

Acceleration of Stiff Chemistry Through TensorFlow and Machine Learning

Dirk Van Essendelft, Kyle Buchheit, Opeoluwa Owoyele, Terry Jordan



June 27, 2019



Solutions for Today | Options for Tomorrow



NETL/CPFD History

Carbonaceous Chemistry for Computational Modeling (C3M)

C3M v.4.0.2

File Tools Help Dev

Gasification
Chemical Looping
Mercury Capture

Fuel Name Arkwright

Proximate Analysis			
Fixed Carbon	Volatile Matter	Ash	Moisture
0.52	0.3863	0.0825	0.0112

Ultimate Analysis				
Carbon	Hydrogen	Oxygen	Nitrogen	Sulfur
0.837802	0.062893	0.054066	0.015447	0.029792

Kinetics

Chemistry Sub-Model	Kinetic Package	Condition
Pyrolysis	MGAS	Condition
Char Oxidation	PCCL	
Moisture Release	CPD	
Gasification	FG-DVC	
Water Gas Shift	HPTR	
Gas Phase Combustion	UQ	
User Defined(BETA)		

Dynamic Report

Devolatilization:

Mass Basis: Volatile Matter -> 0.652142Tar + 0.023289CO + 0.035993CO₂ + 0.052930CH₄ + 0.043403H₂ + 0.061399H₂O + 0.007622HCN + 0.055259H₂S + 0.000000NH₃ + 0.018420C₂H₄ + 0.026677C₂H₆ + 0.022866C₃H₈ + 0.000000C₃H₆

Mole Basis: Volatile Matter -> 0.081324Tar + 0.021867CO + 0.021509CO₂ + 0.086774CH₄ + 0.566252H₂ + 0.089635H₂O + 0.007417HCN + 0.042642H₂S + 0.000000NH₃ + 0.017269C₂H₄ + 0.023333C₂H₆ + 0.014291C₃H₈ + 0.000000C₃H₆

$r_s = 341.6 \cdot \exp(-38534.6 / RT_s) \cdot \epsilon_s \cdot p_s \cdot X_{vm}$

Tar Oxidation:

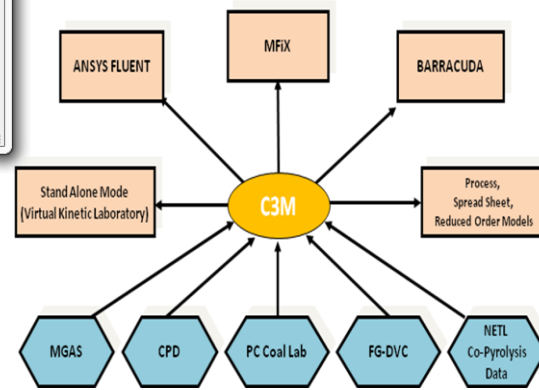
```
# This file was generated by C3M
version 1500

# Material properties ... must be before init
MaterialProp 13
equState 1
compressible 1
material
  formulaDescription "Ash"
  chemFormula "Ash"
  state S
  density 2.931000e+03
  thermalConductivity 1.500000e+00 0.000000e+00 0.000000e+00 0
  massDiffusivity 0.000000e+00 0.000000e+00 0.000000e+00 0
  molecularWeight 1.200000e+02
  ioHiTemp 1.000000e+03
  deltaHf(298) 0.000000e+00
  Cp_h1 1.007750e+03 1.188492e+00 -4.829250e-04 9.365411e-08 -4
  Cp_lo -2.152111e+02 3.048490e+00 1.318064e-03 -4.420546e-06 1
endMaterialProp

# Particles (aka Solids)
Solids 1
solidsDescription 1
solidspecid 1
massPerVol 0.000000e+00
solidMaterial
  materialName "C"
  massFraction 4.342000e-01
endSolidMaterial
solidMaterial
  materialName "Ash"
  massFraction 2.051000e-01
endSolidMaterial
solidMaterial
  materialName "H2O_s"
  massFraction 1.800000e-01
endSolidMaterial
solidMaterial
  materialName "VOL"
  massFraction 1.807000e-01
endSolidMaterial
type
  radius1
  radius2
  dragModel
  typeDrag
  multiplicity
  constant
  coeF1
  coeF2
  agglomeration
  radCut 1.800000e-05
  radFile ""
endAgglomeration
endDragModel
```

```
*****
*****
ftb,double fto, double mv
0.;
position */
cH4 = 0., betaC_cH4 = 0.
cAd_h2 = 0., betaAd_h2o = 0
cH2 = 0., betaC_cH4 = 0.
cAc_h2 = 0., betaAc_h2o = 0
changed via the file : f
= 0.;

*****
*****
NMAX(0) = 0
NW_g(1)
NW_g(2)
NW_g(3)
NW_g(4)
NW_g(5)
NW_g(6)
NW_g(7)
NW_g(8)
NMAX(1)
NW_s(1,1) = 12 # Fixed Carbon (FC)
NW_s(1,2) = 56 # Volatile Matter (VM)
NW_s(1,3) = 18 # Moisture
double mw_co = 28.01055, mw_co2 = 44.00995, mw_ch4 = 16.04303, mw_c2h
mw_c3h8 = 44., mw_c6h6 = 78., mw_h2 = 2.01594, mw_h2o = 18.01534
static double
double heatd;
/* pro
fc_ar
vm_ar
ash_ar
moist_ar
/* ultimate analysis */
carbon = 0;
hydrogen = 0;
oxygen = 0;
```



<https://mfix.netl.doe.gov/c3m/>

NETL/CPFD History

Carbonaceous Chemistry for Computational Modeling (C3M)

We have the
chemistry but solving
it is....

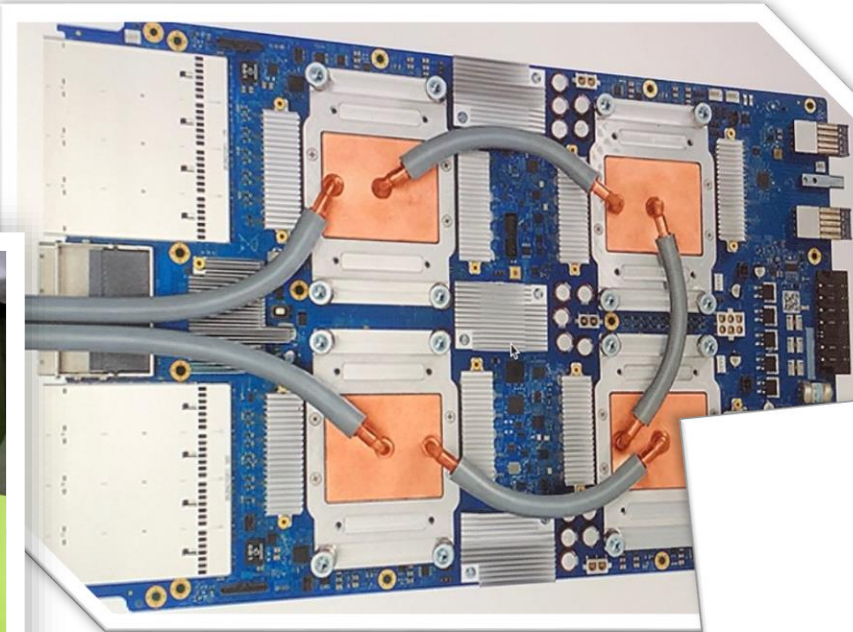
Up to 90% of Reacting CFD is Spent
in Implicit Solvers for Chemistry!!!



