Development of Automated Calibration Tools for Determining Fire Model Input Parameters

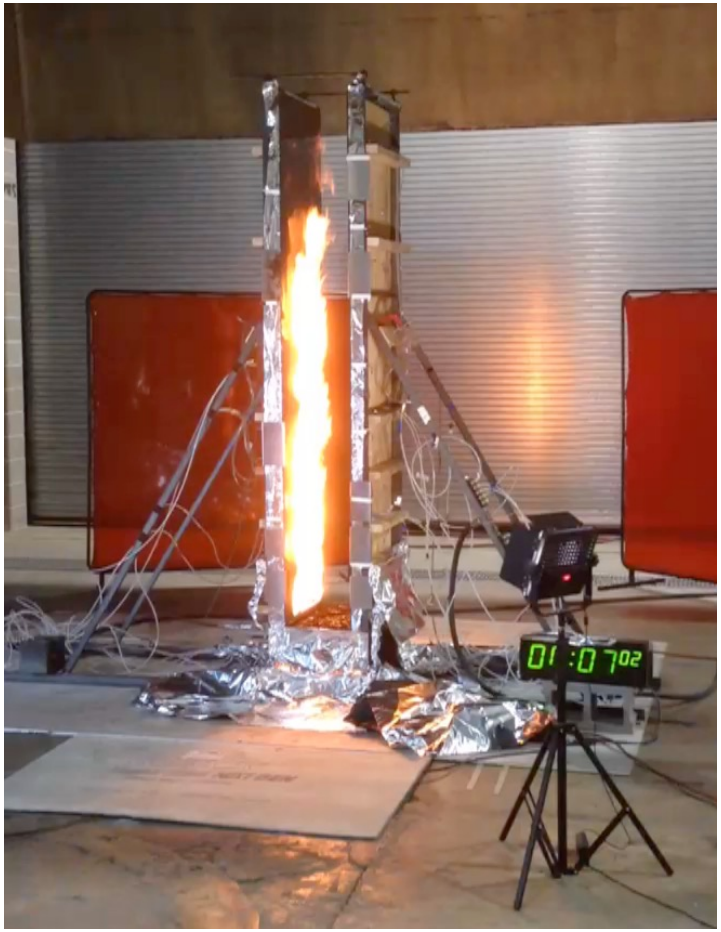
June 5, 2022

Morgan C. Bruns
St. Mary's University (USA)

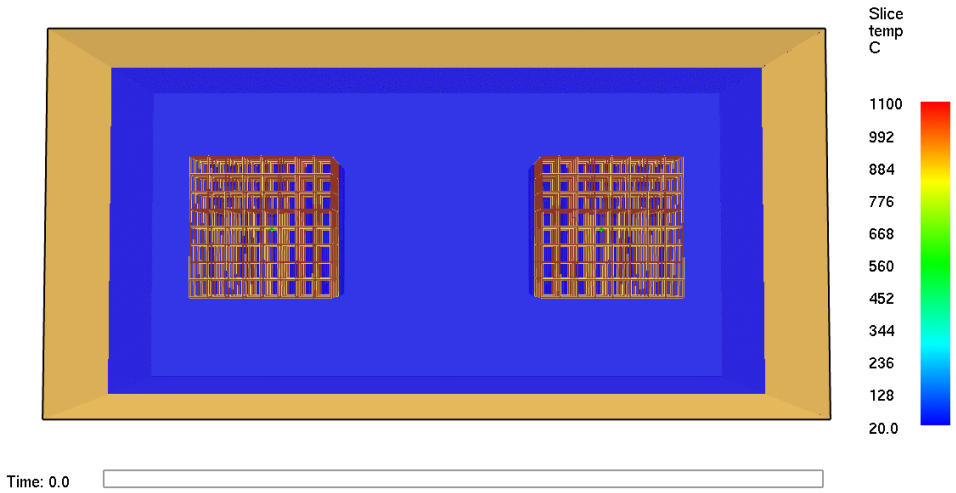
Isaac T. Leventon
National Institute of Standards and Technology (USA)

Predicting Fire Growth

- To engineer safer:
  - Buildings
  - Products
  - Materials
- Accurate predictions require **condensed phase pyrolysis models**



Condensed Phase Challenges



- Physics
  - Multiphase
  - Mechanical deformation
- Numerics
  - Gas phase coupling
  - Multiscale
  - Moving boundary
- **Materials**
  - Many parameters
  - Many materials

How Many Parameters?

