SuperCA++
Supercomputing Applications
Bansko, Bulgaria
April 23-24, 2012

Some Issues in Reliable High-Performance Simulations of Combustion for Engine Applications

Pasqua D'Ambra

pasqua.dambra@cnr.it

Institute for High Performance Computing and Networking

National Research Council of Italy *(ICAR-CNR)*

Joint Project (2005-2010)

ICAR-CNR
Naples Branch

&

Engine Institute
CNR, Naples

High-Performance Algorithms and Software for Diesel Engine Modeling

Collaborations and Credits

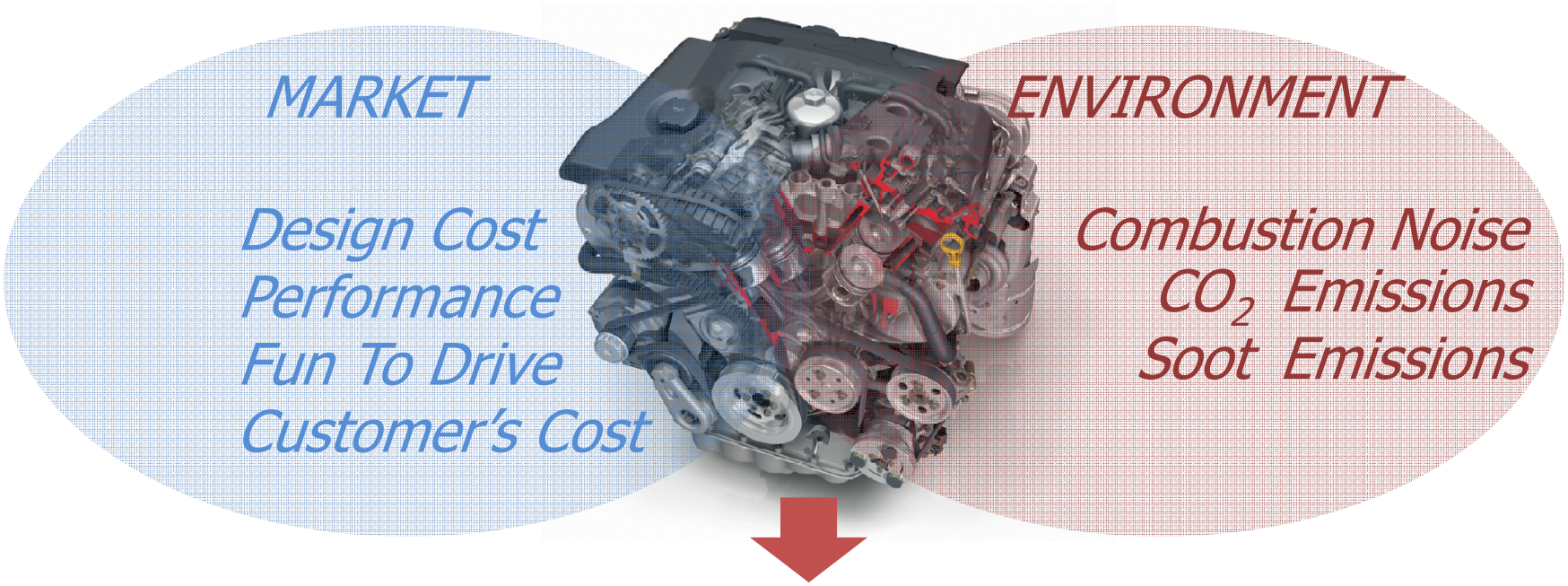
- *Laura Antonelli (ICAR)*
- *Paola Belardini (IM)*
- *Claudio Bertoli (IM)*
- *Stefania Corsaro (ICAR)*
- *Valentina Fraioli (IM)*
- *Francesco Gregoretti (ICAR)*
- *Gennaro Oliva (ICAR)*

Outline

- *Motivation*
- *Mathematical Model & Numerical Algorithm*
- *Reliable Combustion Solver & Adaptivity*
- *Parallel Combustion Software Architecture*
- *Data Distribution & Load Balancing*
- *Some Results*
- *Conclusions &Remarks*

Motivation

Impact of air quality on human health
leads Government to impose strong limits
for engines pollutant emissions



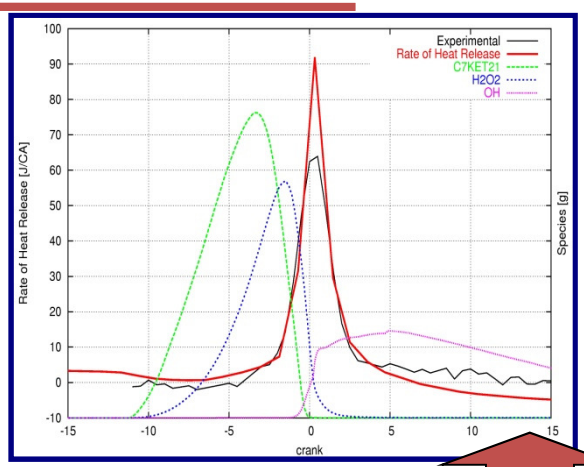
Engine designers make use of **multi-dimensional, multi-physics, multi-scales simulations codes** for matching different requirements

Engine Design Procedure

Mathematical Models (conservation laws)

$$\frac{Dq}{Dt} + divF(q) = C(q)$$

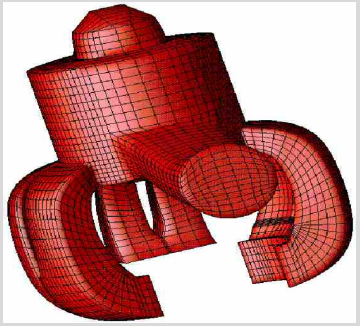
Validation by experimental measures



2

4

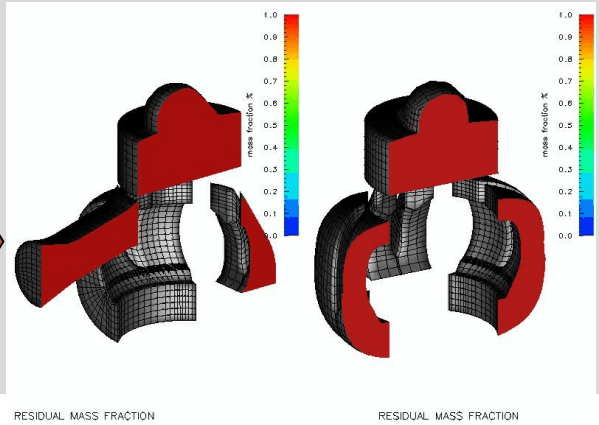
Numerical algorithms & software



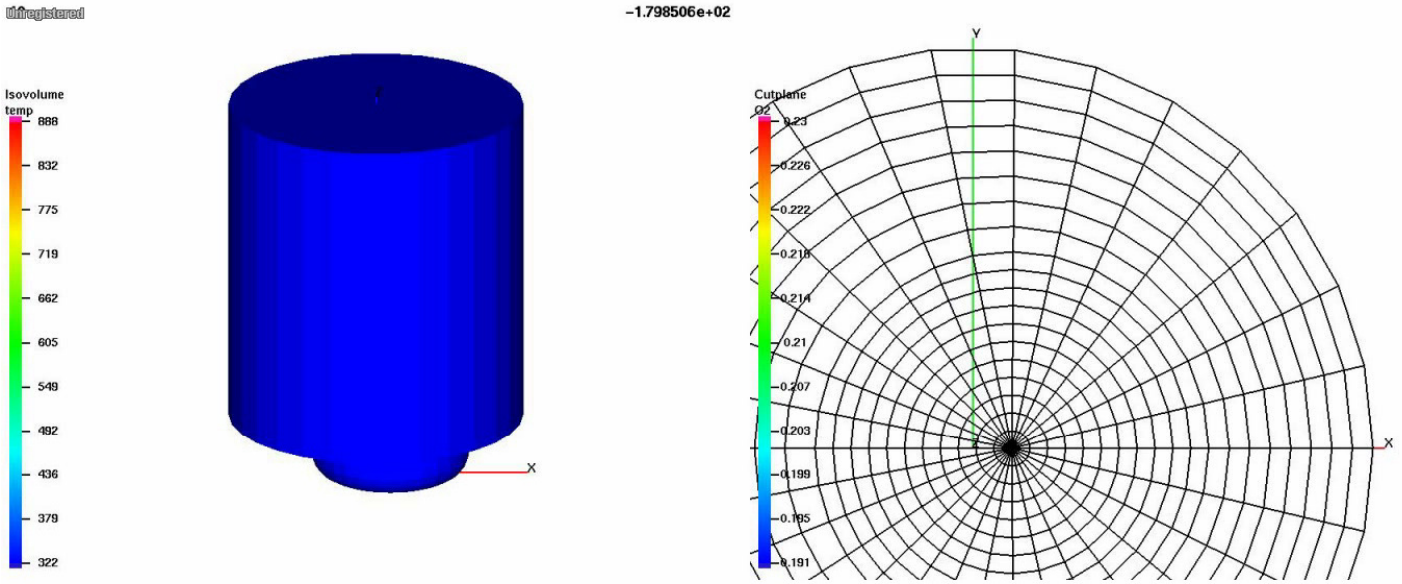
```
set parameters(&N_time_steps, &N_simple_steps)
  for(i=0; i<N_time_steps; i++) {
    delta_t=t(i+1)-t(i); solve chemistry; solve
spray; solve mass density;
    for(j=0; j<N_simple_steps; j++) {
                solve momentum;
                solve energy;
                solve pressure;
   if(test_conv) return;}
solve turbulence; }
```

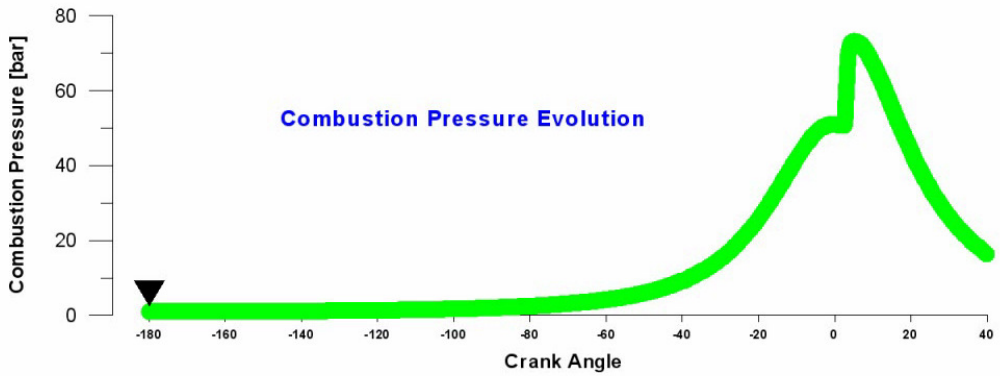
3

Computer Simulations



Main Physical Phenomena





Mathematical Model

Unsteady compressible Navier-Stokes equations for reactive flows

Continuity eq. for species m

$$\frac{\partial \rho_m}{\partial t} + \nabla \bullet (\rho_m \mathbf{u}) = \nabla \bullet \left[\rho D \nabla \left(\frac{\rho_m}{\rho} \right) \right] + \dot{\rho}_m^c + \dot{\rho}^s \delta_{ml}$$