The Computational World Is Changing

Application Specific Integrated Circuits for AI/ML

Mixed Precision
Calculations



Combined Operations

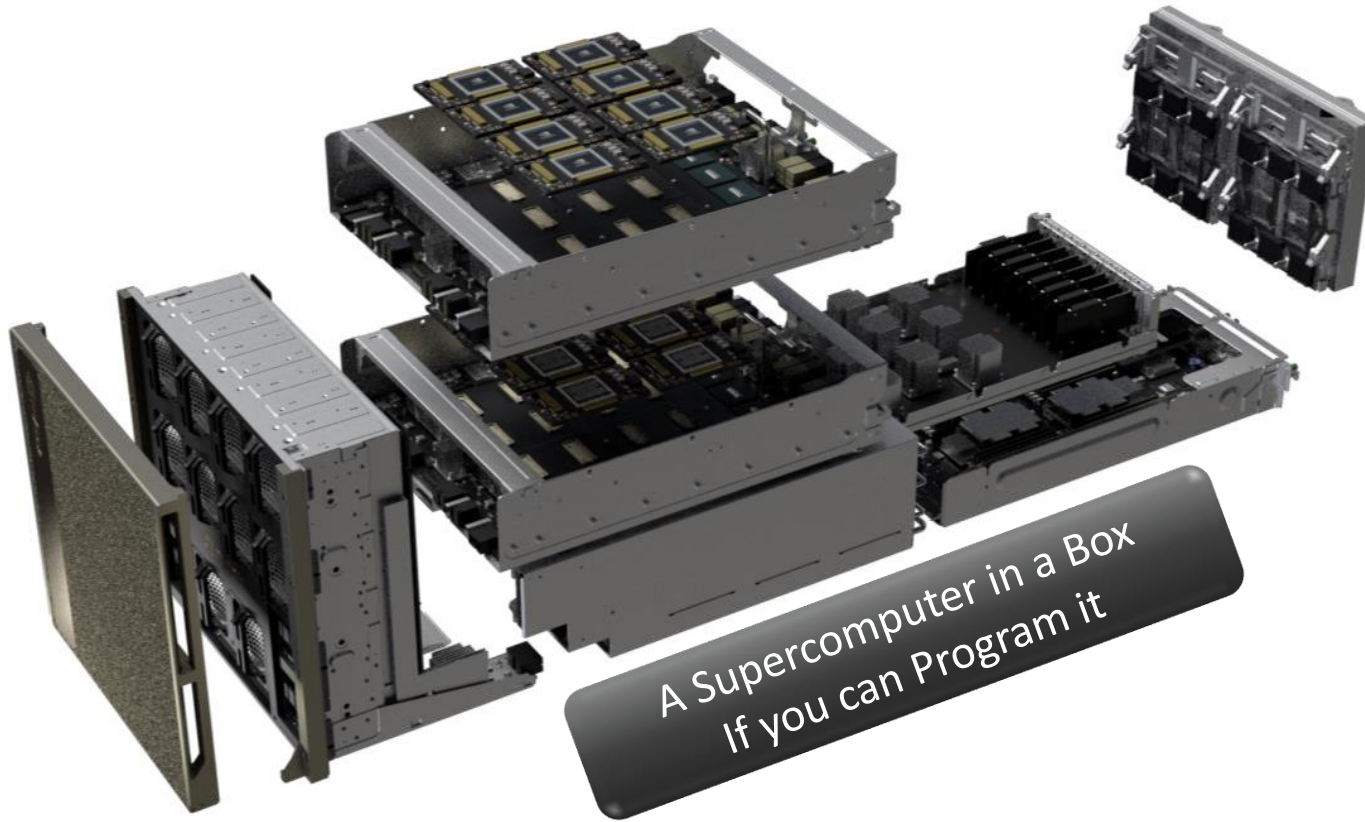
Exceptional Dense Matrix
Multiplication Efficiency



The Computational World Is Changing

Advent of Powerful Multi-Accelerator Hardware

NVIDIA DGX-2



- **2 petaFLOPS per node!!!**
- 16 V100s
- 1.5 TB system ram
- 512 GB GPU ram
- 81920 Cuda Cores
- 10240 Tensor Cores
- NVLink 2.0
- Needs Applications

<https://www.microway.com/preconfiguredsystems/nvidia-dgx-2/dgx-2-explodingdetail/> Used by permission

A new Paradigm

Three Pillars of Future Computation

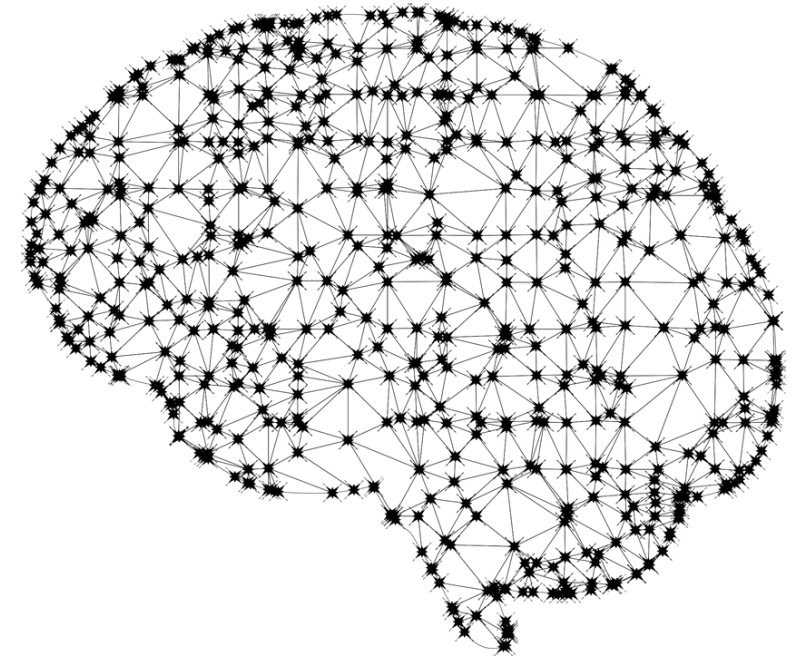
Software



Hardware



Machine Learning



A new Paradigm

Three Pillars of Future Computation

Software



Not Just ML/AI

- A **VERY** powerful and optimized math library
- Hardware agnostic
- Spread Computation over multiple devices and architectures
- No reprogramming for different devices
- Simple to incorporate ML into workflow
- Programs like Numpy
- Can be linked through C to any compiled language