$$\frac{\partial \rho_i}{\partial t} = \dot{m}_i''', \quad i = 1, \ldots N$$

$$\rho c \frac{\partial T}{\partial T} = \nabla \cdot (k \nabla T) + \dot{q}'''$$

$$\dot{m}_i''' = -A_i \rho_i \exp\left(-\frac{E_i}{RT}\right), \quad i = 1, \ldots, N$$

$$\rho c = \sum_{i=1}^{N} \rho_i c_i$$

$$k = f\left(\rho_1, \ldots, \rho_N, k_1, \ldots, k_N\right)$$

$$\dot{q}''' = -\sum_{i=1}^{N} \Delta h_i \dot{m}_i'''$$

$$\rho_i\left(t = 0\right) = \rho_{0,i}, \quad i = 1, \ldots, N$$

Neglecting
- Radiation
- Mass transport
- Charring
- Temperature dependence

At least ~$6N$ material property parameters need to be quantified

How many materials?

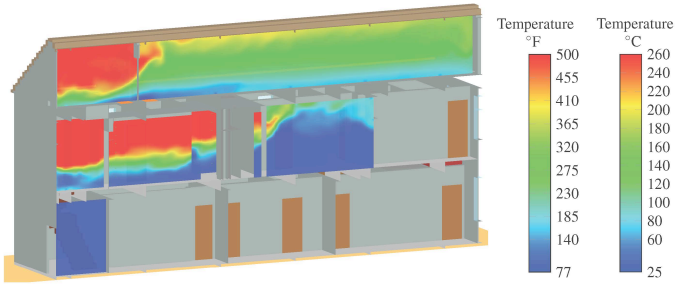
- NFIRS categorizes **38 distinct "types"** of solid materials "First Ignited"
- These "types" are extremely broad categories such as "Plastic", "Rubber", and "Plywood"
- For example, Lyon and Janssens (2005) contains data on **50 common plastics**
- Additional diversity due to processing variability, additives, blends, ageing, etc.

On the order of 10^3 distinct materials relevant to fire growth predictions

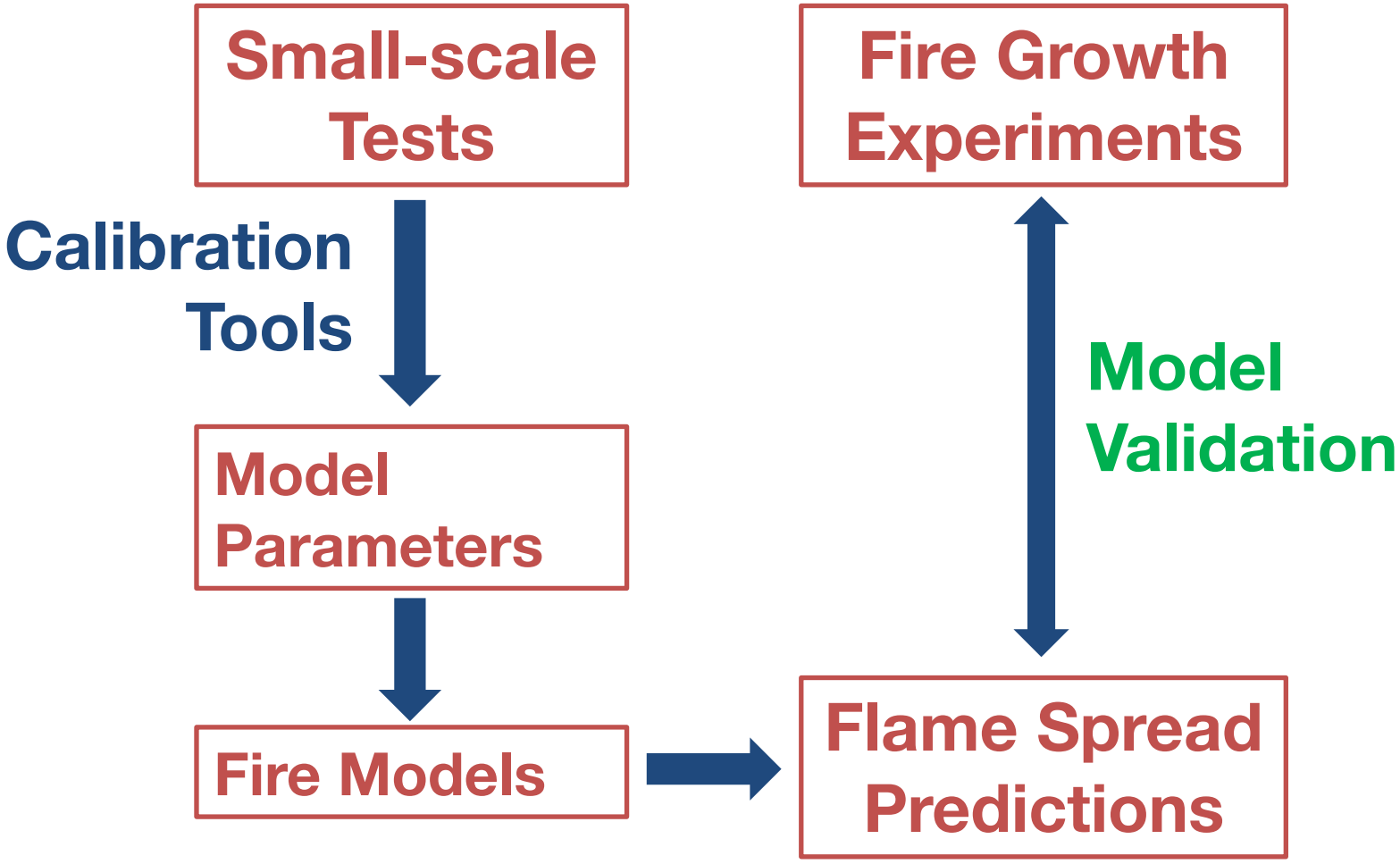
$(10^1$ Parameters) × $(10^3$ Materials)
= **10^4 Properties for reliable fire growth predictions**

