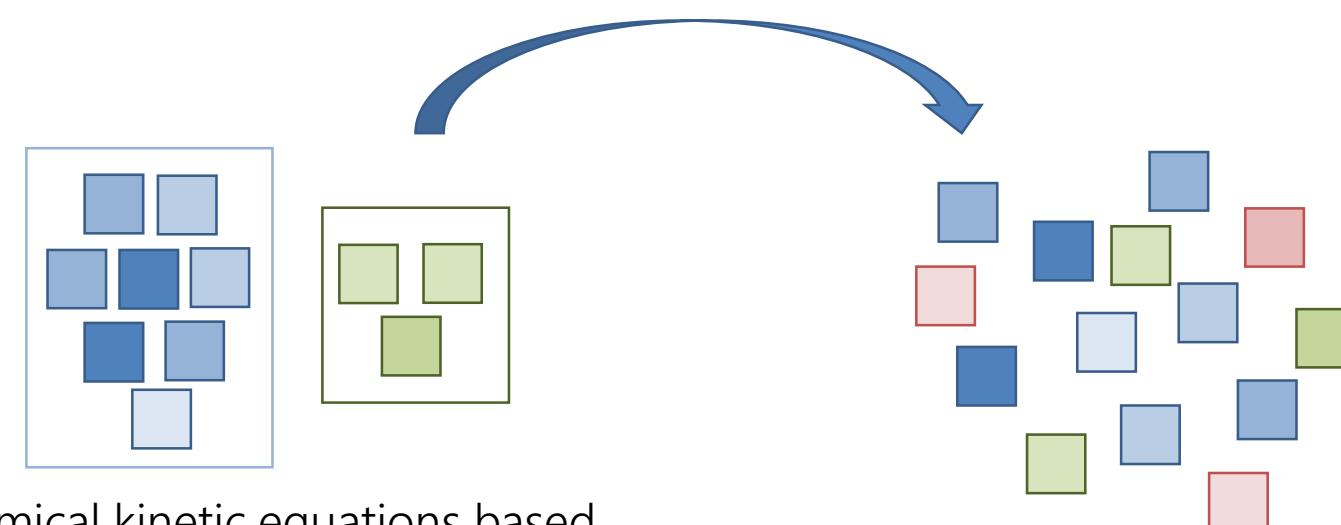
Dynamic Cell Clustering (DCC) (I)

Dynamic Cell Clustering (DCC) dynamically groups/clusters regions of the domain that have similar thermochemical conditions. This reduces the number of detailed chemistry calculations executed at every time step, as calculations are now executed for a group of cells (i.e. the cluster), and not for each and every cell.

1. grouping cells into clusters using an evolutionary data-clustering algorithm



3. mapping the cluster averaged solution back to the individual cells while preserving the initial temperature and species stratification.



2. solving chemical kinetic equations based on cluster averaged state variables



Dynamic Cell Clustering (DCC) (II)

The grouping of computational cells, in the calculation domain, into clusters is achieved by using **clustering algorithms** which identify cells that have similar thermochemical states. Cell temperature and equivalence ratio are typically used as the thermochemical clustering variables.

On which basis can reacting cells be regarded as similar or different?

The chemical kinetic equations are now solved at the cluster and not at the cell level, using averaged values for the state variables. The cluster averaged chemistry solution is then **mapped back** to the individual cells in each cluster.

How to conservatively redistribute the species among the different cells, after integration?

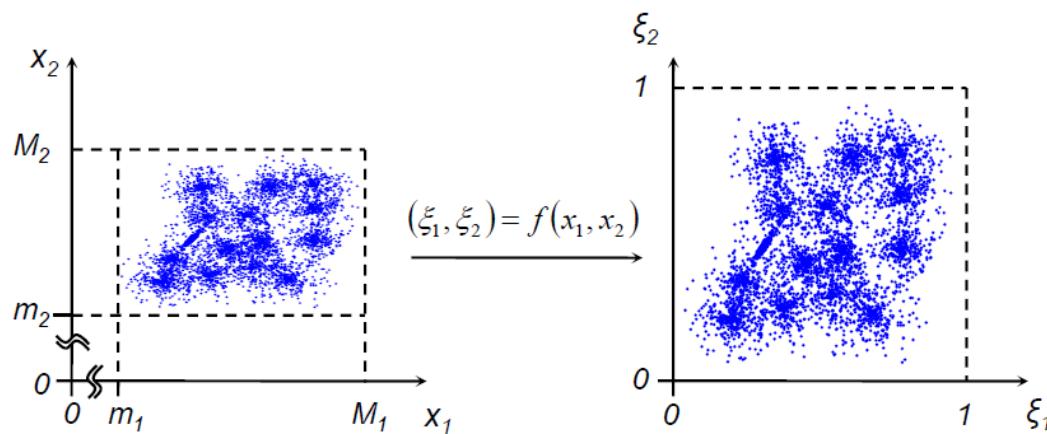
Liang L., Stevens J. G., Farrell J.T., *A Dynamic Multi-Zone Partitioning Scheme for Solving Detailed Chemical Kinetics in Reactive Flow Computations*, Combustion Science and Technology 181(11), p.1345-1371 (2009)

