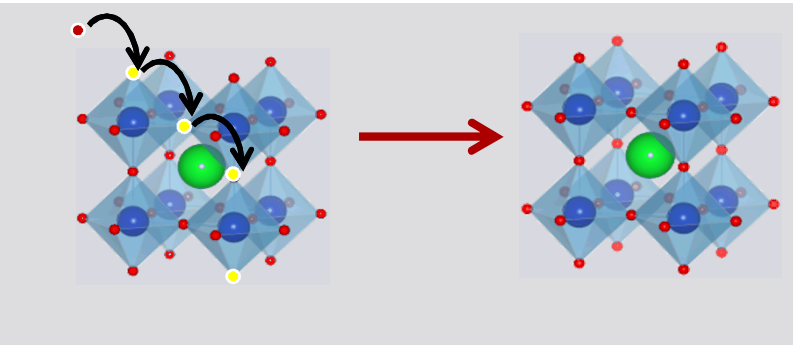
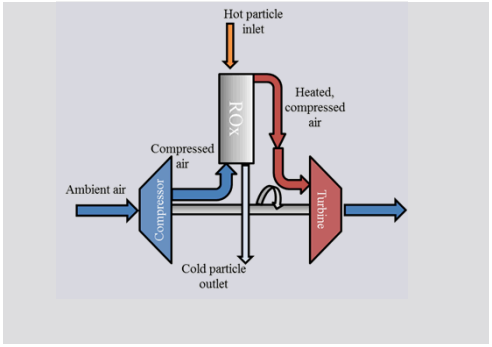
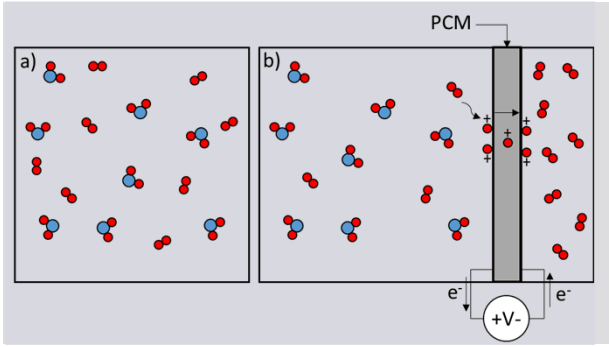
SAND2016-11735C

*Exceptional service in the national interest*

Sandia National Laboratories






# Non-stoichiometric perovskite oxides as high-temperature storage media and their application to concentrating solar power generation and hydrogen production

**Sean M. Babiniec**[1], Andrea Ambrosini[1], Ellen B. Stechel[2], Peter G. Loutzenhiser[3], James E. Miller[1]

1. Sandia National Laboratories
2. Arizona State University
3. Georgia Technical University

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2015-8428 C

Sandia National Laboratories

# MOTIVATION FOR HIGH-TEMPERATURE PEROVSKITE STORAGE MEDIA

# Where can thermal/thermochemical storage make an impact in solar-thermal technologies?

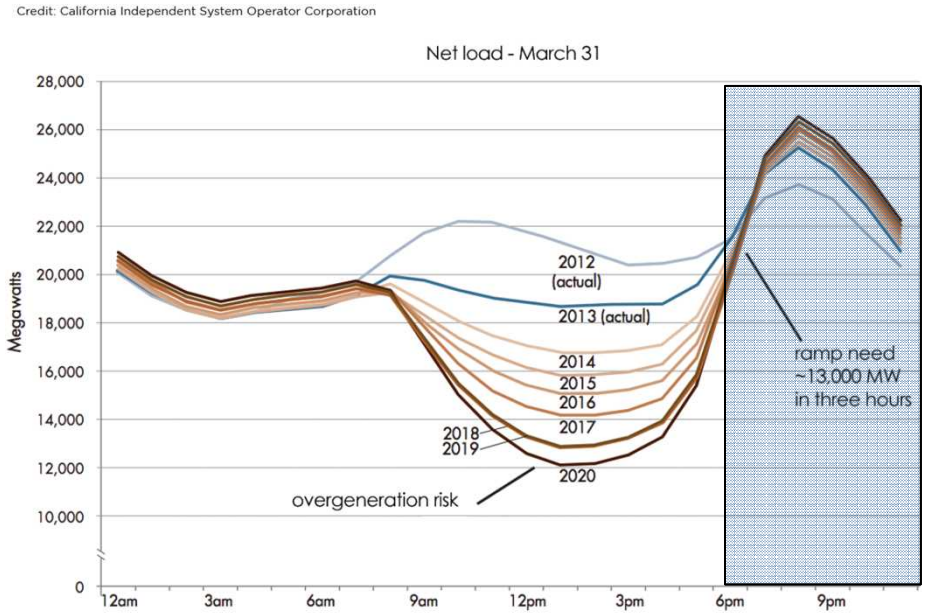
- Concentrating solar power (CSP) electricity generation
  - Intermittent, diurnal nature requires energy storage to:
    1. Increase the duration of electricity generation (increased capacity factor)
    2. Shift the period of operation to peak demand

**BSM6** high temperatures, large chemicalmpotential gradients.

Babiniec, Sean Michael, 11/13/2016

# Where can thermal/thermochemical storage make an impact in solar-thermal technologies?

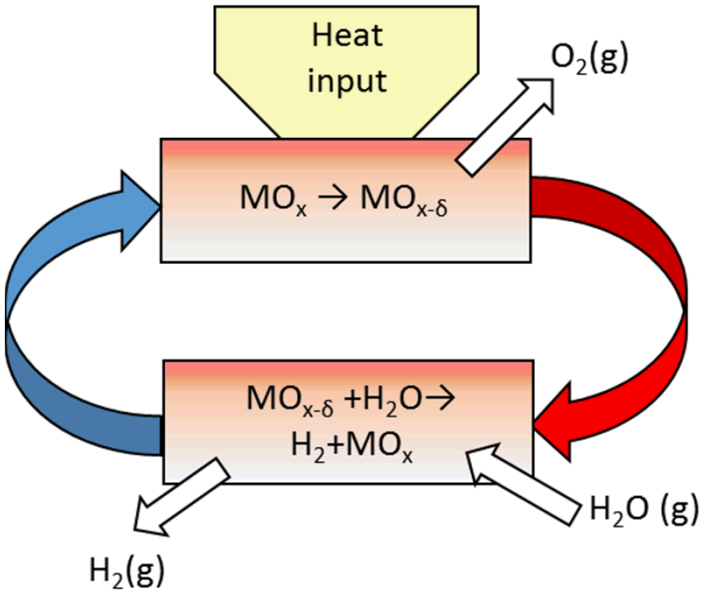
- Concentrating solar power electricity generation
  - Intermittent, diurnal nature requires energy storage to:
    1. Increase the duration of electricity generation (increased capacity factor)
    2. Shift the period of operation to peak demand
       - Requirement to increase solar electricity integration into the U.S. grid!!!




Interested in total storage (sensible + chemical enthalpy)!

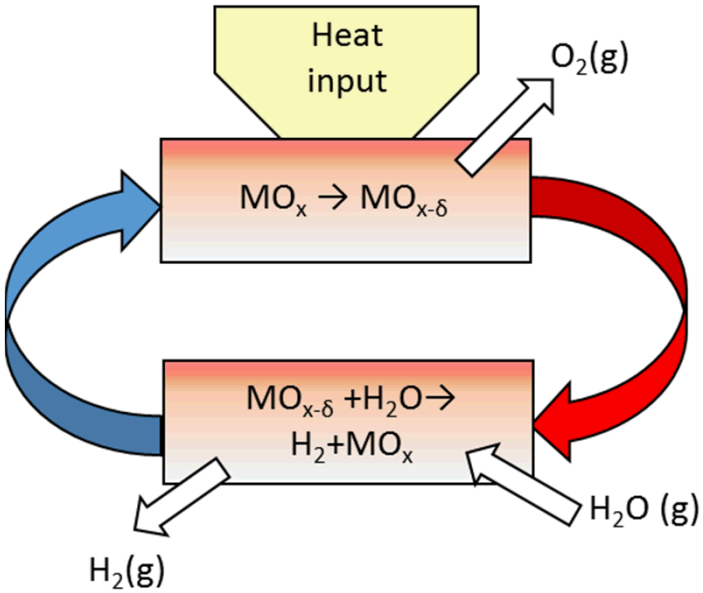
# Where can thermal/thermochemical storage make an impact in solar-thermal technologies?

- Energy carriers in thermochemical cycles
  - Especially of interest for water-splitting hydrogen production



# Motivation for the development of reactive particles

- Energy carriers in thermochemical cycles
  - Especially of interest for water-splitting hydrogen production




Only interested in chemical enthalpy!

Both applications share several desirable materials properties:

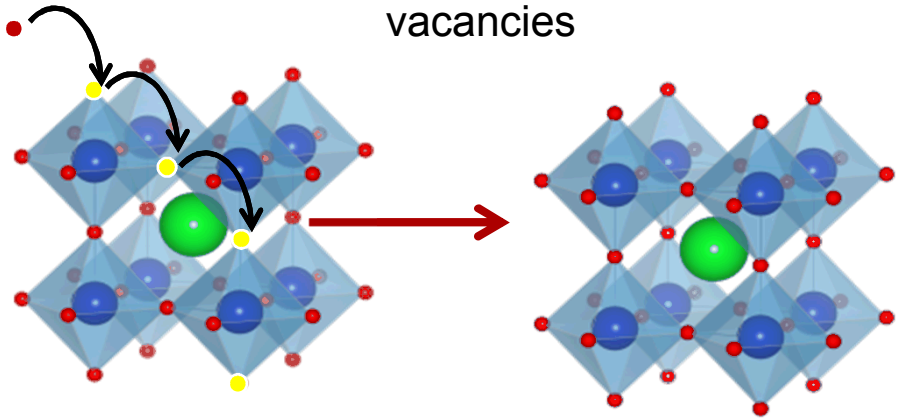
- Tunability of material thermodynamics
- Rapid kinetics
- Adequate oxygen capacity
- Low cost

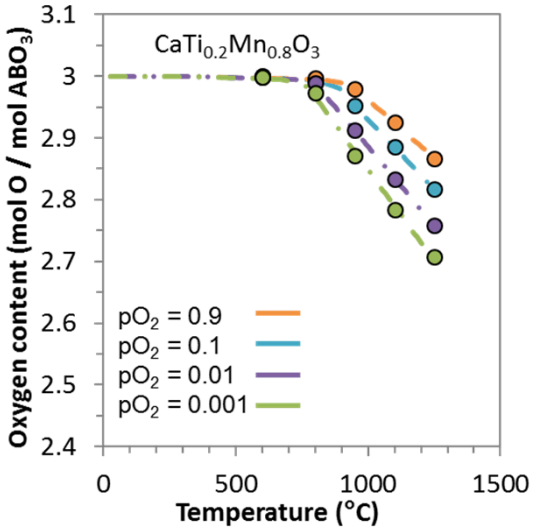
# Perovskites can posses many of the desired qualities

- Materials tunability
  - Dopants easily introduced into A and B sites
- Fast kinetics
  - "Continuous" reaction characteristics
    - No discrete phase change
  - Mixed ionic/electronic conductivity (MIEC)
    - Ambi-polar transport enables oxygen diffusion from bulk to skin or vice versa
- Adequate oxygen capacity
  - Typically observe maximum oxygen nonstoichiometric of δ = 0.2 to 0.5
- Cost can be reduced by using earth-abundant cations
  - Ca, Al, K, etc.

Oxidation: BSM1 $ABO_{3-\delta} + \delta/2O_2(g) \leftrightarrow ABO_3 + \Delta H$

$\delta$ is the site concentration of oxygen vacancies



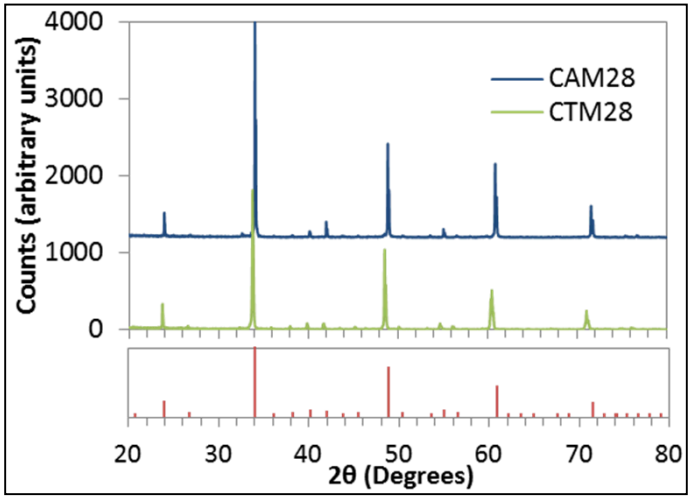


## Slide 7

**BSM1** Change this to reduction instead of oxidation. Easier to discuss what ABO3 is and how vacancies are indroduced.
Babiniec, Sean Michael, 11/13/2016

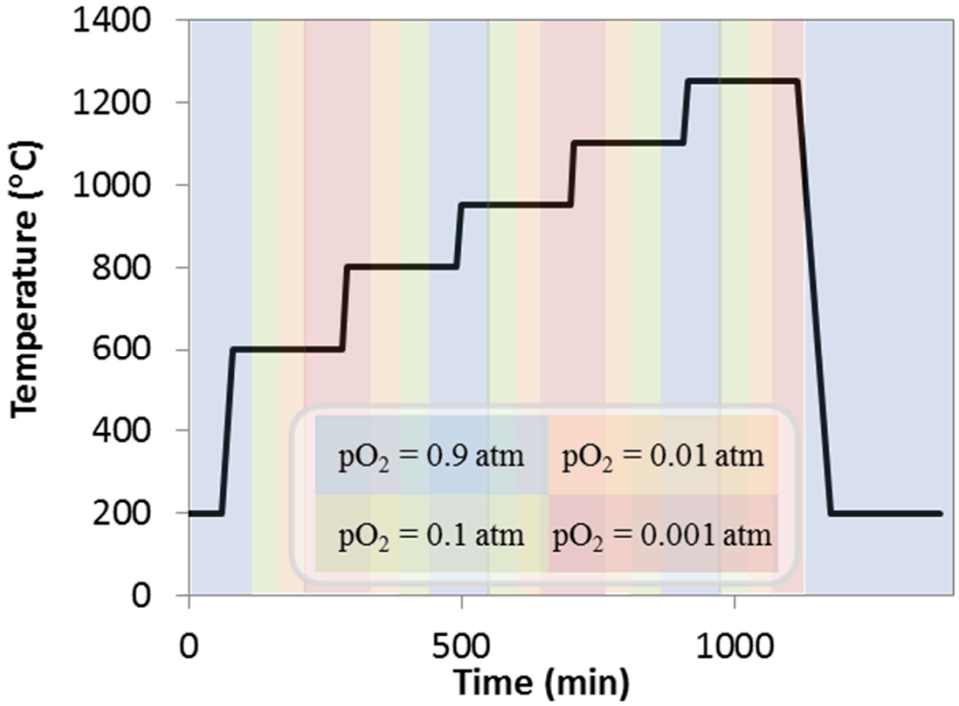
# Synthesis and phase characterization

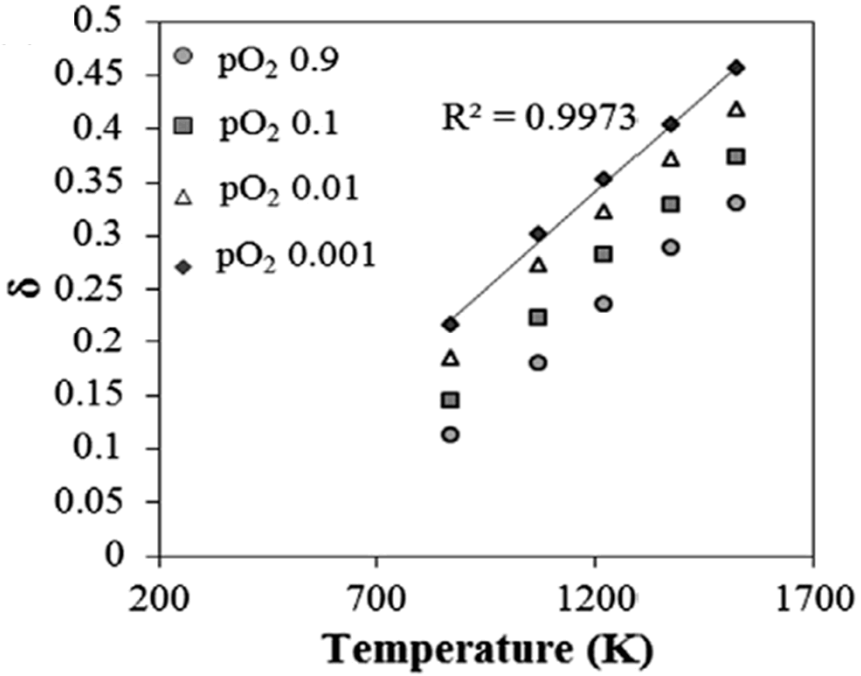
- Materials synthesized using an aqueous (Pechini) method
- X-ray diffraction on pellets was used for phase identification
- Compositions nomenclature: $A_aA'_{a'}B_bB'_{b'}O_{3-\delta}$
  - Ex: $La_{0.7}Sr_{0.3}Co_{0.3}Fe_{0.7}O_{3-\delta}$: LSCM7337
- 2 main materials families investigated
  - $La_aSr_bCo_cM_dO_{3-\delta}$ : M = Fe, Mn
    - ~ 50 compositions
    - Ex: LSCF3773, LSCM2891
  - $Ca_aA'_bX_cMn_dO3-\delta$ : A' = Sr, La, Y; X = Al, Ti, Co
    - ~25 compositions
    - Ex: $CaAl_{0.2}Mn_{0.8}0_{3-\delta}$: CAM28



# Thermogravimetric analysis used to characterize oxygen vacancy concentration

- Equilibrium concentration observed by measuring a change in mass as a function of oxygen partial pressure and temperature
  - 5 temperatures, 4 oxygen partial pressures at each temperature
  - Material allowed ample time to reach a steady-state mass



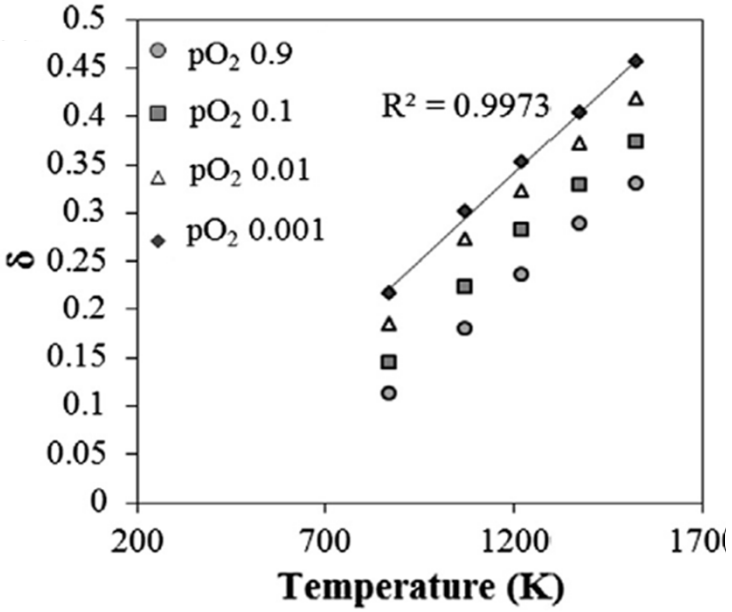



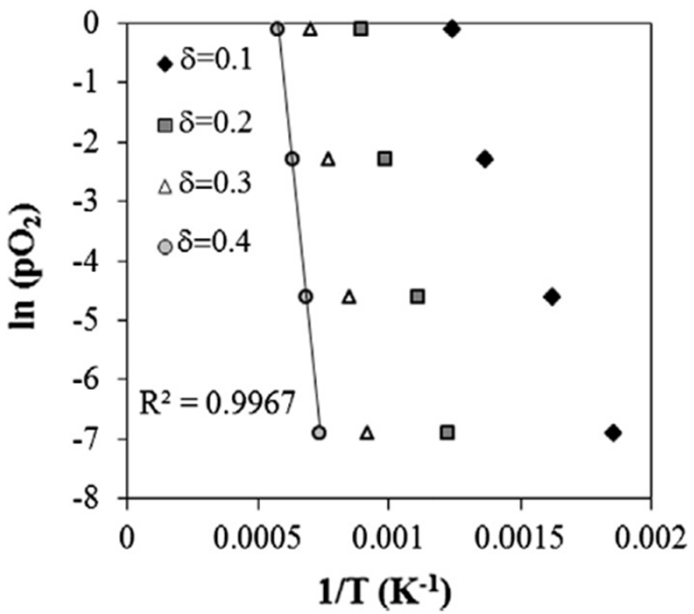
## Slide 9

**BSM2** Is the data too much to explain? Maybe combine this slide with the next without the table?
Babiniec, Sean Michael, 11/13/2016

# Equilibrium TGA data used to estimate thermodynamic parameters

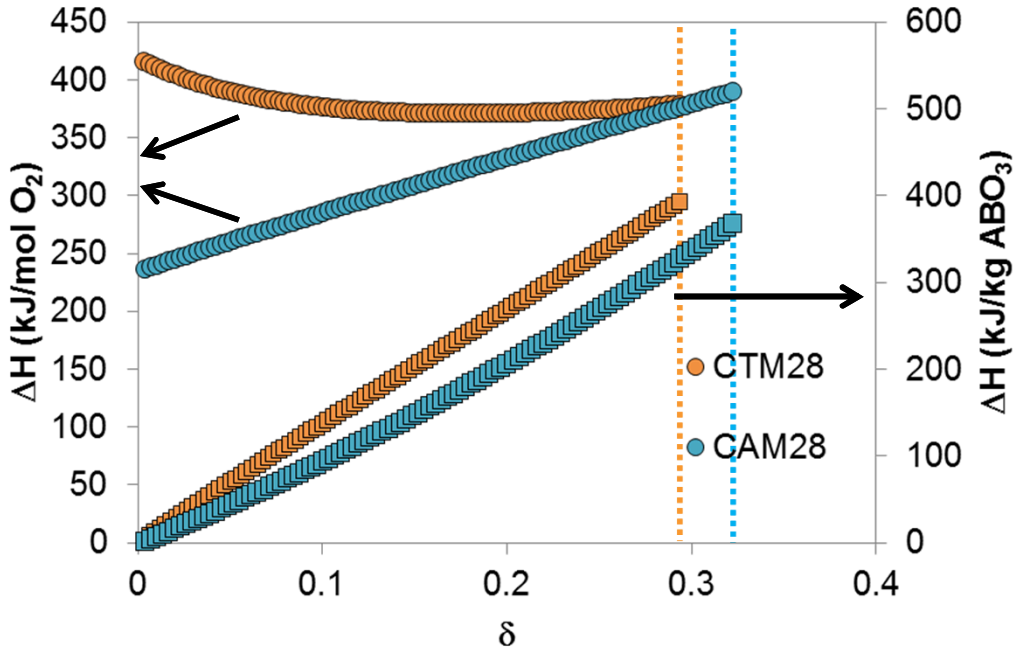
- Thermodynamic parameters extracted by van't Hoff approach:
  - $K = \frac{[ABO_{3-\delta}]^{1/\delta}\ pO_2{}^{1/2}}{[ABO_3]^{1/\delta}}$, assume ratio of solid activities is ≈unity
  - $\ln(pO_2) = 2\frac{-\Delta G_{rxn}}{RT} = 2\left(\frac{1}{T}\cdot\frac{-\Delta H_{rxn}}{R} + \frac{\Delta S_{rxn}}{R}\right)$
    - Enthalpy determined by slope, entropy by intercept for each value of δ