A **Material Property Database** is needed

Material Property Database



- See previous presentation…
- Adopting “Hierarchical” approach
- Critical components
  1. Standard formatting
  2. Standard metadata
  3. **Calibration tools**



Quality of **calibration tools** is ultimately determined in **model validation**—*do the parameters predict fire growth?*

Global Approach

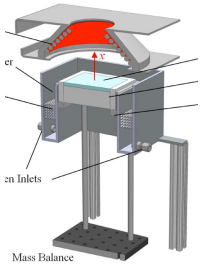
$$A_i, E_i, c_i, k_i, \Delta h_i, \rho_{0,i}$$

Fire Model

Hierarchical Approach

$$A_i, E_i, \rho_{0,i}$$

$$c_i, \Delta h_i$$



$$k_i$$

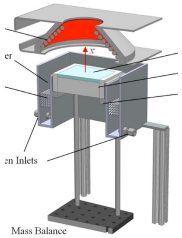
Micro Combustion Calorimeter

$\Delta h_{c,i}$

Thermogravimetric Analysis

$A_i, E_i, \rho_{0,i}$

Controlled Atmosphere Pyrolysis



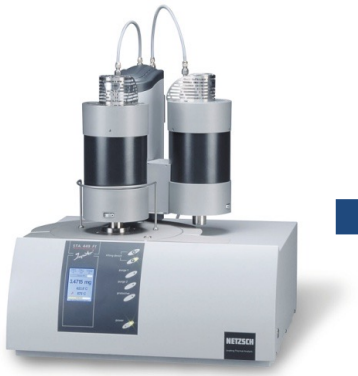
k_i

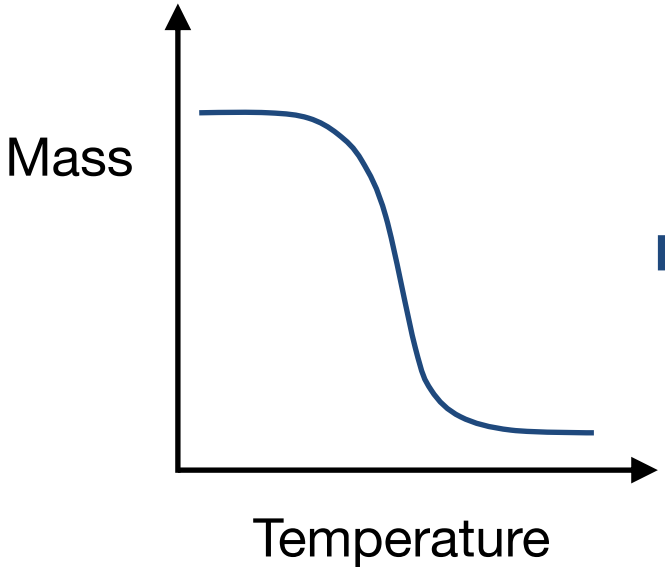
Differential Scanning Calorimetry

$c_i, \Delta h_i$

Analysis of TGA Data

How should we estimate pyrolysis kinetic parameters from raw TGA data?