G.M. Goldin, Z. Ren, S. Zahirovic, *A cell agglomeration algorithm for accelerating detailed chemistry in CDF*, Combust. Theory Model., 13, pp. 721–739 (2009)



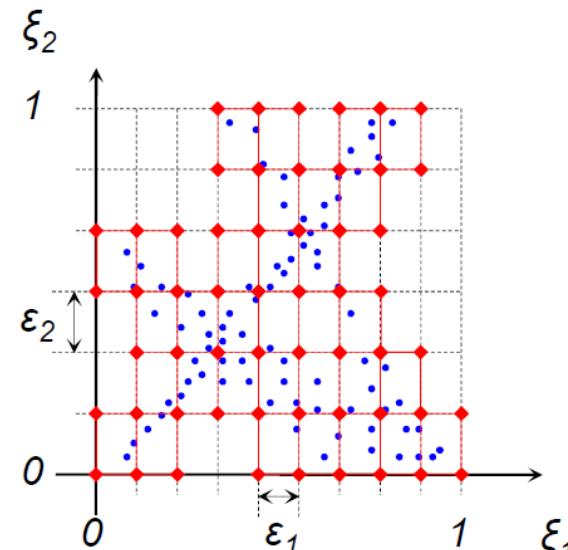
High dimensional cell clustering

- The clustering space is defined as the d-dimensional cell positions in the state space [T; (d-1) mass fractions]
- Normalized to a unity hyper-box



Perini F., High-dimensional, unsupervised cell clustering for computationally efficient engine simulations with detailed combustion chemistry, Fuel 106, p. 344–356 (2013)

Bounding-box clustering



- Cluster initialization as a structured grid
- Each point is contained in a bounding box of 2^d cluster centers
- Clusters have to stay local (bounding-box-constrained k-means algorithm)
- Reduced computational effort (evaluate 2^d distances per point)