Continuity eq. for total mass density

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{u}) = \dot{\rho}^s$$

Momentum eq.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \bullet (\rho \mathbf{u}\mathbf{u}) = -\frac{1}{\alpha^2} \nabla p - A_0 \nabla \left(\frac{2}{3} \rho k \right) + \nabla \bullet \bar{\sigma} + \mathbf{F}^s + \rho \mathbf{g}$$

Energy eq.

$$\frac{\partial \rho I}{\partial t} + \nabla \bullet (\rho I \mathbf{u}) = -p \nabla \bullet \mathbf{u} + (1 - A_0) \bar{\sigma} \div \nabla \mathbf{u} - \nabla \bullet \mathbf{J} + A_0 \rho \varepsilon + \dot{Q}^c + \dot{Q}^s$$

Spray model

Combustion (Reaction) model

Numerical Algorithm

Physical (Linear) Splitting

$$\rho^{n+1} = L(\Delta t)\rho^{n} \qquad \rho^{n} = \rho\ (t_0 + n\Delta t)$$

$$L(\Delta t) \approx L^{S}(\Delta t)L^{C}(\Delta t)L^{CFD}(\Delta t)$$

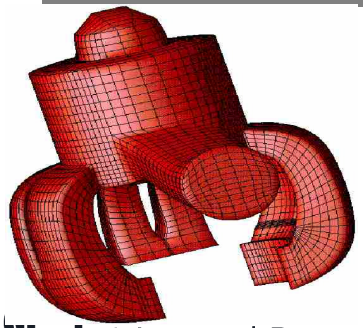
$$\rho^{n+1/3} = L^{S}(\Delta t)\rho^{n}$$

$$\rho^{n+2/3} = L^{C}(\Delta t)\rho^{n+1/3}$$

$$\rho^{n+1} = L^{CFD}(\Delta t)\rho^{n+2/3}$$

L^{S} = *Spray model*
L^{C} = Combustion model
L^{CFD} = *CFD model*

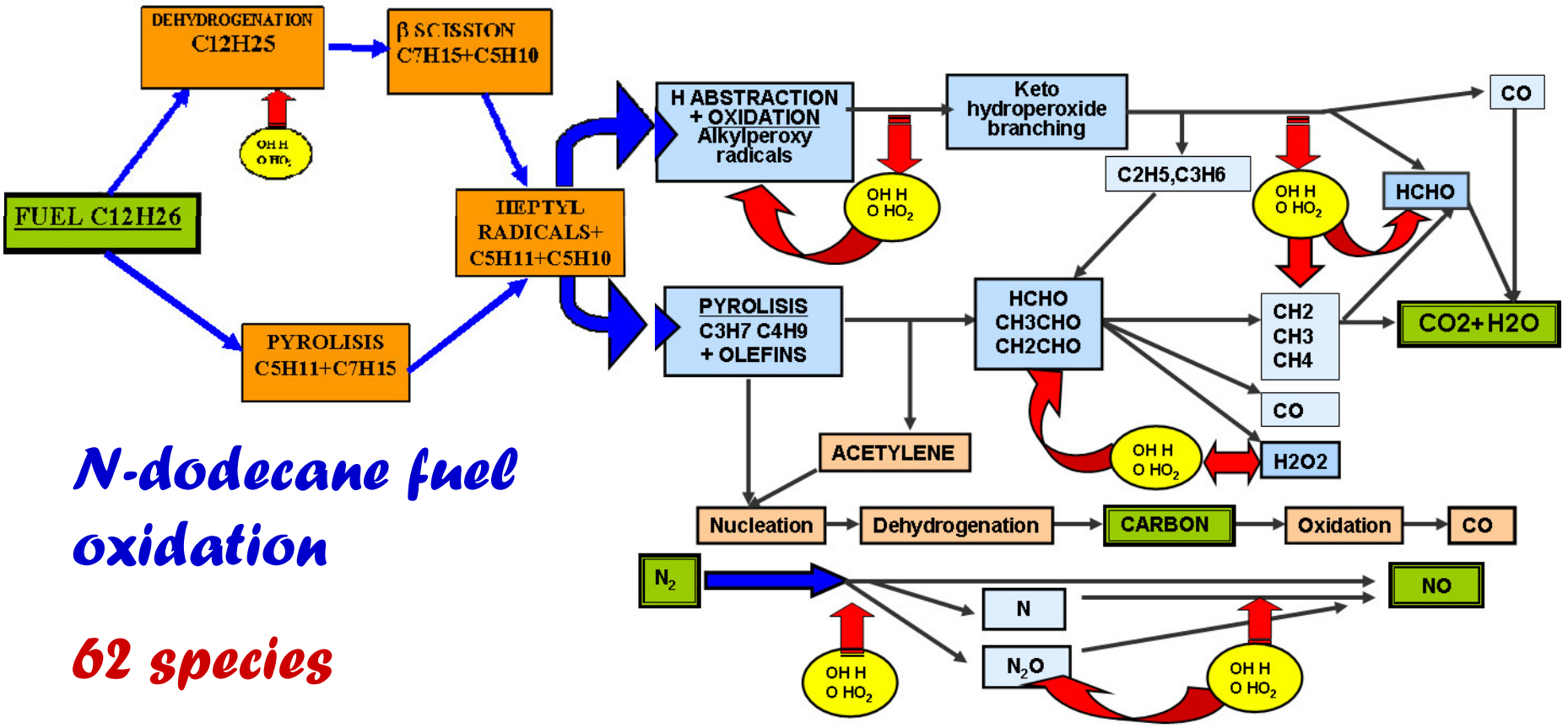
Different physical phenomena are decoupled



Different sub-models are solved independently on 3D computational grid representing engine cylinder

Combustion (Chemical) Model

Gustavsson & Golovitchev (2003)



N-dodecane fuel oxidation

62 species

285 equations

Combustion (Mathematical) Model

an ODE system per each grid cell

$$\dot{\rho}_m = W_m \sum_{r=1}^{R} (b_{mr} - a_{mr}) \dot{\omega}_r (\rho_1, \ldots, \rho_m, T) \qquad m = 1, \ldots, M$$

ρ_m **MASS DENSITY OF SPECIES** m
R **TOTAL NUMBER OF REACTIONS**
M **TOTAL NUMBER OF SPECIES**
W_m **MOLECULAR WEIGHT OF SPECIES** m
a_{mr}, b_{mr} **INTEGRAL STOICHIOMETRIC COEFFICIENTS**
K_{fr}, K_{br}, **COEFFICIENTS DEPENDING ON TEMPERATURE**
a'_{mr}, b'_{mr} , **REACTION ORDERS**

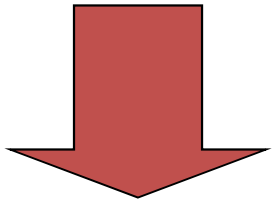
DEPENDING ON CHEMISTRY

inherently parallel model

Current simulation codes

e.g. KIVA3V-II (1999) as well as KIVA4 (2006)

- **Reduced chemical reaction mechanisms**
- Equilibrium assumption for fast chemical reactions
- **Inefficient numerical ODE solvers**



Inadequate for emission predictions in modern engines

Goal of the project

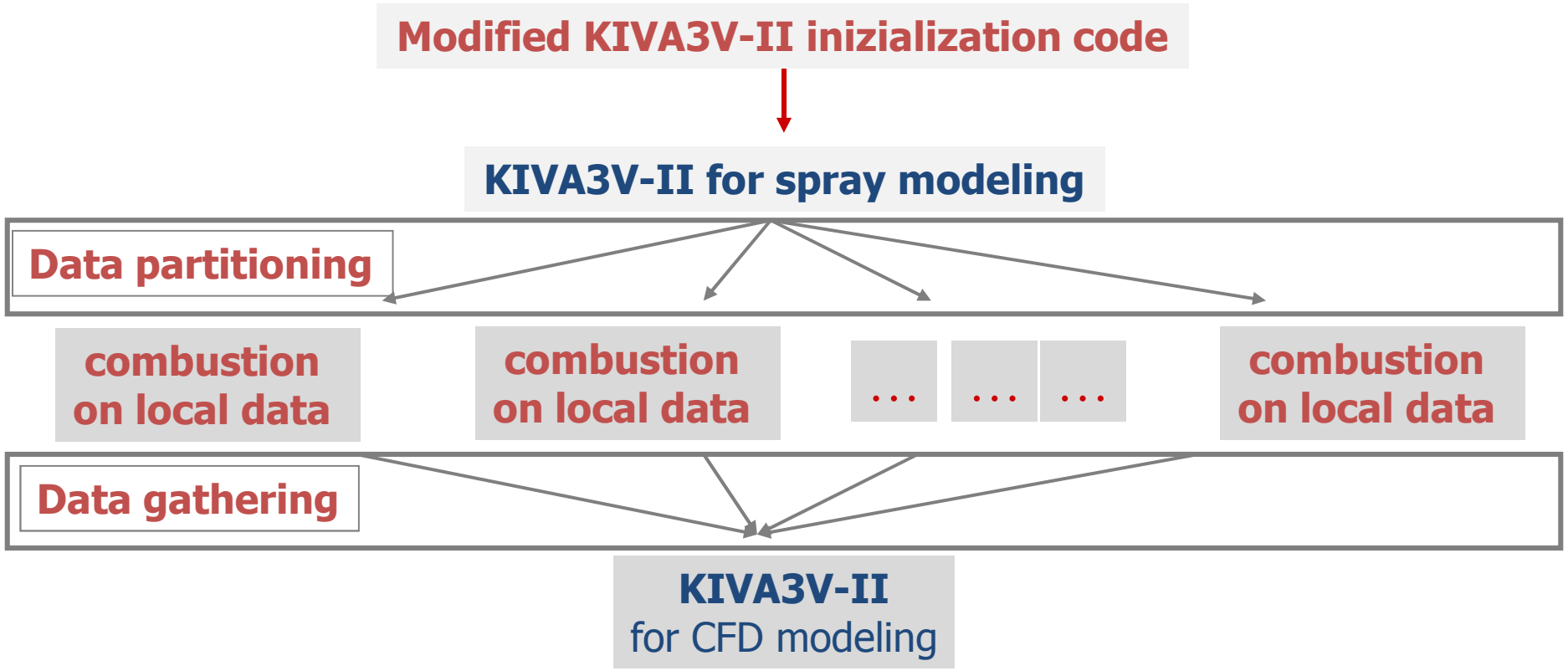
Design and implement **parallel software** for detailed combustion modeling based on **advanced numerical algorithms and software**

✓based on:

- **MPI** library for message passing
- **CHEMKIN-II** package (*Sandia National Laboratories*) for setting ODE systems of combustion
- general-purpose up-to-date **ODE solvers**