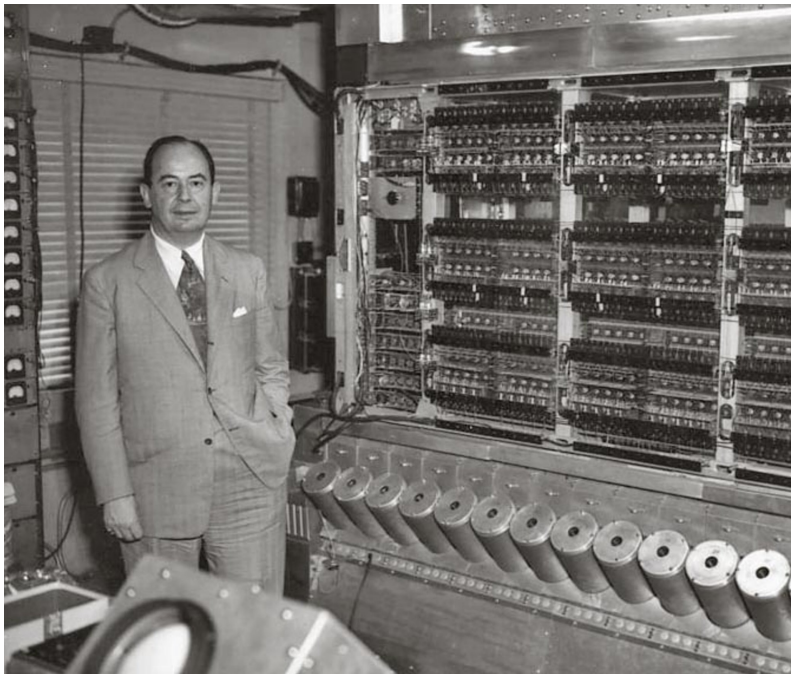


$A_i, E_i, \rho_{0,i}$

Method Requirements

1. Parameters predict data → **Accurate**
2. Many different materials → **Efficient**
3. Many different behaviors→ **Robust**
4. Parameters do not vary → **Consistent**

How to be Consistent



"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

~John von Neumann

1. **No free parameters**
2. **No random numbers**

For given data, the method should always produce the same parameters

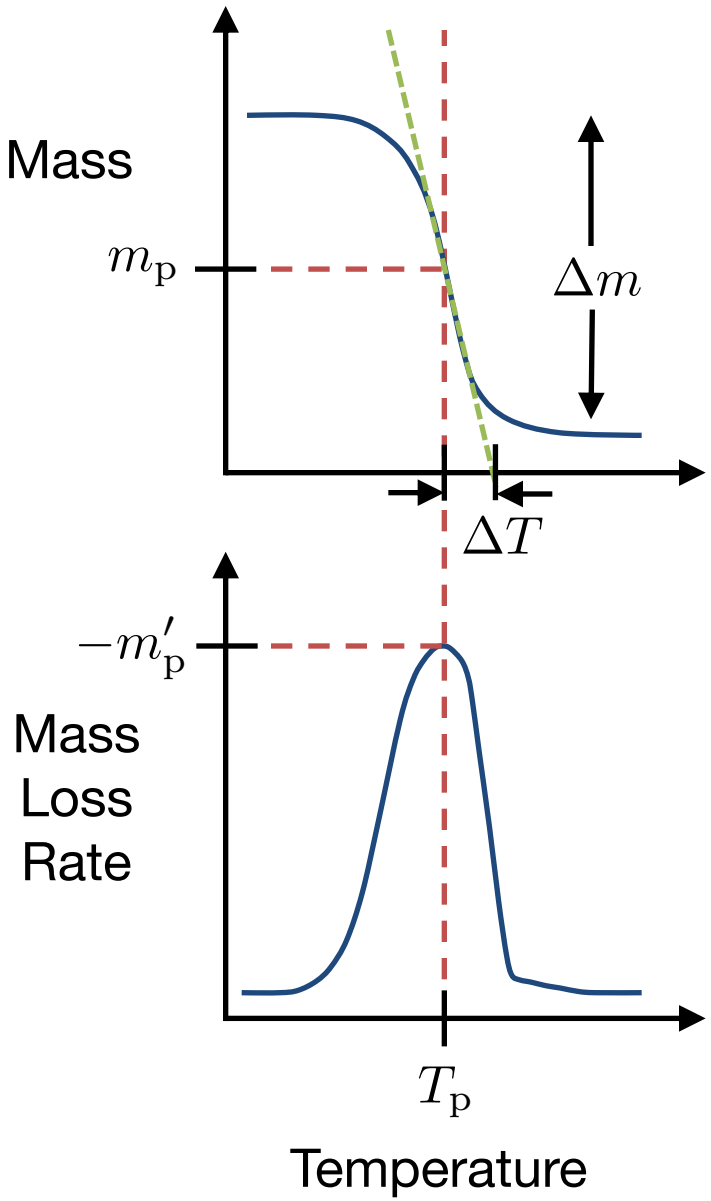
Pyrolysis Model: Independent First-Order Reactions