A new Paradigm

Three Pillars of Future Computation

Hardware

Good for AI = Good for Us

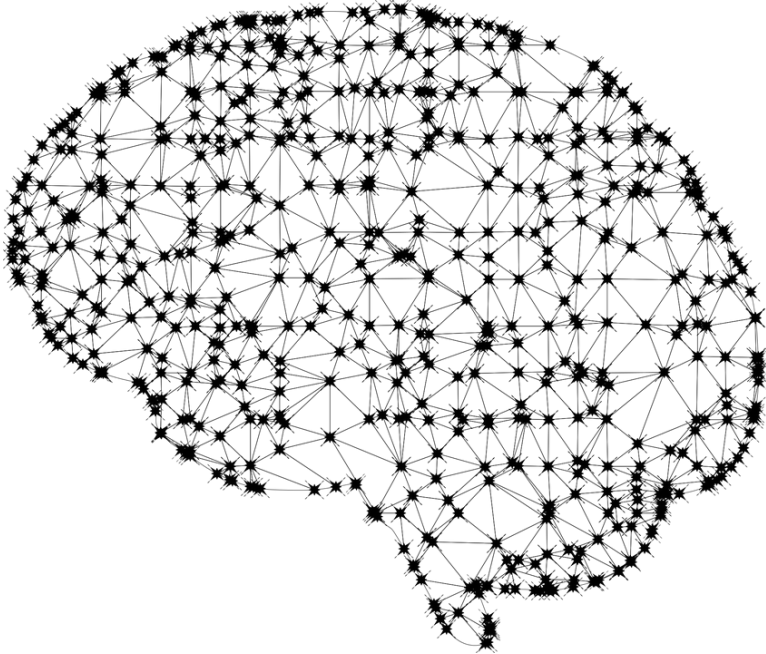
- All Scientific Computing benefits from very fast Dense/Sparse Matrix Multiplication
- Exceptionally high throughput
- Low POWER
- VERY fast RAM
- Interconnect speeds growing
- Hybrid architectures evolving



A new Paradigm

Three Pillars of Future Computation

Machine Learning

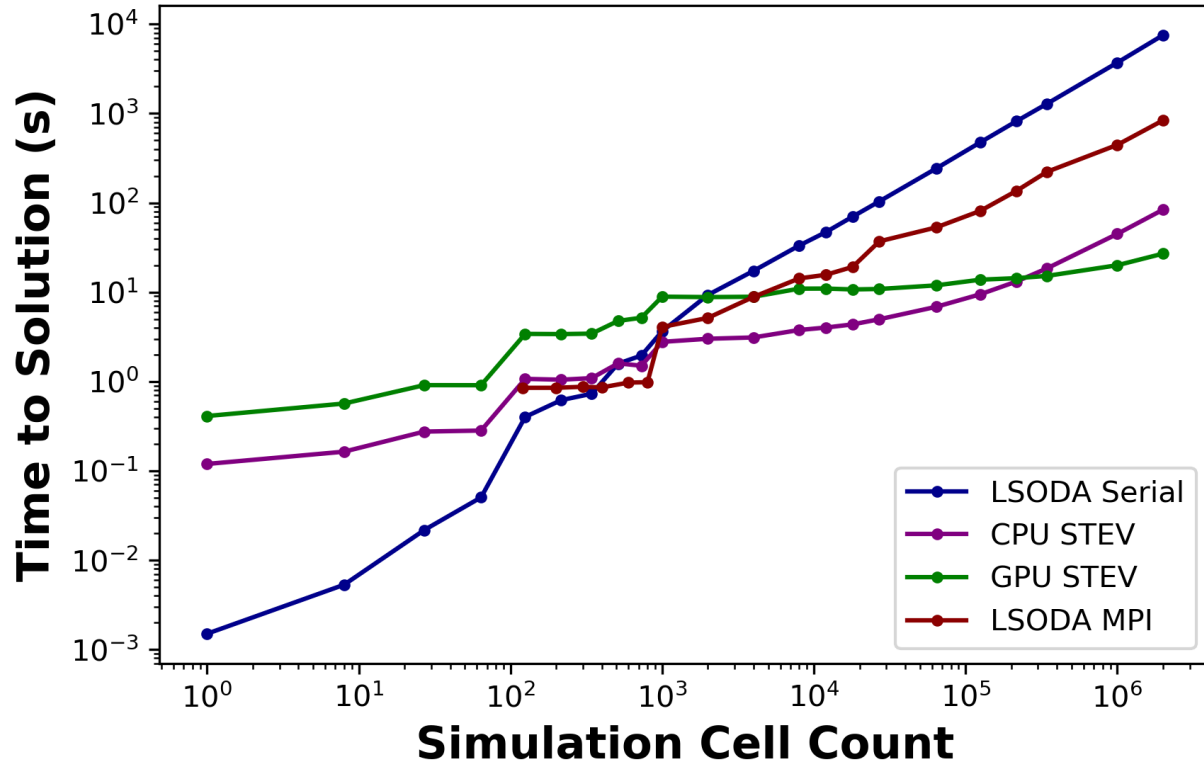


Judicious AI/ML Use

- AI/ML estimates can be very accurate and very quickly attained
- Almost any kind of calculation can be replaced by AI/ML estimators
- Stochastic nature of AI/ML predictions demands use of correctors in scientific computing to maintain accuracy

The Stabilized Explicit Variable-load Solver

The STEV Solver: Speed is Here!



- **Standard C3M G/C Chemistry**
 - 14 Gas Species
 - 4 Solid Species
 - 2 Densities, 2 Temps, 2 Vol Fracs, P_g, dp
 - 11 Reactions
 - PCCL Gasification, Char Combustion, Moisture Release, Devolatilization
 - MGAS Gas Phase Ox, WGS
- **Random Thermochemical IC**
- **Solve to 1 millisecond**

200x Speedup on P100s

The Stabilized Explicit Variable-load Solver

The STEV Solver: Time Step Stabilization

$$r_i^* = \max(-r_i, 1.0 \times 10^{-30}) \quad \forall i \in (1 \dots n)$$

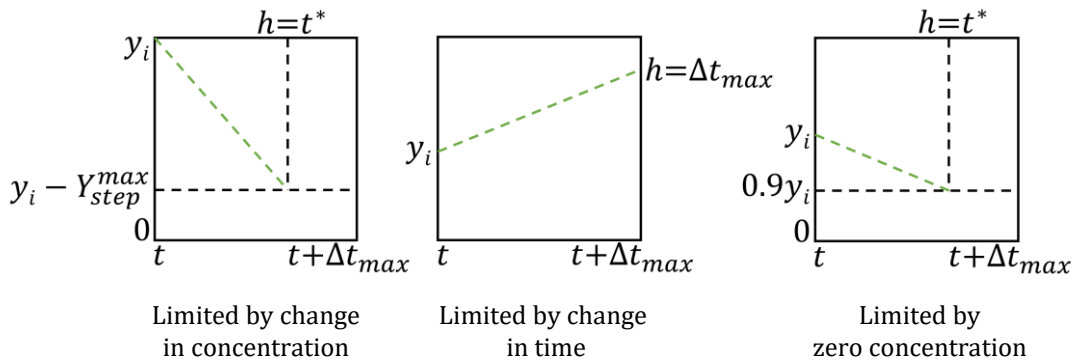
$$y_i^* = \begin{cases} 1.0 & \text{if } y_i \ll 1 \times 10^{-20} \\ 0.9y_i & \text{otherwise} \end{cases}$$