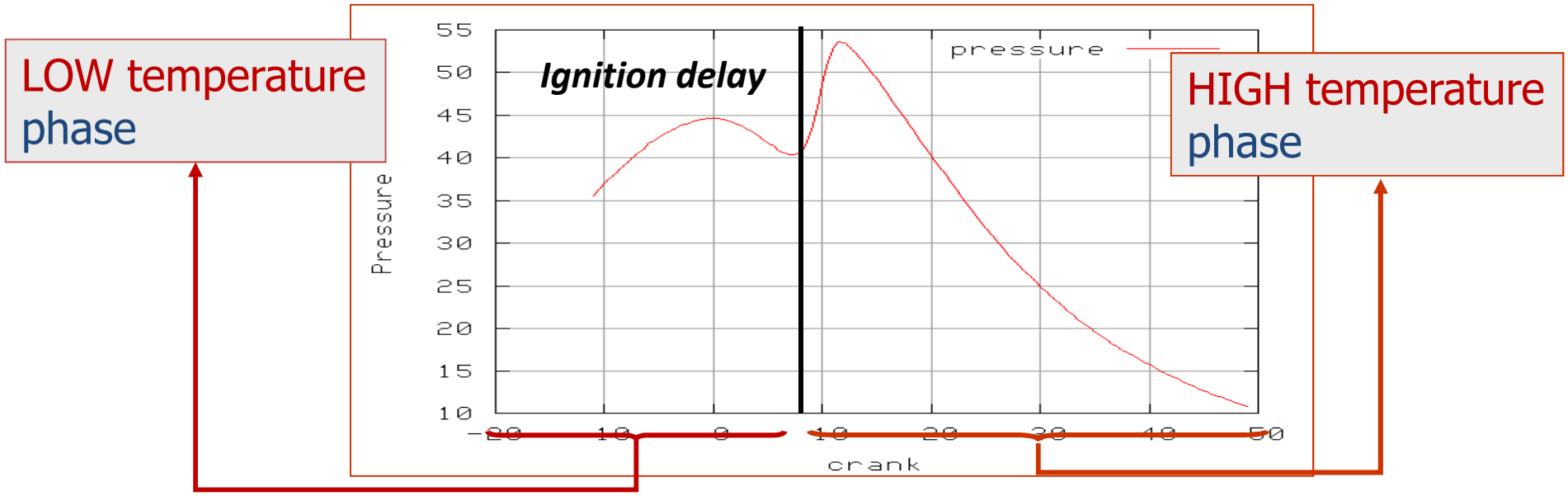
✓interfaced with **KIVA3V-II** (*Los Alamos National Laboratory*) a sequential finite-volume Fortran code for engine simulations;

Parallel Combustion Solver



processes concurrently solve ODE systems

Chemical Issues in Combustion

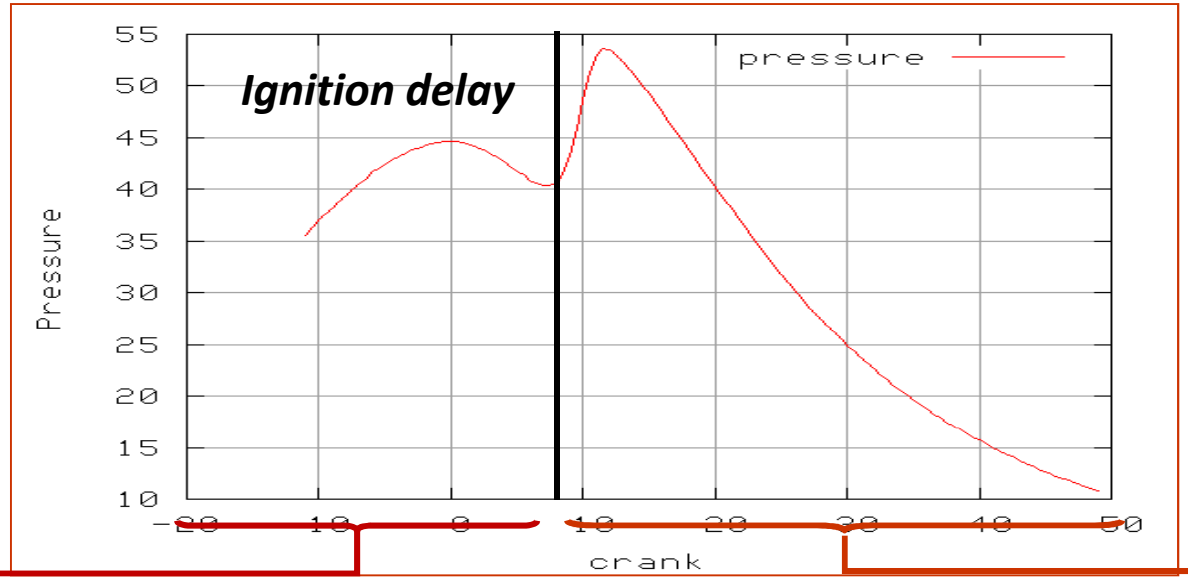


- chemical kinetics involves high number of intermediates with low density and very high reaction rates, specially in the **LOW temperature** phase. These species are fundamental for ignition starting.

- after pressure rise (during **HIGH temperature** phase), kinetic dynamics becomes smooth, and a lot of species are rapidly damped.

Numerical Issues in Combustion

stiff phase



smooth phase

- positivity and total density preserving properties
- high accuracy to efficiently solve components related to intermediates species
- strong stability properties in the highly stiff phase (LOW temperature)
- suitable damping properties in the smooth-phase (HIGH temperature)

ODE SOLVERS

VODE – Brown, Byrne, Hindmarsh (1988)

- Multistep Formulas: Variable Coefficient Backward Differentiation Formulas (BDF)

$$\sum_{i=0}^{q} \alpha_{n,i}\, y_{n-i} + h_n \beta_n \dot{y}_n = 0$$

✓Adaptive in order (from 1 to 5) and time-step choice

SDIRK4 – Hairer, Wanner (1996)

- One-step Formula: Singly Diagonally Implicit Runge-Kutta of order 4

$$\begin{cases} Y_{n,i} = y_{n-1} + h_n \sum_{j=1}^{s} a_{ij} f\left(t_{n-1} + c_j h_n, Y_{n,j}\right) \quad i = 1, \ldots, s \\ y_n = y_{n-1} + h_n \sum_{j=1}^{s} b_i f\left(t_{n-1} + c_j h_n, Y_{n,i}\right) \end{cases}$$

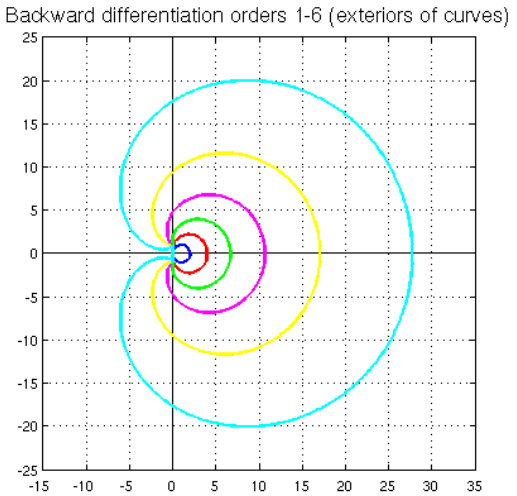
✓Adaptive in time-step choice

VODE: main features

✓ stiffly stable formulas (*Gear*, 1971)

a, c>0, R(z), z∈ C stability function

$$R(z) \supseteq \{z \in C : \mathrm{Re}(z) \le -a\} \cup \{z \in C : -a \le \mathrm{Re}(z) \le 0, -c \le \mathrm{Im}(z) \le c\}$$



courtesy of N. Trefethen

✓ Simplified Newton iterations for non-linear systems at each time
✓Jacobian matrix saved and reused for different time steps
✓Routines from Linpack for direct solution of linear systems

SDIRK4: main features

✓ L-stable formula
(*Ehle*, 1969)

1. A-stable formula
2. $R(z)$, $z \in C$ stability function: $\lim_{z \to \infty} R(z) = 0$

prevents stiff components from being damped out very slowly

✓ Simplified Newton iterations for non-linear systems at each time step

1 Jacobian evaluation and 1 LU factorization per time step

✓Routines from decsol (*Hairer & Wanner*) for direct solution of linear systems

Test case: 16 valves EURO IV Fiat Multijet

Bore	**82.0** [MM]
Stroke	**90** [MM]
Displacement	**475** [CM^3]
Compression ratio	**16.5:1**
Engine speed	**1500** RPM
Valves per cylinder	**4**
Injector	MICROSAC **7** HOLES, ∅**0.140** MM
Injector apparatus	**BOSCH COMMON RAIL III GENERATION**

COMPUTATIONAL GRID: 3D CYLINDRICAL SECTOR,WITH ~ 4500 CELLS

~ 99% OF COMPUTING TIME IN ODE SOLVERS

Testing Platform

Software

- Linux Red Hat 9.0
- GNU Fortran compiler 3.2.2
- LAM MPI 7.0
- KIVA3V-II simulation code by Los Alamos Labs.

Hardware

- Beowulf class cluster operated by IM-CNR
  - ✓ 16 (2.9 GHz) Pentium IV processors, each with 1 GB RAM and 256 KB L2 cache
  - ✓ Fast Ethernet (100 Mbit/sec)

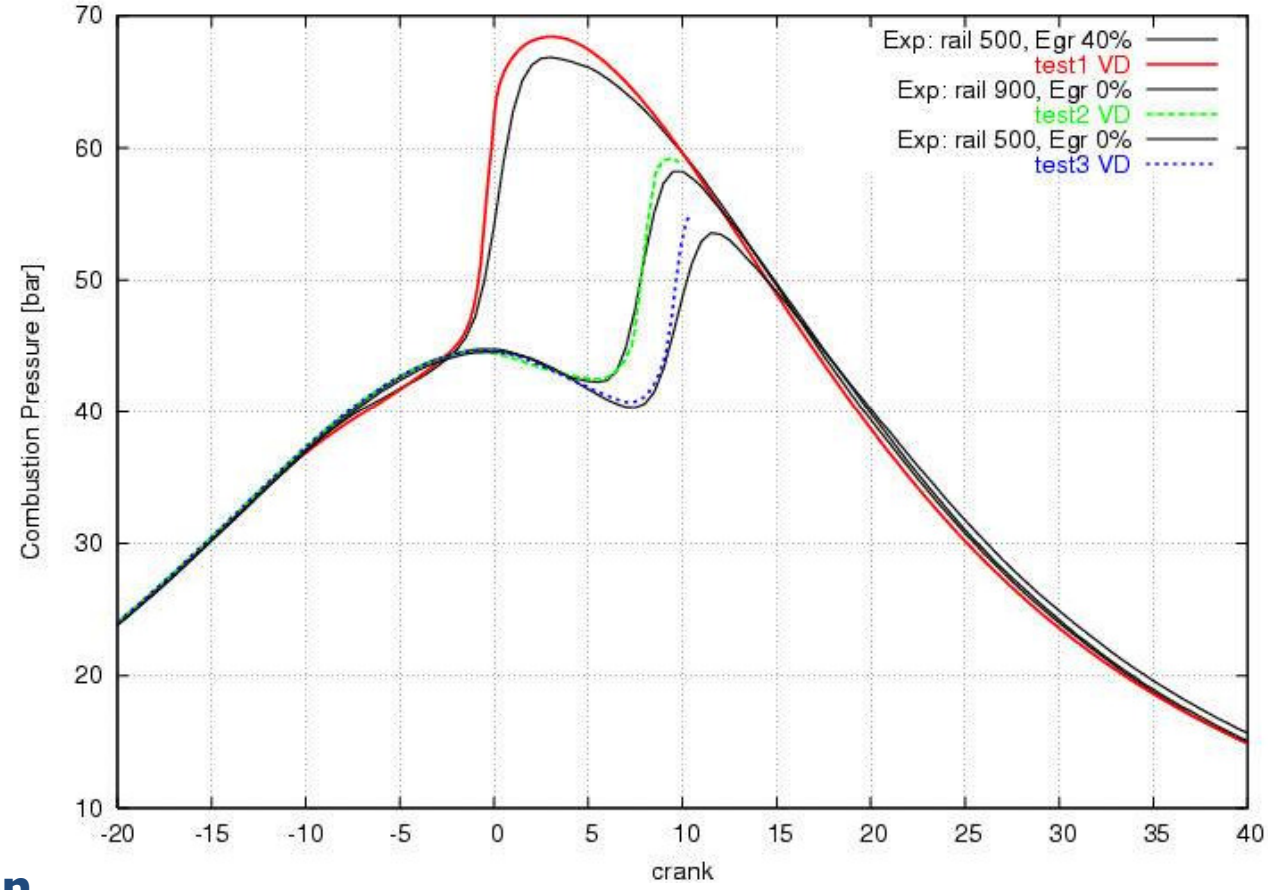
ODE solvers stopping criteria

Absolute tolerances depending on species
Relative tolerances = $\mathbf{10^{-3}}$
10k **maximum number of time steps** per splitting intervals