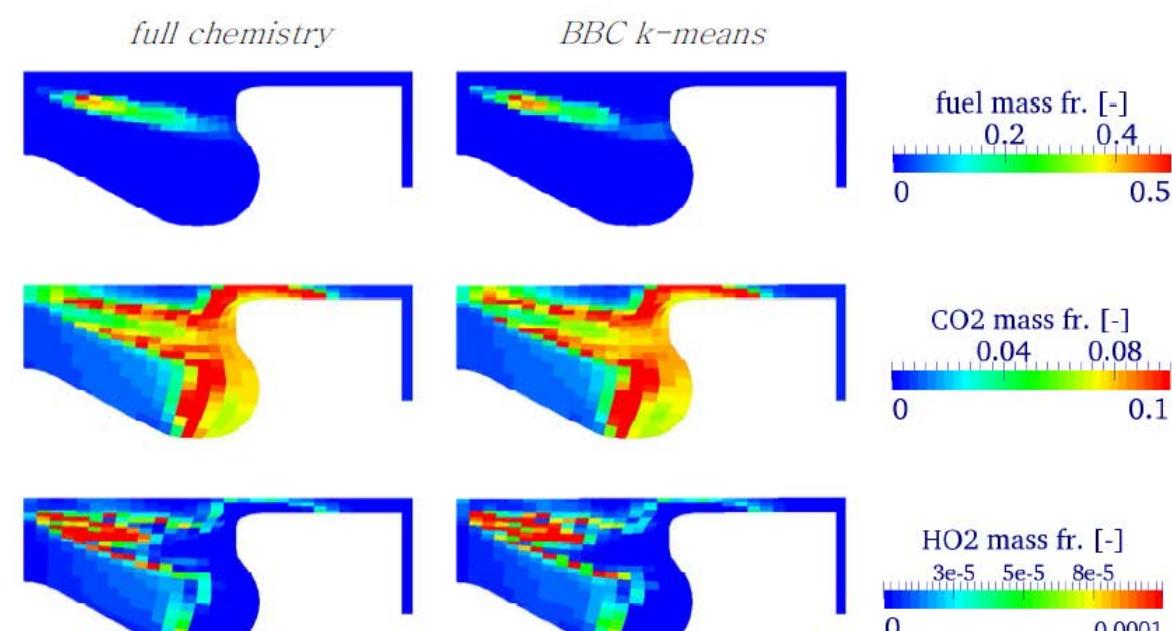
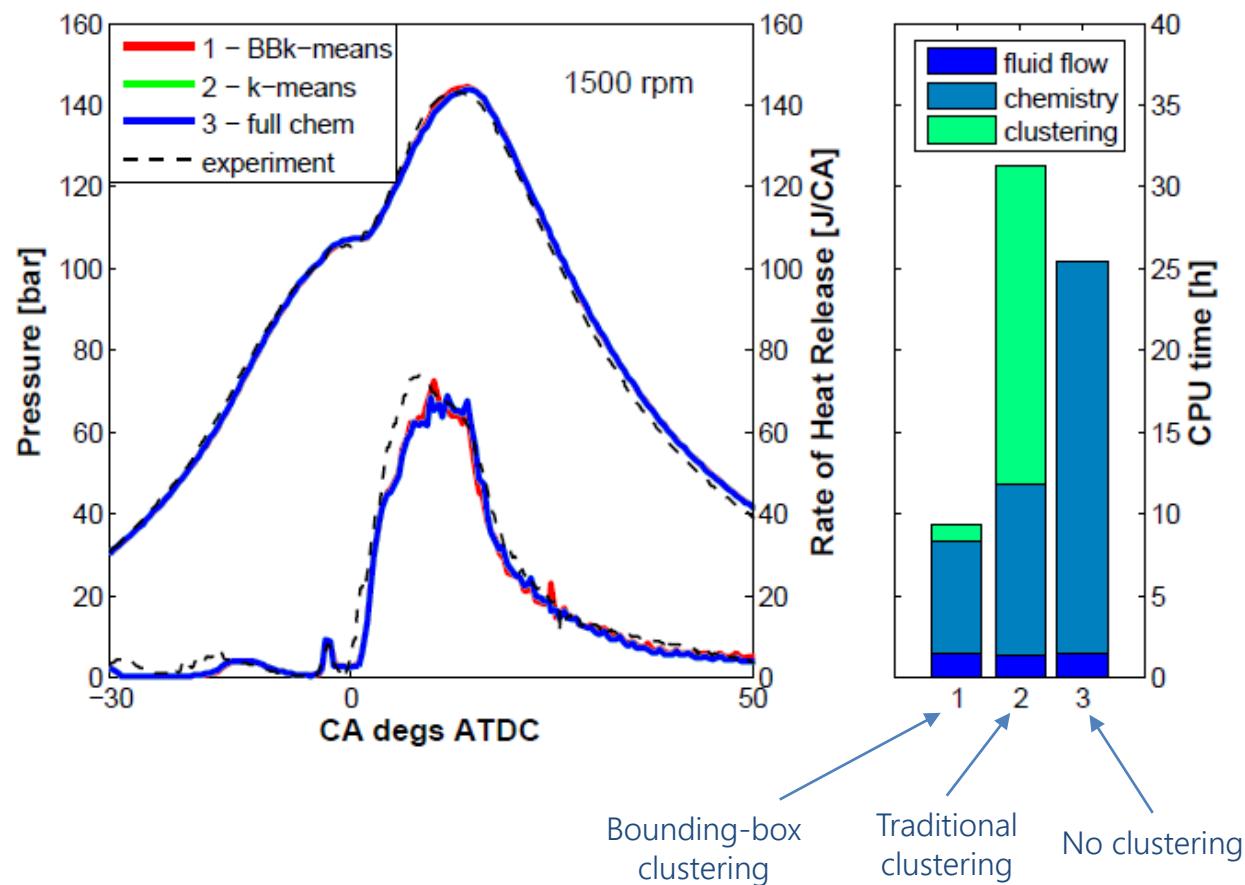
Some results

Fiat 1.3l DI diesel engine (operated with multiple injections)

Cells: ~25,000 (at BDC)

Dimensionality: d=5 (T, nC₇H₁₆, O₂, CO₂, HO₂, H₂O)

Simulations and results from: Perini F., *High-dimensional, unsupervised cell clustering for computationally efficient engine simulations with detailed combustion chemistry*, Fuel 106, p. 344–356 (2013)



