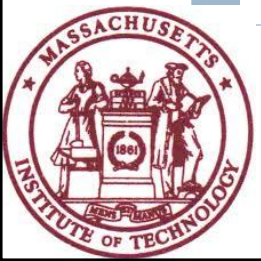
Redesigning Combustion Modeling Algorithms for GPU Computing and Application in Multi-dimensional HCCI engine Simulations with Very Detailed Reaction Mechanisms

Yu Shi and William Green
Chemical Engineering, MIT
O. O. ("Luwi") Oluwole and Hsi-Wu Wong
Aerodyne Research Inc.

ICCS 2011 Manycore and Accelerator-based High-performance Scientific Computing Workshop



Research motivations

- Solving reaction mechanisms (chemical kinetics of fuels) in reacting flow simulations is expensive due to stiff ODEs (i.e. many reaction rates must be computed for many species, many times!)

	A	b	E
ic8h18+h=c8h17+h2	4.38E+07	2.0	7760.0
ic8h18+o2=c8h17+ho2	2.22E+15	0.0	42904.0
c8h17+o2=c8h17oo	1.05E+11	0.0	0.0
ch3+ho2=ch3o+oh	9.60E+13	0.0	30950.0
co+oh=co2+h	1.26E+07	1.3	-758.0
...			
...			
...			

$$\begin{cases} \dfrac{dy_1}{dt} = f_1(T, P, y_1, \cdots y_{N_s}) \\ \vdots \\ \dfrac{dy_{N_s}}{dt} = f_{N_s}(T, P, y_1, \cdots y_{N_s}) \\ \dfrac{dT}{dt} = f_{N_s+1}(T, P, y_1, \cdots y_{N_s}) \end{cases}$$

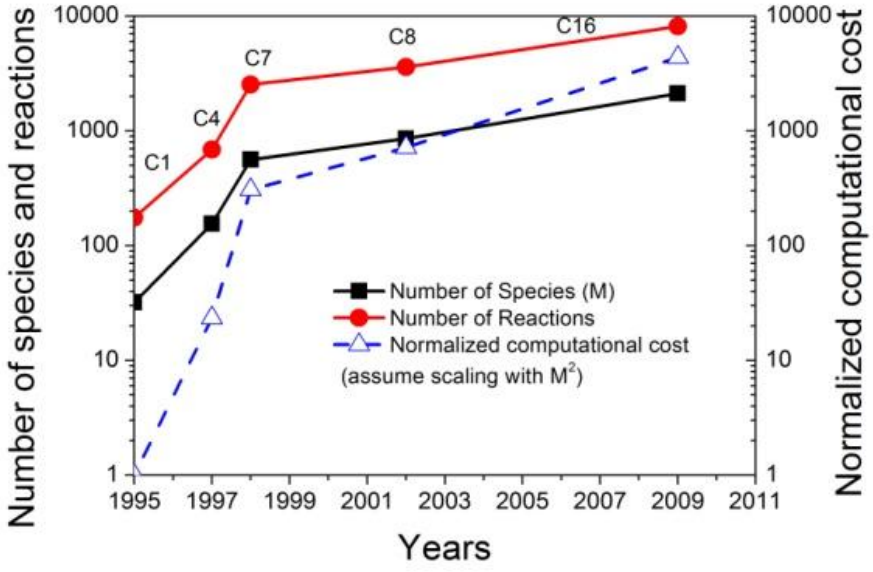
$$f_i = \sum_{j=1}^{N_r} \upsilon_{ij}\omega_j,\ f_{N_s+1} = \frac{\sum_{i=1}^{N_s} u\dot{y}_i}{C_v}$$

$$\omega_j = k_j^f \left(\prod_{i=1}^{N} c_i^{\upsilon_i'} - K_c^{-1} \prod_{i=1}^{N} c_i^{\upsilon_i''}\right),$$

$$k_j^f = P_{cor,j} A_j T^{n_j} \exp\left(\frac{-E_j}{R_0 T}\right)$$

Research motivations

- In reacting flow simulations, over 95% simulation time is spent on solving the stiff ODEs (operator splitting scheme)
- In practice, it is affordable (but still expensive) to solve reaction mechanisms with only a few tens species and hundreds reactions using implicit stiff ODE solvers



Poor scaling of implicit ODE solvers (super-quadratically with species number)

Detailed reaction mechanisms of hydrocarbon fuels contain thousands species and reactions
e.g., C16H34 (hexadecane)
2115 species and 8157 reactions
C11H22O2 (methyl decanoate)
2877 species and 8555 reactions

Research motivations

- There is an urgent need to improve the computational efficiency of stiff ODE solvers to incorporate large detailed reaction mechanisms in reacting flow CFD simulations

Take advantage of many-core structure of GPU and parallel computing

GPU-accelerated chemical kinetics ODE solver (Identify parallelism)

- Solving chemical kinetics problem is essentially to solve stiff ODEs subject to initial conditions

$$y' = \frac{dy}{dt} = f_1(T, y), \quad y(0) = C_1$$

$$T' = \frac{dT}{dt} = f_2(T, y), \quad T(0) = C_2$$

- Implicit backward Euler method for solving stiff ODEs is popular in the community, the existing solvers include DASSL (DASAC in Senkin), DVODE, and etc.

$$\frac{dy}{dt} = \frac{y_n - y_{n-1}}{h} = f_1(T_n, y_n)$$

$$y_n - y_{n-1} - h \times f_1(T_n, y_n) = 0$$

Function evaluation
involves species rate calculation
essentially, reaction rates calculation