Engine operative conditions

	Rail Pressure bar	EGR	Injection timing CA btdc	Injection Period CA	Injected Fuel mg
Test1	500	40%	12.7 btdc	7.3	8.0
Test2	900	0%	2.3 btdc	5.7	8.7
Test3	500	0%	2.1 btdc	7.3	8.5

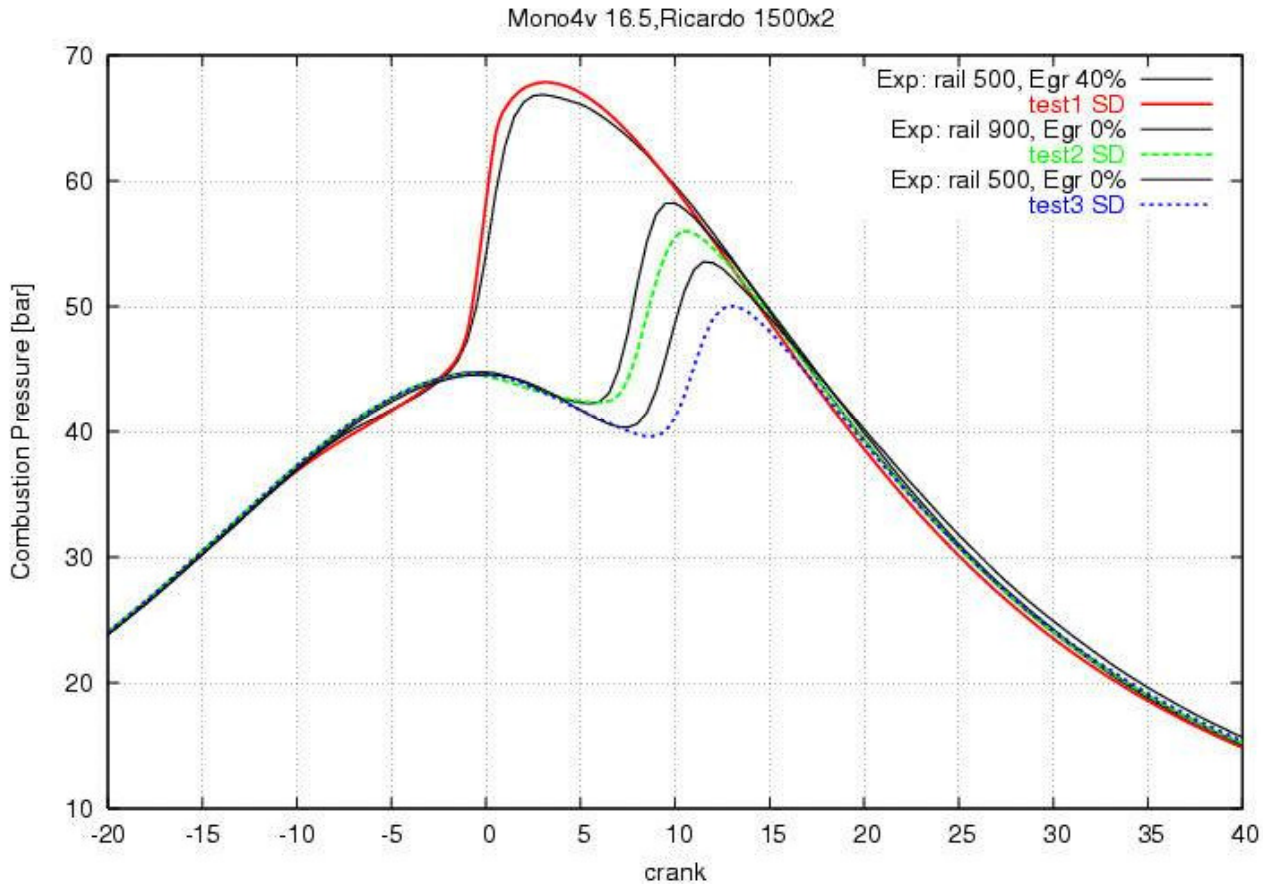
Accuracy: VODE vs SDIRK4



VODE

well predicts ignition
over-estimates pressure rise
does not reach accuracy in high-pressure/high-temperature phase

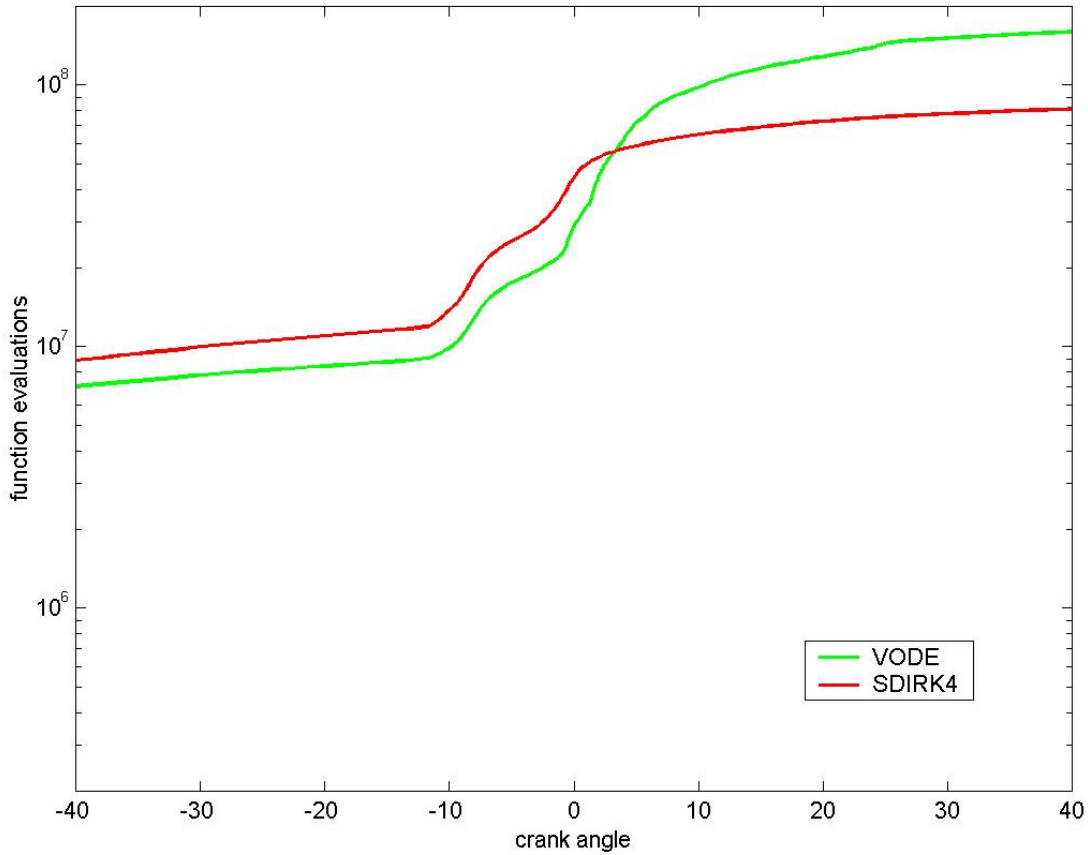
Accuracy: VODE vs SDIRK4 (cont'd)



SDIRK4

delays ignition
under-estimates pressure rise

Efficiency: VODE vs SDIRK4

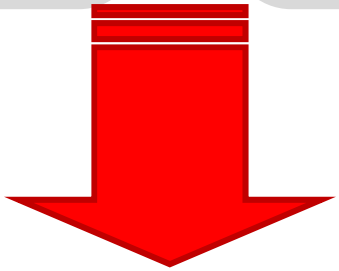


Computational Cost on Test 1
Function Evaluations vs Crank Angle on 1 processor

...one more level of adaptivity

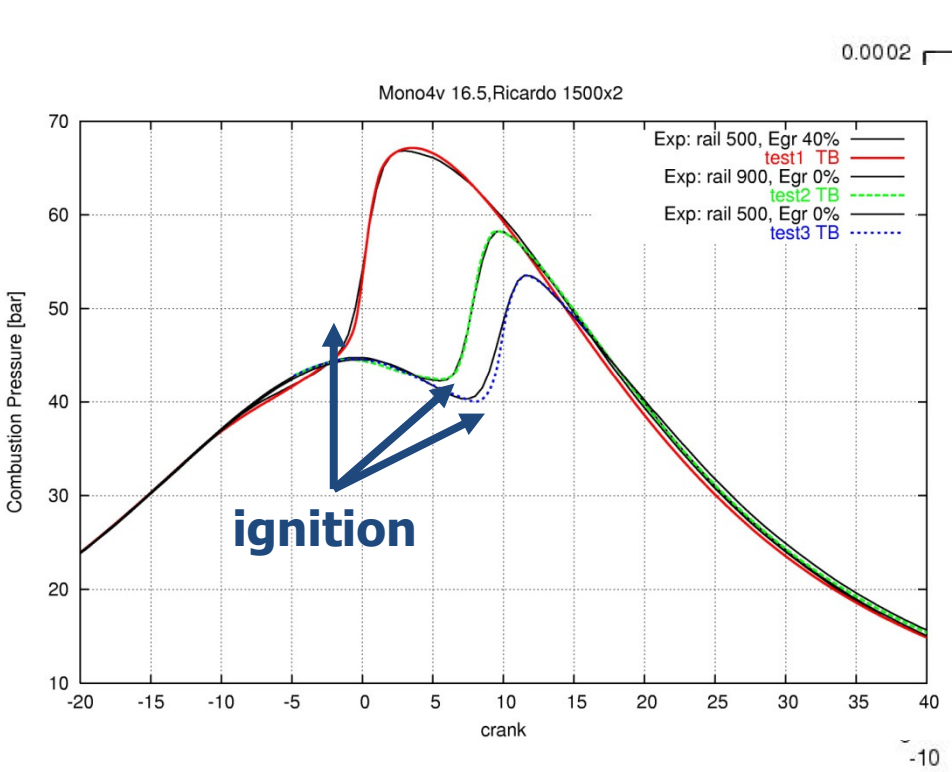
VODE
in stiff
(low temperature/pressure)
simulation phase