$$\text{Reactant} \xrightarrow{k} \nu\text{Char} + (1-\nu)\,\text{Gas}$$

$$k = \left(\frac{A}{\beta}\right)\exp\left(-\frac{E}{RT}\right)$$

$$m' \equiv \frac{\mathrm{d}m}{\mathrm{d}T} = -(1-\nu)\,km, \quad m(T_0) = m_0$$

Appropriateness to be determined by ability to predict fire growth.



Characteristic temperature and mass changes:

$$\Delta T \equiv \frac{m_\mathrm{p}}{-m'_\mathrm{p}}$$

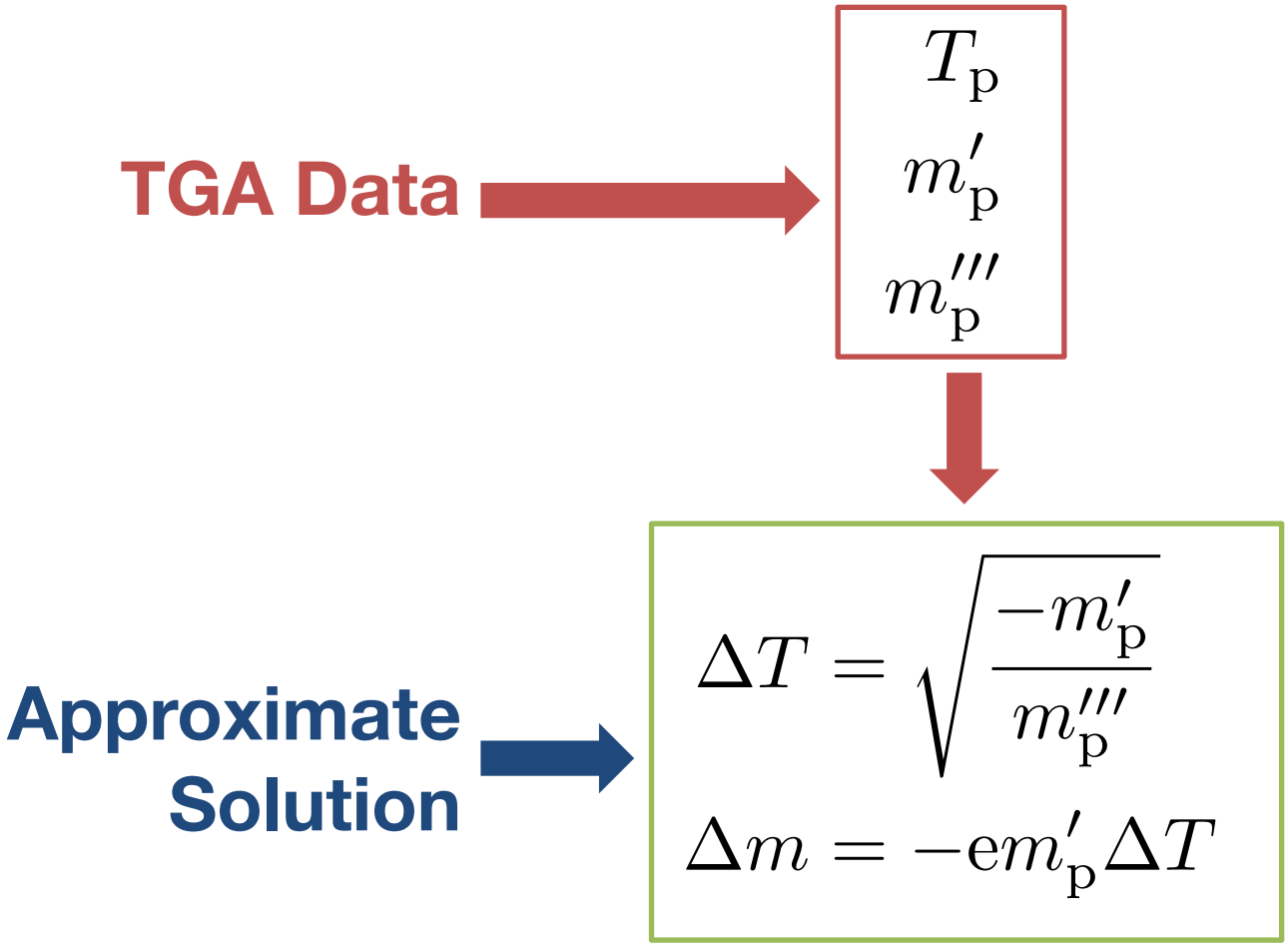
$$\Delta m \equiv m_0 \left(1 - \nu\right)$$