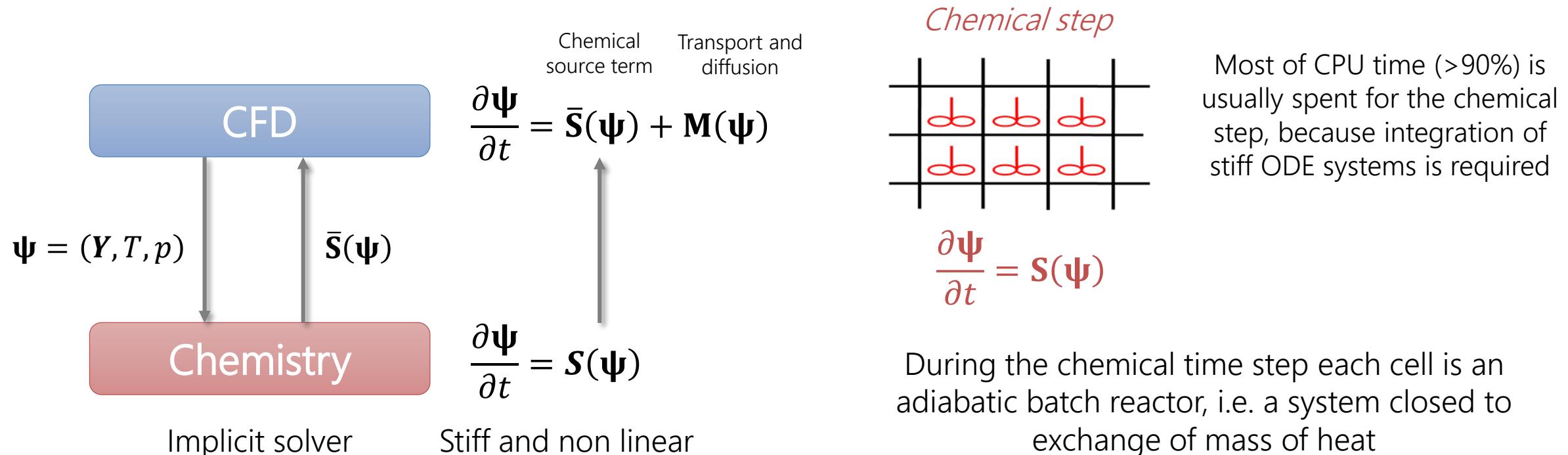
Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks





Dynamic Adaptive Chemistry (DAC) (I)



We can locally reduce the complexity of chemical kinetics according to the operating conditions

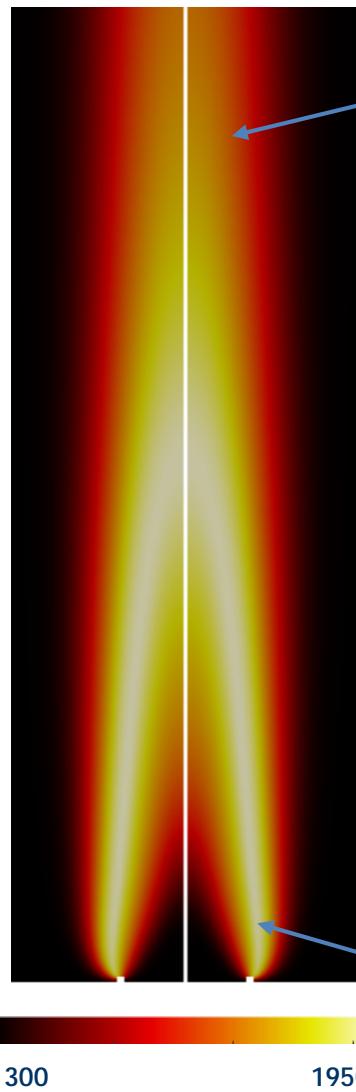
Dynamic Adaptive Chemistry

Adapted from: Contino F. et al., *Tabulation of Dynamic Adaptive Chemistry: A global approach to include detailed mechanisms in engine simulations*, Fifth OpenFOAM Workshop (2011)





Dynamic Adaptive Chemistry (DAC) (II)



Smaller kinetic scheme
for most practical combustion simulation problems, a relatively smaller number of species and reactions participate in the processes of pivotal importance such as heat release and emissions formation

Larger kinetic scheme

Dynamic Adaptive Chemistry

The detailed mechanism is reduced **locally and instantaneously** into accurate sub-mechanisms at each hydrodynamic time step of the calculation ("on the fly")

no a priori information regarding simulation conditions is needed.