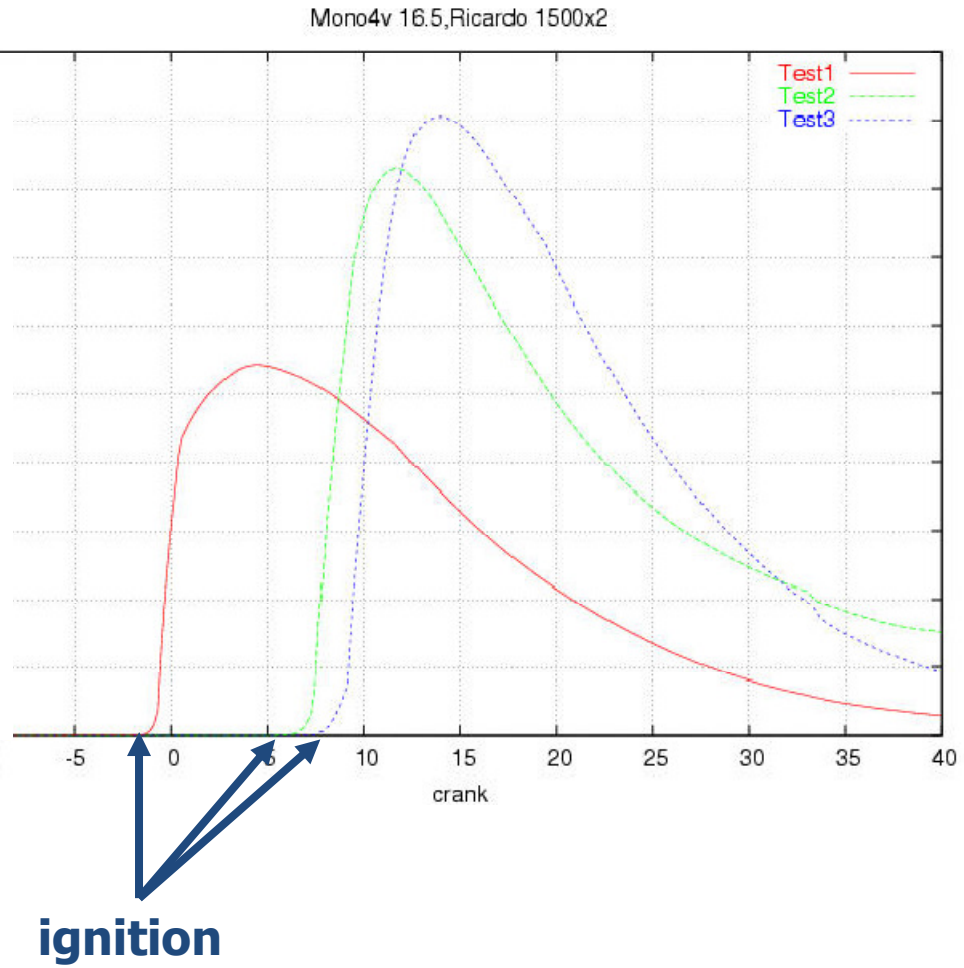
SDIRK4
in smooth
(high temperature/pressure)
simulation phase



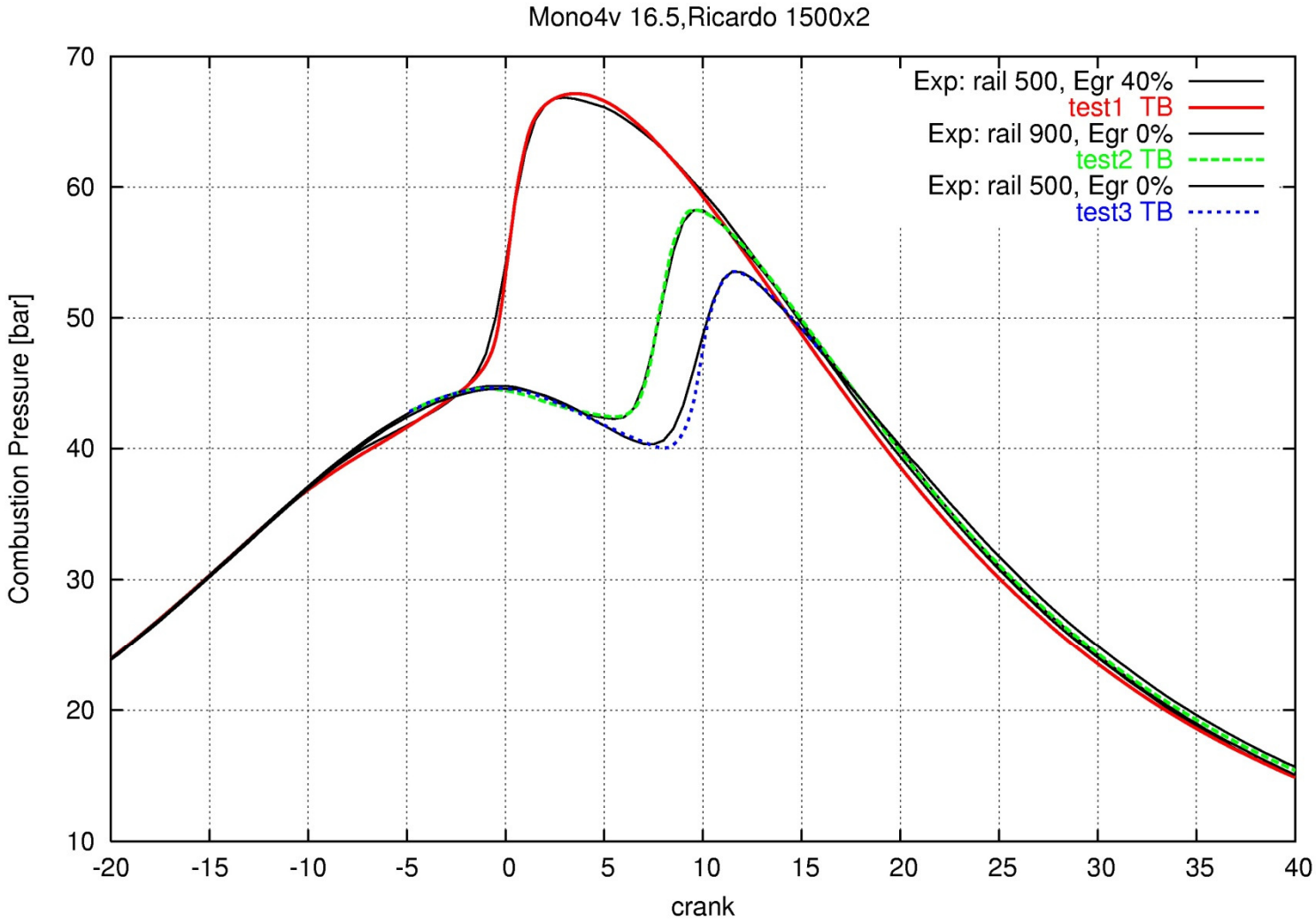
a multi-method ODE component
for improving
reliability and efficiency
of Diesel Combustion Simulations

Method Selection depending on Chemistry

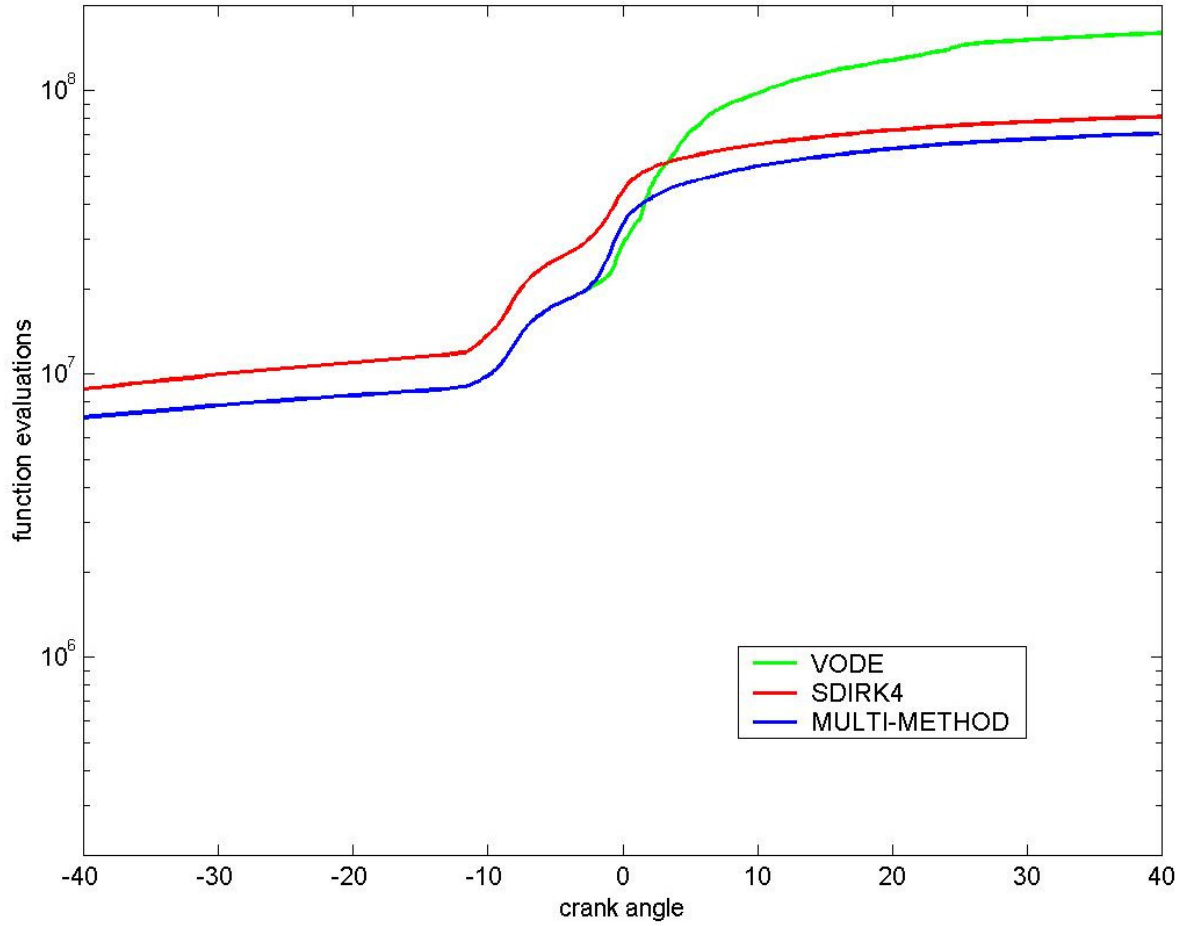




Accuracy of Multi-method Solver



Efficiency of Multi-method Solver



Computational Cost of Combustion on Test 1
Function Evaluations vs Crank Angle on 1 processor