- Newton iteration

$$y_n^{\upsilon+1} = y_n^{\upsilon} - (I - h\frac{\partial f}{\partial y})^{-1}(y_n^{\upsilon} - y_{n-1} - h \times f(T_n, y_n^{\upsilon}))$$

Reactions are independent

Linear system

$$-(I - h\frac{\partial f}{\partial y})^{-1}(y_n^{\upsilon} - y_{n-1} - h \times f(T_n, y_n^{\upsilon})) = \delta$$

$$(I - h\frac{\partial f}{\partial y})\delta = -(y_n^{\upsilon} - y_{n-1} - h \times f(T_n, y_n^{\upsilon}))$$

$$y_n^{\upsilon+1} = y_n^{\upsilon} + \delta$$

Solving the linear equation system
Heavily rely on LU decomposition
on LHS

Matrix elements are independent

These two parts take over 97% time when solving large mechanisms

GPU-accelerated chemical kinetics ODE solver

- Reaction rate evaluations

C kernel function to evaluate reaction rates in parallel

(with Fortran wrapper)

- Accelerated linear algebra

  CULA library is employed to supersede LAPACK LU factorization subroutine which is originally used in most implicit ODE solvers, such as DASAC and DVODE tested in the present study

- Apply GPU-accelerated DASAC and DVODE solvers to Senkin code (Chemical kinetics simulation code in Fortran)

Redesigning Combustion Modeling Algorithms for the Graphics Processing Unit (GPU): Chemical Kinetic Rate Evaluation and Ordinary Differential Equation Integration, Combustion and Flame, 2011 (Accepted)

Comparison study

- Constant-volume ignition simulations with CPU, CPU+GTX260, CPU+Tesla C2050

  - Test conditions: stoichiometric fuel/air mixtures under 1200 K and 20 bar for 0.003 second simulation time
  - Hardware: Intel I7-930 (only use single-core)
    GTX 260 (Capacity 1.3)
    Telsa C2050 (Capacity 2.0, ECC off (5% performance boost))
  - Compiler: Intel Fortran + GCC (NVCC) + O3 optimization, on Ubuntu 64bit system

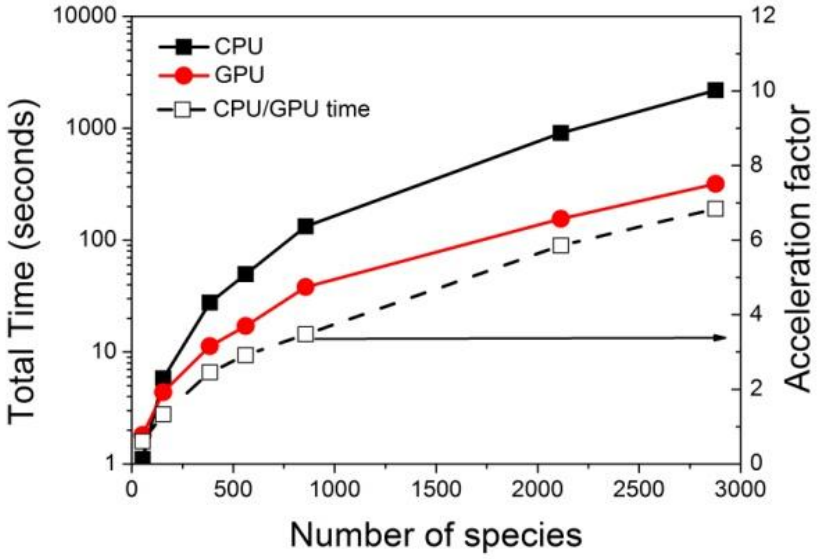
Comparison study

- Tested reaction mechanisms

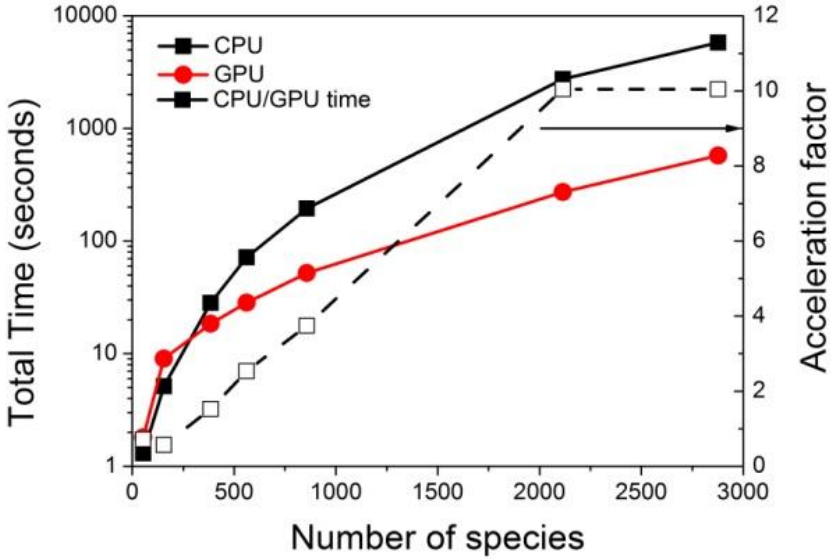
Reaction mechanism	Species number	Reaction number
GRI-mech 3.0	53	325
n-butane	155	689
n-pentane	385	1896
n-heptane	561	2539
Iso-octane	857	3606
n-hexadecane	2115	8157
methyl decanoate	2877	8555

Comparison study (CPU VS GTX 260)

Simulation time (left Y axis) and Speed-up factor (right Y axis)