Analysis of peak condition yields:

$$E = \frac{RT_\mathrm{p}^2}{\Delta T}$$

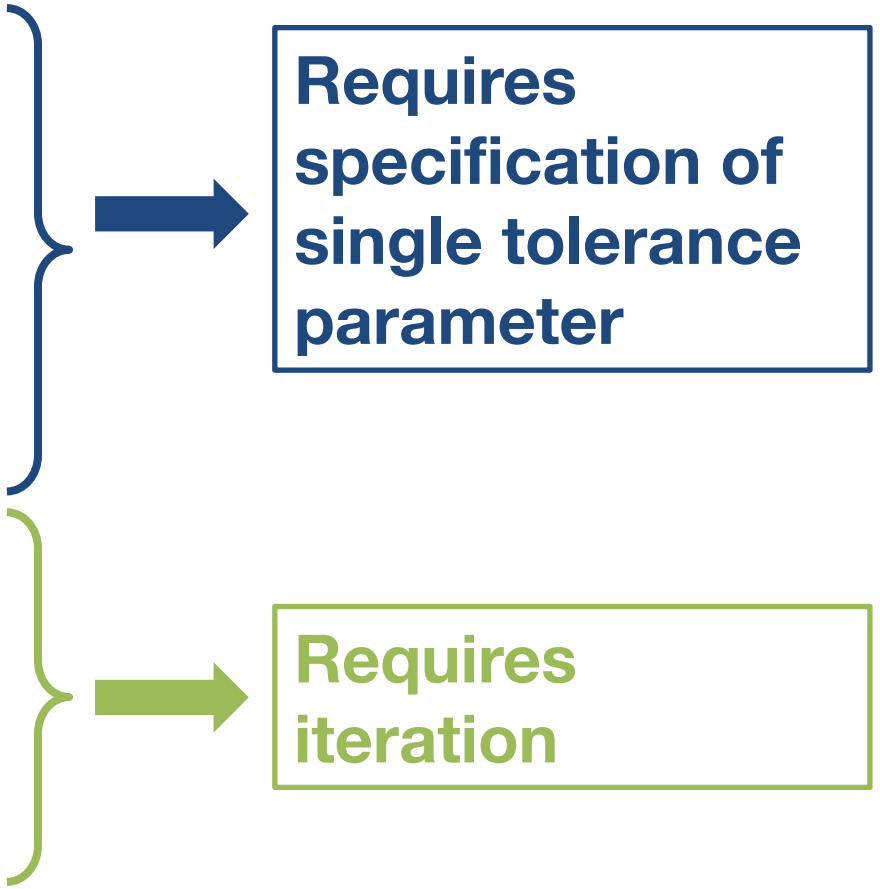
$$A = \frac{\beta}{\Delta T} \exp\left(\frac{T_\mathrm{p}}{\Delta T}\right)$$

Estimating Parameters



Some Details

1. Smoothed data derivatives are found using Savitzky-Golay filter
2. “Small” mass loss rate peaks are neglected
3. Algorithm corrects for overlapping reactions
4. Mass changes corrected to conserve mass