Engine Simulation Software

KIVA3V-II numerical procedure

```
SET N_time_steps, N_nonlin_steps
for s=1 to  N_time_steps do
  1. SOLVE spray;
  2. SOLVE COMBUSTION;
  3. SOLVE mass density;
  4. for j=1 to N_nonlin_steps do
          SOLVE momentum;
          SOLVE energy;
          SOLVE pressure;
          if (test_conv) return;
     endfor
  5. SOLVE turbulence;
  6. SOLVE advection;
endfor
```

SOFTWARE STRUCTURE

```
1. KIVA3V-II for inizialization
and spray solver;
GRID PARTITIONING
SCATTER grid cells data by MPI
```

```
2. CHEMKIN II for ODE systems
setting
  2.1 if phase == AUTOIGNITION
      ODE solver is VODE
  2.2 if phase == COMBUSTION
      ODE solver is SDIRK4
```

```
GRID GATHERING
GATHER grid cells data by MPI
3-6. KIVA3V-II for CFD solver;
```

Parallel Computing Issues

Combustion solver deals with:

- detailed kinetic schemes for accurate simulation of combustion
- local physical conditions
- numerical adptivity of solvers

many sources of
computational load imbalance

grid partitioning is a key issue
for efficient parallel simulations

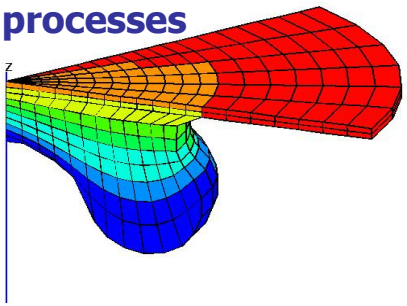
Grid Partitioning for Combustion

Different strategies to be flexible at changes in:

- chemistry model & solvers
- grid size at run time (due to piston movement)
- hw/sw platform (possible heterogeneity)

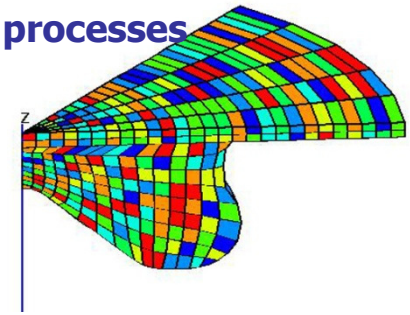
Block partitioning

8 processes

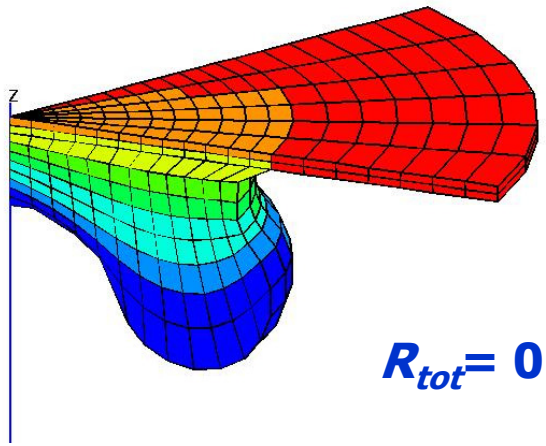


Random partitioning

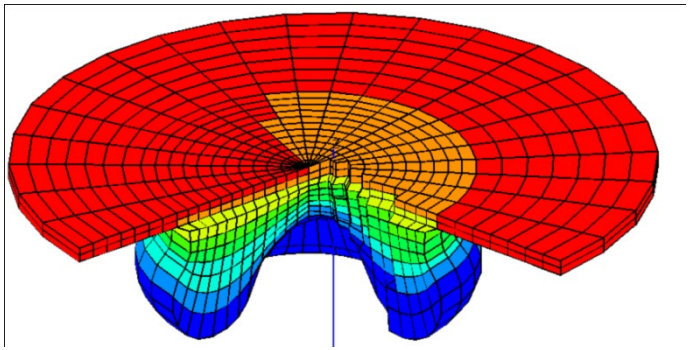
8 processes



Performance: Block Partitioning



R_{tot}= 0.4

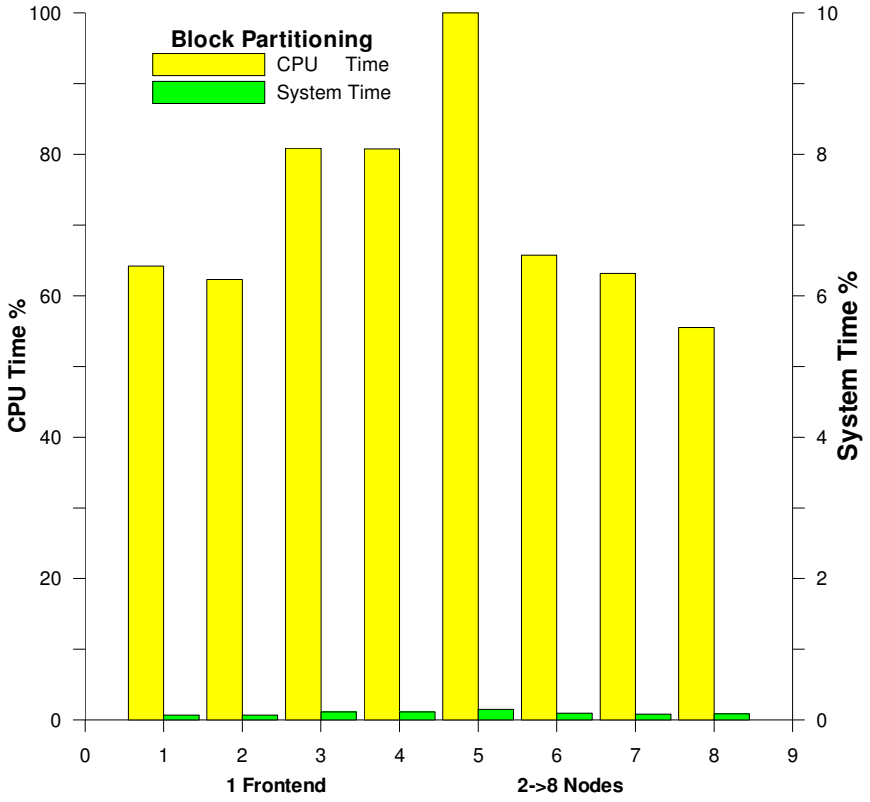


SIMULATION CYCLE:
FROM 5 CA BTDC TO 40 CA ATDC

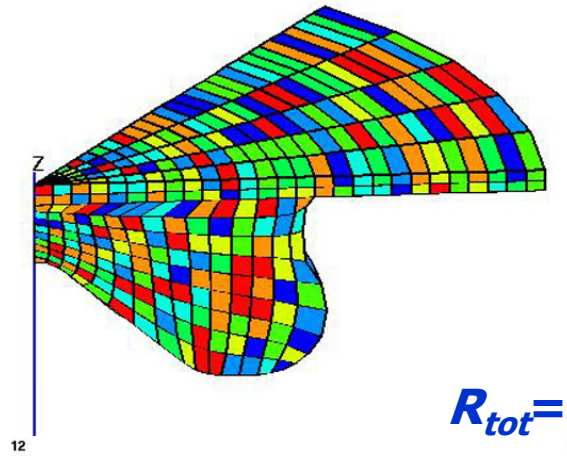
$$R_{tot} = \left(T_{\max}^{tot} - T_{med}^{tot}\right) / T_{\max}^{tot} \quad \text{load imbalance}$$

$$T_{\max}^{tot} = \text{max execution time among processes}$$

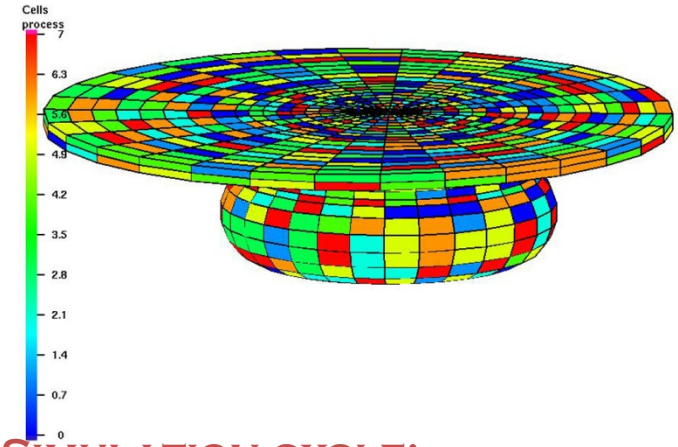
$$T_{med}^{tot} = \text{average execution time among processes}$$



Performance: Random Partitioning



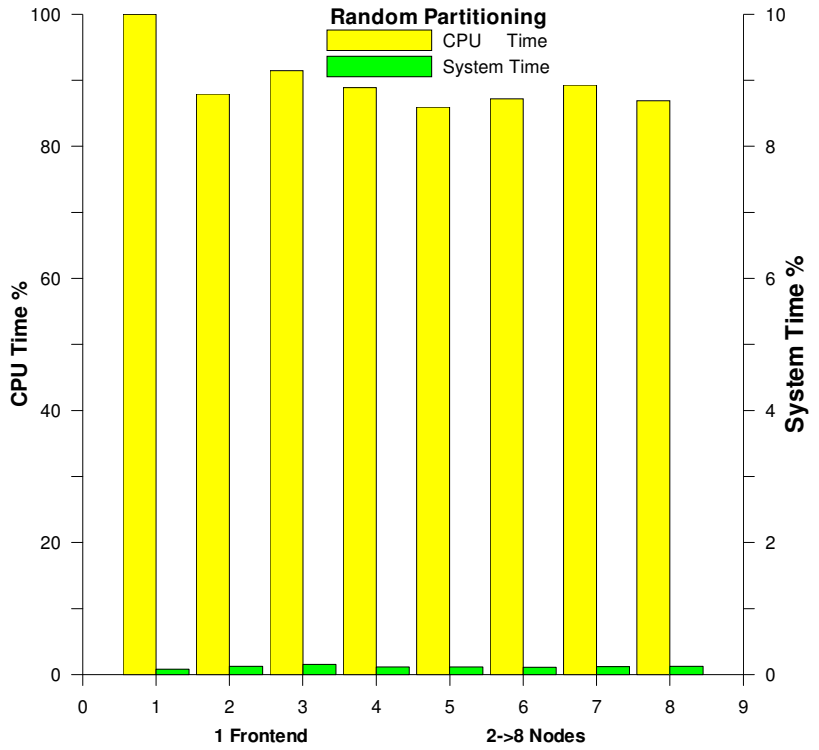
R_{tot}= 0.1



$$R_{tot} = \left(T^{tot}_{\max} - T^{tot}_{med}\right) / T^{tot}_{\max} \quad \text{load imbalance}$$