$$t^* = \min_i \left(\frac{\min(Y_{step}^{max}, y_i^*)}{r_i^*} \right)$$

$$\Delta t_{max} = t_f \cdot \delta^{max}$$

$$h = \min(t^*, \Delta t_{max})$$

- Choose time step based on change in concentrations
 - Y_{step}^{max} , primary tolerance control (0.0001)
 - δ^{max} , helps ignition delay accuracy (0.01)
 - y_i^* helps prevent negative overstepping
- Ensures solver only works hard during stiff conditions
- Essentially makes solver stable for unlimited stiffness with irreversible reactions provided numeric over/underflow is not reached



The Stabilized Explicit Variable-load Solver

The STEV Solver: Small Species Stabilization



$$y_i^{**} = y_i \gamma(y_i)$$

$$\gamma(y_i) = \frac{y_i}{(\alpha + y_i)}$$

- In most cases, small species concentrations are not important
 - $\gamma(y_i) \approx 1$ when $y_i \gg \alpha$
 - $\gamma(y_i) \approx 0$ when $y_i \ll \alpha$
 - $\alpha = 1 \times 10^{-5}$, typically safe value
- Helps reduce stiffness in limit of small species and transition to zero (prevents negative overstepping)
- May not be appropriate for all kinetic models

The Stabilized Explicit Variable-load Solver

The STEV Solver: Reversible DFT Stabilization

$$r_{store}^i = r_{rev}h$$

$$D_{rev} = |DFT(r_{store}^i)|$$

$$f_{ring} = \frac{D_{rev}^0}{\max(D_{rev}^j |_{n < j < N})}$$

$$rOF_0 = 1.0$$

$$rOF^* = \begin{cases} 1.0/r_{fac} & \text{if } f_{ring} < Tol_{ring} \text{ and } D_{rev}^0 < Tol_{freq} \\ r_{fac} & \text{otherwise} \end{cases}$$

$$rOF_{t+1} = \min(rOF_t * rOF^*, 1.0)$$

$$r_{rev}^{eff} = r_{rev} * rOF_t$$

- **Detect ringing and equilibrium based on the history of reaction progress and slow down reaction in response**
 - History of 13 most recent reaction progress changes
 - Calculate f_{ring} by comparing high/low frequency magnitudes of DFT
 - Update a rate oscillation factor
 - Calculate effective rate
 - Tol_{ring} (0.6), Tol_{freq} (0.0014), r_{fac} (1.03)
- **Works well as long as there aren't large numbers of stiff AND reversible AND competitive reactions**

The Stabilized Explicit Variable-load Solver

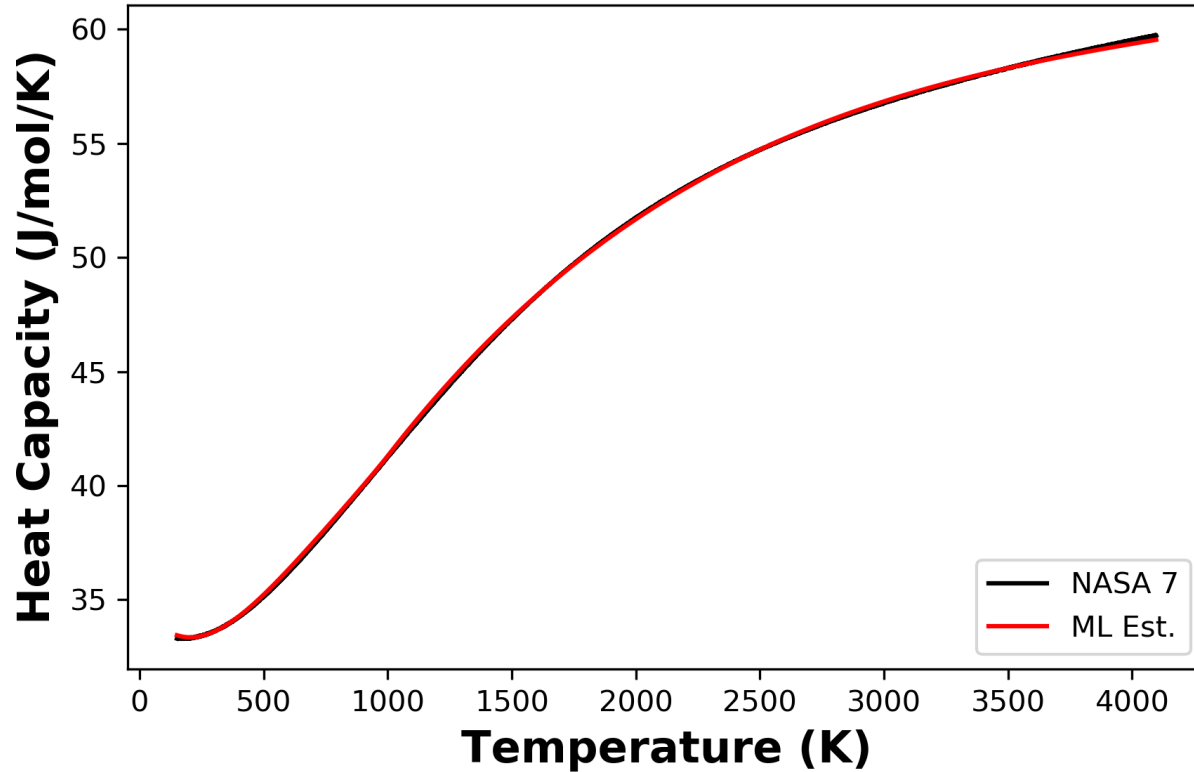
The STEV Solver: Variable-Load



- All cells evolve at their own time step
- Unfinished cells are gathered and removed from active calculation when final time is reached
- Finished cells are gathered and written to storage variable
- Allows for maximum vectorization, warp convergence, and minimum computational effort