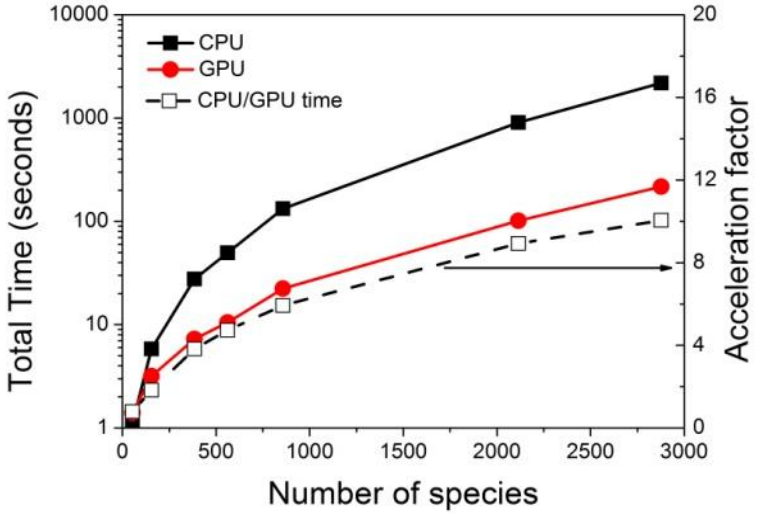
DASAC solver with Senkin



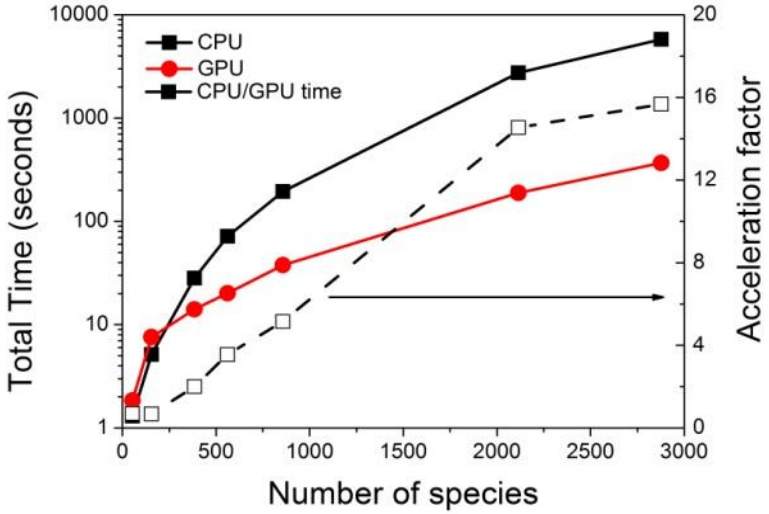
DVODE solver with Senkin

Comparison study (CPU VS Tesla C2050)

Simulation time (left Y axis) and Speed-up factor (right Y axis)



DASAC solver with Senkin

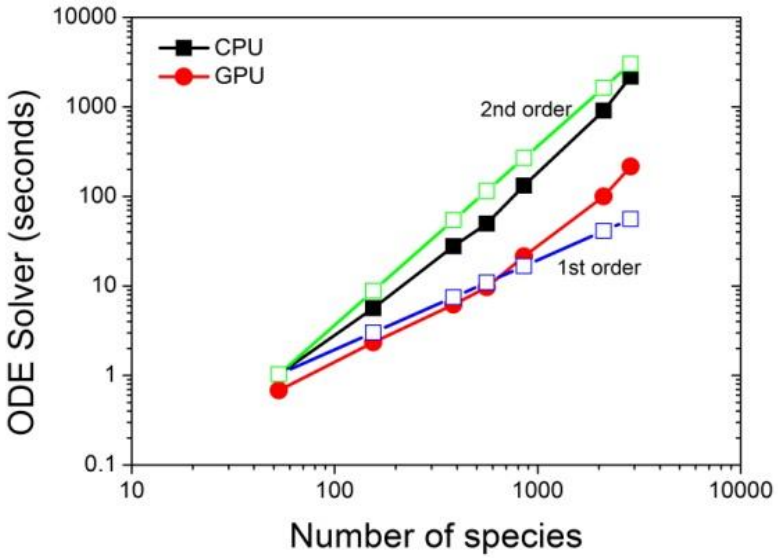


DVODE solver with Senkin

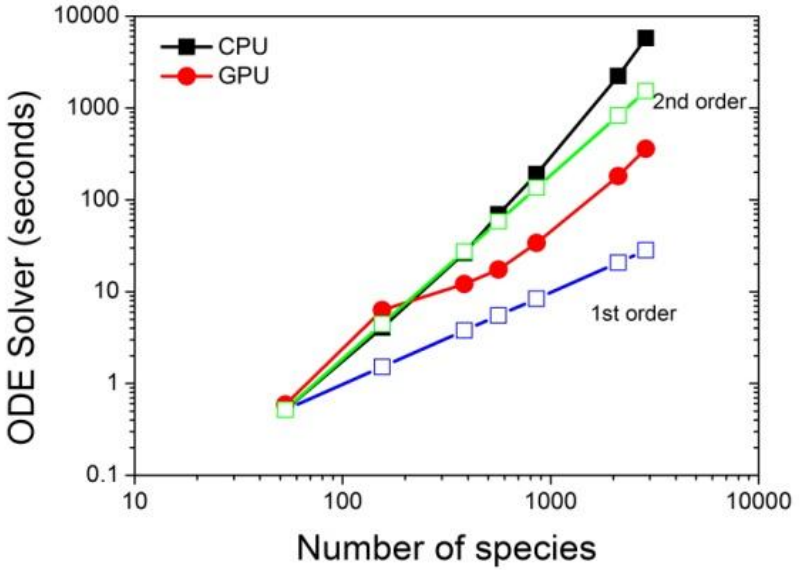
Tesla C2050 simulation only ~ 1.5 to 1.8 times faster than GTX 260?!