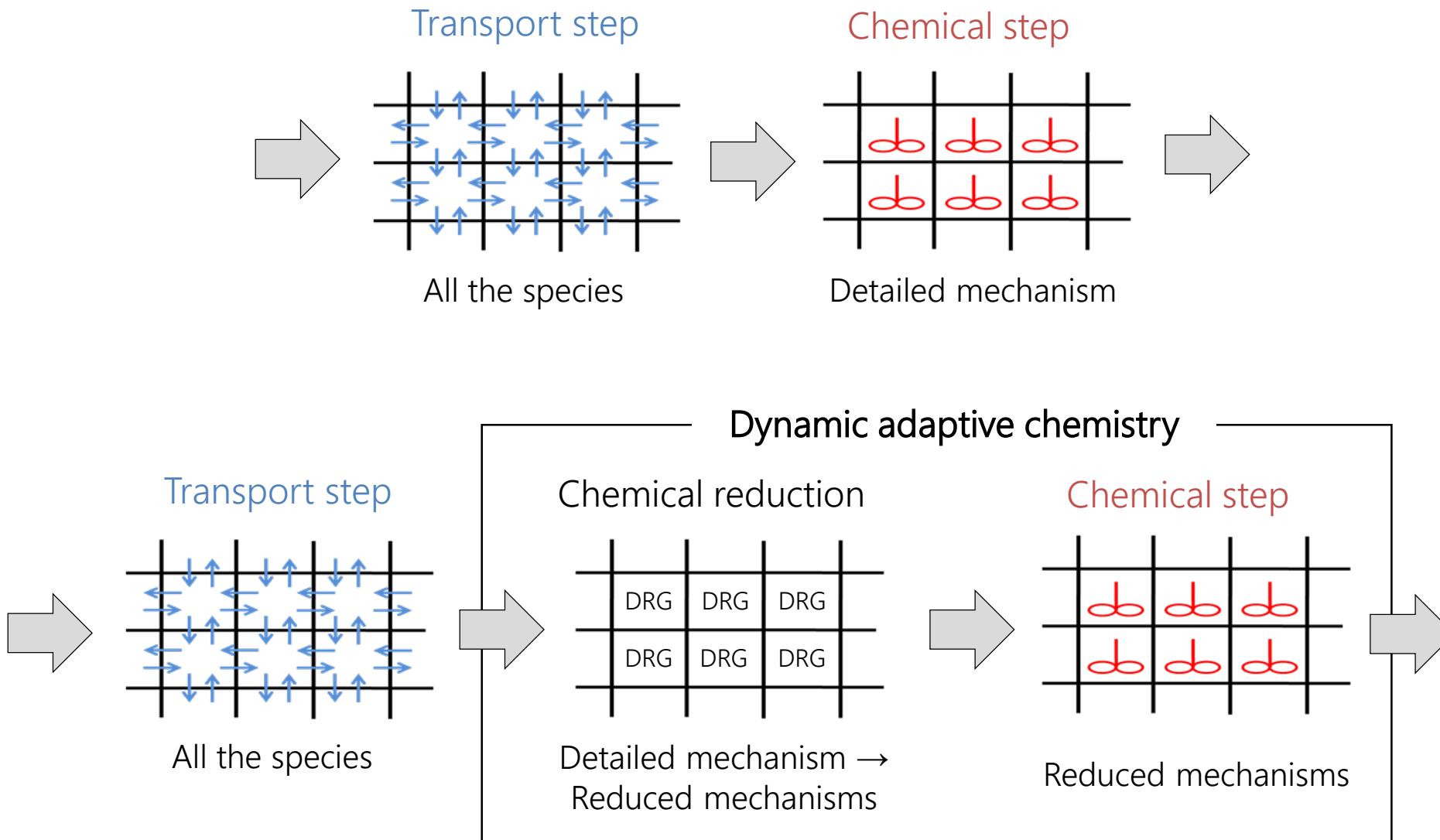
For comprehensiveness, more species is better
For computational cost, less species is better

Liang L., Stevens J.G., Farrell J.T., *A dynamic adaptive chemistry scheme for reactive flow computations*, Proceedings of The Combustion Institute, 32, p. 527–534 (2009)

Liang L., Stevens J.G., Raman S., Farrell J.T., *The use of dynamic adaptive chemistry in combustion simulation of gasoline surrogate fuels*, Combustion and Flame, 156, p. 1493–1502 (2009)