$$T^{tot}_{\max} = \text{max execution time among processes}$$

$$T^{tot}_{med} = \text{average execution time among processes}$$



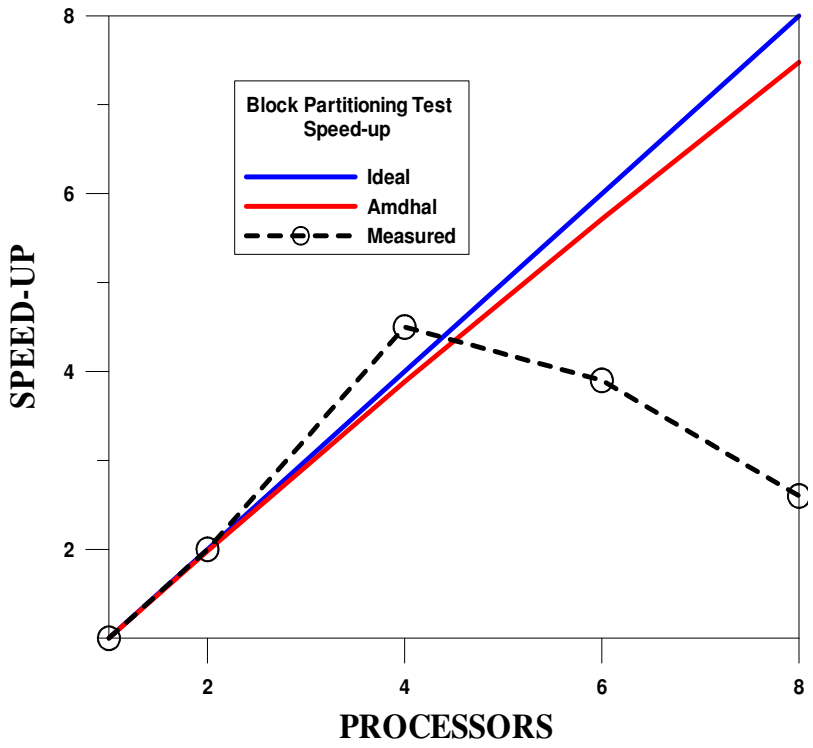
SIMULATION CYCLE:

FROM 5 CA BTDC TO 40 CA ATDC

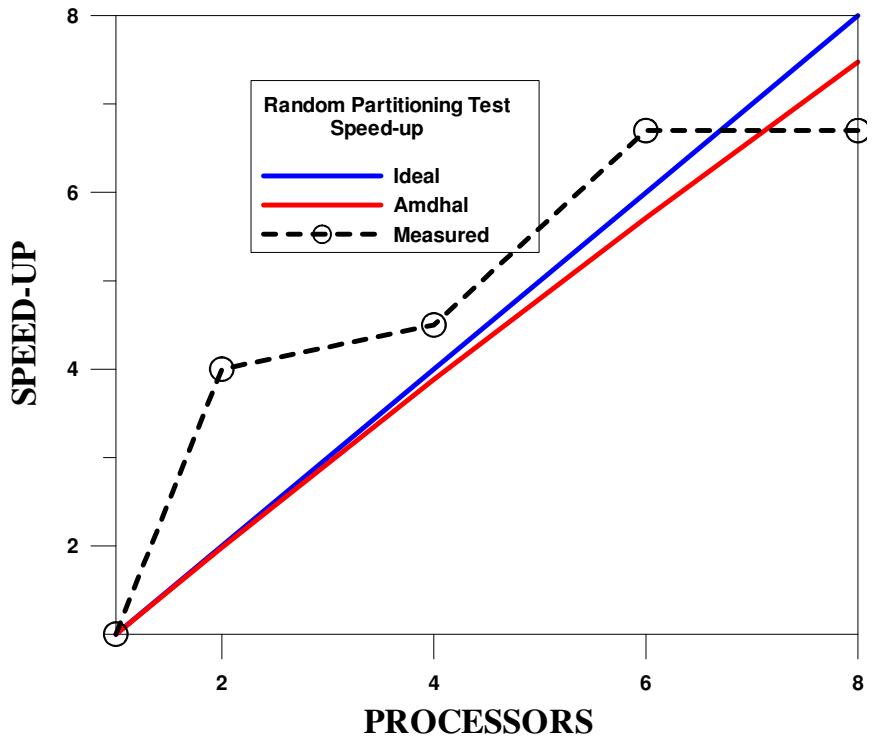
National Research Council of Italy

Parallel Combustion

Block-partitioning



Random-partitioning



super-linear speedup in case of good partitioning

Grid Partitioning for Efficient Engine Simulation

RANDOM PARTITIONING
is a good strategy
for **reducing load imbalance**
in parallel combustion
but

parallel PDEs solvers needed for CFD modeling
(e.g. KIVA-4MPI)
require interaction among contiguous cells at each time-step
i.e.

BLOCK PARTITIONING
minimizes T_{com}/T_{calc}
(surface/volume effect)

Dynamic Grid Partitioning

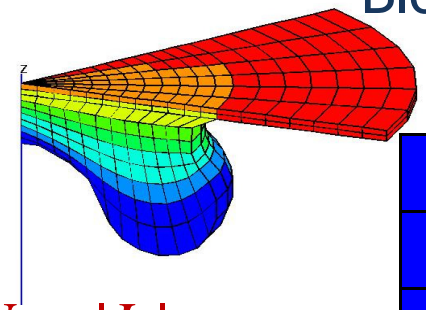
introduce in
BLOCK PARTITIONING
DYNAMIC WORKLOAD BALANCING
for preserving contiguousness of grid cells
and improving efficiency of computation

estimate at run time load imbalance
and possibly re-map grid cells by
a non-uniform block partitioning

Process Workload Measure

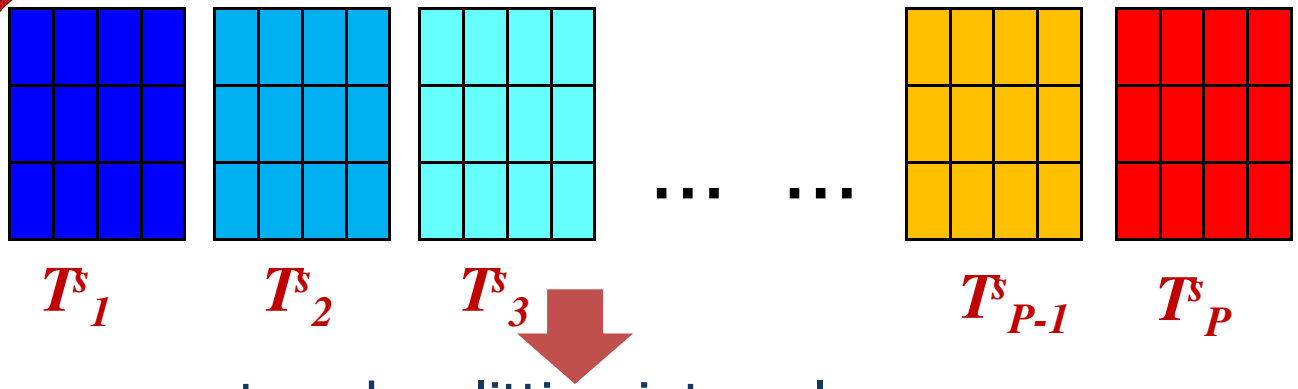
Block partitioning

Let be N total number of grid cells with indices in $I=\{1,2,3,\ldots,N\}$ and P number of processes. Block partitioning produces Λ, a uniform partition of I, with p subsets I_p of consecutive indices



$N_p = |I_p|$

$\Lambda = \{I_p\}_{p=1,\ldots,P}$



at each splitting interval s

combustion workload measure of process p

$$T_p^s = \sum_{i \in I_p} T^s(i)$$

$T^s(i)$ **execution time of combustion on grid cell i**

Dynamic Partitioning: Basic Idea

at the end of each splitting interval

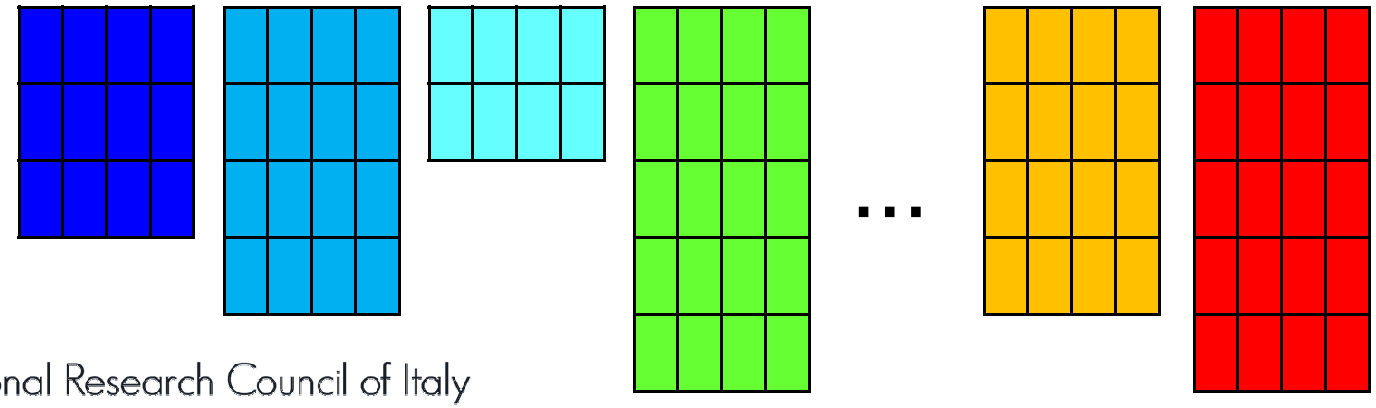
possibly define a new partition Λ^{new} with

a suitably computed dimension

for reducing "spread" among processes workload

non-uniform block partitioning

depending on combustion workloads



Dynamic Load Balancing Algorithm

Let be: $T_p^{old} = \sum_{i \in I_p^{old}} T^s(i)$ $\quad T^s_{\max} = \max_p T_p^{old}$ $\quad T^s_{med} = \sum_p T_p^{old} \; / \; P$