Comparison study (CPU VS Tesla C2050)

Computational time scaling with the species number



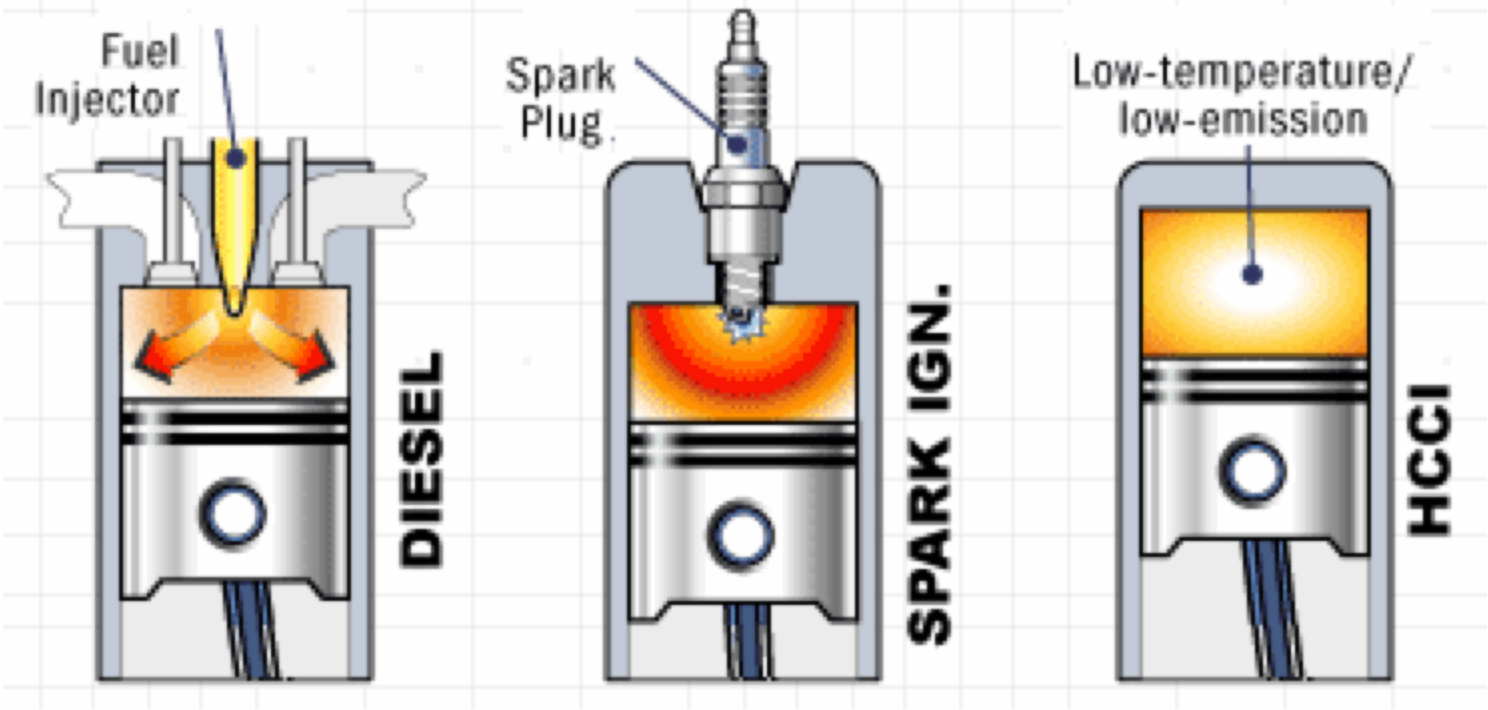
DASAC solver with Senkin



DVODE solver with Senkin

Application in HCCI engine simulations

- Homogeneous Charge Compression Ignition Engine



Combustion performance and emissions are mainly characterized by the chemical kinetic behavior of fuels in HCCI engines

On-the-fly mechanism reduction method + GPU-assisted chemistry solver

- Apply graph theory based model reduction method to reduce the number of species and reactions on-the-fly in engine combustion simulations
  - At each simulation time step, species and reaction number vary from thousands to tens.
  - GPU-assisted ODE solver is called to solve large dynamically-formed mechanisms
  - CPU-only ODE solver is called to solver small mechanisms
  - Threshold: 300 species and 1000 reactions