The Stabilized Explicit Variable-load Solver

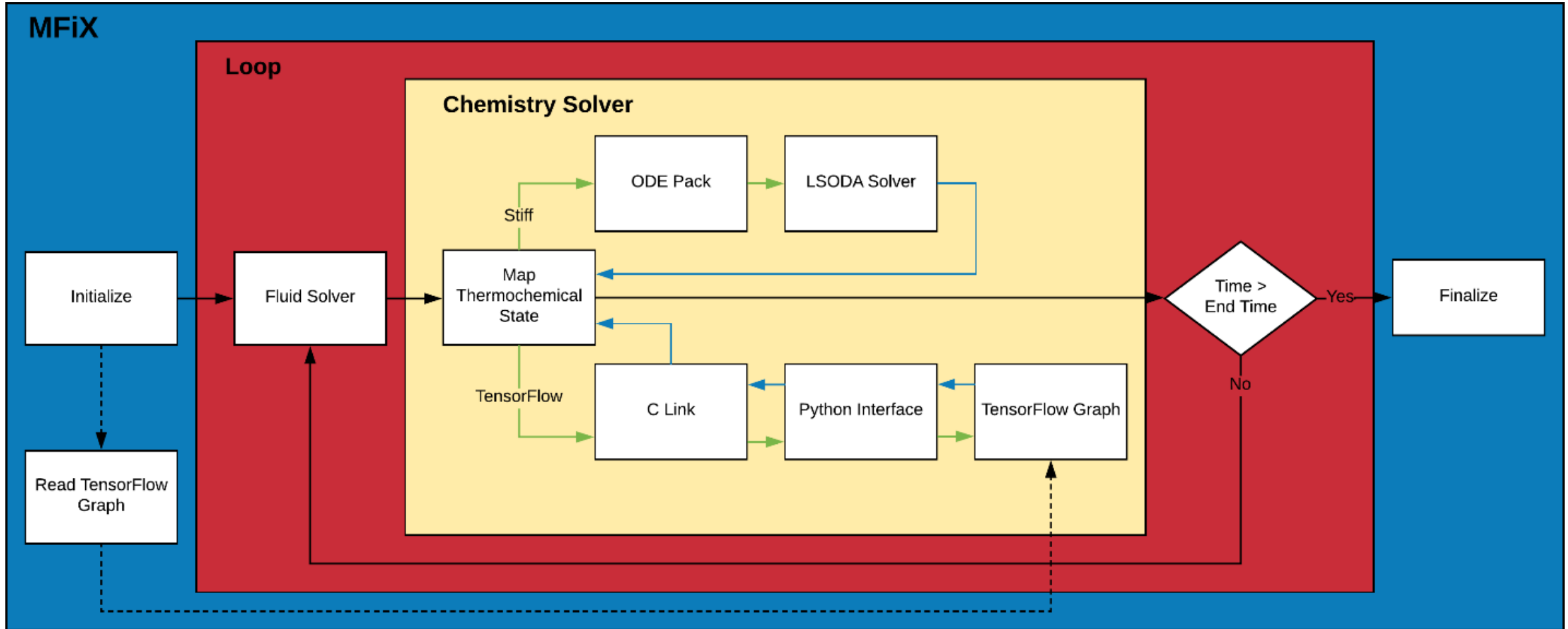
The STEV Solver: Laminar Flame Example



- Estimate thermodynamic properties (CP and H) with a shallow network
- Use Levenberg-Marquardt Second order solver for training
- Trains in a couple minutes
- Saves anywhere from 10-50% of computational effort per solver iteration depending on model
- Insignificant difference in accuracy

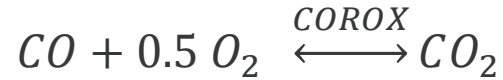
The Stabilized Explicit Variable-load Solver

The STEV Solver: Linking to CFD



The Stabilized Explicit Variable-load Solver

The STEV Solver: Laminar Flame Example



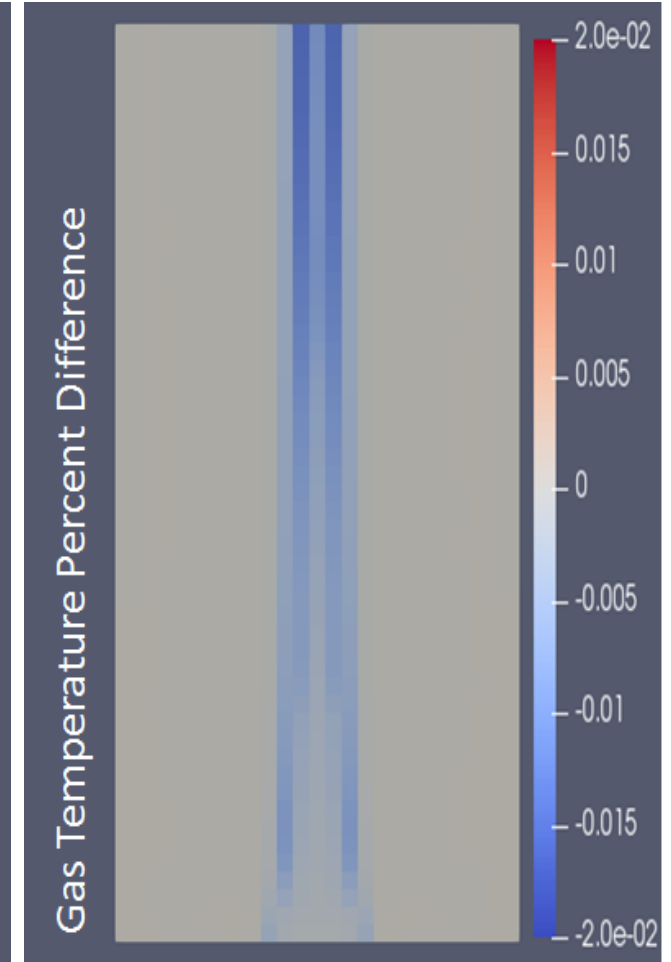
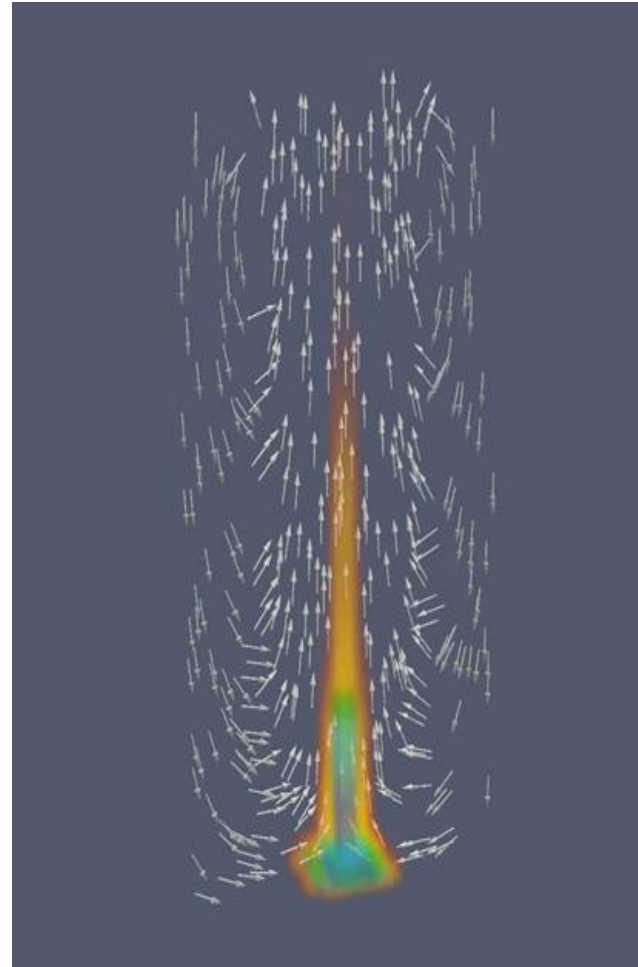
$$r_{MEPOX} = k_{MEPOX} e^{-E_a/RT} C_{CH_4}^{0.5} C_{O_2}^{0.65}$$