# Enthalpies from van't Hoff are given for a specific oxygen non-stoichiometry

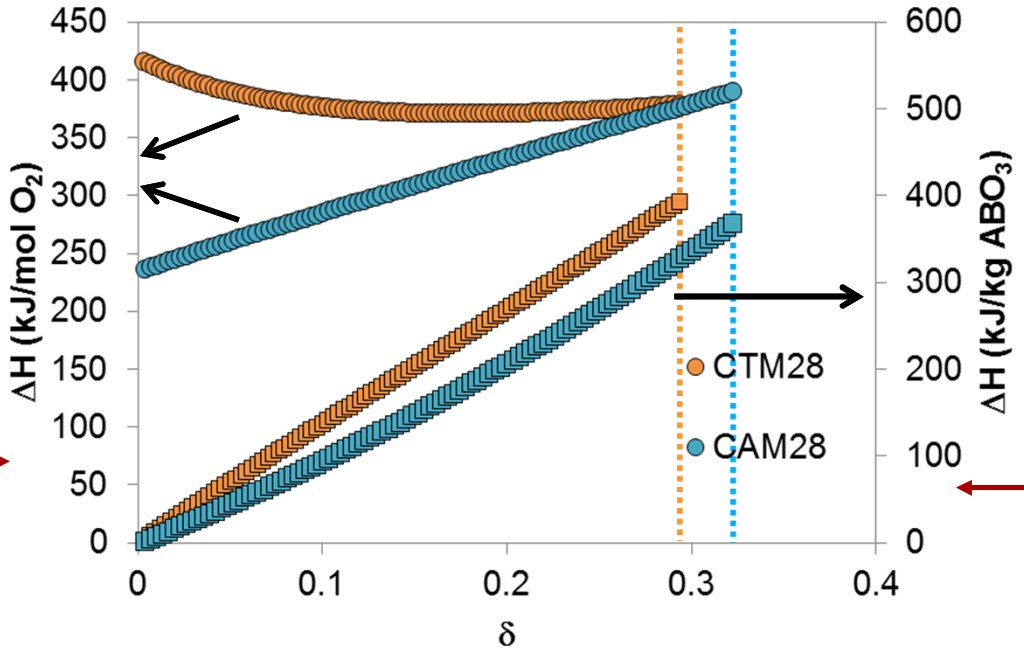
- Partial molar: describes energy to remove a mole of $O_2$ at a specific δ
- Enthalpies must be integrated over δ to describe continuous reaction by series of discrete reactions



# Enthalpies from van't Hoff are given for a specific oxygen non-stoichiometry

- Partial molar: describes energy to remove a mole of $O_2$ at a specific δ
- Total energy storage: enthalpies must be integrated over δ to describe continuous reaction by series of discrete reactions

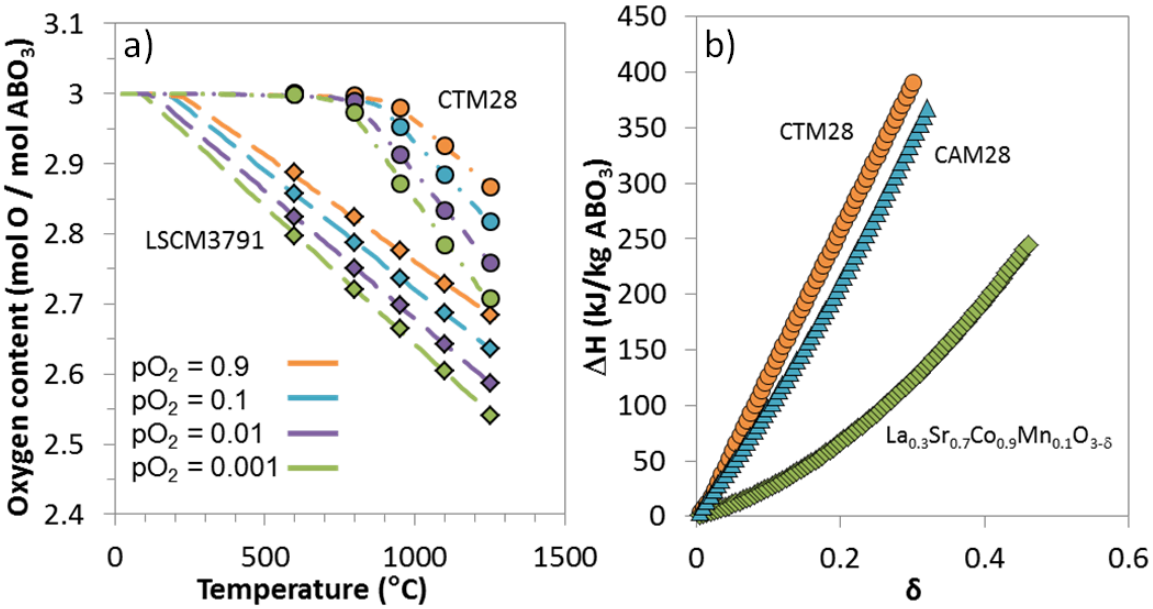
Important for chemical manipulations (steam-splitting, etc.)




Important for energy storage

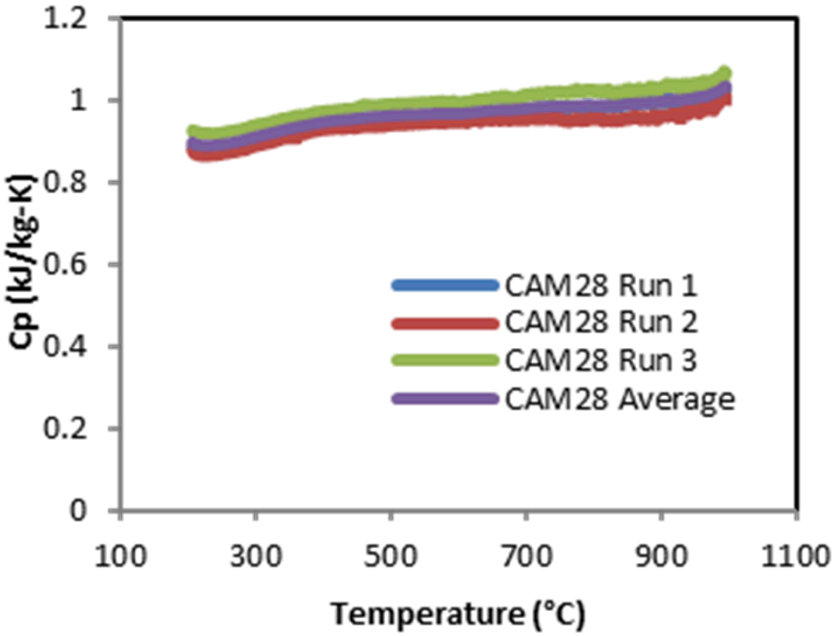
# thermodynamic limits which inhibit further increases in reaction enthalpy

- $ABO_3 + \Delta H \leftrightarrow ABO_{3-\delta} + \delta/2\ O_{2(g)}$
  - Gas species dominates entropy term (largest # degrees-of-freedom)
    - Entropy change between two solid compositions likely << entropy in diatomic gas particle
- At equilibrium (onset of reduction) $\Delta G_{red} = 0 = \Delta H_{red} - T\ \Delta S_{red}$,
  - For two compositions at equilibrium with nearly equal entropy, a change in reduction enthalpy necessitates a change in reduction temperature



# Total storage capacity

- Heat capacity similar for materials of the same family
  - Same number of atoms in lattice, similar molecular weight




$\Delta H_{sens} = \int_{T1}^{T2} C_p(T)dT$ ~ 875 kJ/kg between 200 and 1250 °C

$\Delta H_{total} = \Delta H_{sens} + \Delta H_{reaction} \sim 900 + 350 = 1350\ kJ/kg$

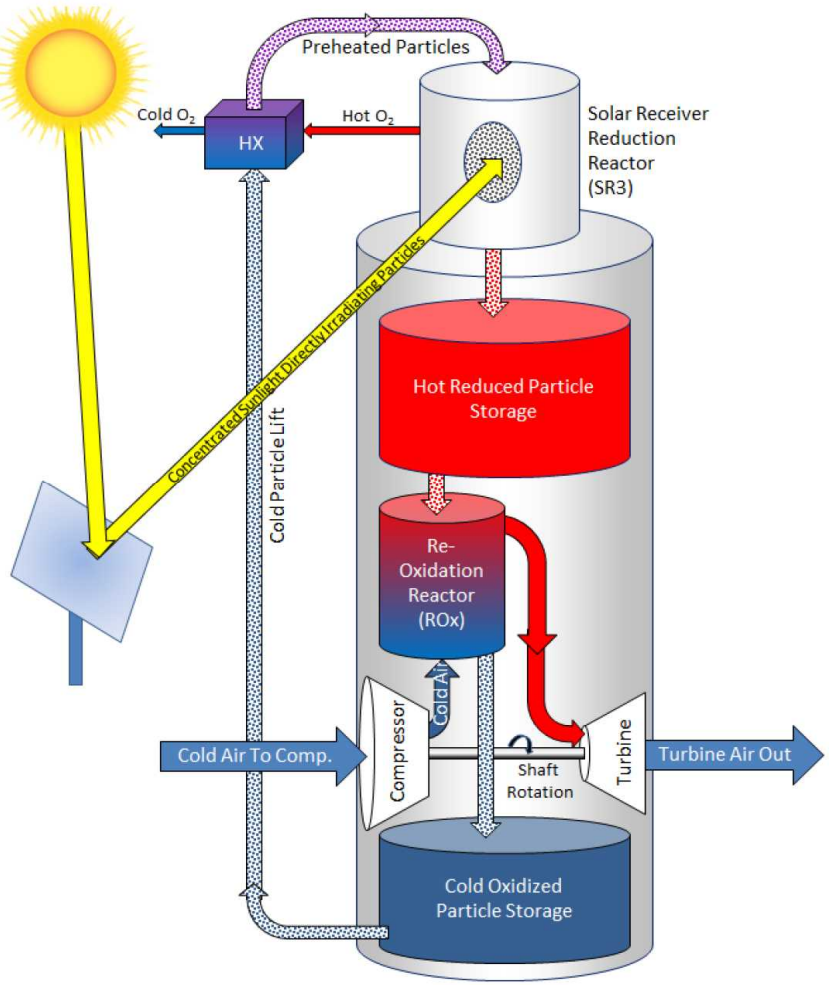
Molten salt energy density ~ 400-450 kJ/kg

Sandia National Laboratories

# APPLICATIONS FOR HIGH-TEMPERATURE PEROVSKITE STORAGE MEDIA

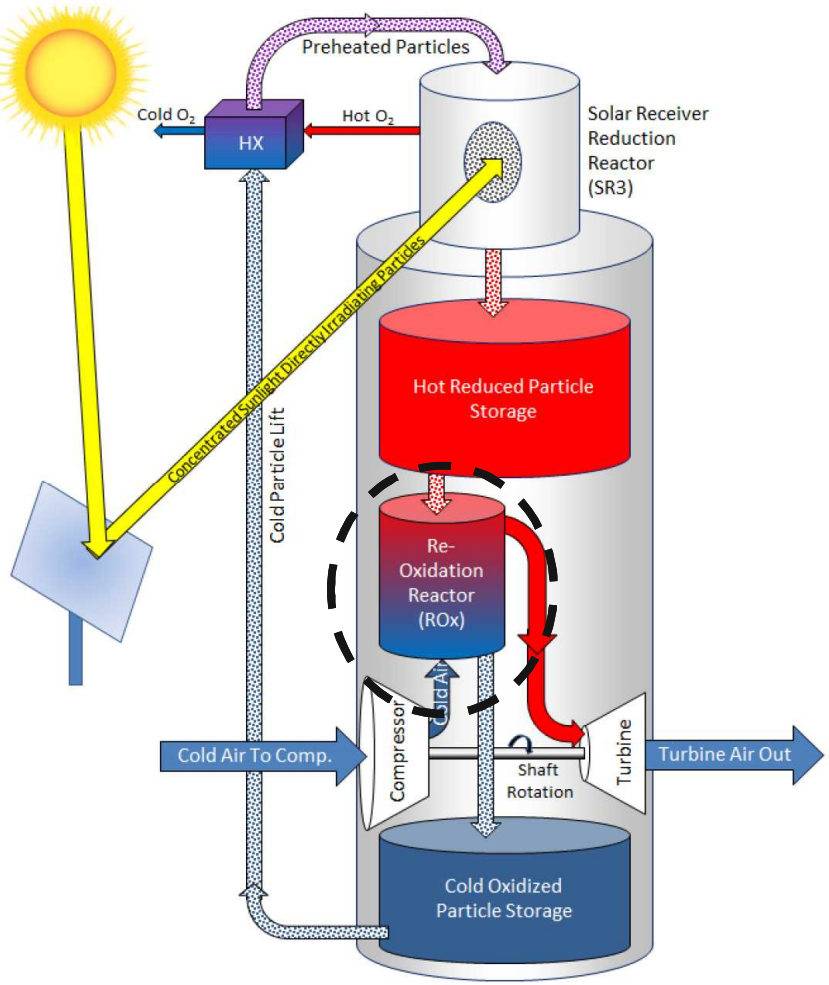
# Concentrating Solar Power

- Closed moving particle, open air-Brayton cycle proposed
  - Direct irradiation of thermal storage media
    - Mitigate re-radiation as compared to high-temperature indirect receivers
  - Requires no storage of gasses
  - Requires no condensation of working fluid
  - Working fluid assumes multiple roles
    - Heat transfer fluid and reactant stream
  - Compression of air favorably shifts re-oxidation reaction thermodynamics
  - Takes advantage of high storage temperatures
    - Air-Brayton turbines operating between 900-1200 °C

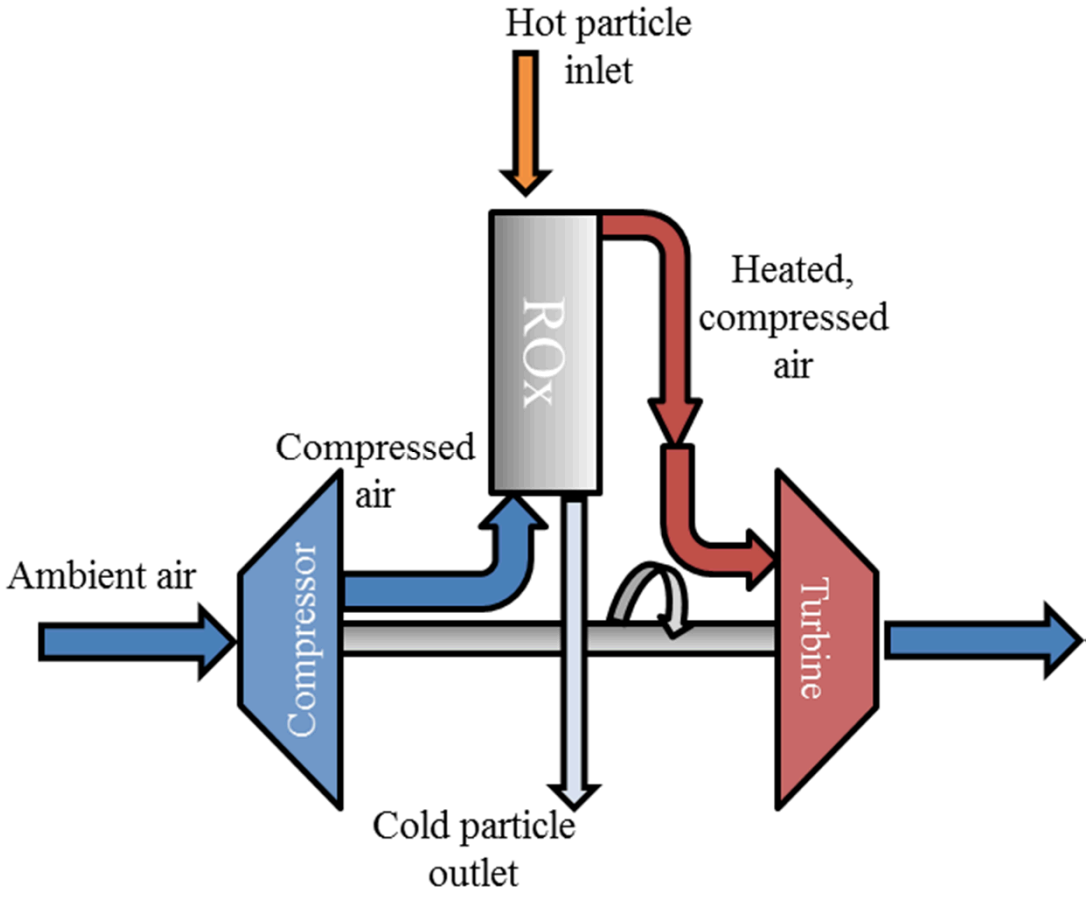


# Concentrating Solar Power

- Closed moving particle, open air-Brayton cycle proposed
  - Direct irradiation of thermal storage media
    - Mitigate re-radiation as compared to high-temperature indirect receivers
  - Requires no storage of gasses
  - Requires no condensation of working fluid
  - Working fluid assumes multiple roles
    - Heat transfer fluid and reactant stream
  - Compression of air favorably shifts re-oxidation reaction thermodynamics
  - Takes advantage of high storage temperatures
    - Air-Brayton turbines operating between 900-1200 °C



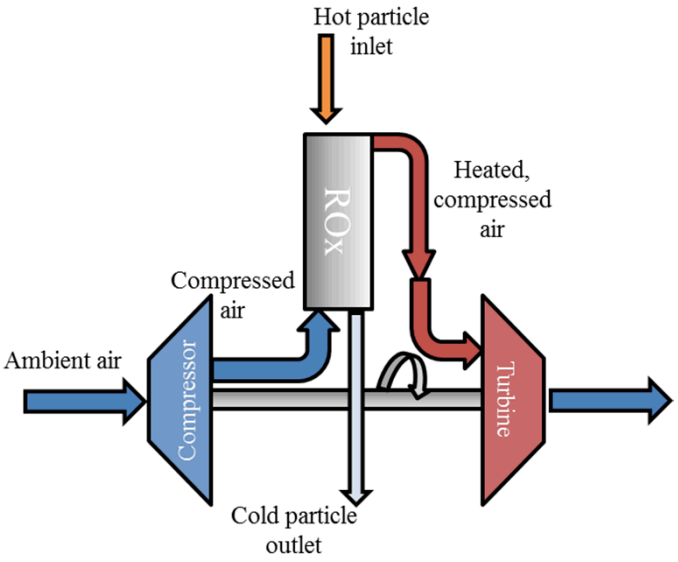
# Recovery of heat: Powering the Air-Brayton turbine