Engine specifications and test conditions

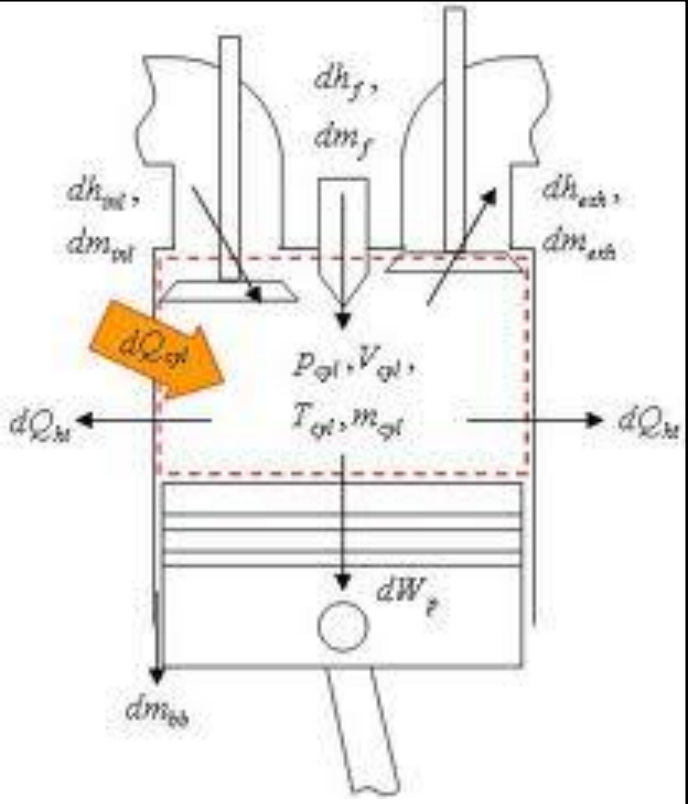
Bore (cm)×Stroke (cm)	13.716×16.51	
Compression Ratio	16.1	
IVC	143 BTDC	
EVO	130 ATDC	
Speed (rev/min)	821	1737
Intake temperature (K)	350	350
Intake pressure (bar)	1.9	1.4
Equivalence ratio	0.2	0.6
Load (bar)	5	11
EGR (%)	0	0

Caterpillar 3400E

Single-zone simulations

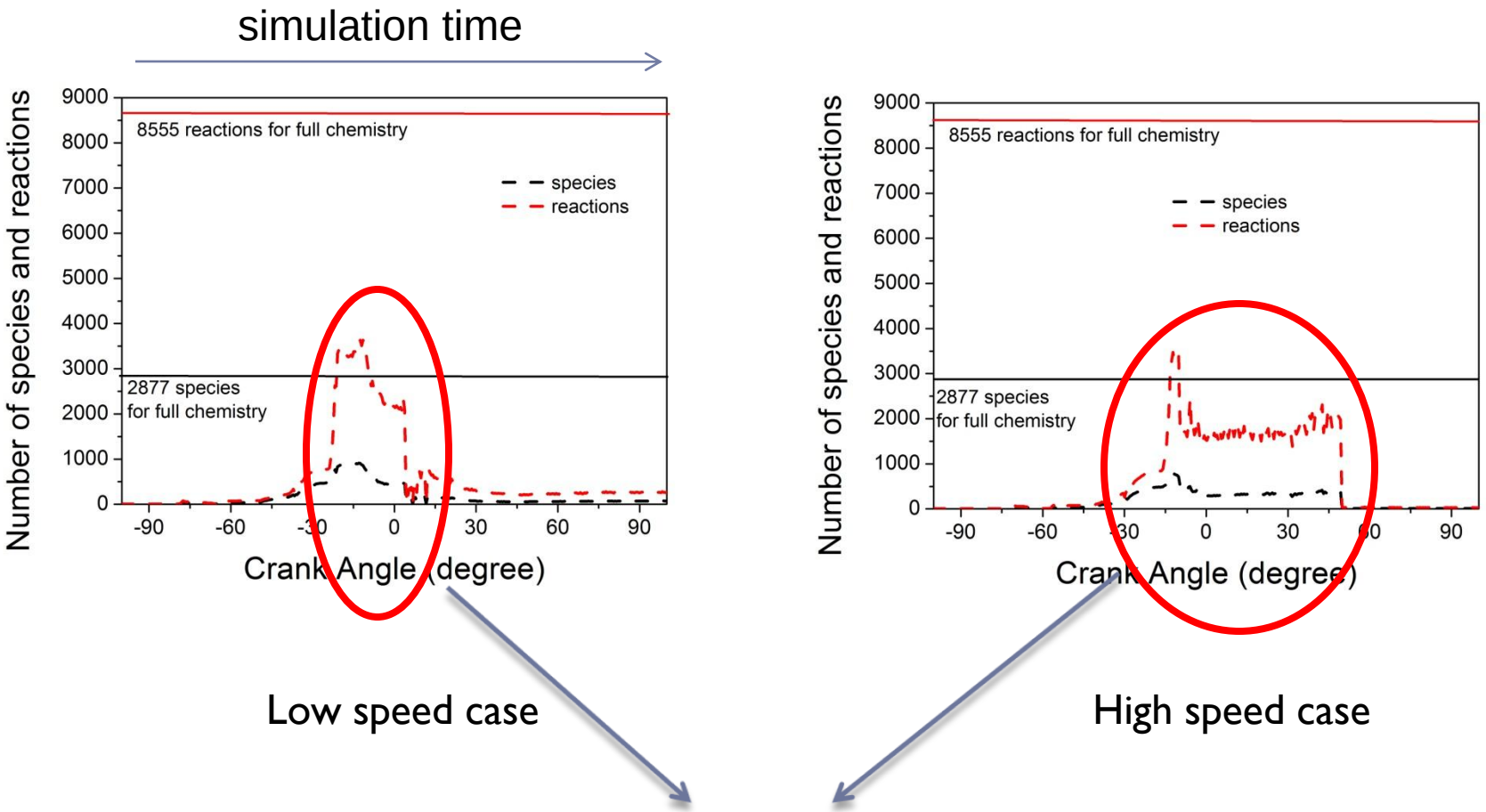
Solve ODEs with varied volume

Detailed methyl decanoate (biodiesel) mechanism with 2877 species and 8555 reactions