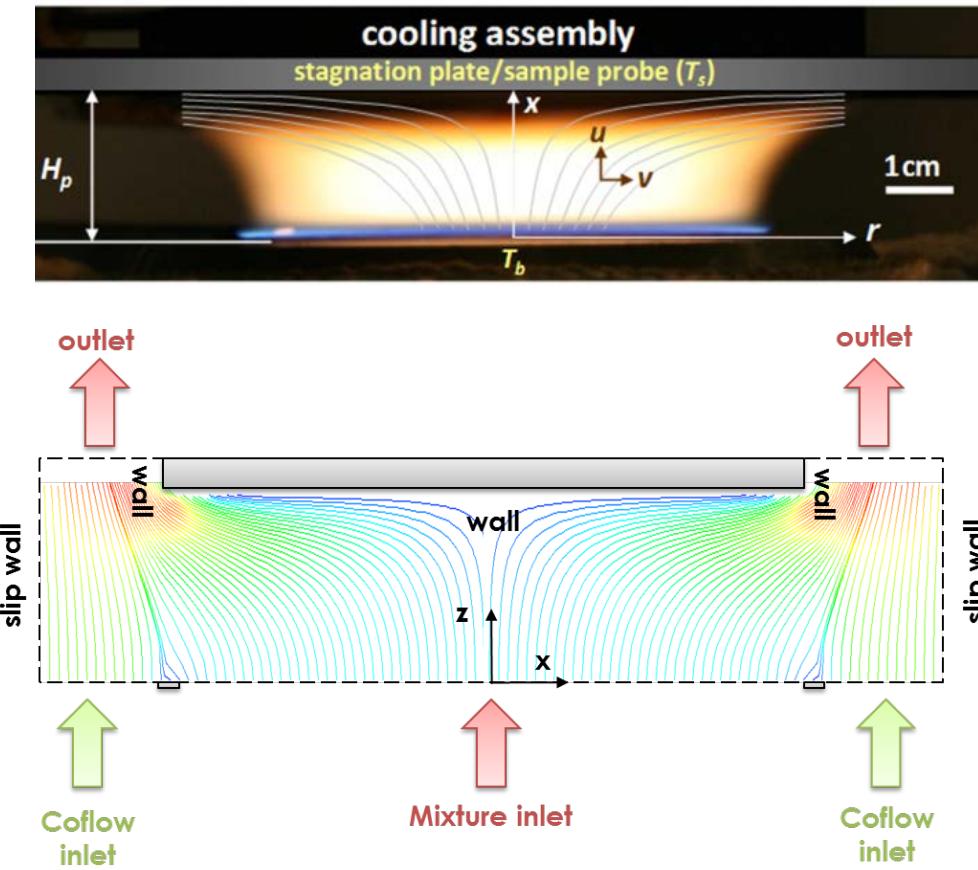


Operator-splitting and DAC



An example: a premixed, laminar flame

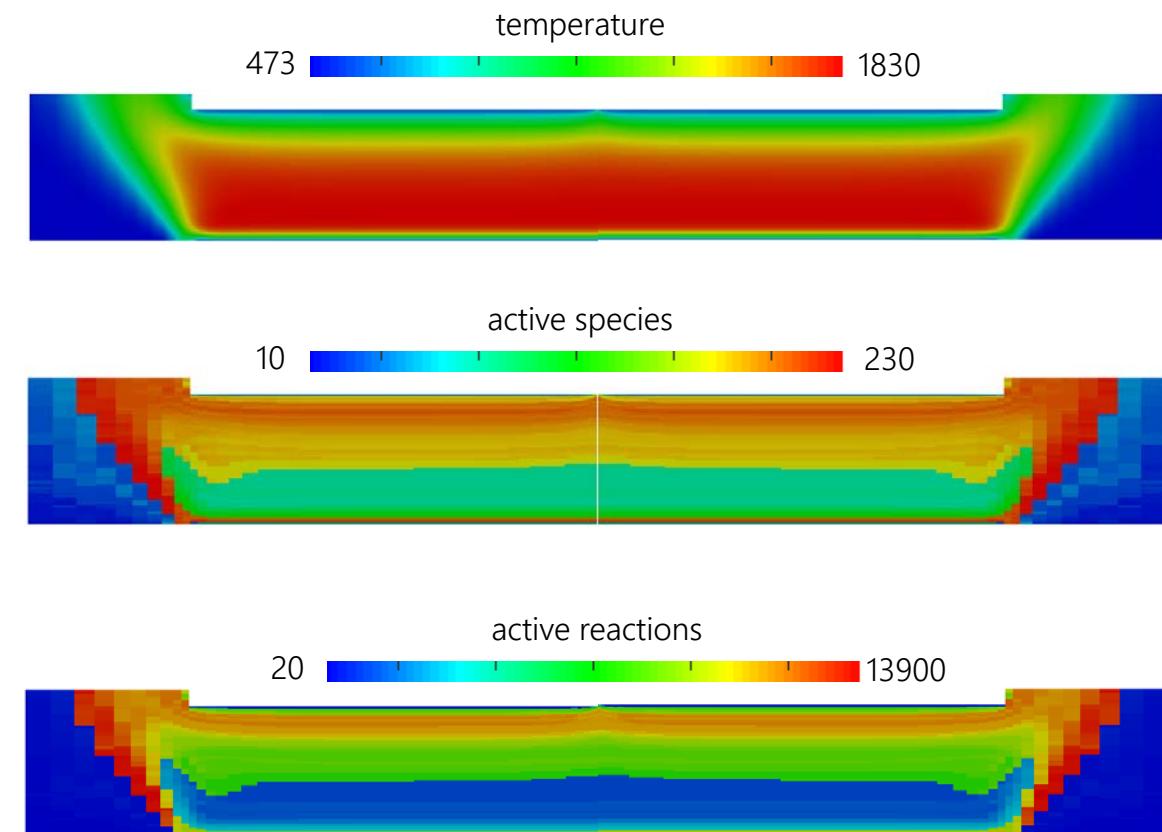
Burner-Stabilized Stagnation Flame



Saggese C. et al., Modeling Study of Probe-Induced Effects on Soot Sampling in Laminar Premixed Flames in a Benchmark Burner-Stabilized Stagnation Flame, Manuscript in preparation