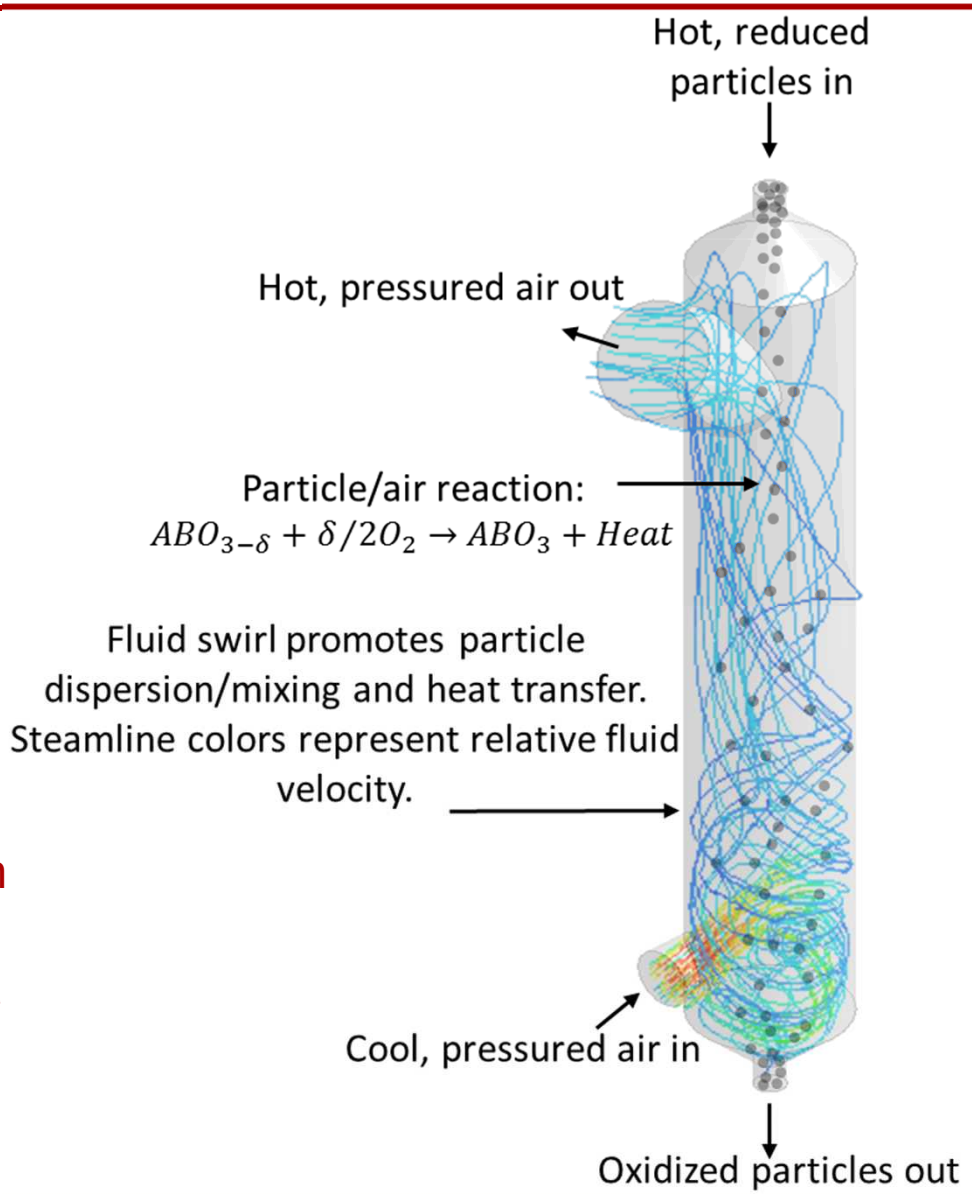



Counter-flow air/particle operation desired

- Allows maximum reactor outlet temperature
- Must mitigate entrainment

# Recovery of heat: Powering the Air-Brayton turbine

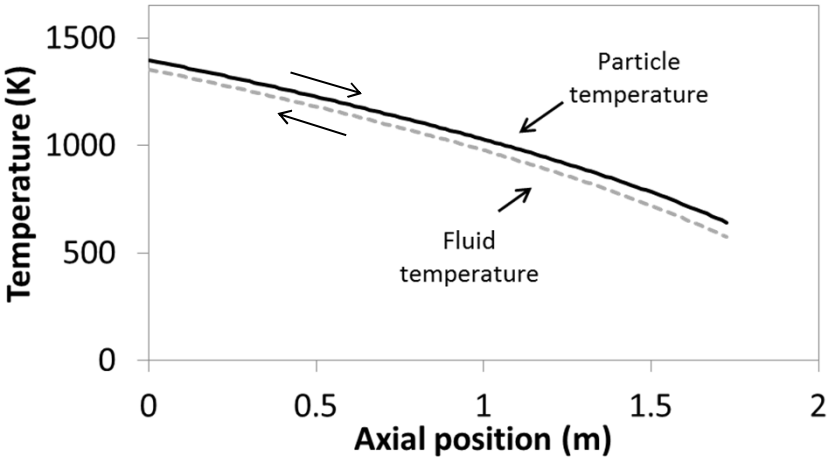


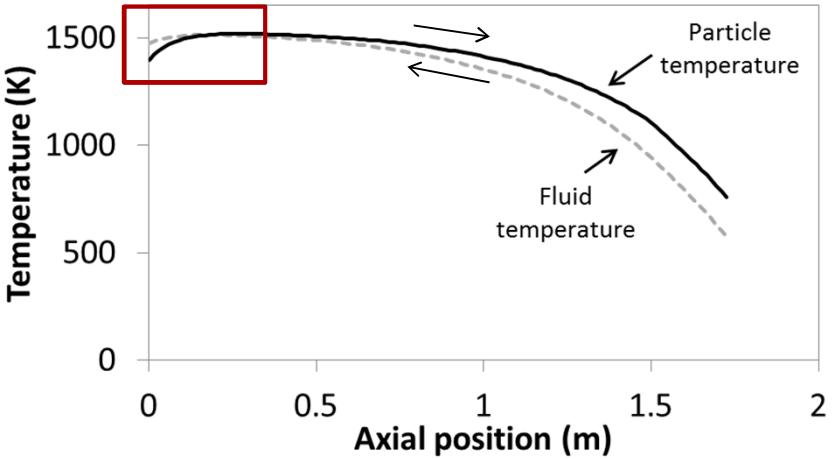



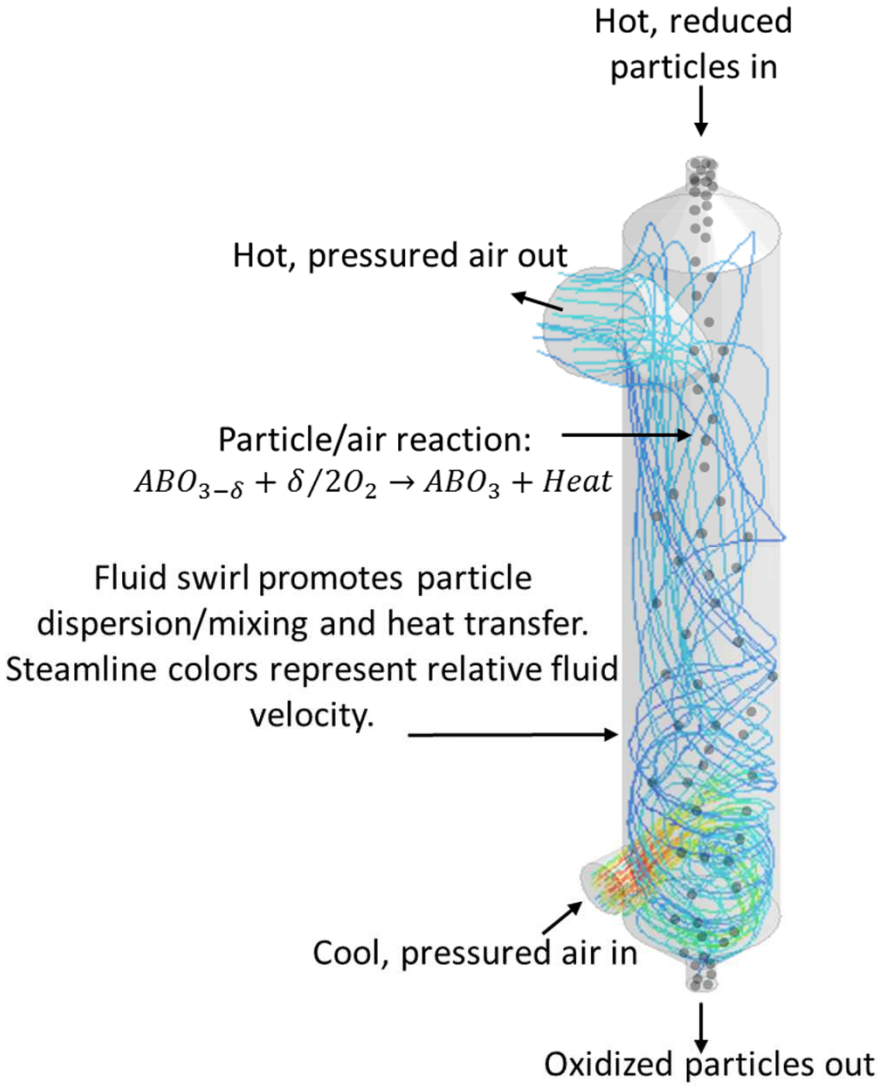

Counter-flow air/particle operation desired

- Allows maximum reactor outlet temperature
- Must mitigate entrainment

# Recovery of heat: Powering the Air-Brayton turbine



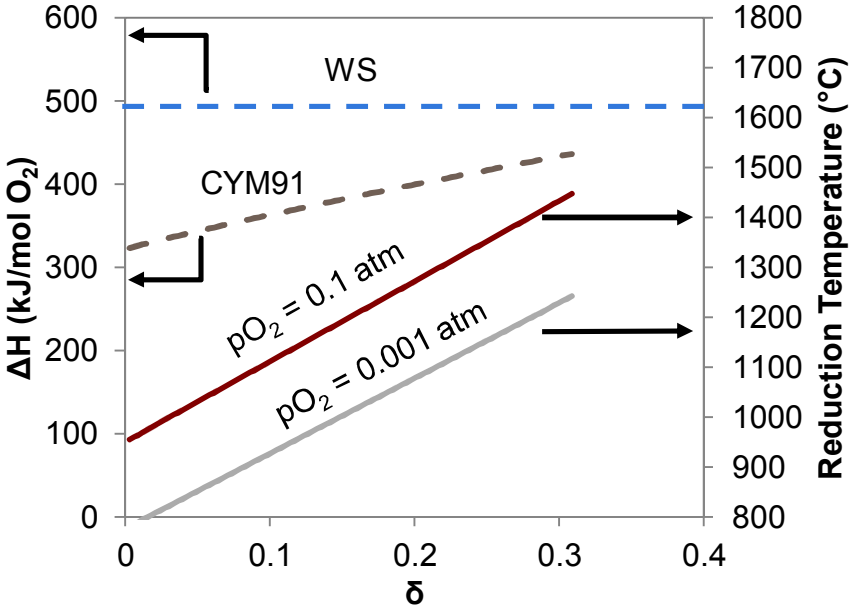



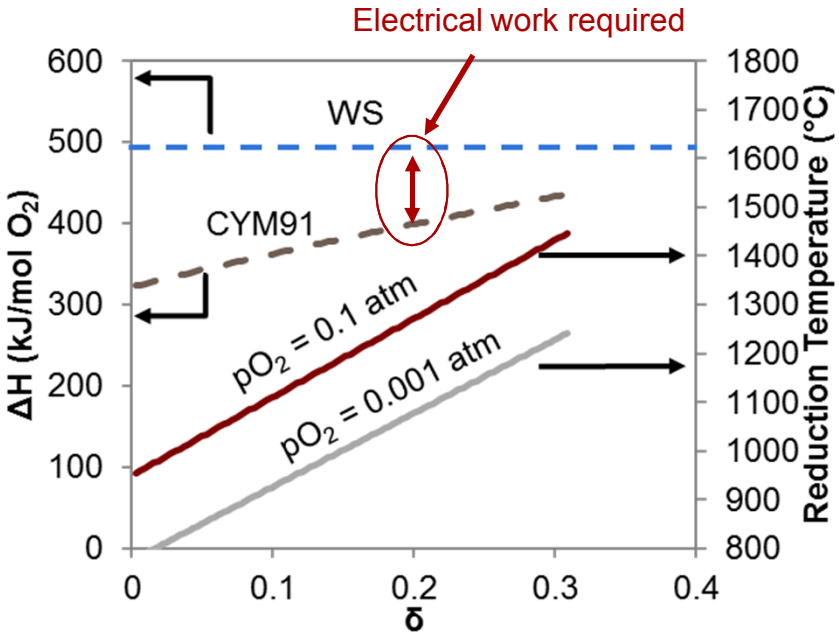

Babiniec *et al*., ASME Power & Energy 2016, Paper No. ES2016-59646, pp. V001T04A021; doi:10.1115/ES2016-59646

# Hydrogen production

- Main goal: Alleviate extreme environments required in typical ceria-based cycles
- Want to harness advantages of our materials
  - Fast kinetics
  - High oxygen vacancy capacities relative to ceria
  - Decreased reduction temperature
- Typically, perovskite MIECs cannot maintain these advantages while enabling spontaneous water splitting
  - High ΔH required pushes reduction temperature higher



# Enabling perovskite MIEC water splitting – coupled thermochemical/electrochemical hybrid cycle



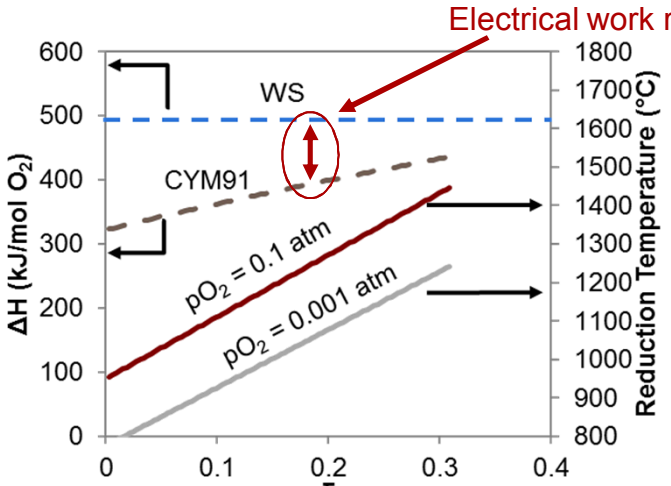

Picture and reference from Palumbo patent

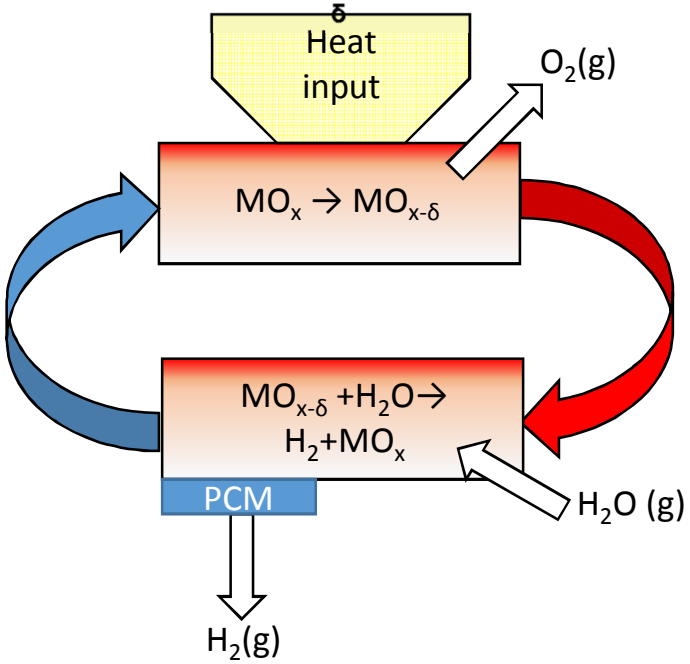
$$H_2O \leftrightarrow 1/2\,O_2 + \delta H_2 \quad \Delta G = 2?? \text{ kJ/mol-O}$$

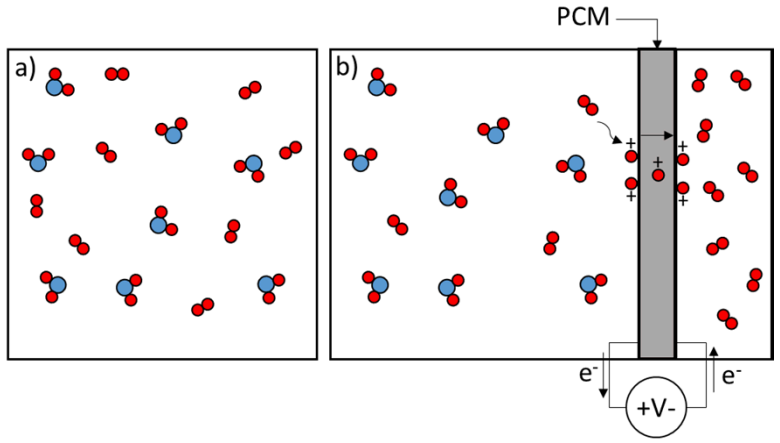
Direct electrolysis utilizing the chemical energy stored in the reduced solid-oxide:

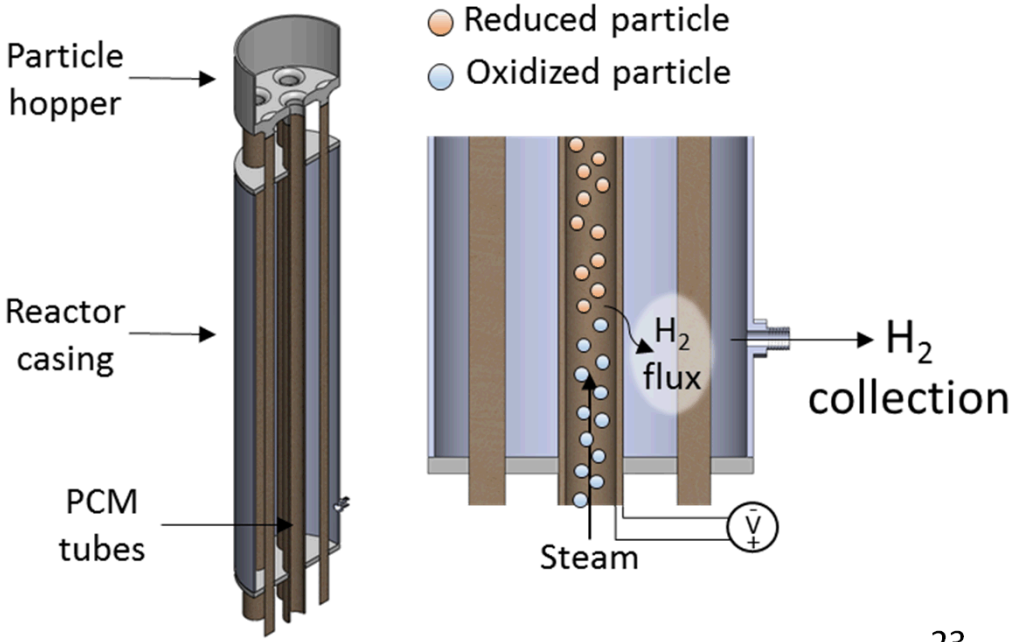
$$MO_{x-\delta} + H_2O \leftrightarrow MO_x + \delta H_2 \quad \Delta G \sim 10\text{-}50 \text{ kJ/mol-O}$$

# Enabling perovskite MIEC water splitting – coupled thermochemical/electrochemical hybrid cycle



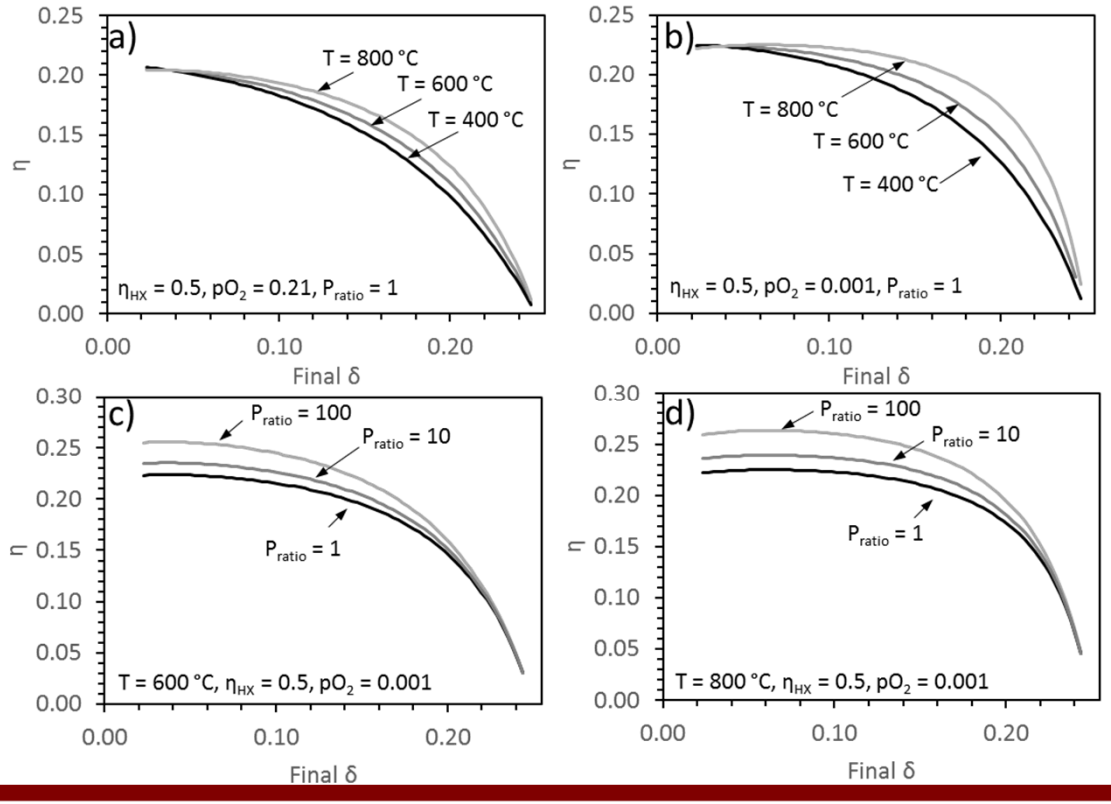



# Promising efficiencies predicted

- Parametric thermodynamic study completed
  - System efficiencies including particle/steam thermodynamics
    - Losses: Thermal-to-electric, heat recovery losses, pumping losses, boiler losses, etc.
  - Provisional patent filed, manuscript prepared for submission



# Conclusions

- Identified two applications for high-temperature reactive particles
- Identified perovskites as a viable candidate as material for particles
- Discussed the synthesis and characterization of perovskites
  - Apparent limit in reduction enthalpy
  - Identified CAM28, CYM91 as highest reaction enthalpies with reduction temperature < 1300 °C
- Demonstrated methods for how such particles could be used
  - Counter-flow particle/air reoxidation reactor for Air-Brayton turbines
  - Coupled thermochemical/electrochemical cycle for hydrogen production

# Acknowledgments

This work is supported by the U.S. Department of Energy, SunShot Initiative, under award number DE-FOA-0000805.


Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

# Questions?

# Supplementary material

Sandia National Laboratories

**Table 1. Redox properties of LSCF materials**

| Material | MW (g/mol) | Single Phase? | $T_{red}$ in air (°C) | $\delta_{max}$ (1250 °C, $pO_2$ = 0.001) | Reduction enthalpy @ $\delta_{max}$ (kJ/kg-$ABO_3$) |
|---|---|---|---|---|---|
| $La_{0.2}Sr_{0.8}Co_{0.8}Fe_{0.2}O_3$ (LSCF2882) | 204.2 | Y | 357 | 0.48 | 220 |
| $La_{0.3}Sr_{0.7}Co_{0.9}Fe_{0.1}O_3$ (LSCF3791) | 209.6 | Y | 352 | 0.46 | 264 |
| $La_{0.3}Sr_{0.7}Co_{0.7}Fe_{0.3}O_3$ (LSCF3773) | 209.0 | Y | 348 | 0.45 | 262 |

**Table 2. Redox properties of CXM materials**

| Material | MW (g/mol) | Single Phase? | $T_{red}$ in air (°C) | $\delta_{max}$ (1250 °C, $pO_2$ = 0.001) | Reduction enthalpy @ $\delta_{max}$ (kJ/kg-$ABO_3$) |
|---|---|---|---|---|---|
| $Ca_{0.9}Y_{0.1}MnO_3$ (CYM910) | 147.9 | Y | 966 | 0.31 | 399 |
| $CaAl_{0.2}Mn_{0.8}O_{2.9}$ (CAM28) | 135.8 | Y | 759 | 0.29 | 280 |
| $Ca_{0.9}Y_{0.1}Al_{0.2}Mn_{0.8}O_3$ (CYAM9128) | 142.3 | Y | 800 | 0.31 | 339 |
| $CaZr_{0.2}Mn_{0.8}O_3$ (CZrM28) | 150.3 | N | 867 | 0.27 | 297 |

# Design challenges and limiting factors

- Thermo/hydrodynamic challenges
  - Counterflow operation limits gas velocity
    - Must avoid particle entrainment
  - Spatial particle homogeneity important for heat transfer
  - Pressure drop must be limited
    - Fluid delivered to turbine at T and P near design point

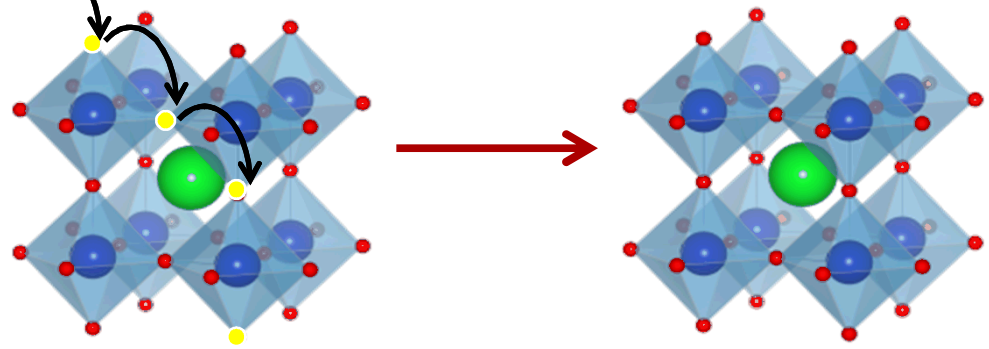
Tightly-coupled thermal-fluid problem:
Need to control particle motion and residence time in a way that minimizes pressure drop and maximizes heat transfer to deliver turbine gas at ≥ 1200 °C.