Methods	Slow speed (in minutes)	High speed (in minutes)
Full Chemistry (CPU)	784.85	827.21
Full Chemistry (GPU)	41.23	43.41
On-the-fly reduction (GPU/CPU)	3.47	2.72

Single-zone simulations (species number)

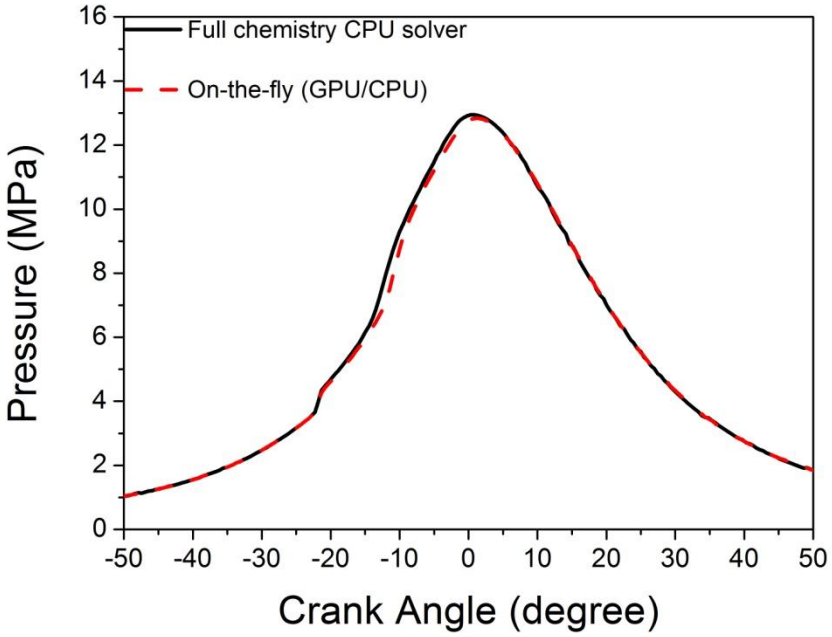


Low speed case

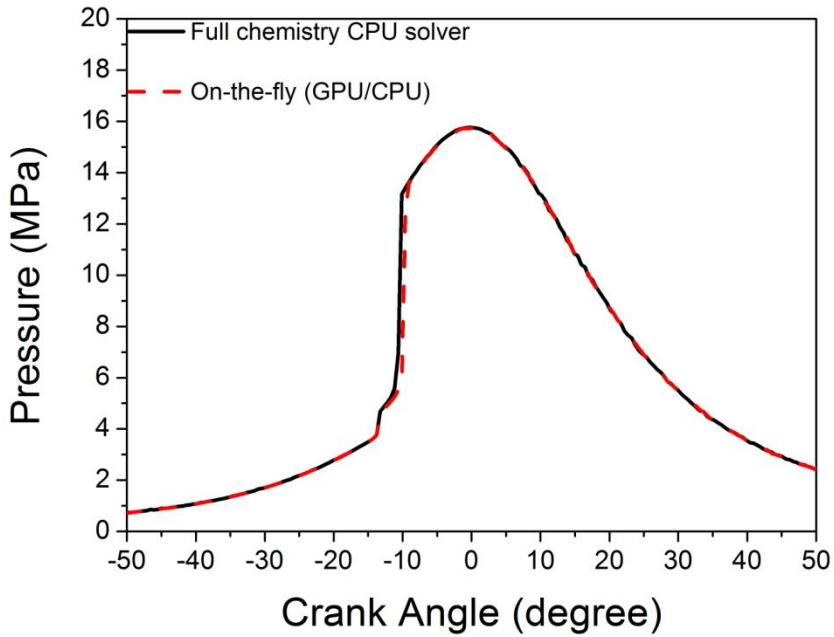
High speed case

GPU-assisted chemistry solver

Single-zone simulations (pressure trace)