$$r_{COROX} = k_{COROX} T^{0.7} e^{-E_a/RT} \left(C_{CO} C_{O_2}^{0.5} - \frac{C_{CO_2}}{K} \right)$$

$$K = \left(\frac{1 \text{ atm}}{RT} \right)^{\Delta N} e^{-\Delta G_{rxn}/RT}$$

Reaction	k	E_a (cal/mol)
MEPOX	4.9E+09	35500
COROX	2.00E+08	12000

Reaction set based on the following units: cm, s, mol, cal

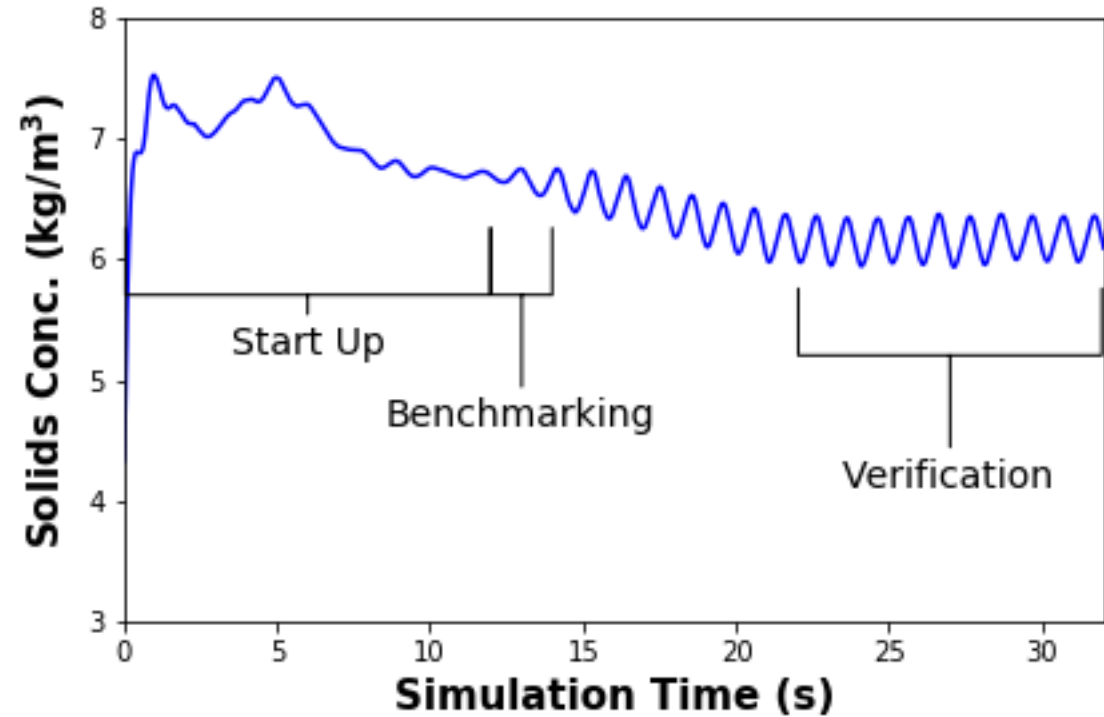
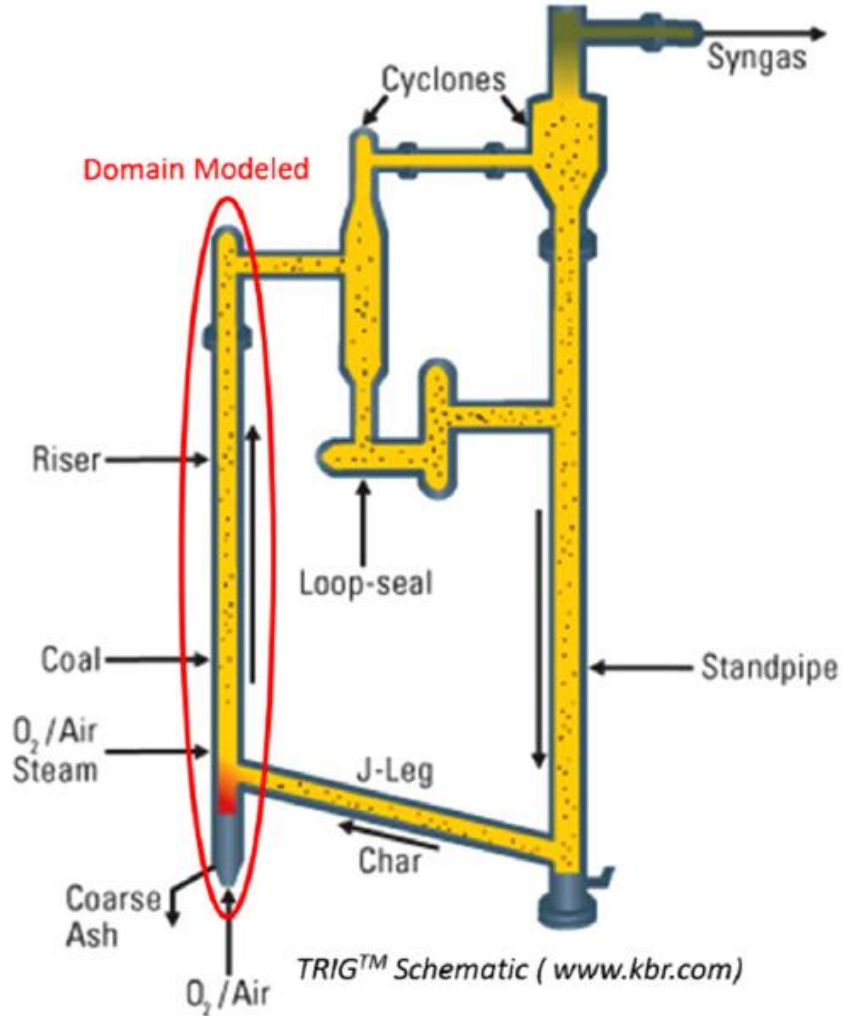