# Reaction must be understood before modeling can be considered

- Cyclic reduction/oxidation of perovskite oxides ($ABO_3$)
  - NOT a discrete phase change! Continuous across all thermodynamic states!
    - Material does not decompose (i.e. A → B + C)
    - Oxygen vacancies are generated as oxygen is evolved from the solid while reducing and vice-versa when oxidizing

Oxidation: $ABO_{3-\delta} + \delta/2O_2(g) \leftrightarrow ABO_3 + \Delta H$

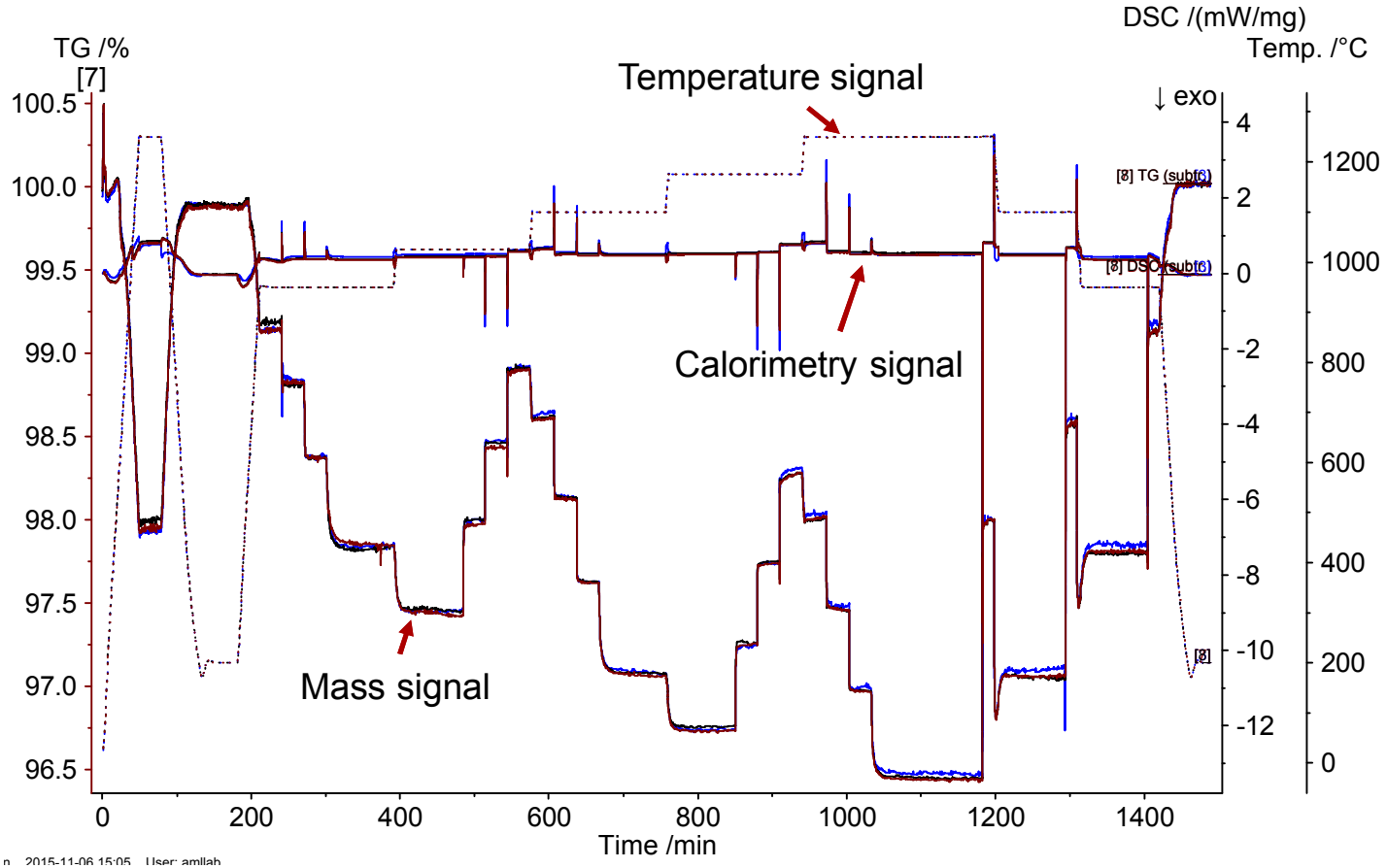
$\delta$ is the site concentration of oxygen vacancies



Reaction is reversible, oxidation is fast (seconds) in strongly-oxidizing conditions

# Characterizing reaction enthalpy Measurements

We can measure oxygen loss and uptake as a function of thermodynamic state (temperature, $pO_2$) using thermogravimetric analysis (TGA)



# Characterizing Reaction Enthalpy
# Thermodynamics

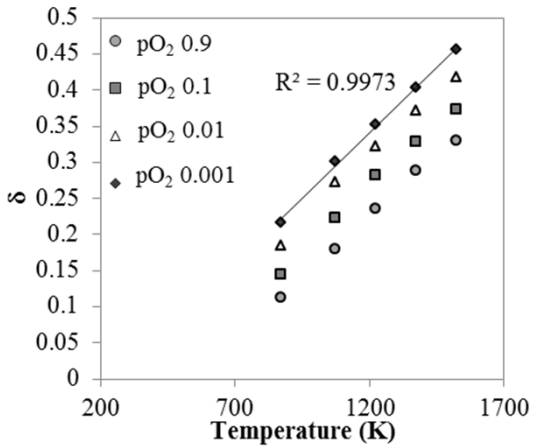
Sandia National Laboratories

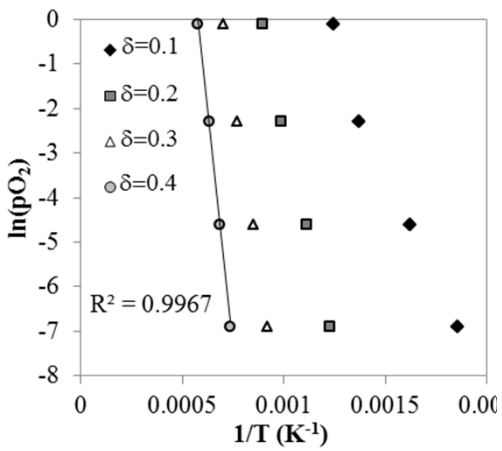
Reduction: $ABO_3 + \Delta H \leftrightarrow ABO_{3-\delta} + \delta/2 O_2(g)$

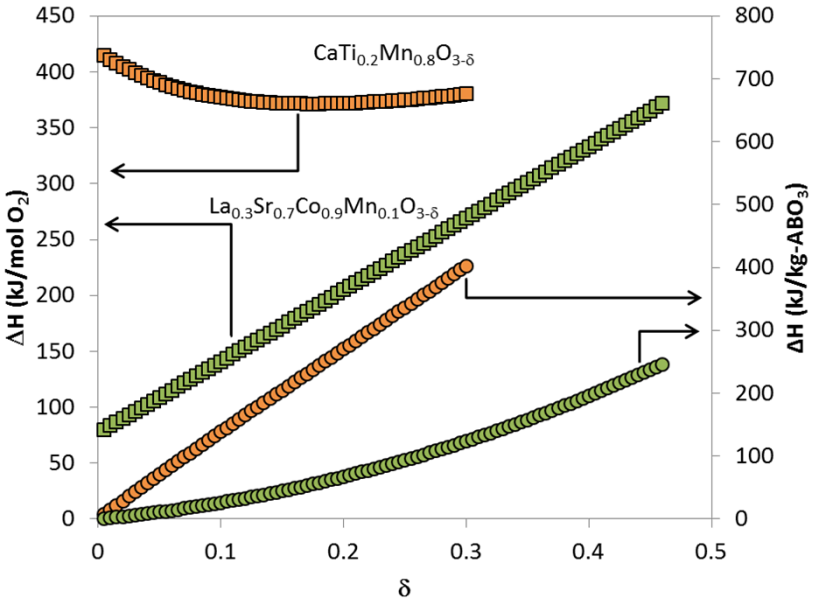
$$K = \frac{ABO_{3-\delta}^{1/\delta} \cdot pO_2^{1/2}}{ABO_3^{1/\delta}} = e^{\frac{-\Delta G_o}{RT}}$$

$$pO_2^{1/2} = e^{\frac{-\Delta G_o}{RT}} \quad \leftarrow \Delta G_o = \Delta H_o - T\Delta S_o$$

$$\ln(pO_2) = 2\left(\frac{-\Delta H_o}{RT} + \frac{\Delta S_o}{R}\right)$$



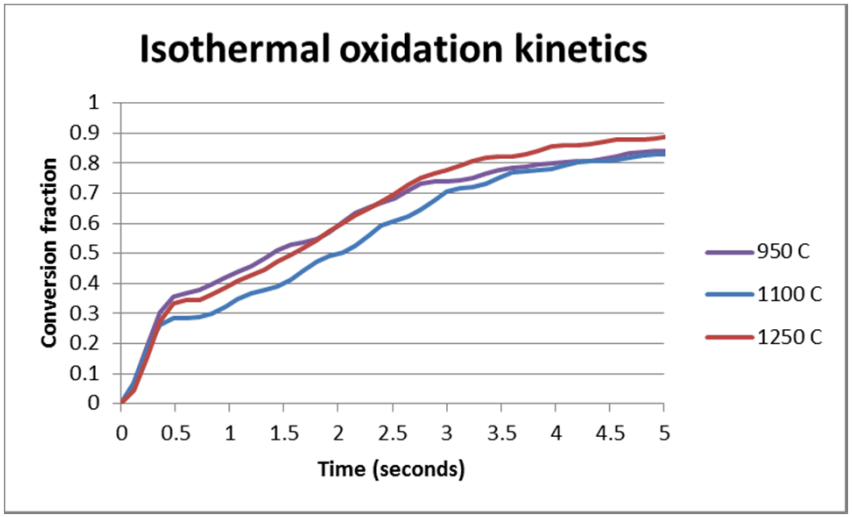



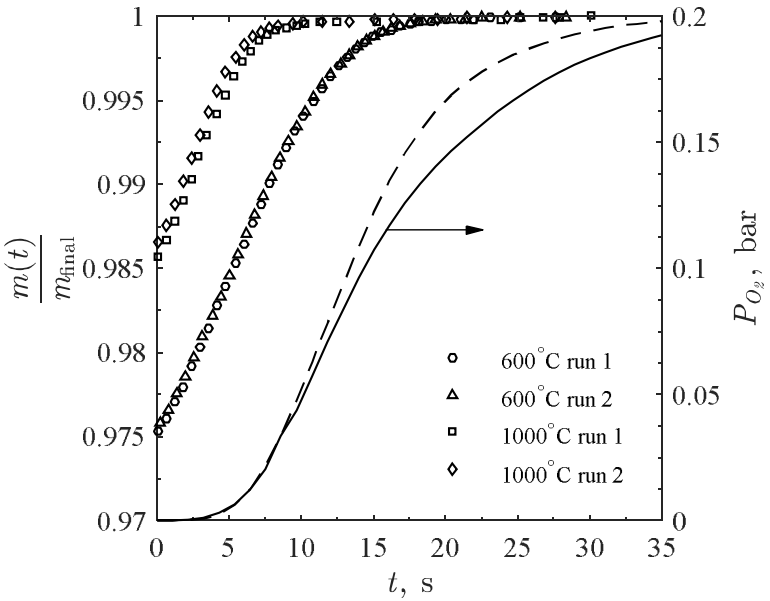



# Measuring kinetics

- Attempted to measure oxidation kinetics in TGA by isothermal gas switch from $pO_2$ = 0.001 to 0.9 atm
  - Reaction rate is significantly faster than gas switchover time
  - Rate is thermodynamically limited a time scale of multiple seconds

$$conversion\ fraction = \frac{\delta - \delta_{in}}{\delta_{eq} - \delta_{in}}$$




Data courtesy Prof. Loutzenhiser, Georgia Tech

# Psuedo-equilibrium kinetic description

- Mechanism for reaction implemented using psuedo-equilbrium kinetics (quasi-thermodynamically limited)

The new value for δ due to oxidation is modeled as:

$$\delta_{new} = \delta_{current} - \beta(\delta_{current} - \delta_{equilibrium})$$

where $\delta_{equilibrium}$ is well described using curve fits from the equilibrium TGA experiments.

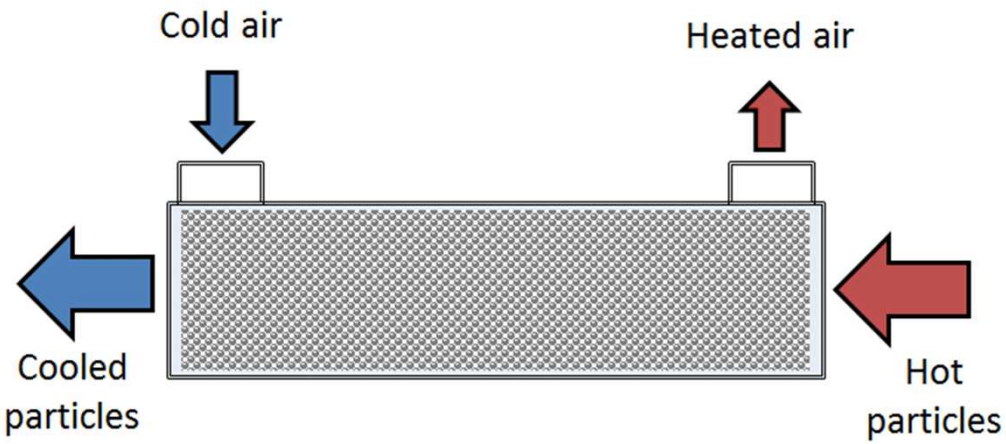
- The change in delta can be used to calculate gaseous oxygen consumption and heat generation.
- The β term describes the "progress" towards equilibrium, and can be used to adjust the reaction rate.

- What we have now
  - ΔH(δ)
  - δ(T, $pO_2$)
  - r(T, $pO_2$, δ)

Used to calculate $\delta_{eq}$

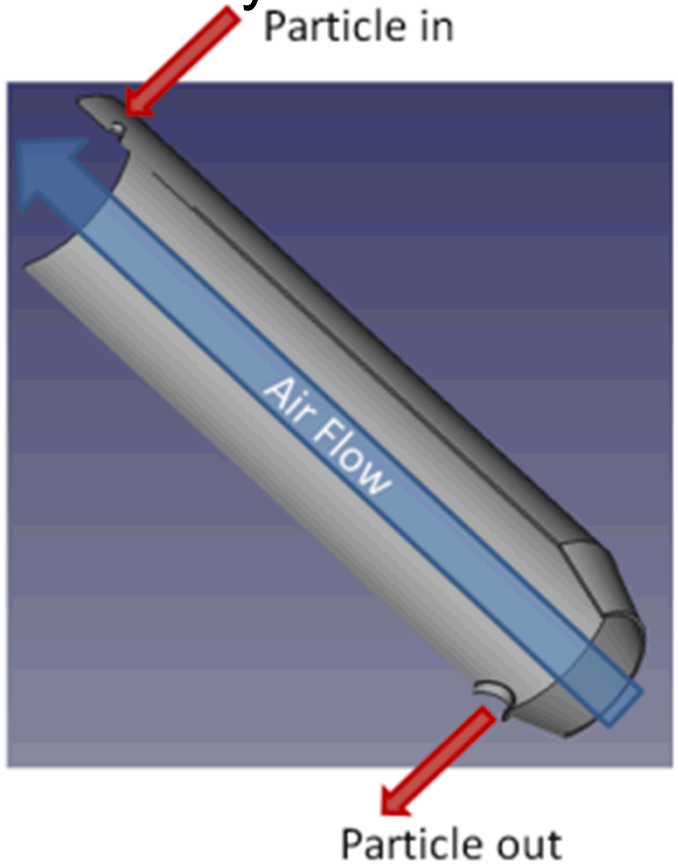
# Two reactor configurations considered

## Moving packed-bed



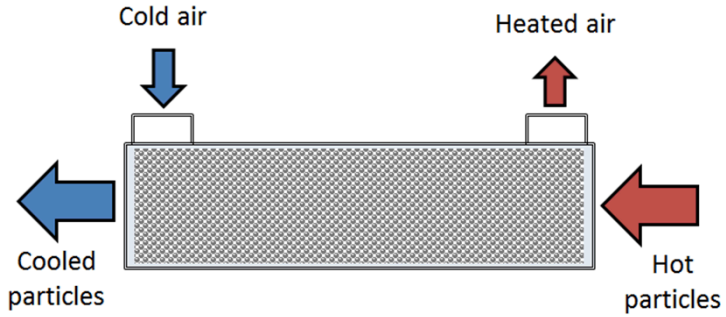

Pros: Particle velocity/residence time control, spatially homogenous
Cons: Pressure drop high if fluid velocity too high, axial heat transfer through solids phase

## Gravity-driven flow



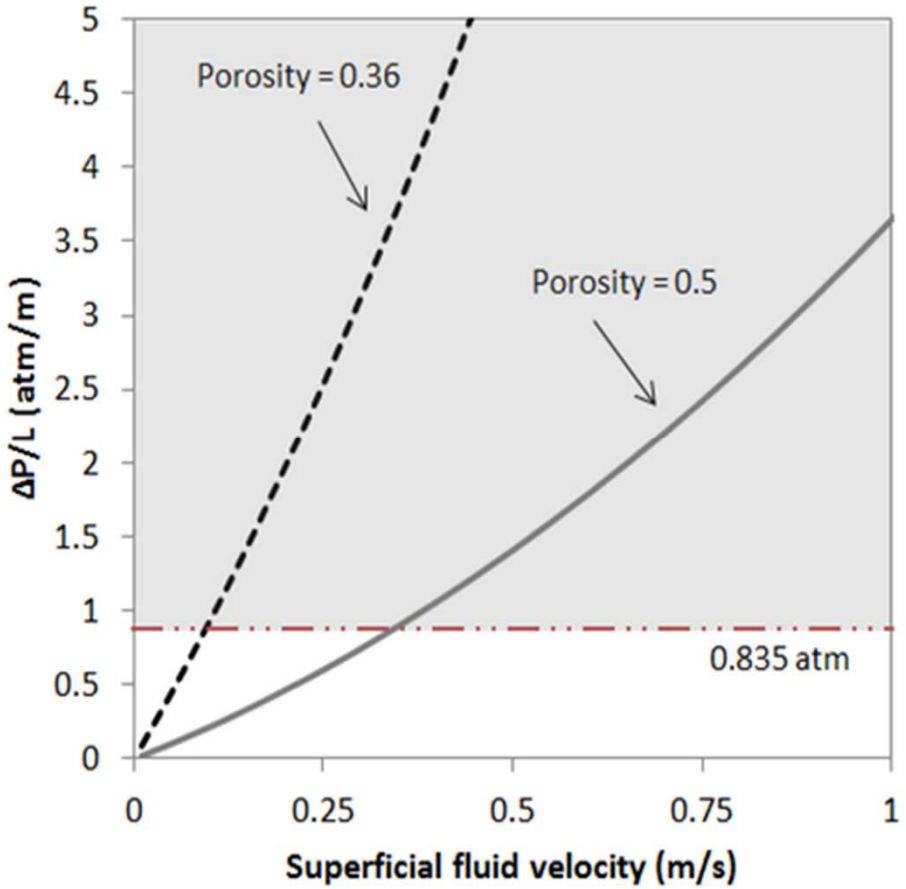

Pros: Can operate in disperse particle regime, low pressure drop
Cons: Limited particle control, entrainment must be avoided

# Moving packed-bed: Pressure drop severely limits allowable fluid velocity

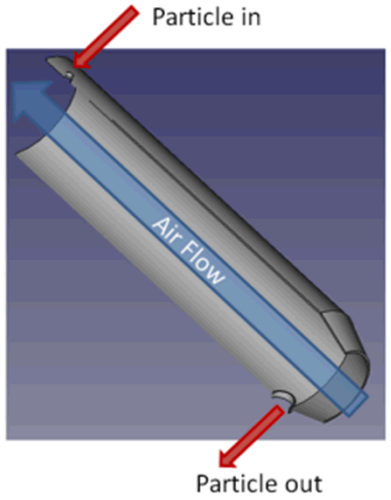



$$\frac{\Delta P}{L} = \frac{150\mu}{{d_p}^2}\frac{(1-\varphi)^2}{\varphi^3}v_s + \frac{1.75\rho}{d_p}\frac{(1-\varphi)}{\varphi^3}{v_s}^2$$

Low fluid velocity leads to large reactor diameter and extremely slow particle flow (~30 min residence time in some cases)