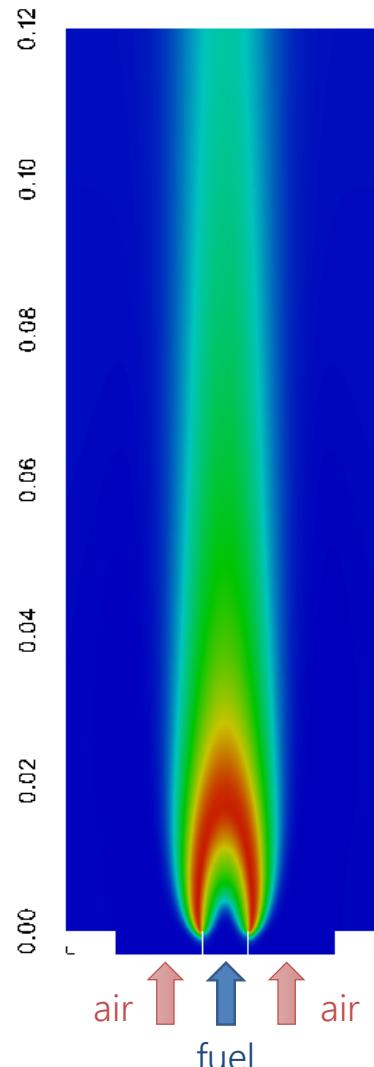
Detailed kinetic mechanism: 292 species and \sim 15,800 reactions

Saggese et al., *Kinetic Modeling of Particle Size Distribution of Soot in a Premixed Burner-Stabilized Stagnation Ethylene Flame*, Combustion and Flame, Accepted





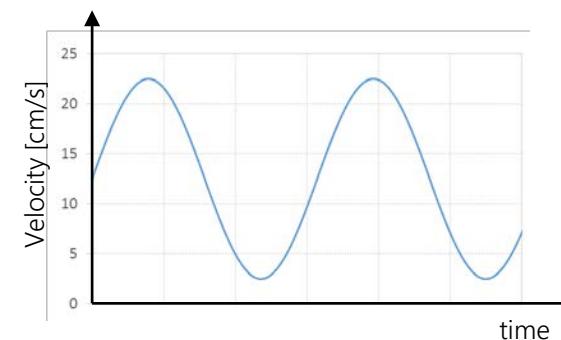
An example: a pulsating, non-premixed flame