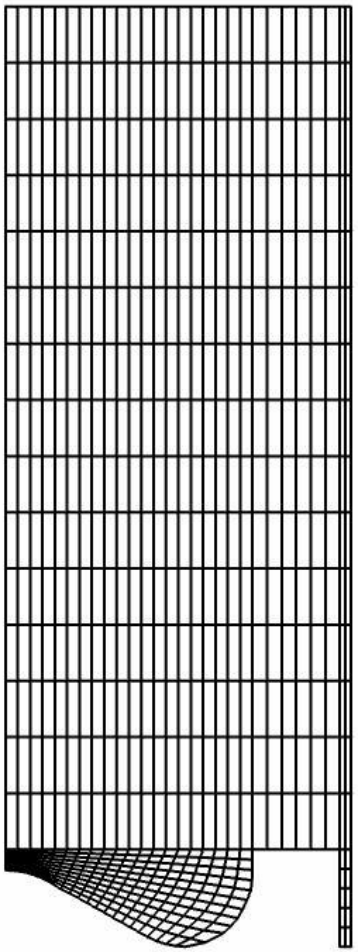


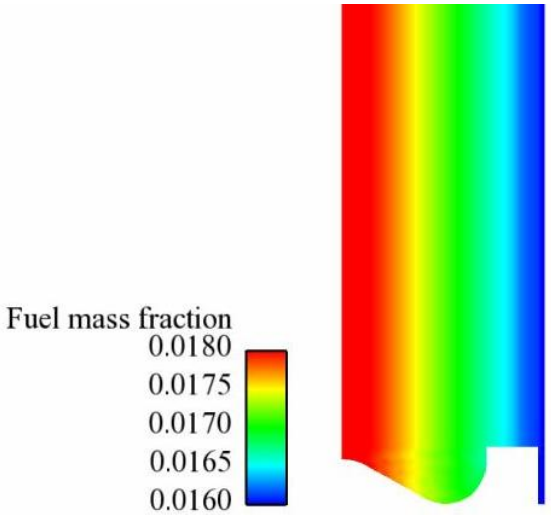
Low speed case



High speed case

Multi-dimensional simulations



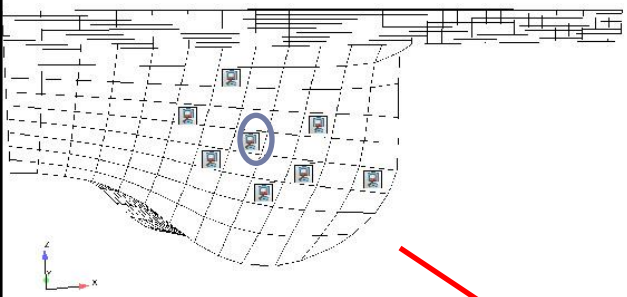


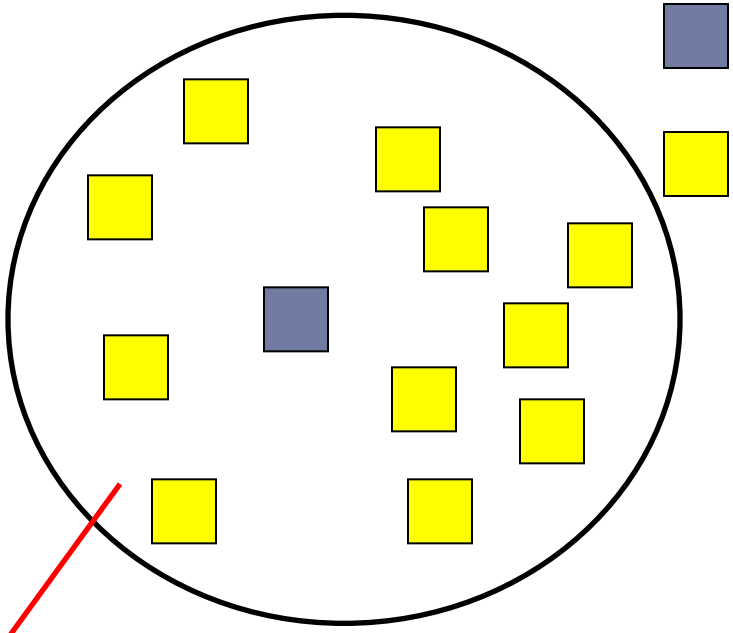
650 cells using KIVA-Chemkin code with GPU-assisted DVODE solver

Detailed methyl decanoate (biodiesel) mechanism with 2877 species and 8555 reactions

Adaptive multi-grid chemistry model

Mapping cells to a group
(2-D illustration)





Center cell

Thermodynamically similar cells
(cells with similar temperature and equivalence ratio ϕ)

$$\phi = \frac{2C^{\#}_{-CO_2} + H^{\#}_{-H_2O}/2 - z'C^{\#}_{-CO_2}}{O^{\#}_{-CO_2-H_2O} - z'C^{\#}_{-CO_2}}$$

Computational domain

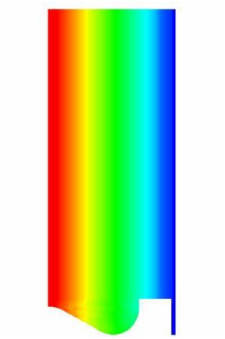
Chemkin Solver
(Using on-the-fly reduction and GPU-assisted ODE solver)

mass conservation and species gradient preservation

Remapping back to individual cells

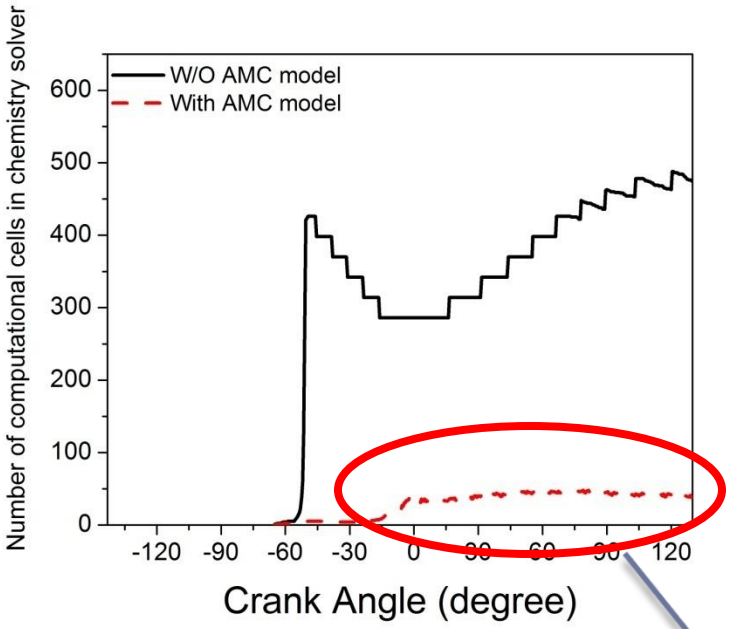
Multi-zone simulations (AMC model + on-the-fly reduction + GPU)