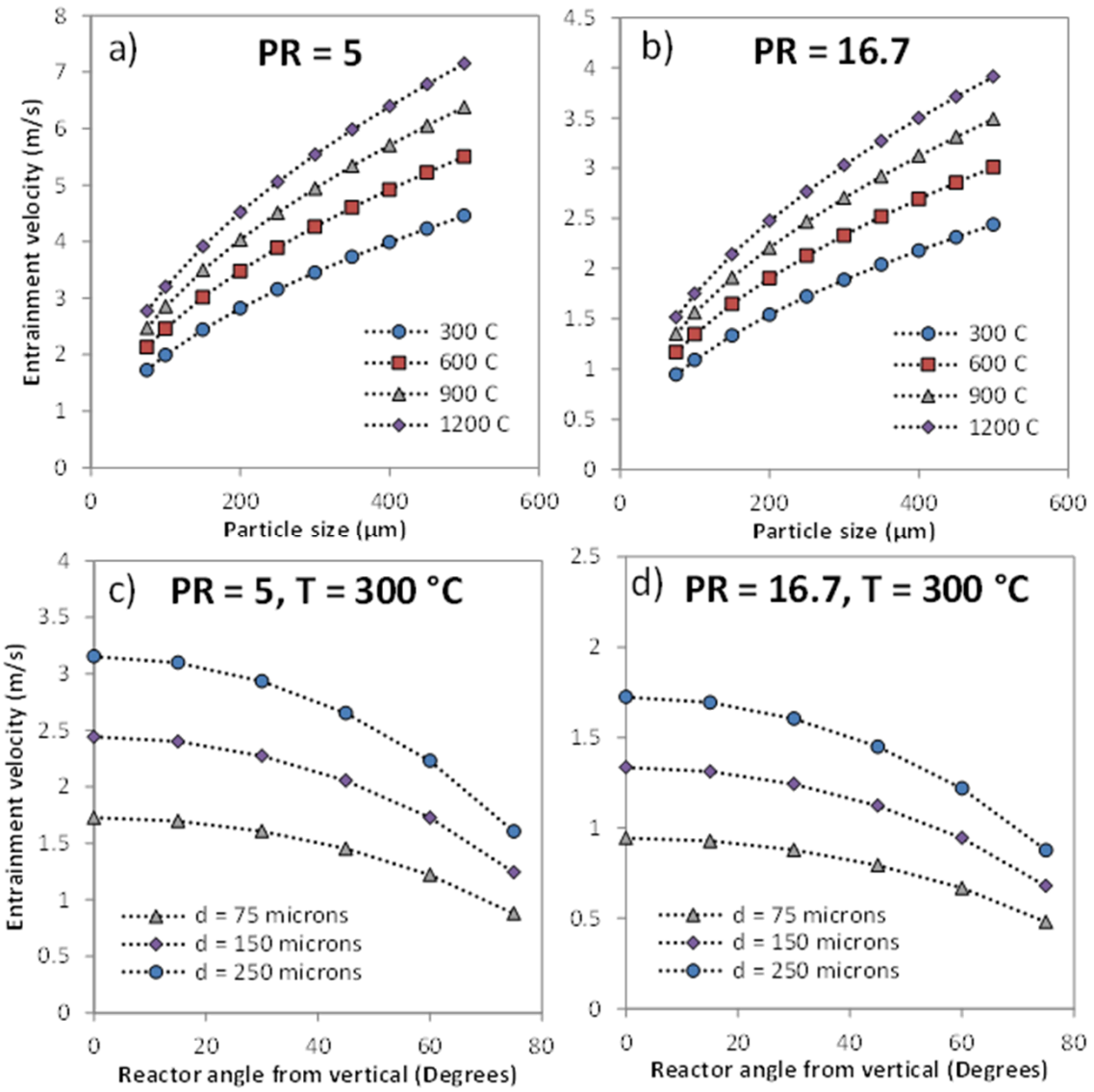


# Gravity-driven flow: Minimal particle control, but higher limiting velocities

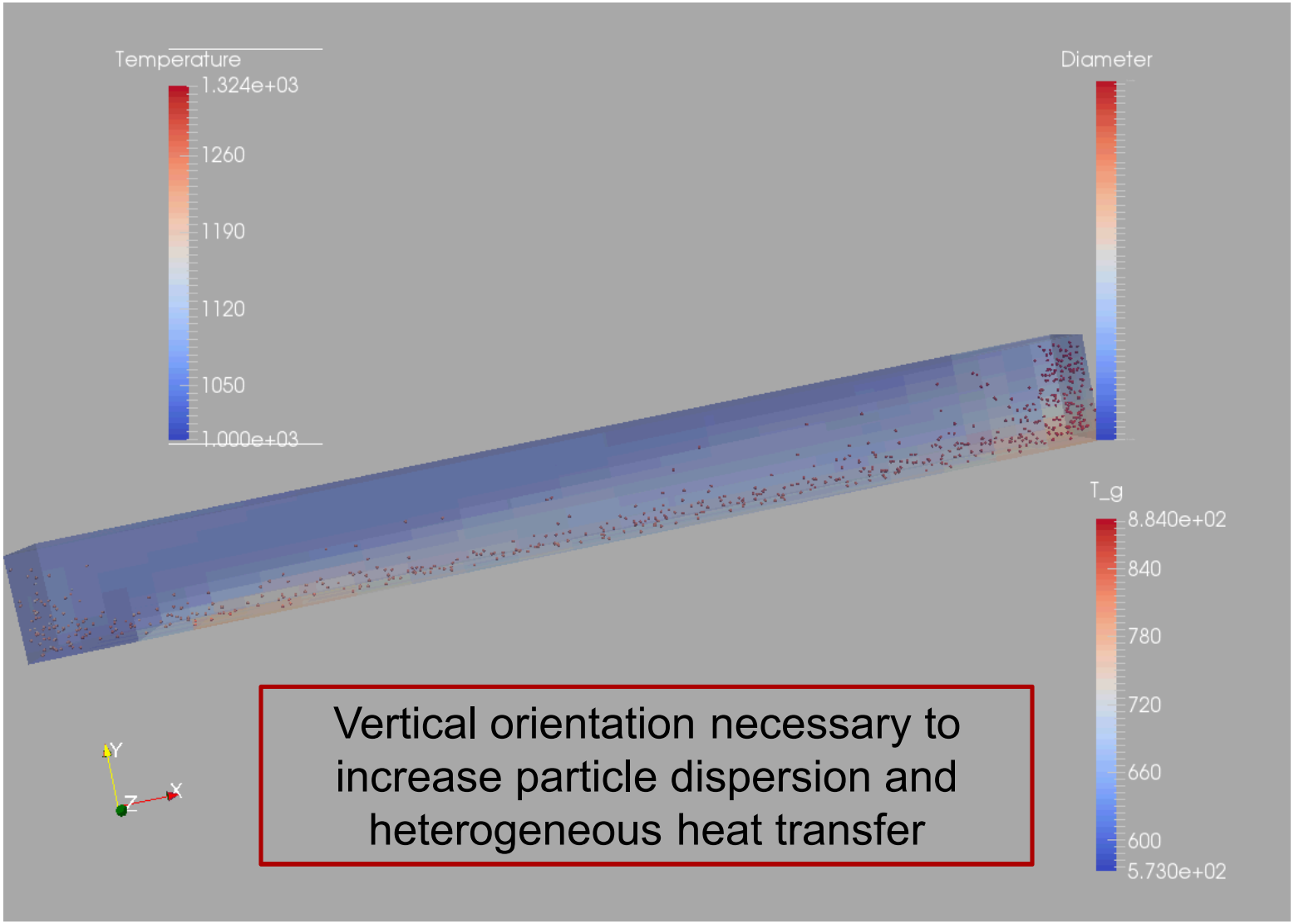



$$v_f = \sqrt{\frac{\cos(\theta) 2 g V_p}{C_D A_x} \left( \frac{\rho_p - \rho_f}{\rho_f} \right)}$$

Designing reactor hydrodynamics to slow particle is risky, very sensitive to particle size variation



# Sloped reactor exhibits poor heat transfer



# 1-D model development

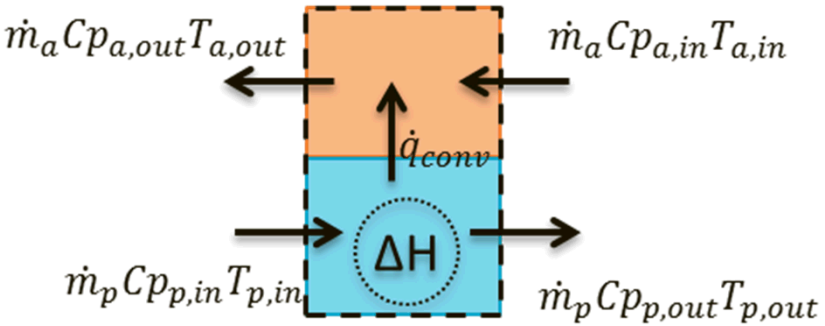
$$\frac{dT_f}{dx} = -\frac{h(T_p - T_f)A_c}{\dot{m}_f c_{p,f}} + \frac{(1 - F_{p,f})\Delta H_{rxn}}{c_{p,f}\, dx}$$

$$\frac{dT_p}{dx} = -\frac{h_s(T_p - T_f)A_c}{\dot{m}_p c_{p,p}} + \frac{F_{p,f}\, \Delta H_{rxn}}{c_{p,p}\, dx}$$

$$\frac{du_p}{dt} = g\underbrace{\frac{\rho_p - \rho_f}{\rho_p}} - \frac{(u_p + u_f)^2 Cd \cdot Ax \cdot \rho_f}{2V_p \rho_p}$$

Buoyancy term $\approx 1$

$$dt = \frac{dx}{u_p}$$




Assumptions:

- Steady-state
- Adiabatic (insulating reactor walls), Axial conduction is negligible (Pe >>1)
- Convection-dominated particle heat transfer (lumped capacitance)
- Particle volume fraction constant along reactor length

# Heterogeneous heat transfer coefficient

- Gunn correlation used (D.J. Gunn *Int. J. Heat Mass Transfer*, 1977)

$$Re_s = \frac{d_p(v_f + v_p)\rho_f}{\mu_f}$$

$$Nu_s = (7 - 10\alpha_f + 5\alpha_f^2)(1 + 0.7Re_s^{0.2}Pr^{1/3}) + (1.33 - 2.4\alpha_f + 1.2\alpha_f^2)Re_s^{0.7}Pr^{1/3}$$

$A_i$ = interface area density = $\frac{6\alpha_p}{d_p}$ Particle surface area per cell volume

$$h_s = \frac{k_f}{d_p} A_i \cdot Nu_s$$

$$q''' = h_s(T_p - T_f) \quad \text{in W/m}^3$$

# Python script

```
### Discretized ROx model #############
### Sean Babiniec ######################
### NOTES: Particle acceleration added#
###        Stochastic ability added ###
#######################################

import numpy as np
import matplotlib.pyplot as plt
#from scipy.integrate import odeint

### All units in SI ###################################
###### things needed to be brought into function ######
#Tp in
#Tf in
#PR
#L
#r
#dpart
#mdotf
#mdotp
#deltain
#TPf
#Beta

Rgas=8.314
MWair = 28.9/1000.
### geometry (assume cylinder) #####
L = 0.5 #m - reactor length
r = 0.019 #m - reactor radius

PR = 1.0 #atm

vfpart = 0.01 # volume fraction particles

deltain = 0.3 #[-] - inlet delta
rhop = 4402.7 #kg/m^3 - particle density
Prox = PR *101325 # Pa
#### Reaction properties #####
TPf = 0.5 #Particle to fluid heat generation factor

P = 1 # number of prob sims

Tfluid_P = np.zeros(P)
dpart_P = np.zeros(P)
Tpartin_P = np.zeros(P)
Tfluidin_P = np.zeros(P)
mdotp_P = np.zeros(P)
mdotf_P = np.zeros(P)
hfactor_P = np.zeros(P)

for sto in range(0, P):
    dpart_avg = 500.00000000e-6 #m - particle diamter (assume spheres)
    sigma_dpart = 175.00000000e-6
    dpart =np.random.normal(dpart_avg,sigma_dpart)
```

```
Tpartin_avg = 1323 #K - particle inlet temperature
sigma_Tpartin = 10 #K
Tpartin = np.random.normal(Tpartin_avg,sigma_Tpartin) #K
Tfluidin_avg = 670 #K - air inlet temperature
sigma_Tfluidin = 2 #K
Tfluidin = np.random.normal(Tfluidin_avg,sigma_Tfluidin) #K
mdotp_avg = 0.00225 #kg/s - particle flow rate
sigma_mdotp = 4.80e-6
mdotp = np.random.normal(mdotp_avg, sigma_mdotp)
mdotf_avg = 0.0015 #kg/s - air flow rate
sigma_mdotf = 4.98e-6
mdotf = np.random.normal(mdotf_avg, sigma_mdotf)
hfactor_avg = 1
sigma_hfactor = 0.1
hfactor = np.random.normal(hfactor_avg, sigma_hfactor)
if hfactor < 0.:
    while hfactor < 0.:
        hfactor = np.random.normal(hfactor_avg, sigma_hfactor)

Tpartin_P[sto] = Tpartin
Tfluidin_P[sto] = Tfluidin
mdotp_P[sto] = mdotp
mdotf_P[sto] = mdotf
dpart_P[sto] = dpart
hfactor_P[sto] = hfactor

###################################
######## Procedure ################
###################################
#1. Make guess array
#2. Solve discritized equation using guess array when information unknown
#3. Guess = New
#4. Iterate until converged


######### calculate invariant parameters ##########
Ax = np.pi*r**2.
Vp = 4.0/3.0*np.pi*(0.5*dpart)**3.0 #m^3
Ap = np.pi/4.0*(dpart)**2.0 #m^2
Cd=0.47
g = 9.81 #m/s^2

Aint = 6.*vfpart/dpart # interfacial area per volume (m^2/m^3)

########### initialize arrays ######################

N = 250 # number of spatial steps
x = np.zeros(N+1) # spatial array
dx = L/(N) # discrete length
Tfluid_g = np.zeros(N+1) # Fluid T guess array
Tfluid_n = np.zeros(N+1) # Fluid T new array
Tpart_g = np.zeros(N+1)  # Particle T guess array
Tpart_n = np.zeros(N+1)  # Particle T new array
```

# Python script

```
##########################################
# Temperature-dependent variables functions#
##########################################

def rhoair(Temp,Prox): #kg/m^3
    rho=Prox/(Rgas/MWair*Temp)
    return(rho)
def cpf(Temp): #J/kg-K
    cpf = 1.9327e-10*Temp**4.-7.9999e-7*Temp**3.+0.0011407*Temp**2.-0.4489*
    return(cpf)
def cpp(Temp):  #J/kg-K
    cpp = (0.0772*np.log(Temp-273.15)+.4747)*1000.
    return(cpp)
def kf(Temp):  #W/m-K
    k=1.5207e-11*Temp**3-4.8574e-8*Temp**2+1.0184e-4*Temp-0.00039333
    return(k)
def muf(Temp):  #Sutherland approximation
    mu=0.00001827*(291.15+120.)/(Temp+120.)*(Temp/291.15)**(3./2.)
    return(mu)

def Red(Temp, dpart, vfluid, vpart, Prox):
    Red = dpart*(vfluid+vpart)*rhoair(Temp,Prox)/(muf(Temp))
    return(Red)

def Pr(Temp):
    Pr=cpf(Temp)*muf(Temp)/kf(Temp)
    return Pr
def Nu(Temp,vfpart, dpart, vfluid, vpart, Prox): #Gunn correlation
    Nu=(7.0-10.0*(1.-vfpart)+5.*(1.-vfpart)**2.0)*\
        (1.0+0.7*Red(Temp,dpart,vfluid,vpart,Prox)**(0.2)*Pr(Temp)**(1./3.)
        +(1.33-2.4*(1.-vfpart)+1.2*+(1.-vfpart)**2.0)*Red(Temp,dpart,vfluid
    return Nu
def H_AB03(delta):
    Rxn=(787.6*delta**2.0+897.06*delta)*1000.
    return Rxn
#print 'Nu', Nu(1480.,vfpart, dpart,0.19200 , 0.25, 16.7*101325.)
#print 'rhof', rhoair(1480., 16.7*101325.)
#print 'Red', Red(1480., dpart, 0.19200, 0.25, 16.7*101325.)
#print 'muf', muf(1480.)
# Are these all needed? Can just calculate in loop
vpart_i = mdotp/(rhop*np.pi*r**2.*vfpart) # particle velocity IGNORES accel
Conv_per_sec = 3.0 #assume 80% conversion every second. This can be an inpu
#Beta = Conv_per_sec*dx/vpart # Beta calculates conversion per cell

##### Initialize other arrays #########
vfluid= np.zeros(N+1)  # calculate these
vpart= np.zeros(N+1)
Beta= np.zeros(N+1)
rho_f = np.zeros(N+1) # Fluid density
k_fluid = np.zeros(N+1) # Fluid thermal conductivity
Pr_f = np.zeros(N+1)    # Prandtl number
#Re_f = np.zeros(N+1)  # Fluid-cylinder Reynolds number
Re_d = np.zeros(N+1)  # Fluid-particle Reynolds number
a = np.zeros(N+1)
```

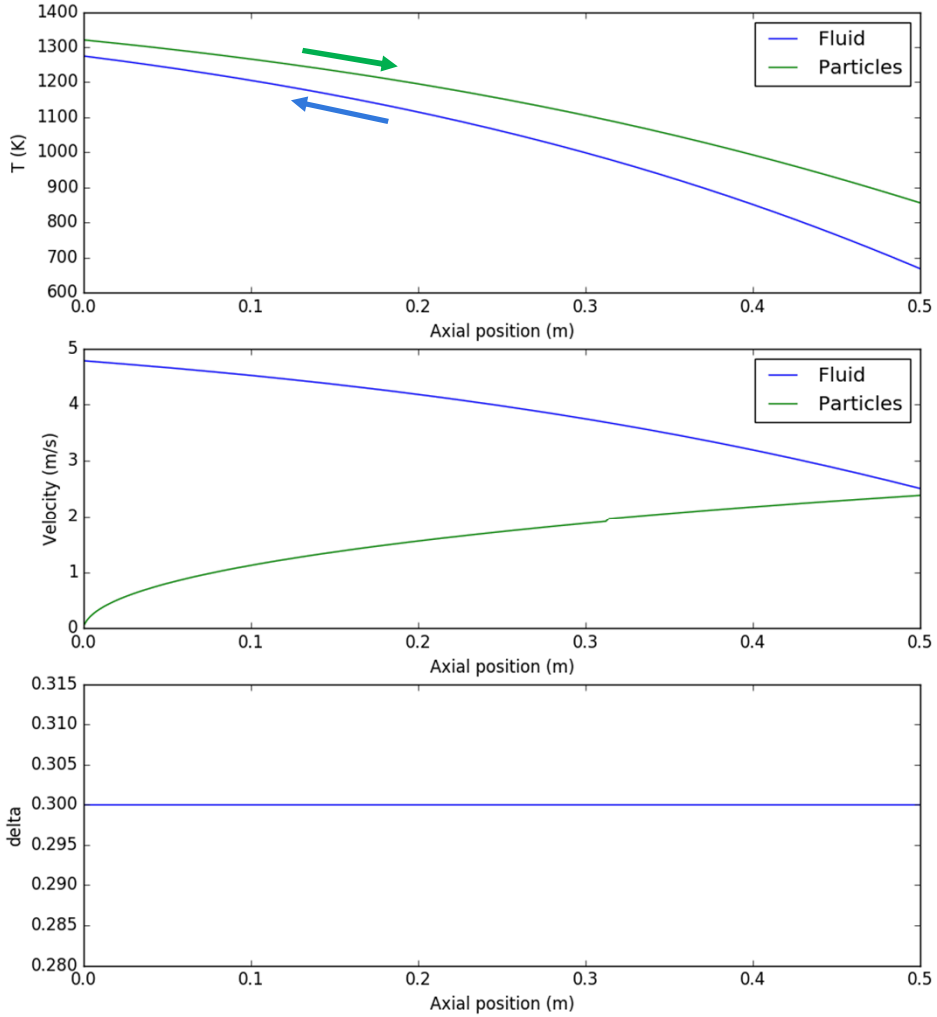
```
b = np.zeros(N+1)
#Nu = np.zeros(N+1)
h = np.zeros(N+1)  #6*kfluid*Nu/dpart * Aint
delta = np.zeros(N+1)
delta_eq = np.zeros(N+1)
delta_new = np.zeros(N+1)
H_rxn = np.zeros(N+1)
Tpart_n[0] = Tpartin
Tfluid_n[N] = Tfluidin
error =1.
iteration = 0.
for ee in range (0,N+1):
    Tpart_g[ee] = Tpartin-250.
    Tfluid_g[ee] = Tfluidin+250.
    vfluid[ee] = 0.7
    h[ee] = 600.
    delta[ee] = deltain
    H_rxn[ee] = 500.0
    delta_eq[ee] = 0.3
    vpart[ee] = vpart_i
    Beta[ee] = Conv_per_sec*dx/vpart_i
for i in range (0,N+1):
    x[i] = i*dx

while error > 1e-6:

    error = 1e-12
    T_tot = 0.
    res_tot = 0.
    tpartres = 0.
    for k in range (1,N): #Loop for inner cells.
            Tpart_n[k]=(TPf*H_rxn[k]/cpp(Tpart_g[k])+h[k]*dx*np.pi*r**2./(m
            Tfluid_n[k] = ((1.-TPf)*H_rxn[k]/cpf(Tfluid_g[k])+h[k]*dx*np.pi

    #Value for air temp at air outlet and particle temp at particle outlet
    Tfluid_n[0] = ((1.-TPf)*H_rxn[0]/cpf(Tfluid_g[0])+h[0]*dx*np.pi*r**2./(
    Tpart_n[N]=(TPf*H_rxn[N]/cpp(Tpart_g[N])+h[N]*dx*np.pi*r**2./(mdotp*cpp

    for conv in range (0,N+1):
        res_sq =np.abs(Tfluid_n[conv]-Tfluid_g[conv])/(Tfluid_n[conv])
        res_tot = res_tot + res_sq
        T_tot = T_tot + Tfluid_n[conv]
        error = res_tot/float(N)
    #print error
    H_tot=0.
    #update guesses and h delta H_rxn
    for z in range (0, N+1):
        Tfluid_g[z]=Tfluid_n[z]
        Tpart_g[z]=Tpart_n[z]
        vfluid[z] = mdotf/(rhoair(Tfluid_g[z],Prox)*np.pi*r**2.)
        h[z] = kf(Tfluid_g[z])*Nu(Tfluid_g[z],vfpart, dpart, vfluid[z], vpa
        b[z] = (Cd*Ap*rhoair(Tfluid_g[z],Prox))/(2.0*Vp*rhop)
#delta[0] must always be deltain
        if z >0:
```

# Python script

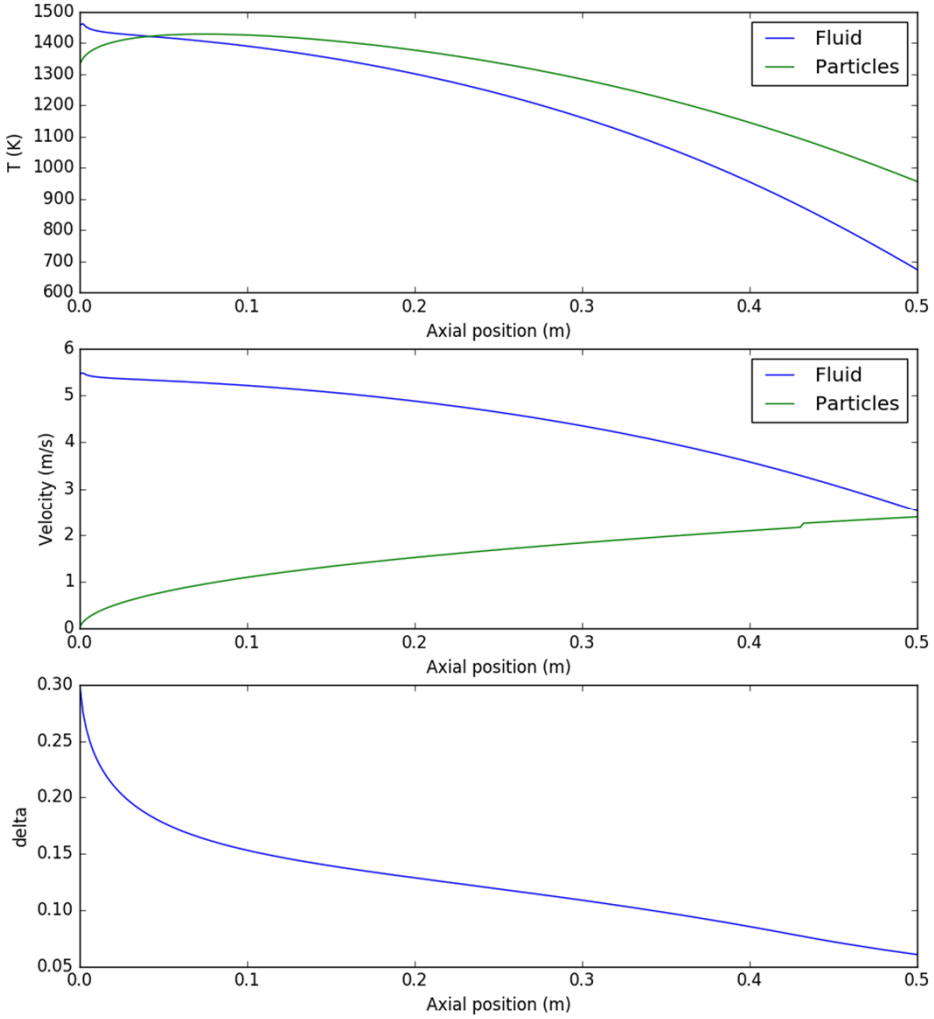
```
                if Tpart_g[z]*0.00032-0.34919 > 0.:
                    delta_eq[z] = Tpart_g[z]*0.00032-.34919
                    vpart[z] = (-2.*b[z]*vfluid[z]*dx+vpart[z-1]+np.sqrt(4.*b[z]
                else:
                    delta_eq[z] = 0.
                Beta[z] = Conv_per_sec*dx/vpart[z]
                delta[z] = delta[z-1]-(delta[z-1]-delta_eq[z])*Beta[z]
                H_rxn[z]= H_ABO3(delta[z-1])-H_ABO3(delta[z])
                H_tot = H_tot+H_rxn[z]

            iteration = iteration + 1
            # print 'iteration', iteration
        Tfluid_P[sto] = Tfluid_n[0]
        print sto, dpart, vpart[N]

        for s in range (0,N+1):
            tpartres = tpartres+dx/vpart[s]
        print tpartres

print dpart_P, Tfluid_P
#print 'delta', delta[N], H_tot, Tpartin
plt.subplot(321)
plt.plot(dpart_P[:], Tfluid_P[:], ".")
plt.xlabel('Particle diameter (m)')
plt.ylabel('Fluid outlet T (K)')

#
plt.subplot(322)
plt.plot(mdotf_P[:], Tfluid_P[:], ".")
plt.xlabel('Fluid mdot (kg/s)')

plt.subplot(323)
plt.plot(mdotp_P[:], Tfluid_P[:], ".")
plt.xlabel('Particle mdot (kg/s)')
plt.ylabel('Fluid outlet T (K)')

plt.subplot(324)
plt.plot(Tpartin_P[:], Tfluid_P[:], ".")
plt.xlabel('Particle inlet T (K)')

plt.subplot(325)
plt.plot(Tfluidin_P[:], Tfluid_P[:], ".")
plt.xlabel('Fluid inlet T (K)')
plt.ylabel('Fluid outlet T (K)')

plt.subplot(326)
plt.plot(hfactor_P[:], Tfluid_P[:], ".")
plt.xlabel('hfactor')

plt.show()
```