$$R_s = \left(T^s_{\max} - T^s_{med}\right) / T^s_{\max}$$

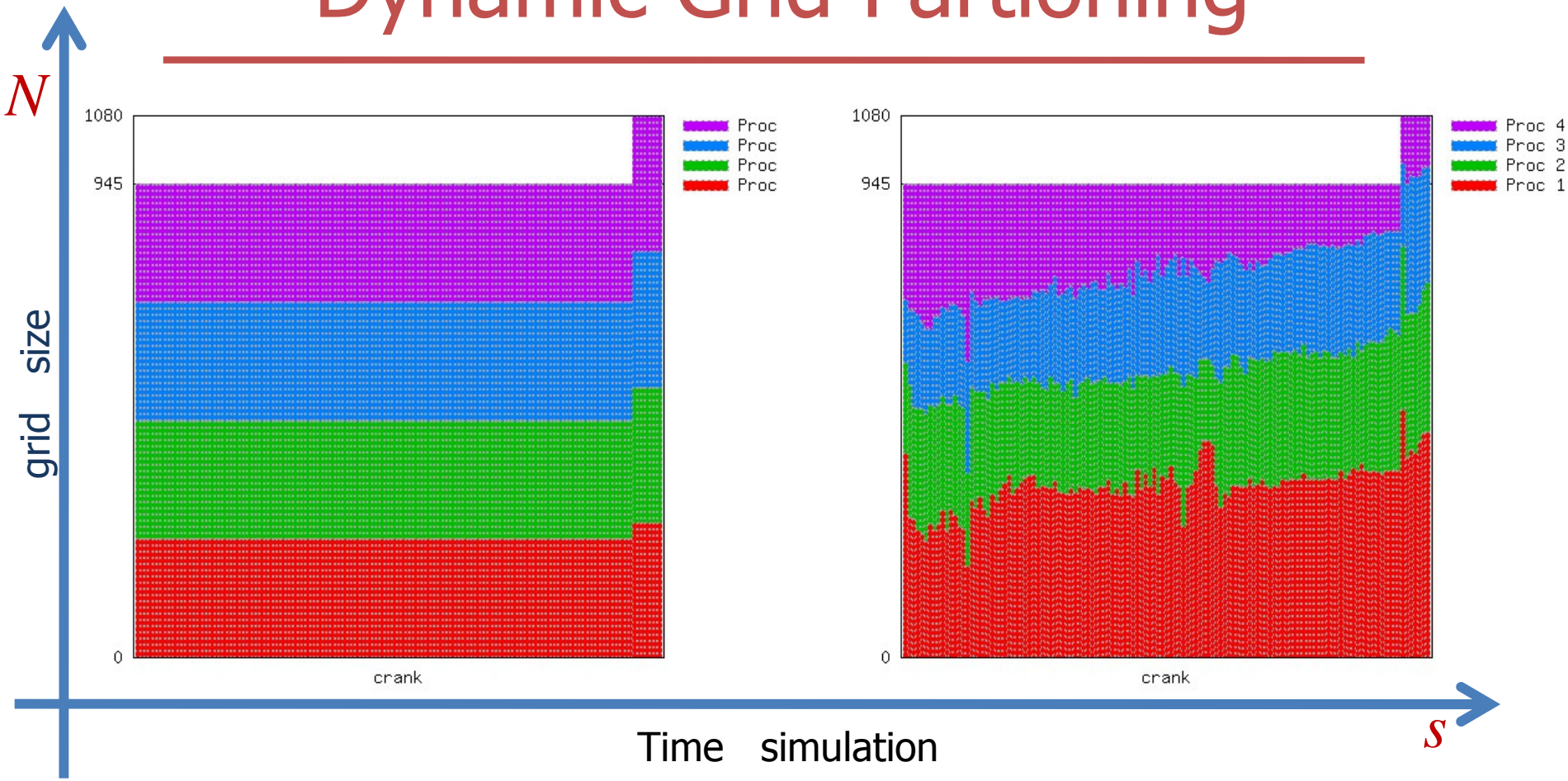
computational load imbalance at the splitting interval s

If R_s > r with threshold 0<r<<1
a new partition of consecutive grid indices is obtained

$$\Lambda^{new} := \{\mathrm{I}^{new}{}_{\mathrm{p}}\}_{p=1,\ldots,P} \quad \text{where} \quad T_p^{new} = \sum_{i \in I_p^{new}} T^s(i) \le T^s_{med}$$

then a **re-distribution of grid cells data according with the new partition** is applied using a **greedy algorithm** for local work flow scheduling

Dynamic Grid Partioning



Block partitioning

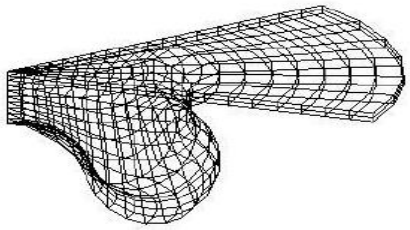
Block partitioning with dynamic load balancing

Testbed

HP XC 6000 Linux cluster

- 64 bi-processor nodes
  - Intel Itanium 2 Madison processor
  - 1.4 Ghz clock frequency
  - 4 GB RAM
- Quadrics QsNetII Elan 4 network (bandwidth of 900 MB/sec, latency 5 *μsec*)
- HP Linux for High Performance Computing 3 (kernel 2.4.21)
- GNU Compiler Collection 4.2
- HP MPI implementation 2.01

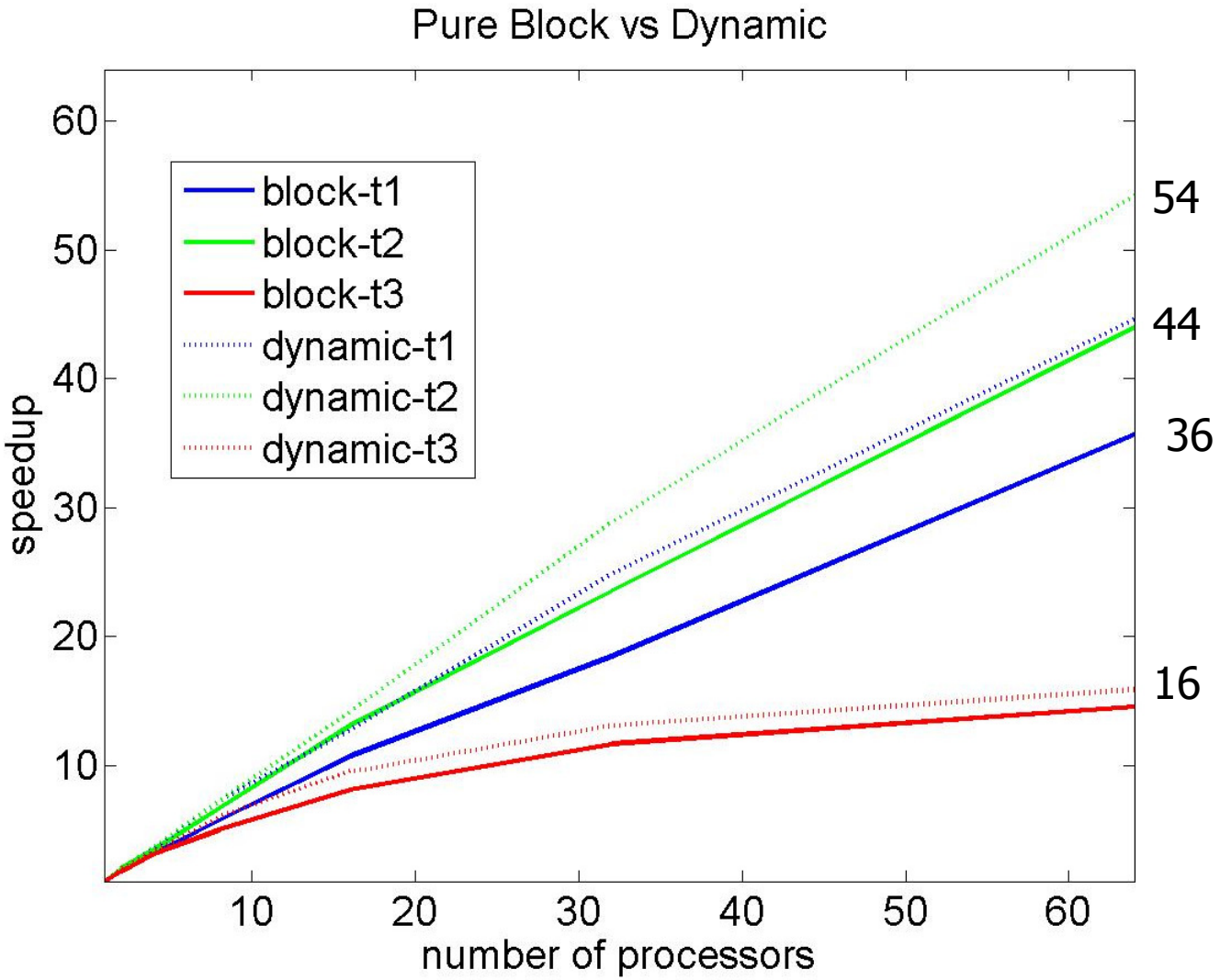
COMPUTATIONAL GRID: 3D CYLINDRICAL SECTOR,WITH ~ 4500 CELLS



Simulation cycle: **from injection to 40 CA ATDC**

Grid size during simulation: min:945 max: 1755

Block Partitioning vs Dynamic Partitioning



Some Conclusions and Remarks

- **Detailed combustion** for engine simulation **is a typical context for HPC**
- Reliable and efficient predictive simulations require **advanced and adaptive solvers**
- **Data partitioning and load balancing are main issues** to manage adaptivity in a parallel environment
- Flexible software framework should be adapt to **variability in test cases** (operative conditions and grid sizes), **hardware platform** (processor heterogeneity), **software platform** (interfacing with different components needed for different sub-models)

References

1. L. Antonelli, P. D'Ambra, Dynamic Load Balancing for High-Performance Simulations of Combustion in Engine Applications, Proc. of the 19th EUROMICRO Conf. on PDP, IEEE CS Pub., 2011.
2. L. Antonelli, M. Briani, P. D'Ambra, V. Fraioli, Positivity Issues in Adaptive Solutions of Detailed Chemical Schemes for Engine Simulations, Communications to SIMAI Congress, Vol. 3, 2009.
3. L. Antonelli, P. Belardini, P. D'Ambra, F. Gregoretti, G. Oliva, A Distributed Combustion Solver for Engine Simulations on Grids, Journal of Computational and Applied Mathematics, Vol. 226, 2, 2009.
4. L. Antonelli, P. D'Ambra, F. Gregoretti, G. Oliva, and P. Belardini, A Parallel Combustion Solver within an Operator Splitting context for Engine Simulations on Grids, in Proc. of LSSC'07, LNCS Series, Vol. 4818, 2007.
5. P. Belardini, C. Bertoli, S. Corsaro, P. D'Ambra, V. Fraioli, Application of a Multi-Method ODE Software Component for Simulations of Ultra-Low Emissions Diesel Engines, in Proc. of Fisita'06, P027, 2006.
6. P. Belardini, C. Bertoli, S. Corsaro, P. D'Ambra, Introducing Combustion-Turbulence Interaction in Parallel Simulation of Diesel Engines, in Proc. of HPCC'06, M. Gerndt, D. Kranzlmuller eds., LNCS series, Vol. 4208, 2006.
7. P. Belardini, C. Bertoli, S. Corsaro, P. D'Ambra, The Impact of Different Stiff ODE Solvers in Parallel Simulations of Diesel Combustion in Proc. of HPCC'05, L.T. Yang, O. Rana, B. Di Martino, J.J. Dongarra eds., LNCS series, Vol. 3726, 2005.
8. P. Belardini, C. Bertoli, S. Corsaro, P. D'Ambra, Multidimensional Modeling of Advanced Diesel Combustion System by Parallel Chemistry, in Proc. of Society Automotive Engineers World Congress, SAE Inc. Pub., 2005.
9. P. Belardini, C. Bertoli, S. Corsaro, P. D'Ambra, Parallel Simulation of Combustion in Common Rail Diesel Engines by Advanced Numerical Solution of Detailed Chemistry, in Applied and Industrial Mathematics in Italy, M. Primicerio, R. Spigler, V. Valente eds., World Scientic Pub., 2005.

Thanks
for your attention