Matched LSODA to within 0.02%

The Stabilized Explicit Variable-load Solver

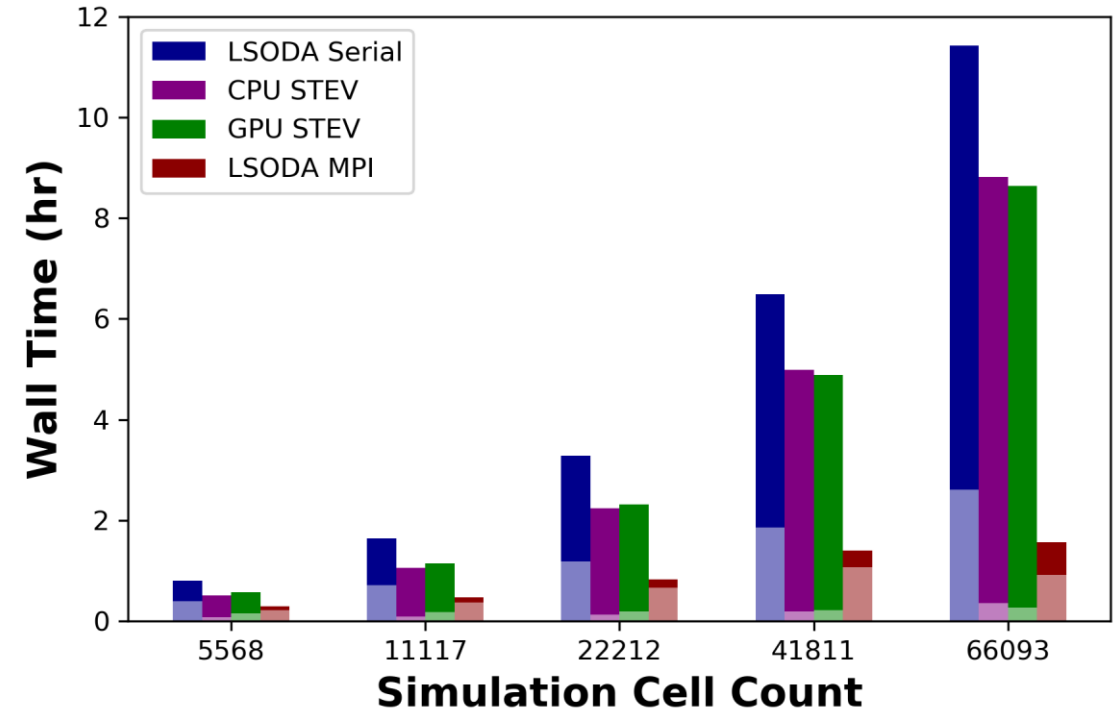
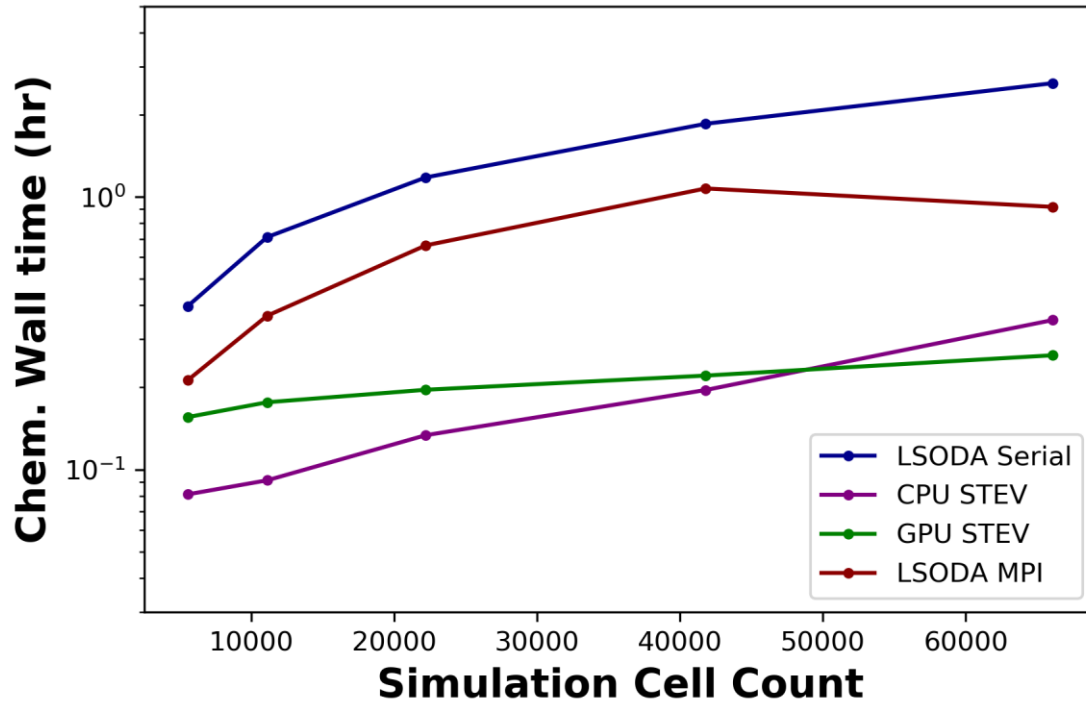
The STEV Solver: PSDF Gasifier

- Simulated the riser of the PSDF TRIG gasifier at multiple grid resolutions
- Standard C3M G/C model
 - See arXiv paper for full mechanism details