# 1-D model results

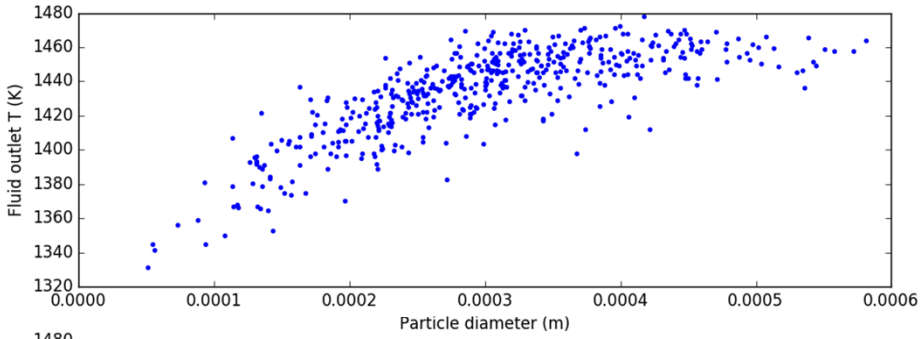
## Without exothermic reaction

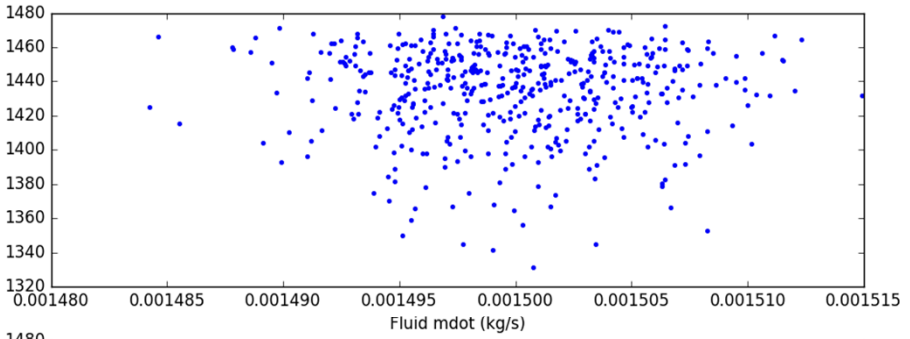


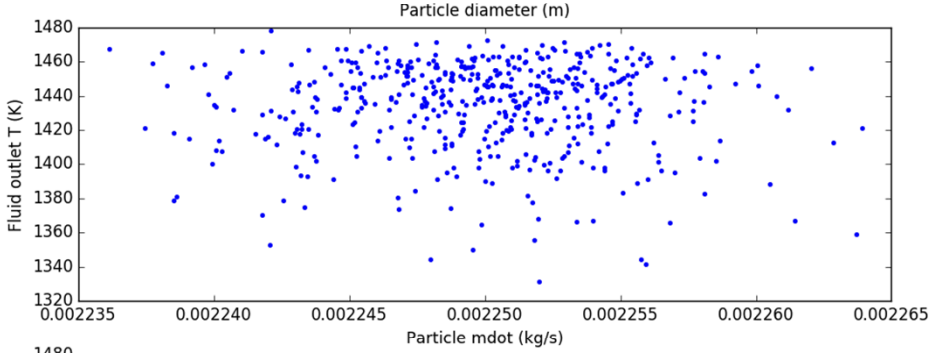

## With exothermic reaction

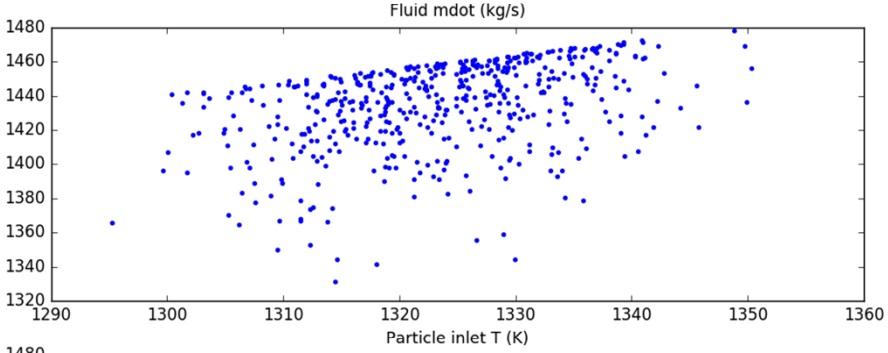


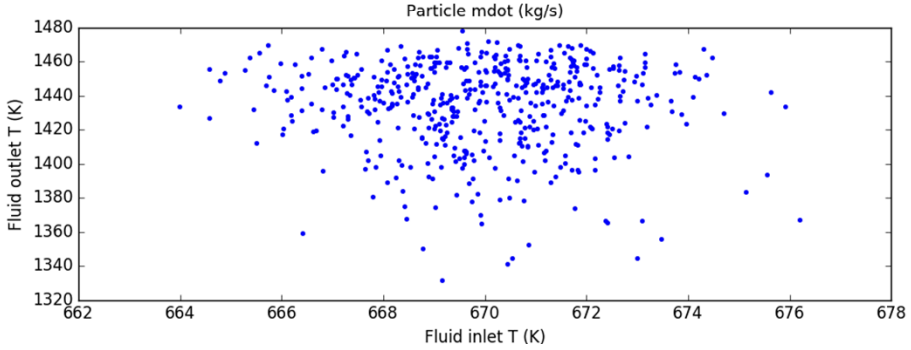
# Stochastic investigation - PR 16.7 – 500 simulations

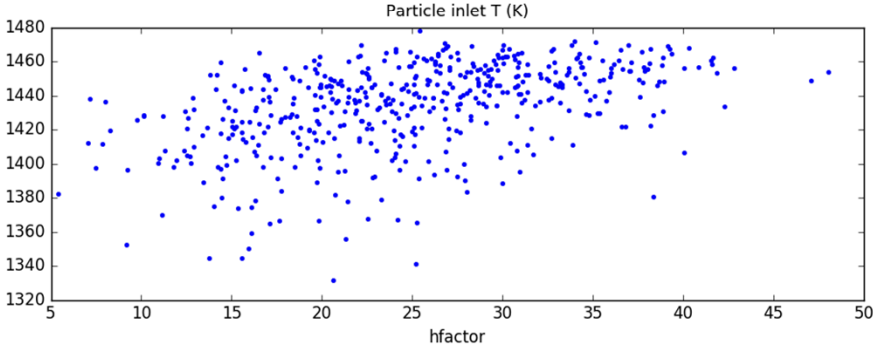


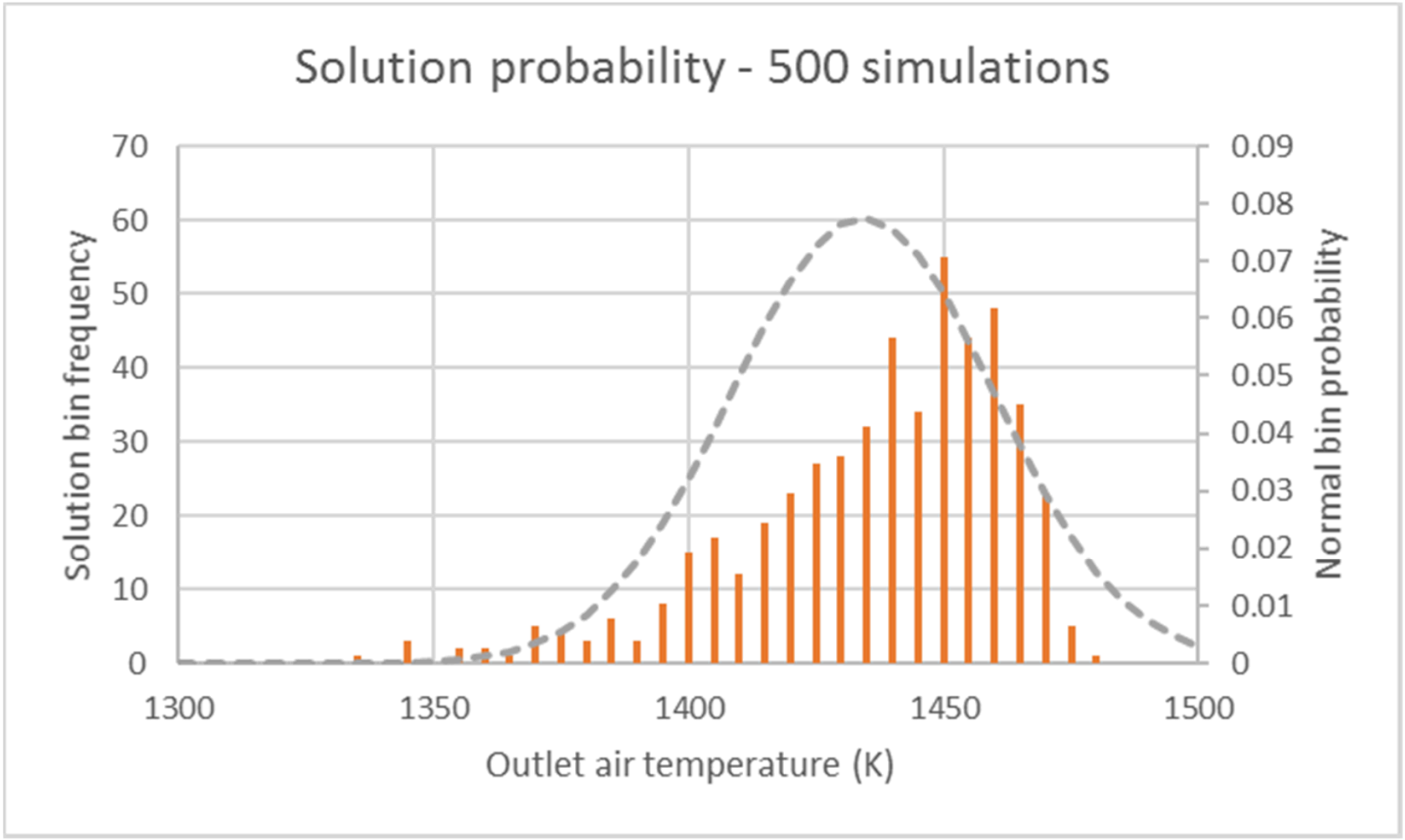



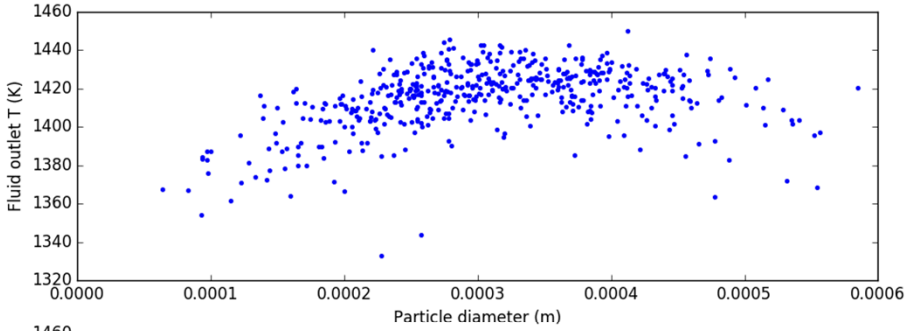



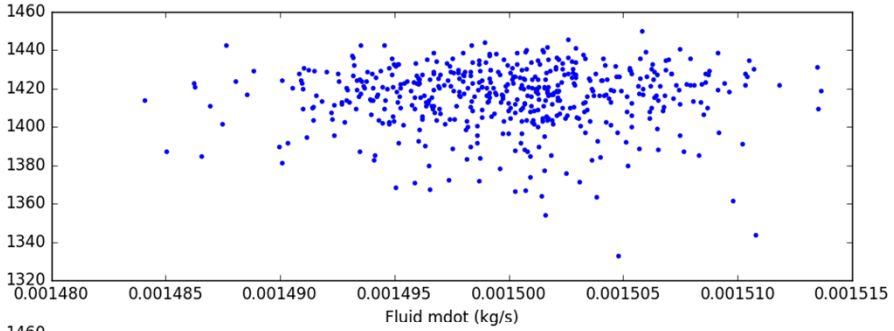



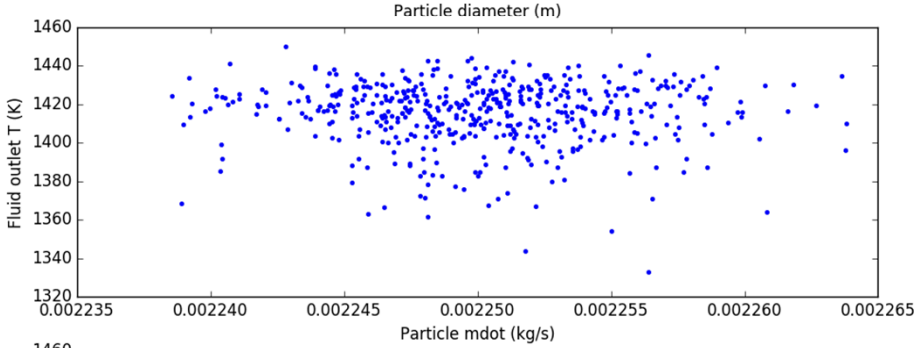
# PR 16.7 – 500 simulations

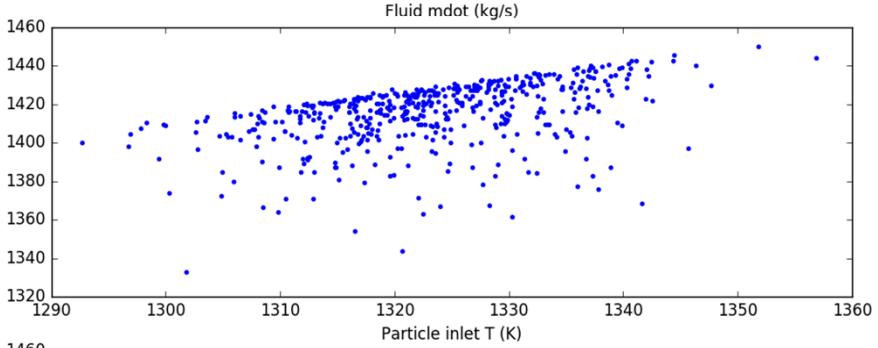


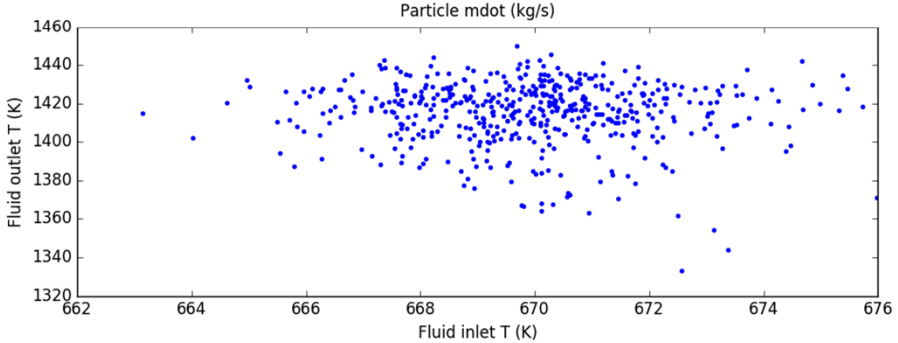
# Stochastic investigation - PR 1.0 – 500 simulations

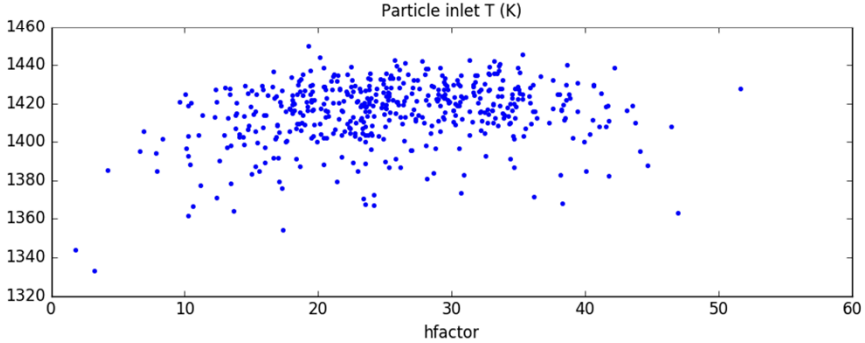


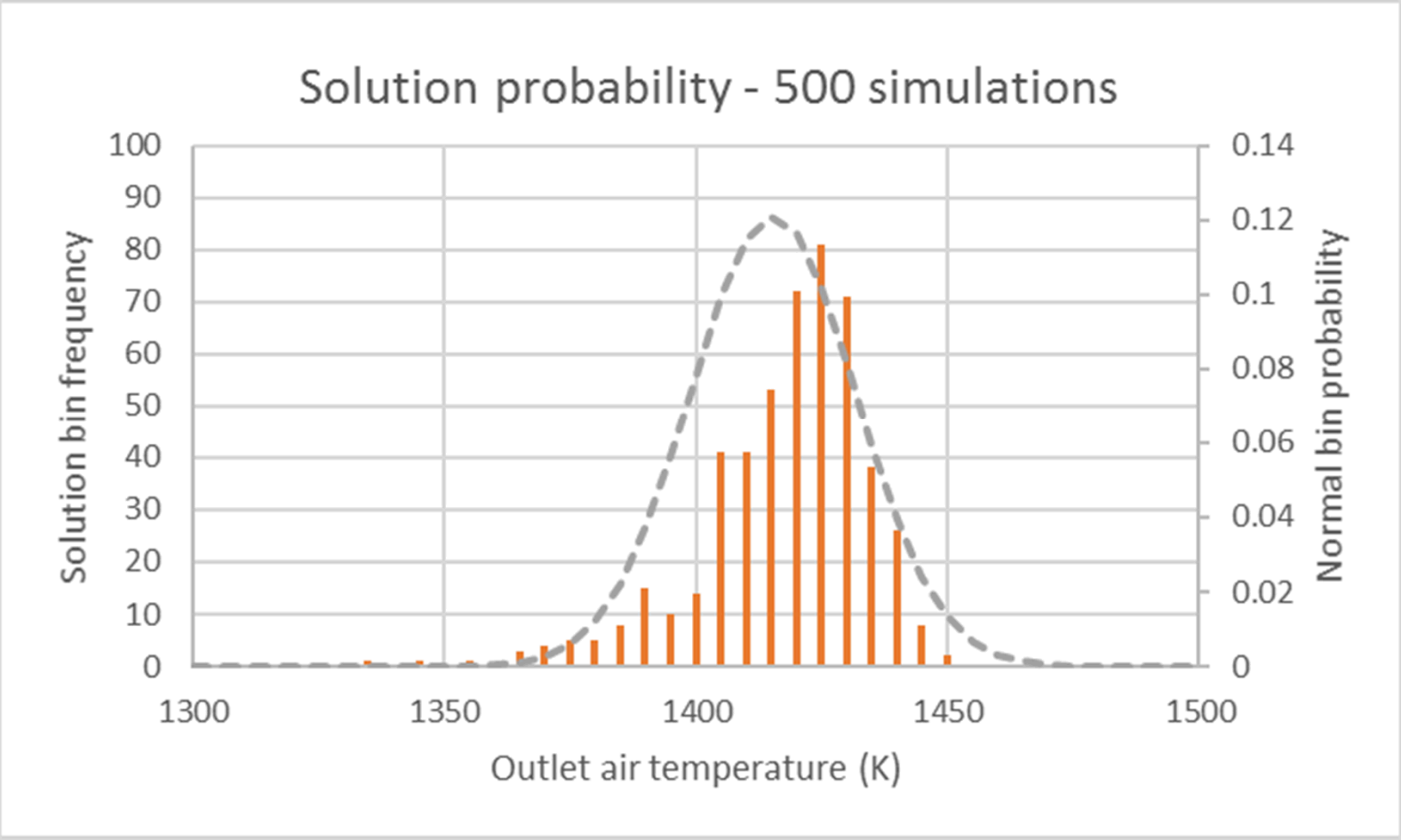



# PR 1.0 – 500 simulations



# Description of 3-D model

- ANSYS Fluent commercial code is used to model the fluid properties and the interactions between fluid and particle phase
  - Eulerian-Eulerian approach taken – particle phase approximated as a "fluid"
    - Same conservation equations used for both phases
    - Constitutive relationships describe solid particle phase properties
      - Granular kinetic theory
      - Momentum transfer between particle and fluid phase,
      - Heat transfer between particle and fluid phases
      - Particle "shear stress" described by frictional stress
    - Reaction implemented using a custom-written UDF
      - Reaction treated as quasi-thermodynamically limited

# Eulerian-Eulerian approximation

Two sets of conservation equations coupled through volume fractions for each phase

- Continuity:

$$\frac{\partial}{\partial t}(\alpha_i \rho_i) + \nabla \cdot (\alpha_i \rho_i v_i) = 0$$