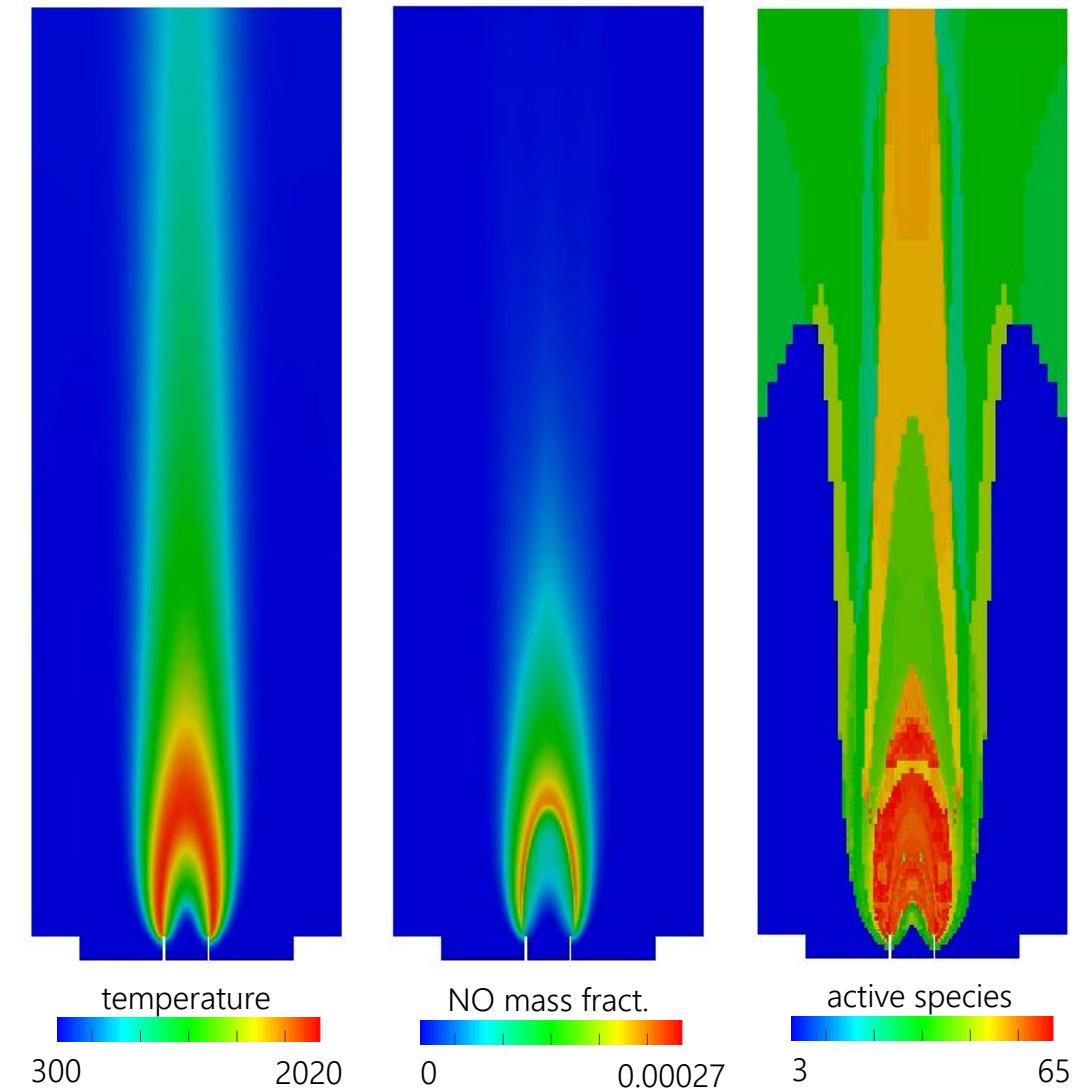
POLIMI kinetic mechanism
Species: 115
Reactions: 2141

Fuel mixture: 34% C₂H₄, 66% N₂
Coflow stream: 21% O₂, 79% N₂

The transient behavior is
induced by a **10 Hz** perturbation
in the fuel velocity profile:



In cooperation with D. Di Fiore and A.
Parente (Université Libre de Bruxelles)



Outline

1. Introduction: the need of detailed kinetic mechanisms
2. The Jacobian matrix: the keystone of simulations with detailed mechanisms
3. Exploiting sparsity of detailed mechanisms
4. 1D simulations with detailed kinetics
5. Computation Cost Minimization (CCM)
6. Reduction of detailed kinetic mechanisms
 1. Directed Relation Graph (DRG)
 2. Quasi Steady State Assumption (QSSA) and stiffness removal
 3. Diffusivity reduction
7. Coupling CFD and detailed kinetic mechanisms
 1. Operator-splitting methods
 2. Reducing the number of reacting environments (RNA & DDC)
8. Combining techniques: Dynamic Adaptive Chemistry (DAC)
9. Final remarks



Additional topics (not covered here)

- ISAT (In Situ Adaptive Tabulation) and storage/retrieval algorithm [Pope, 1997]
- TDAC (Tabulated Dynamic Adaptive Chemistry) [Contino et al., 2011]
- PCA (Principal Component Analysis) [Parente et al., 2013]
- GPU Computing (Graphical Processor Units) [Shi et al., 2011, Niemeyer et al., 2014]
- Exponential ODE integrators [Bisetti, 2012]
- ...

Parente, A., Sutherland, J.C., *Principal component analysis of turbulent combustion data: Data pre-processing and manifold sensitivity*, Combustion and Flame 160 (2), pp. 340-350 (2013)

Contino et al., *Coupling of in situ adaptive tabulation and dynamic adaptive chemistry: An effective method for solving combustion in engine simulations*, Proceedings of The Combustion Institute, 33, p. 3057-3064 (2011)

Shi Y. et al., *Accelerating multi-dimensional combustion simulations using GPU and hybrid explicit/implicit ODE integration*, Combustion and Flame 159, p. 2388-2397 (2012)

Niemeyer K.E. et al., *Accelerating moderately stiff chemical kinetics in reactive-flow simulations using GPUs*, Journal of Computational Physics, 256:854–871 (2014)

Bisetti F., *Integration of large chemical kinetic mechanisms via exponential methods with Krylov approximations to Jacobian matrix functions* Combustion Theory and Modelling 16(3), p. 387–418 (2013)

Pope S.B., *Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation*, Comb. Theory and Modelling, 1, 41-63 (1997)