Stratified charge

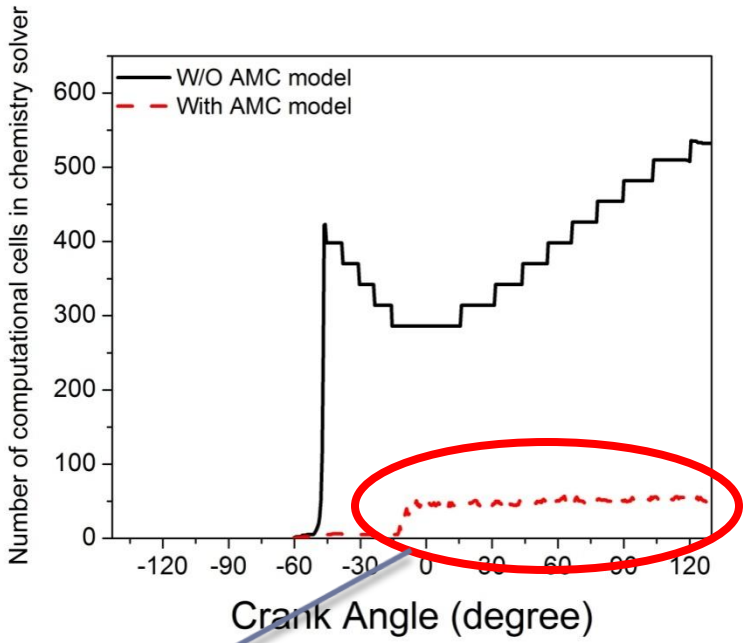


Methods	Slow speed case (in hours)	High speed case (in hours)
W/O AMC model (GPU/CPU)	166.04	56.64
AMC model (GPU/CPU)	14.73	10.75

Multi-dimensional simulations (zone number)



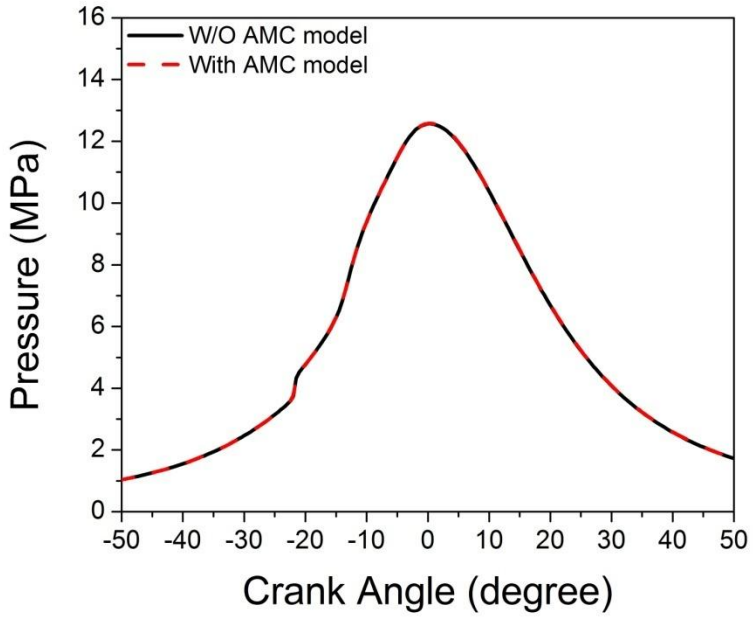
Low speed case



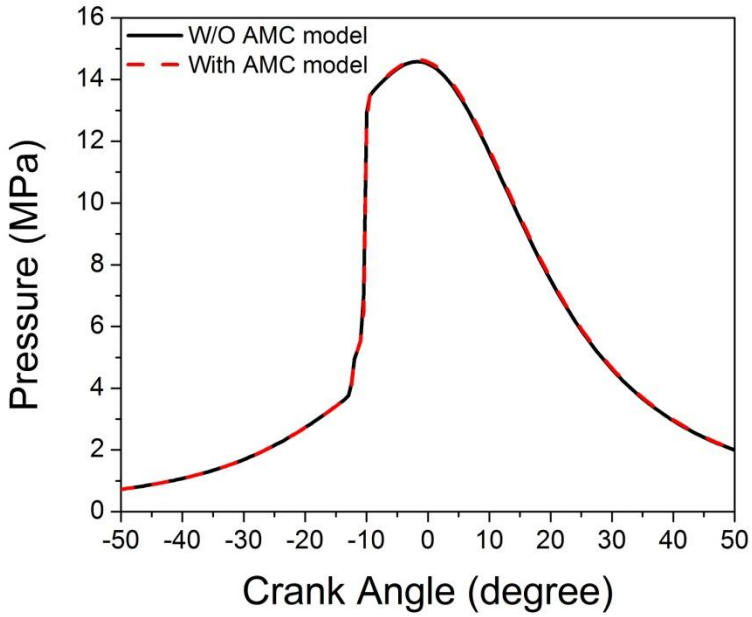
High speed case

Solve much less grids

Multi-dimensional simulations (pressure trace)



Low speed case



High speed case

Conclusions

- Significant simulation acceleration are obtained by exploiting GPUs in combustion simulations.

- The computational time is seen to scale less than quadratically with the number of chemical species in the kinetic mechanism when using the GPU, as compared to the super-quadratic scaling normally seen with CPU-only chemical kinetics computations.

Conclusions (Cont.)

- The combination of the on-the-fly model reduction scheme with the GPU/CPU hybrid ODE solver accelerated the single-zone HCCI engine simulations with the detailed methyl decanoate mechanism by up to 300-fold!

- By using the AMC model together with the model reduction method and the GPU/CPU ODE solver, we completed 2-D HCCI engine simulations within only 15 hours on a PC with a Tesla C2050 GPU card

Email:
Yu Shi, yushi@mit.edu
Luwi Oluwole, oluwoleo@aerodyne.com