- Momentum:

$$\frac{\partial}{\partial t}\left(\alpha_f \rho_f \boldsymbol{v_f}\right) + \nabla \cdot \left(\alpha_f \rho_f \boldsymbol{v_f} \boldsymbol{v_f}\right) = -\alpha_f \nabla p + \nabla \cdot \tau_f + \alpha_f \rho_f \boldsymbol{g} + K_{sf}\left(\boldsymbol{v_s} - \boldsymbol{v_f}\right)$$

$$\frac{\partial}{\partial t}(\alpha_s \rho_s \boldsymbol{v_s}) + \nabla \cdot (\alpha_s \rho_s \boldsymbol{v_s} \boldsymbol{v_s}) = -\alpha_s \nabla p_s + \nabla \cdot \tau_s + \alpha_s \rho_s \boldsymbol{g} + K_{sf}\left(\boldsymbol{v_f} - \boldsymbol{v_s}\right)$$

- Energy:

$$\frac{\partial}{\partial t}(\alpha_i \rho_i H_i) + \nabla \cdot (\alpha_i \rho_i \boldsymbol{v_i} H_i) = \nabla \cdot \alpha_i k_{i,eff} \nabla T_i - h_{ij}\left(T_i - T_j\right)$$

# Constitutive relationships – Heat, Momentum

- Momentum exchange between phases (Gidaspow)

$$K_{sf} = \begin{cases} \frac{18}{Re_s}\left[1 + 0.15\left(\alpha_f Re_s\right)^{0.687}\right]\frac{\alpha_s \rho_f \left|v_s - v_f\right|}{d_s} & \alpha_f > 0.8 \\ \frac{150\alpha_s\left(1-\alpha_f\right)\mu_f}{\alpha_f d_s^2} + 1.75\frac{\alpha_s \rho_f \left|v_s - v_f\right|}{d_s} & \alpha_f \leq 0.8 \end{cases}$$

- Heat exchange (Gunn)

$$Nu_s = \left(7 - 10\alpha_f + 5\alpha_f^2\right)\left(1 + 0.7 Re_s^{0.2} Pr^{1/3}\right) + \left(1.33 - 2.4\alpha_f + 1.2\alpha_f^2\right) Re_s^{0.7} Pr^{1/3}$$

$$h_{ij} = \frac{k_f}{d_p} A_i [Nu_s]$$ $A_i$ = interface area density = $\frac{6\alpha_p}{d_p}$ Particle surface area per cell volume

$$Q_{12} = h_{12}(T_1 - T_2)$$ (volumetric heat rate)

# Constitutive relationships – Granular theory

- Granular energy treated analogous to molecular diffusion, "granular temperature"

$$\Theta_s = \frac{1}{3} u_{s,i} u_{s,i}$$ Granular temperature function of particle velocity

$$\frac{3}{2}\left[\frac{\partial}{\partial t}(\alpha_s \rho_s \Theta_s) + \nabla \cdot (\alpha_s \rho_s v_s \Theta_s)\right] = (p_s I + \tau_s) : \nabla v_s + (\kappa \nabla \Theta_s) - \gamma_{\Theta_s} - 3K_{sg}\Theta_s$$

Transport equation accounts for diffusion of kinetic energy, collisional dissipation, and solid-fluid exchange

$$\kappa = \frac{15 d_s \rho_s \alpha_s \sqrt{\Theta_s \pi}}{4(41 - 33\eta)}\left[1 + \frac{12}{5}\eta^2(4\eta - 3)\alpha_s g_{0,ss} + \frac{16}{15\pi}(41 - 33\eta)\eta \alpha_s g_{0,ss}\right]$$

$$\gamma_{\Theta_s} = \frac{12(1 - e_{ss}^2) g_{0,ss}}{d_s \sqrt{\pi}} \rho_s \alpha_s^2 \Theta_s^{1.5} \qquad \eta = \frac{1}{2}(1 + e_{ss})$$

$$g_{0,ss} = \left[1 - \left(\alpha_s / \alpha_{s,max}\right)^{1/3}\right]^{-1}$$

# Constitutive relationships – Solid viscosity

Effective solid "viscosity" needed to calculate shear stress tensor in momentum continuity equation

$$\mu_s = \mu_{s,kin} + \mu_{s,col} + \mu_{s,fr}$$

$$\mu_{s,kin} = \frac{10\rho_s d_s \sqrt{\pi\Theta_s}}{96\alpha_s(1+e_{ss})g_{0,ss}}\left[1+\frac{4}{5}\alpha_s g_{0,ss}(1+e_{ss})\right]^2$$

$$\mu_{s,col} = \frac{4}{5}\alpha_s\rho_s d_s g_{0,ss}(1+e_{ss})\left(\frac{\Theta_s}{\pi}\right)^{1/2}$$

# Treatment of reaction by UDF

- Oxidation kinetics are fast, thermodynamically limiting in our experiments – Accurate oxidation kinetic rates unavailable
  - Mechanism for reaction implemented using psuedo-equilbrium kinetics (quasi-thermodynamically limited)

The new value for δ (for oxidation) due to oxidation is modeled as:

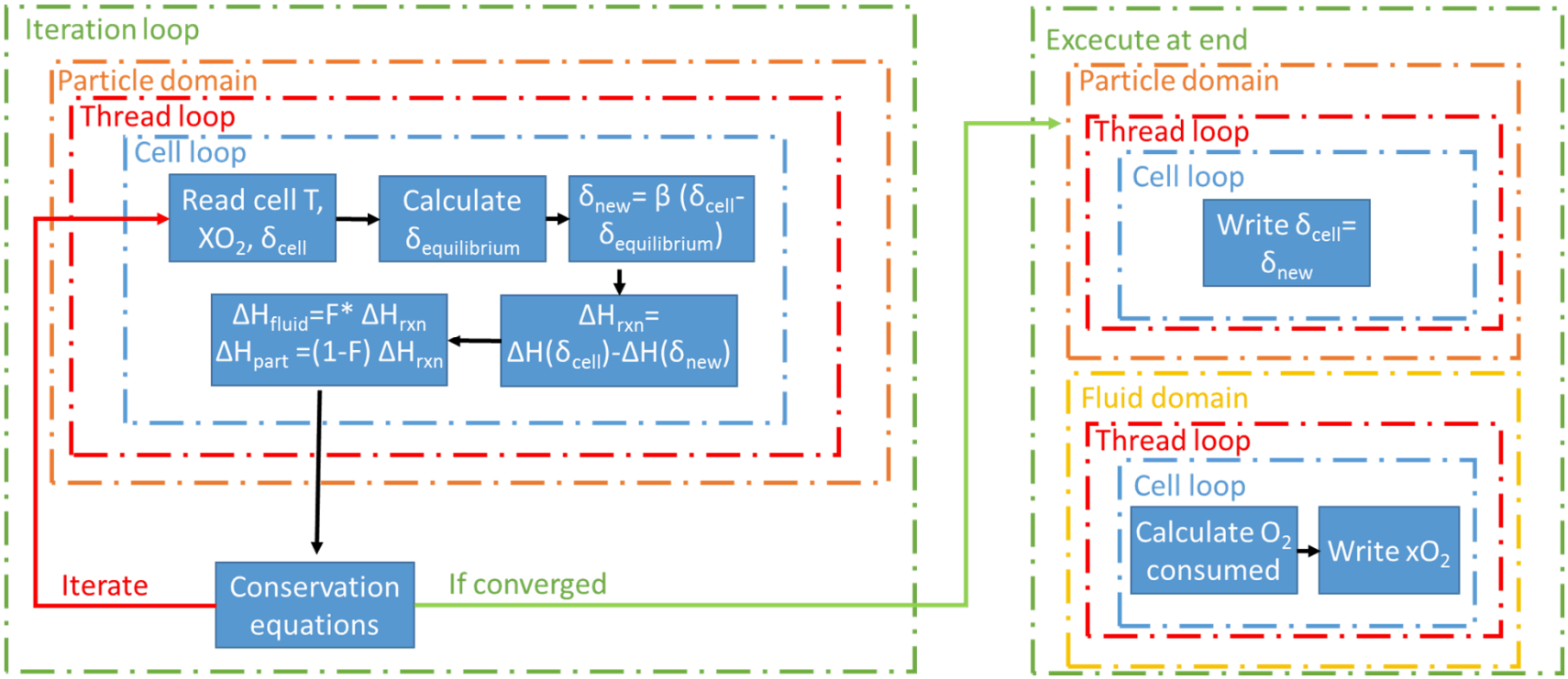
$$\delta_{new} = \delta_{current} - \beta(\delta_{current} - \delta_{equilibrium})$$

- The change in delta in the computational cell is used to calculate gaseous oxygen consumption and heat generation in each computational cell.
- The β term describes the "progress" towards equilibrium in each cell, and can be used to adjust the reaction rate.

To ensure equal reaction rates at any time step:

$$\beta = \frac{\alpha}{second} * current\_timestep$$

# UDF flow diagram




Simplified flow diagram of custom reaction UDF algorithm

# Testing UDF for appropriate behavior

Sandia National Laboratories

## Testing the reaction UDF for accuracy

Engineering estimation of behavior:

$\Delta H_{\delta=0.3}$= 340,000 J/kg

$m_{CAM28} = 4.4027 \times 10^{-7}$ kg

$m_{air} = 1.2 \times 10^{-6}$ kg

$n_{air} = 4.15 \times 10^{-5}$ mol

$n_{O2} = 8.72 \times 10^{-6}$ mol

$c_{p,air} \sim c_{p,CAM28} = 1100$ J/kg-K

$$\Delta T = \frac{\Delta H \, m_{air}}{(m_{air} + m_{CAM28}) \, c_p} = \sim 84.5 \, K$$

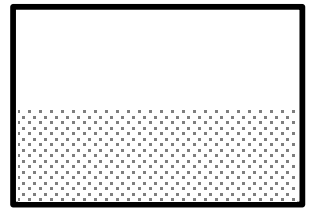
Expected $T_{equil} \sim 384.5$ K

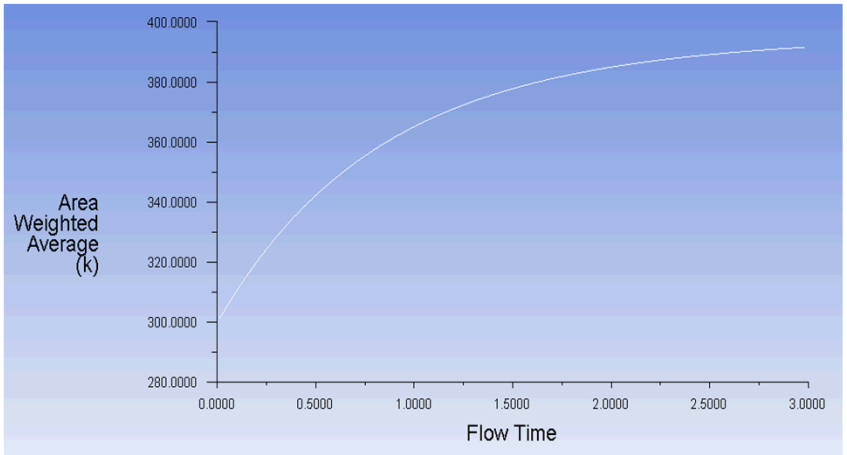
$n_{O2,\,cons} = 4.86 \times 10^{-7}$ mol

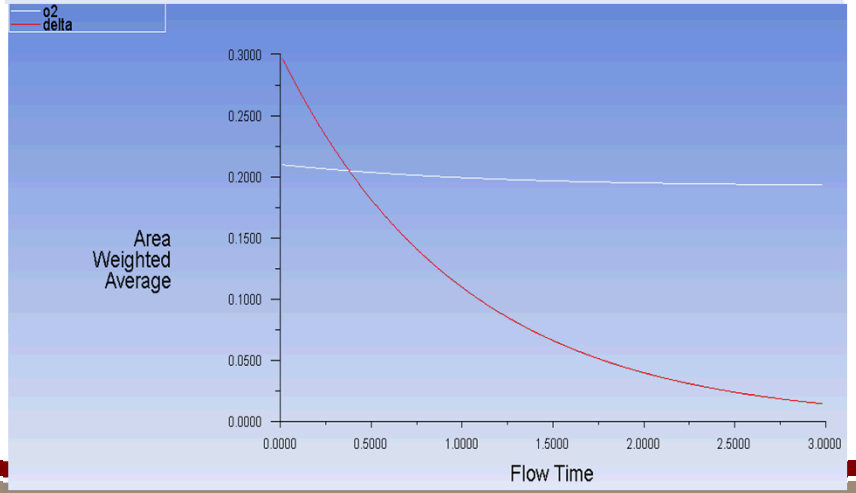
Expected mole fraction$_{O2}$ = 0.198

Simulation results:
Average temp of particles and air = 391.5 K
Mole fraction$_{O2}$ = 0.19

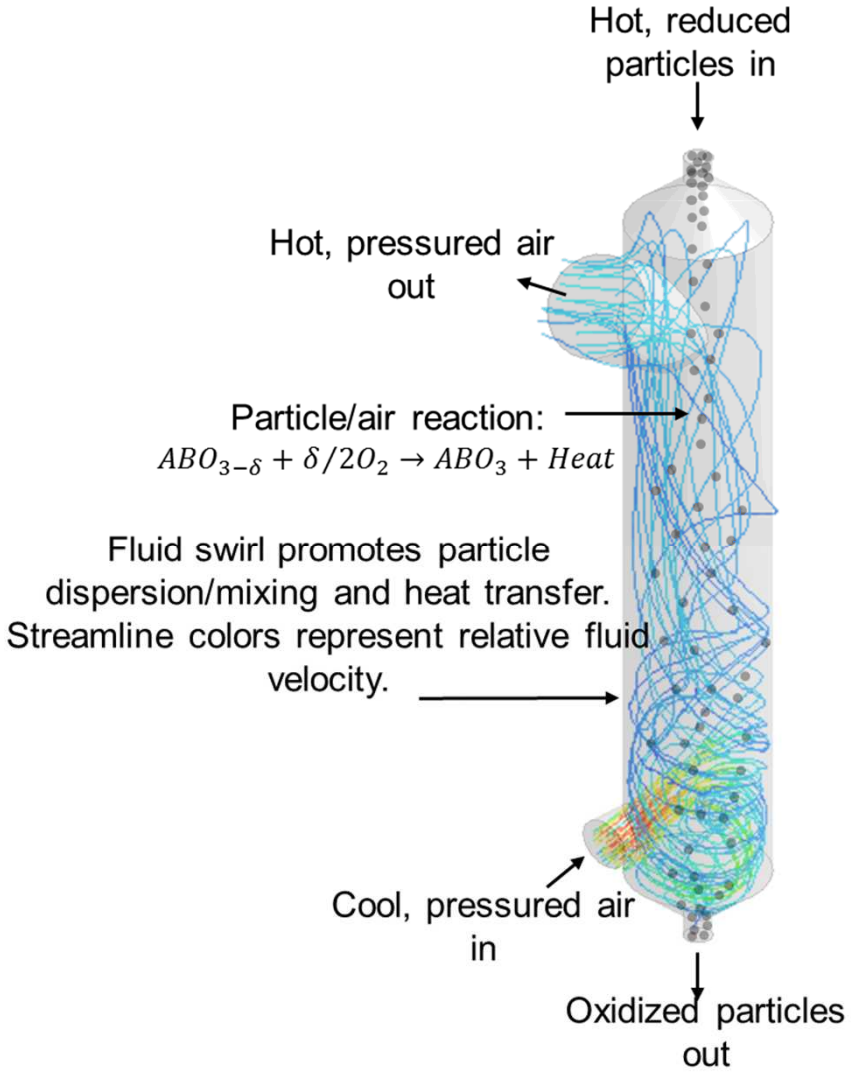
Start at t=0 with known particle and air mass in container, particle δ = 0.3, $T_{particle}$ and $T_{air}$ = 300 K





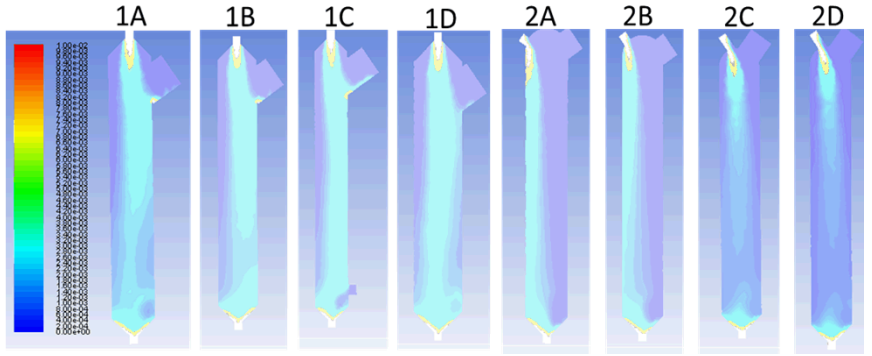



# Optimizing the 3-D reactor geometry




A parametric analysis was conducted on the design geometry

- 8 total geometries analyzed
  - Varied diameters:
    - Main tube
    - Inlets and outlets
  - Varied particle inlet, air outlet locations
- Particle reaction NOT included
- Recorded:
  - Outlet temperatures of particles and air
  - Pressure drop across reactor

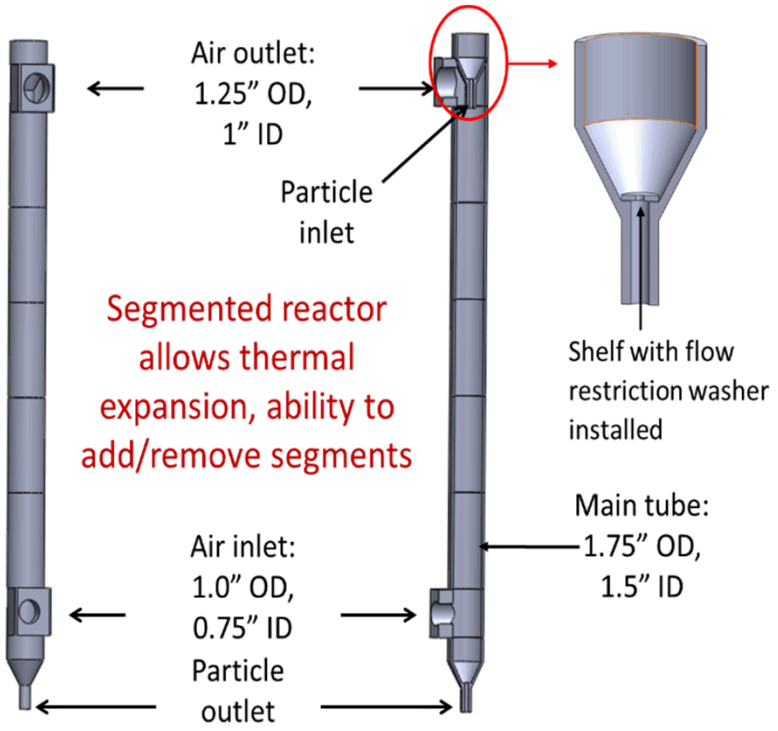



Steady state particle volume fraction

# Realization of ROx geometry

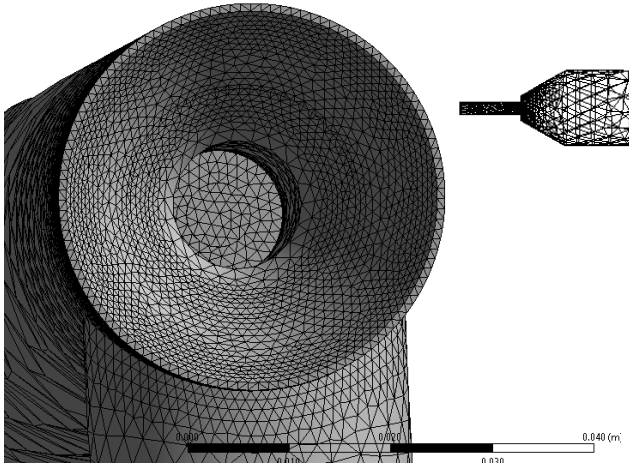
Geometry must be modified for fabrication feasibility

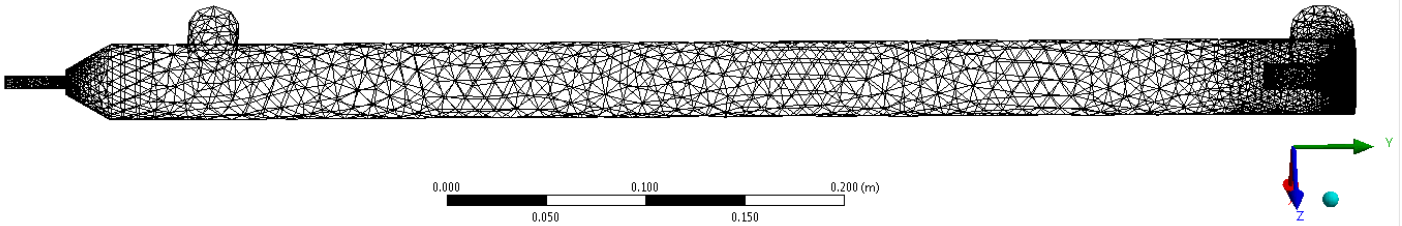
- Single monolith part not feasible due to size and complexity
- Segmented reactor approach taken
  - Several smaller parts easier to manufacture via additive manufacturing (Robocasting) or by off-the-shelf tubes
  - Thermal stresses alleviated by expansion between segments
  - Mullite chosen as construction material
    - Operation up to 1400 °C
    - Good thermal shock resistance
    - Low thermal conductivity compared to alumina
    - Inexpensive



# 3-D CAD/Meshing, simulation boundary conditions

- Meshing done using ANSYS Workbench meshing software
  - Default settings used on main tube
  - Smaller features refined using “edge meshing” refinement
- Total number of elements = 72,408

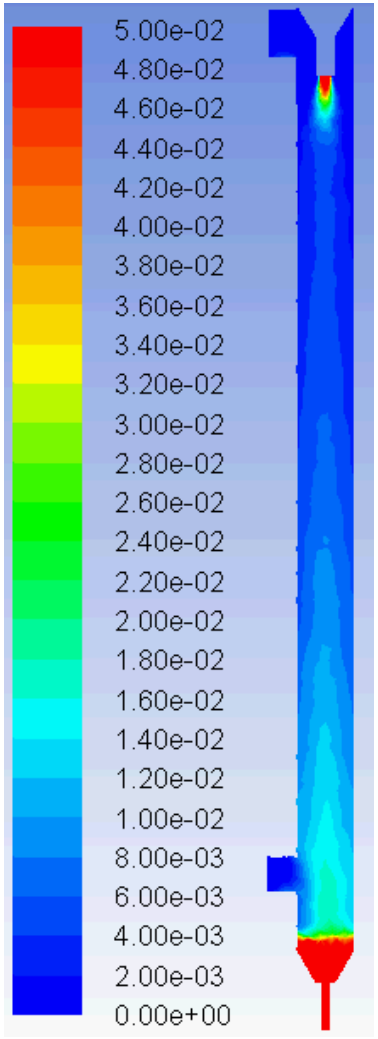




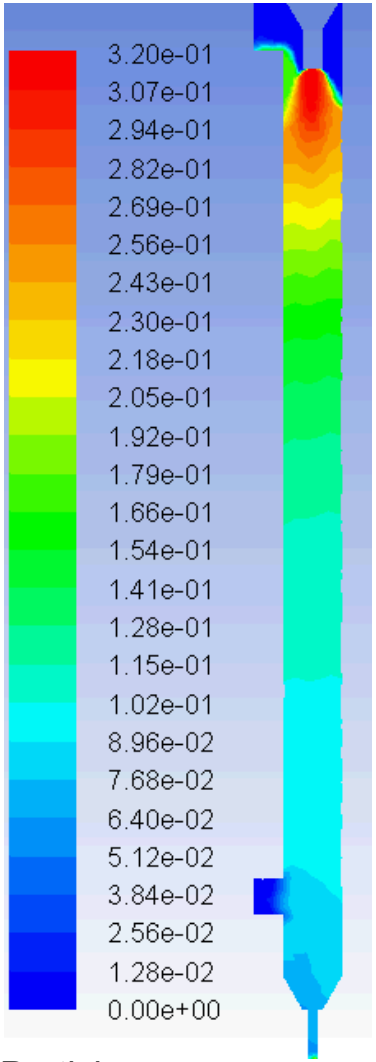

Boundary conditions:

- Air in: $\dot{m} = 0.003$ kg/s, 670 K, $pO_2$ = 0.21 atm, P = 16.7 atm
- Air out: Pressure outlet, P = 16.7 atm
- Particle in: $\dot{m} = 0.0045$ kg/s, 1323 K, δ = 0.32
- Particle out: Only particles removed

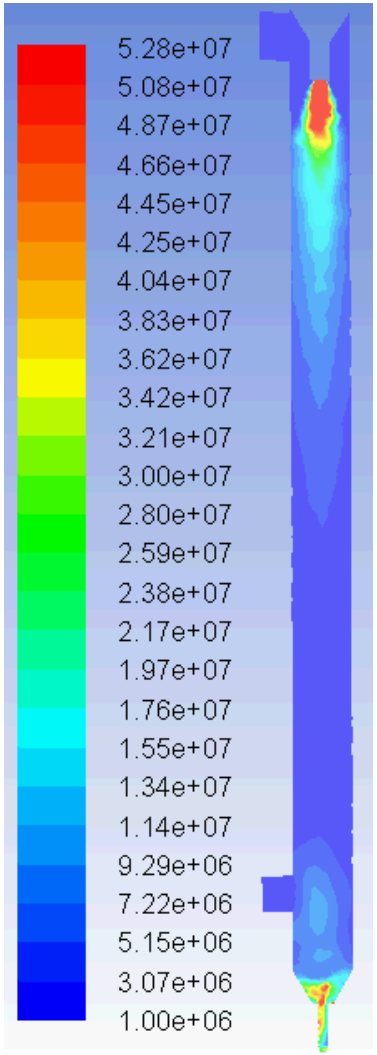
# 3-D results – Good particle/fluid dispersion



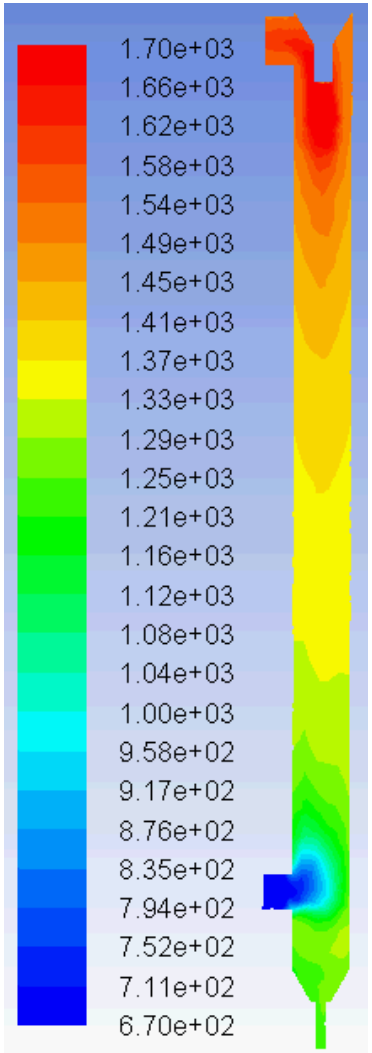

Particle volume fraction




Particle oxygen non-stoichiometry (δ)

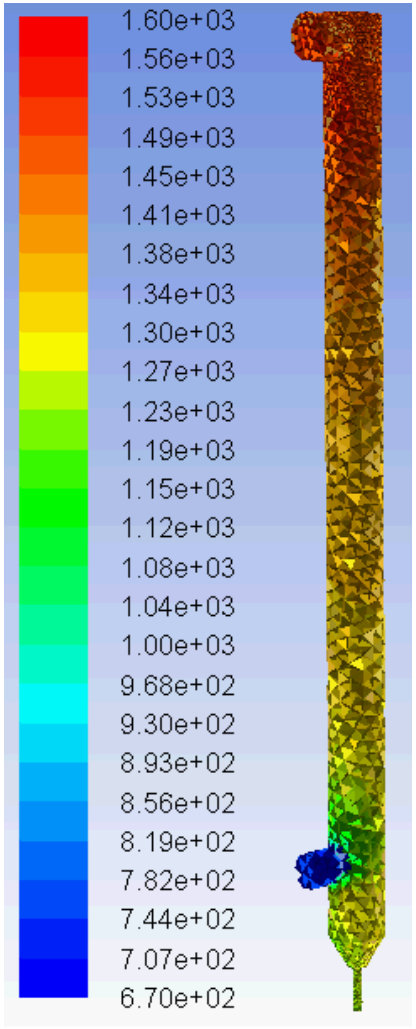



Exotherm (W/m$^2$)

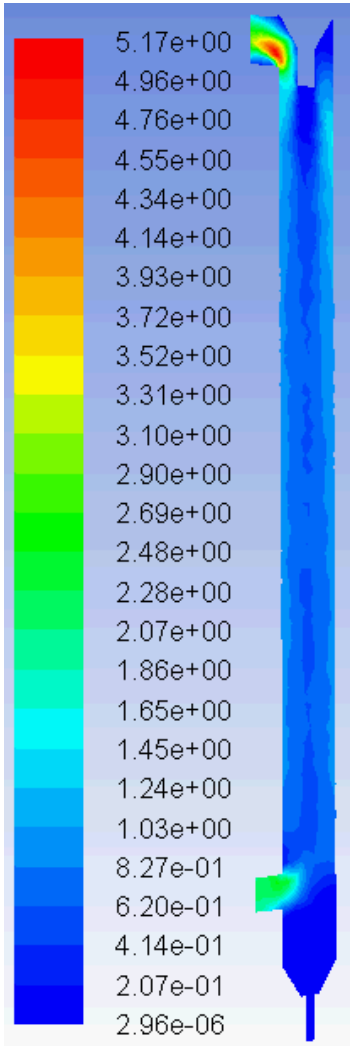



Fluid Temperature (K)

# 3-D results – Heat transfer metrics




Fluid Temperature (K)




Fluid Pressure (Pa)

| Parameter | Value |
|---|---|
| Air outlet temp (°C) | 1275.45 ± 23.98 |
| Particle outlet temp (°C) | 965.97 ± 23.50 |
| δ out | 0.071 ± 0.005 |
| Effectiveness | 78.2% |
| Power transferred to fluid | 3.05 kW |

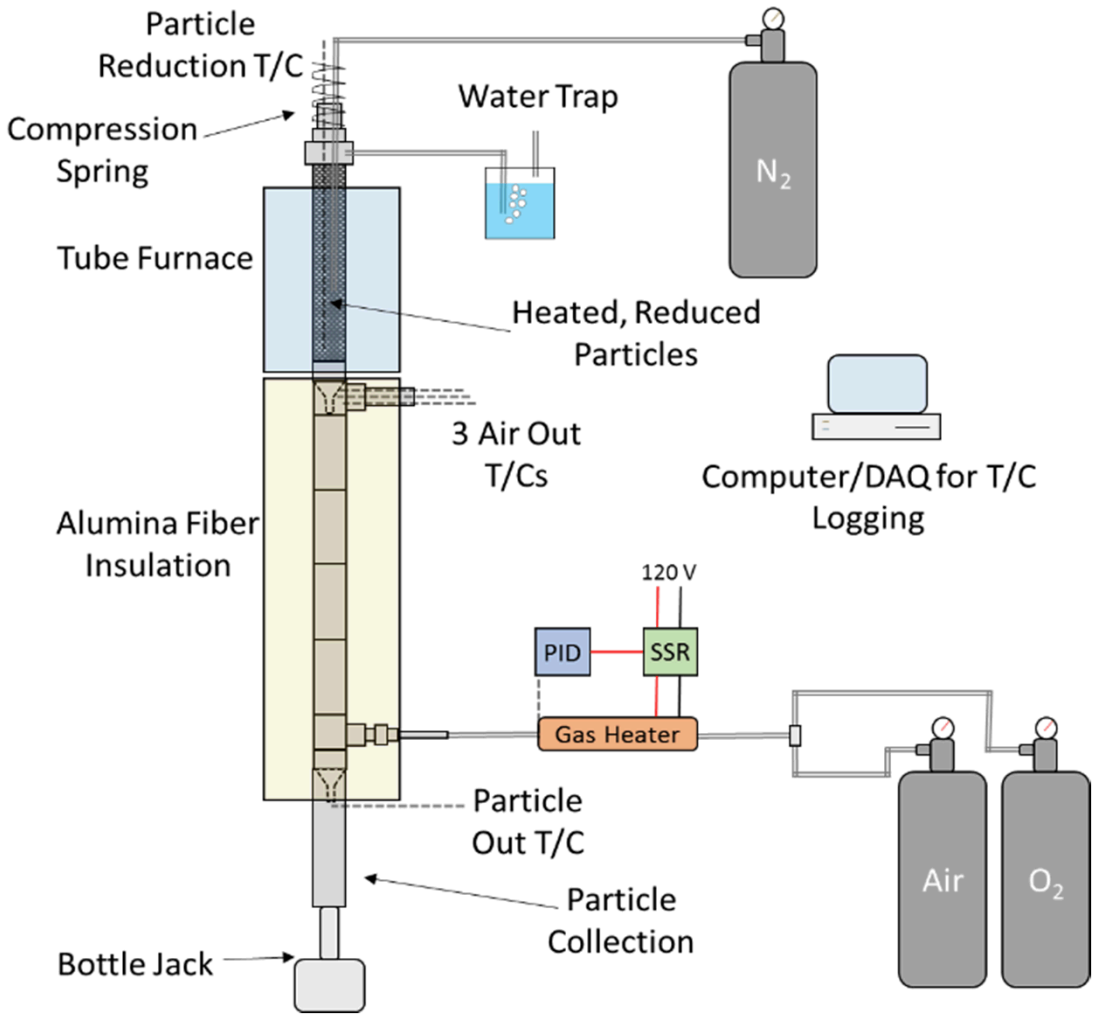
$$\varepsilon = \frac{\dot{q}}{\dot{q}_{max}} = \frac{\dot{m}_f \bar{c}_{p,f}\left(T_{f,in} - T_{f,out}\right)}{C_{min}\left(T_{p,in} - T_{f,in}\right) + \dot{m}_p \Delta H_{rxn}}$$

where $C_{min}$ is the lesser value between $\dot{m}_p \bar{c}_{p,p}$ and $\dot{m}_f \bar{c}_{p,f}$.

# Bench-scale instrumentation flow diagram

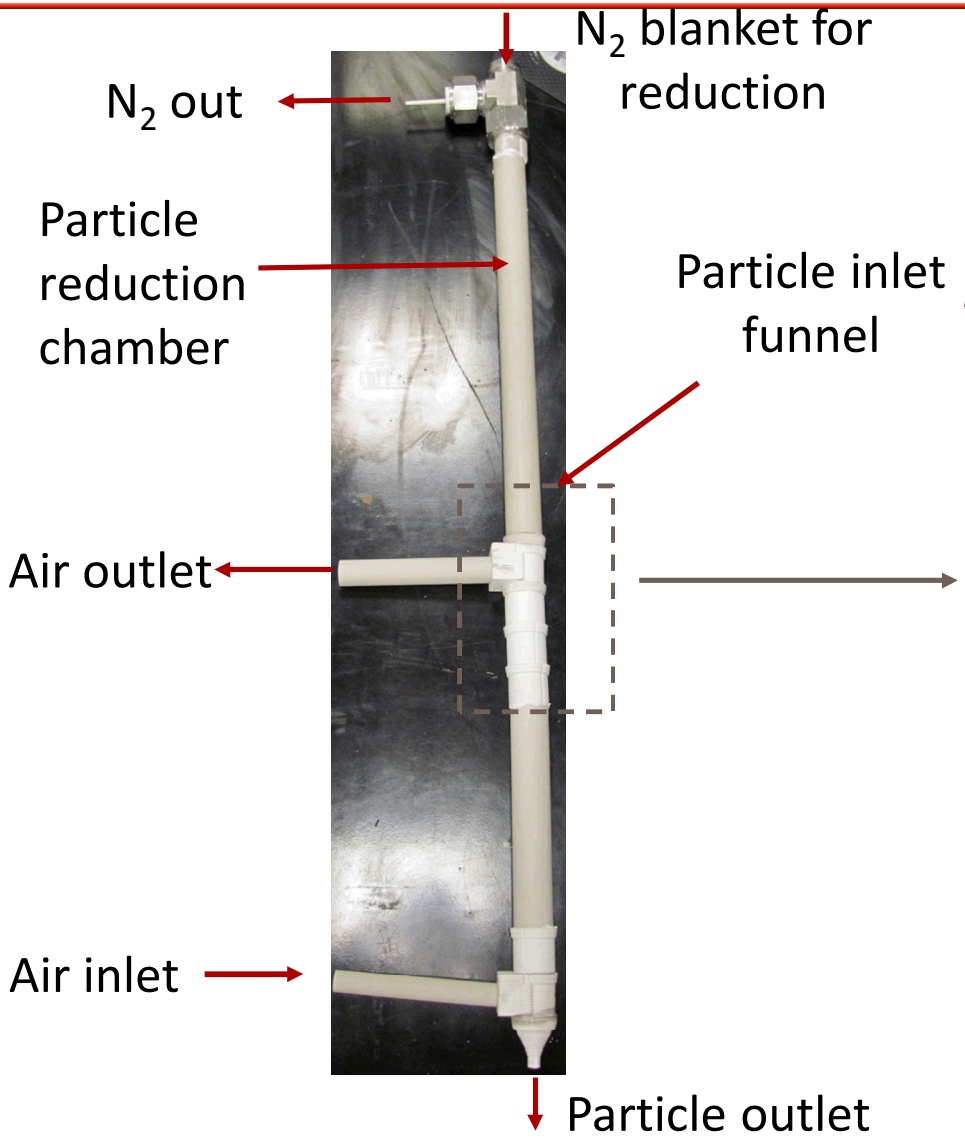
Reactor instrumentation:

- Mass flow controllers
  - Air, oxygen blend
  - Nitrogen blanket
- Inline gas heater
  - Simulates compressor preheat
- Thermocouples
  - 3 fluid outlet
  - 1 particle outlet
  - 1 particle reduction
- DAQ for T/C logging

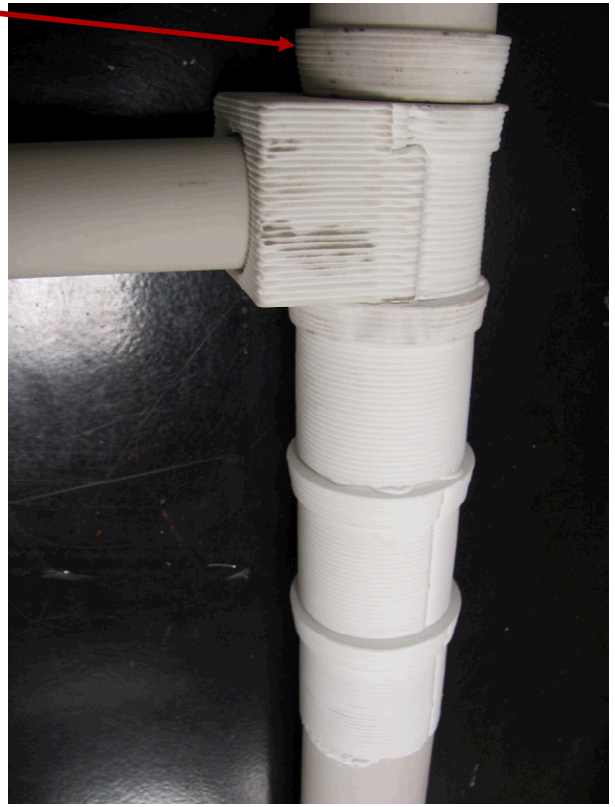



One “floating” thermocouple has been added to probe insulation temperature in multiple spots to provide a heat loss estimate

# Mock-up of ROx reactor




Detail of Robocasting segments



# Future work – data collection for model validation

| $L_{reactor}$ (fluid in to fluid out) | $\dot{m}_p/\dot{m}_f$ = 1.0 | $\dot{m}_p/\dot{m}_f$ = 1.5 | $\dot{m}_p/\dot{m}_f$ = 2.0 |
|---|---|---|---|
| 20 inches | Inert<br>$pO_2$ = 0.21 atm<br>$pO_2$ = 0.9 atm | Inert<br>$pO_2$ = 0.21 atm<br>$pO_2$ = 0.9 atm | $pO_2$ = 0.21 atm<br>$pO_2$ = 0.9 atm |
| 14 inches | | Inert<br>$pO_2$ = 0.21 atm | |

- Model validation approach
  - Validate and modify particle/fluid heat transfer coefficients using data from the inert test (at two different lengths)
  - Infer particle oxidation rate using the performance difference between $pO_2$ runs of 0.21 atm and 0.9 atm
- Use validated model to scale geometry to 100 $kW_{th}$ reactor

# Acknowledgements

This work is supported by the U.S. Department of Energy, SunShot Initiative, under award number DE-FOA-0000805.

Special thanks to SNL contributors

Dr. Jim Miller, project PI (1815)

Dr. Andrea Ambrosini, materials lead (6124)

Dr. Eric Coker, thermal measurements lead (1815)

Thank you for your attention!

Questions?

# Supplementary slides

C:\Users\smbabin\Fluent models\5kw Rox_files\dp0\FFF-14\Fluent\Verified_Transient_2016_07_26.c 1

```
/* Sean Babiniec */
/*UDF for oxidation*/
/*March 16,| 2016*/
/*Species conservation added*/
#include "udf.h"

/* Constants:*/
//#define BETA  /*Progress factor towards equilibrium*/
#define FtPfactor 0.8 /*Heat applied to fluid/heat applied to particle*/
#define m 0.000332 /*fitting factors for delta vs temp at PR 16.7*/
#define b -0.34919
float DDELT;

float Dnew;
float Dold;
float mdot;
float molC2change;
float molC2old;
float molC2new;
float BETA;

/*Source term in units of kg/m^3. Equation: Source = H(delta new) - H(delta old))/current_timestep*/

/*Execute at end to store new delta value in memory as old delta value*/
DEFINE_EXECUTE_AT_END(Delta)
{
    BETA = 12. / (1. / CURRENT_TIMESTEP);
    Domain *d;
    Domain *dom;
    Thread *t;
    cell_t c;
    d = Get_Domain(3);   /* particle domain*/
    dom = Get_Domain(2); /*fluid domain*/
    float eqdelt;
    thread_loop_c(t, d) /*Loop through particle threads and adjust mass fraction*/
    {
        if (FLUID_THREAD_P(t))
        {
            begin_c_loop(c, t)
            {
                Dold = C_YI(c, t, 0);
                C_UDMI(c,t,0) = C_VOLUME(c, t)*C_R(c, t)*C_VOF(c, t) / 0.1358*(Dold - C_UDMI(c,t,3)) /
    2.0; /*May need to write this to C_UDMI(c,t,0) to make sure there is a value for each cell passed to
    the next loop*/
                C_YI(c, t, 0) = C_UDMI(c, t, 3);
            end_c_loop(c, t)
            }

        }
    }

    thread_loop_c(t, dom) /*Loop through fluid threads and adjust mass fraction*/
    {
        begin_c_loop(c, t)
        {
            molO2old = C_YI(c, t, 0)*C_R(c, t)*C_VOLUME(c, t)*C_VOF(c, t) / 0.036;
            molO2new = molC2old - C_UDMI(c, t, 0);
            if (molC2new < 0.0)
            {
                molC2new = 0.0;
            }
            else
            {
                molC2new = molC2old - C_UDMI(c, t, 0);
            }
            C_YI(c, t, 0) = (molC2new*.036) / (C_R(c, t)*C_VOLUME(c, t)*C_VOF(c, t));
```

p. 1

p. 2

p. 3

C:\Users\smbabin\Fluent models\5kw Rox_files\dp0\FFF-14\Fluent\Verified_Transient_2016_07_26.c 2

```
            /*printf("%f", molC2change/molO2old);*/
            end_c_loop(c, t)
        }
    }
}


DEFINE_ADJUST(pO2, d)
{
    Thread *t;
    cell_t c;
    d = Get_Domain(2); /*fluid domain*/

    thread_loop_c(t, d)
    {
        begin_c_loop(c, t)
            C_UDMI(c, t, 2) = C_YI(c, t, 0);
        end_c_loop(c, t)
    }



}
DEFINE_SOURCE(particle, cell, thread, dS, eqn)
{
    real source;
    real face;
    BETA = 12. / (1. / CURRENT_TIMESTEP);
    float eqdelt;

    if (C_UDMI(cell, thread, 2) > 0.08)
    {


        Dold = C_YI(cell, thread, 0);

        eqdelt = m*C_T(cell, thread) + b;
        if (eqdelt < 0.0)
        {
            eqdelt = 0.0;
        }
        else
        {
            eqdelt = m*C_T(cell, thread) + b;
        }

        Dnew = Dold - BETA*(Dold - eqdelt);
        if (Dnew < 0.0)
        {
            Dnew = 0.0;
        }
        else
        {
            Dnew = Dold - BETA*(Dold - eqdelt);
        }
        C_UDMI(cell, thread, 3) = Dnew;
        C_UDMI(cell, thread, 1) = 1000.0 * ((873.68*Dold*Dold + 863.26*Dold) - (873.68*Dnew*Dnew + 863.26*
    Dnew)) *C_VOF(cell, thread)*C_R(cell, thread)*C_VOLUME(cell, thread) / (C_VOLUME(cell, thread)*
    CURRENT_TIMESTEP);
        source = C_UDMI(cell, thread, 1) *(1. - FtPfactor);
        dS[eqn] = 0.; //-C_UDMI(cell, thread, 1) *(1. - FtPfactor) / (CURRENT_TIMESTEP*CURRENT_TIMESTEP);
    }
    else
    {
        C_UDMI(cell, thread, 3) = C_YI(cell, thread, 0);
```

C:\Users\smbabin\Fluent models\5kw Rox_files\dp0\FFF-14\Fluent\Verified_Transient_2016_07_26.c 3

```
        source = 0.0;
        dS[eqn] = 0.0;
    }


        //printf("part %f", source);
    return source;

}

DEFINE_SOURCE(fluid, cell, thread, dS, eqn)
{
    real source;

        source = C_UDMI(cell, thread, 1) *FtPfactor;
        dS[eqn] = 0.; //-C_UDMI(cell, thread, 1) *(FtPfactor) / (CURRENT_TIMESTEP*CURRENT_TIMESTEP);

        //printf("fluid %f", source);
    return source;

}
```