The Stabilized Explicit Variable-load Solver

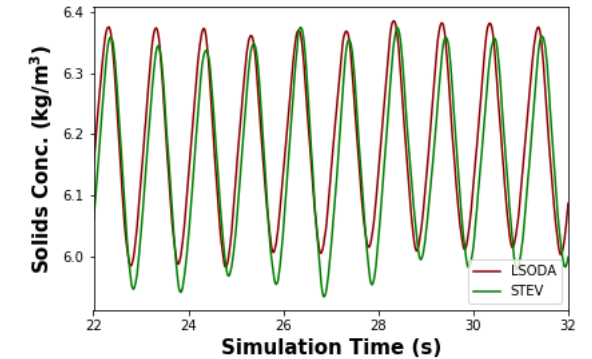
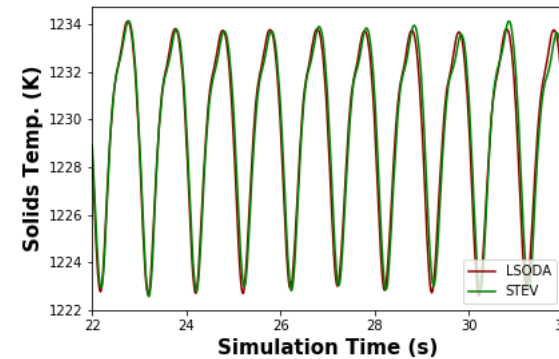
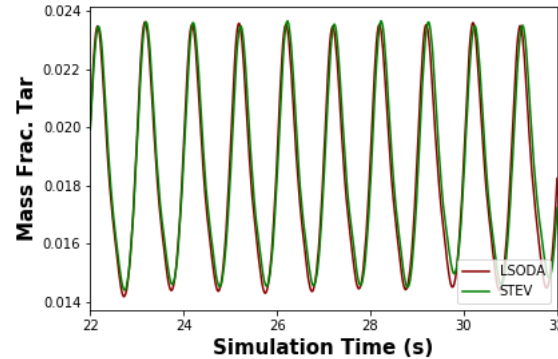
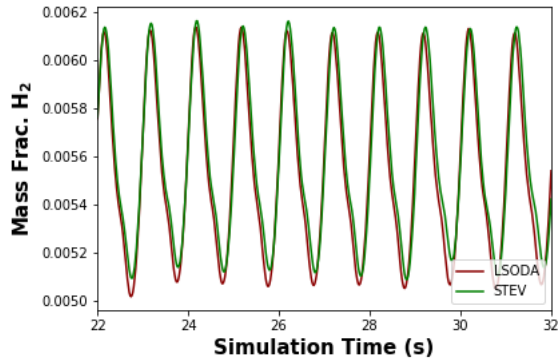
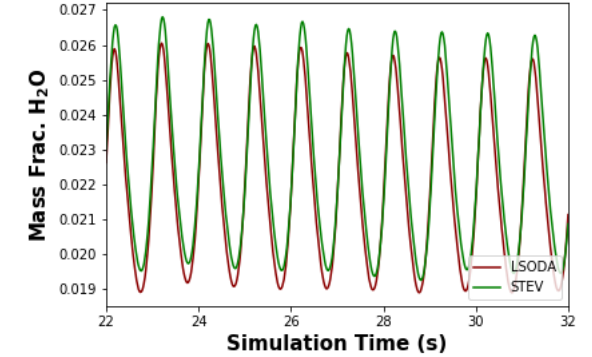
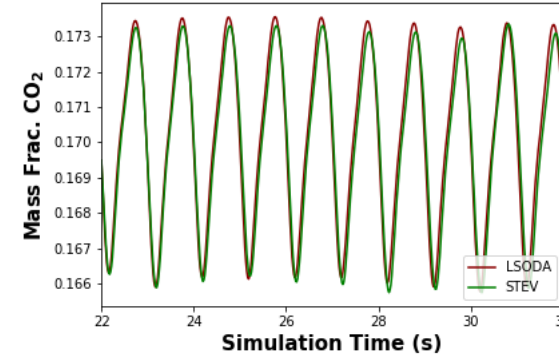
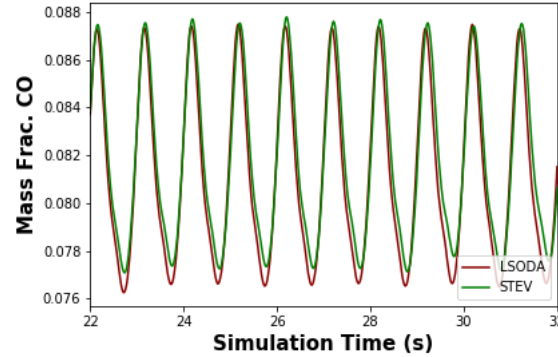
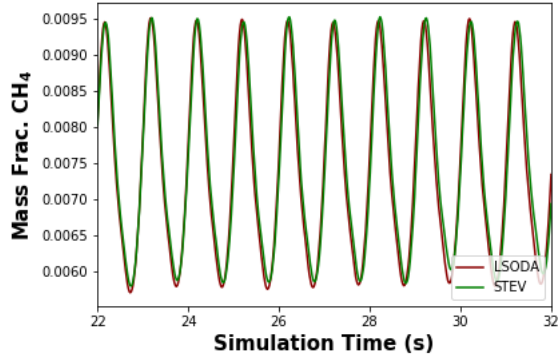
The STEV Solver: PSDF Gasifier



~10x faster than LSODA in Serial
~5x faster than LSODA in MPI

The Stabilized Explicit Variable-load Solver

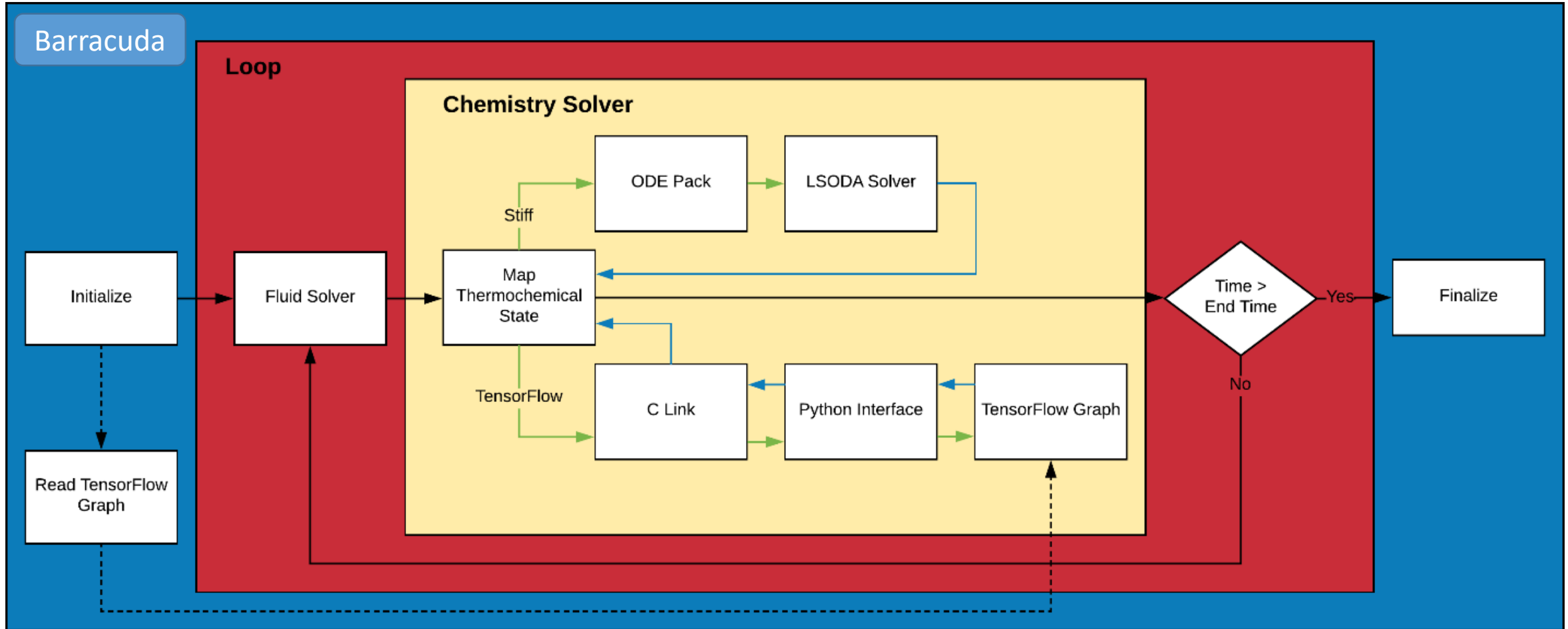
The STEV Solver: PSDF Gasifier



Matched LSODA runs to within 2.8% relative error

The Stabilized Explicit Variable-load Solver

The STEV Solver: Linking to Barracuda



Questions??