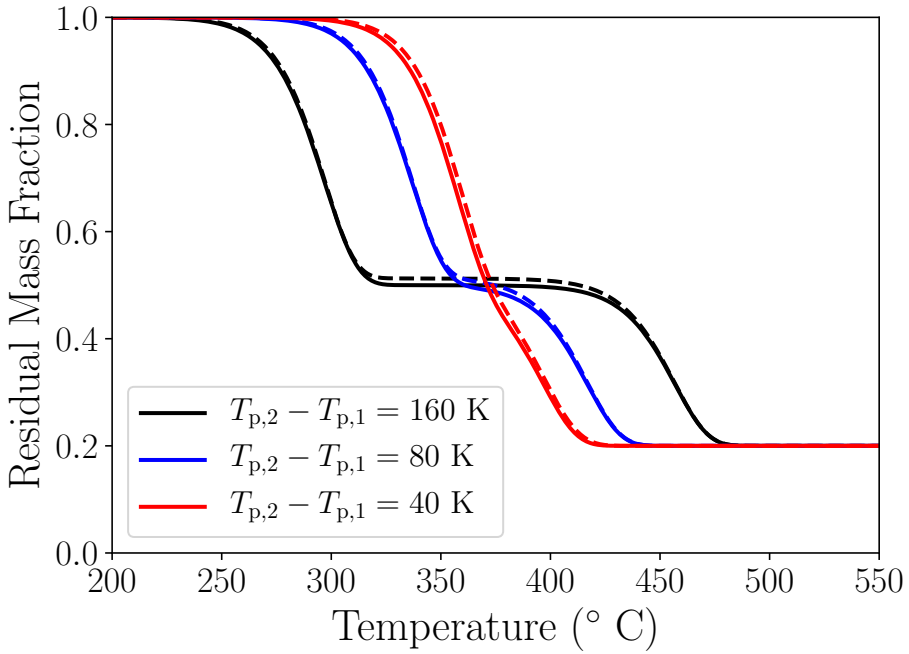
TGA Fit Verification

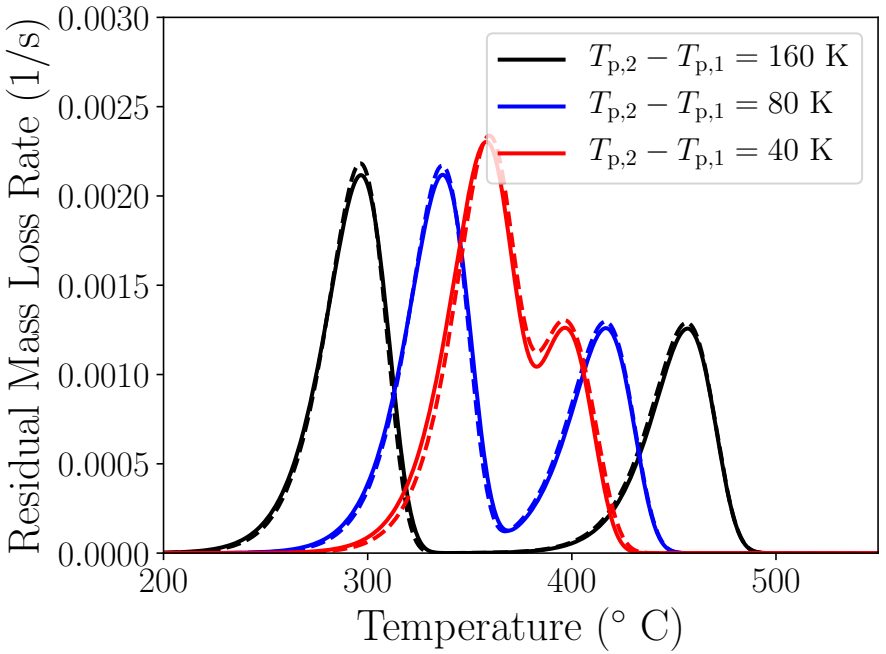
1. Assume **kinetic parameters**
2. Generate TGA data
3. Use algorithm to find assumed parameters

Purpose:

1. Check implementation
2. Test validity of approximate solution

Two Reactions Verification





Closer fit for more separated reactions

Validation

Purpose:

1. Test algorithm with real TGA data

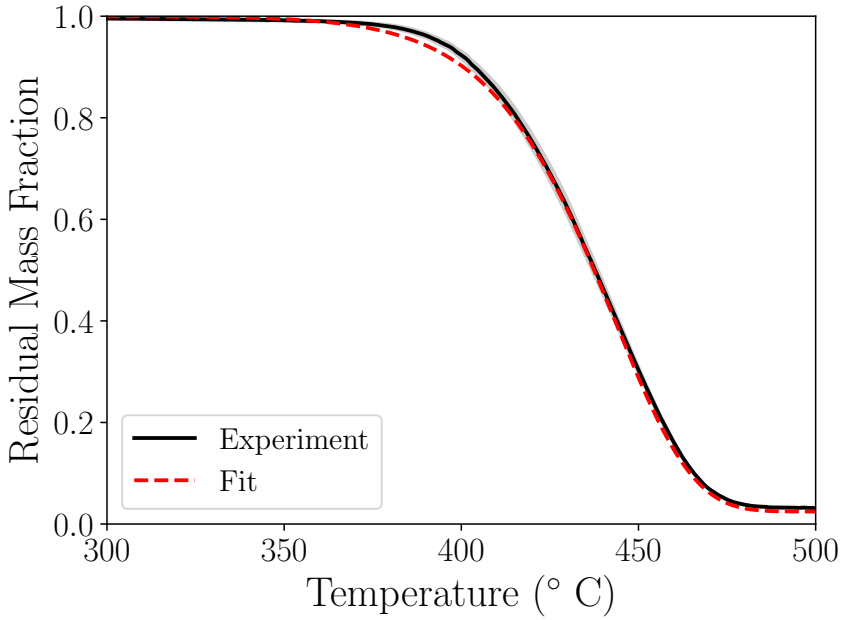
Test Materials:

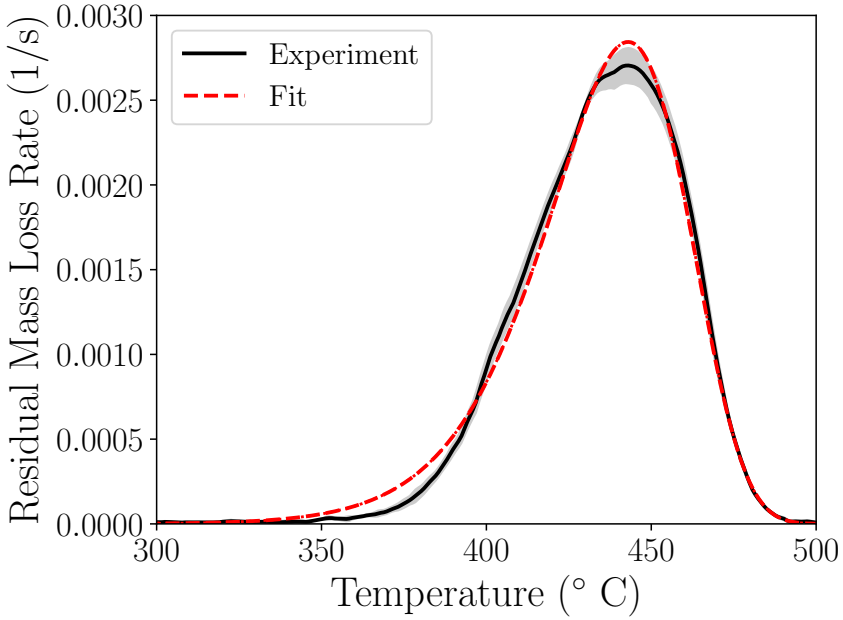
1. Nylon 6,6
2. Flexible polyurethane (PU) foam
3. Polyvinyl Chloride (PVC)

Procedure:

- In nitrogen
- Samples: 3-5.5 mg
- Isothermal heating for 20-30 min
- Dynamic heating at 10 K/min

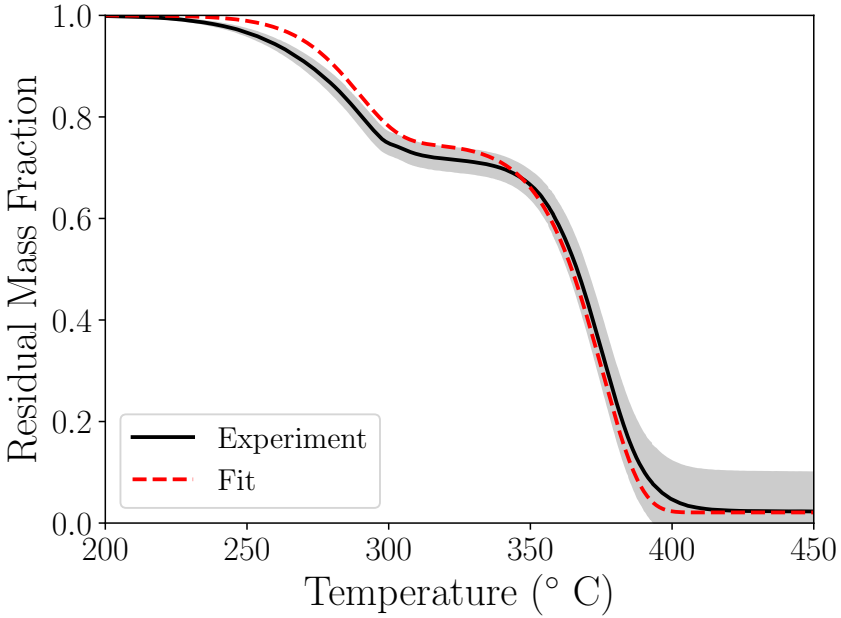
Validation: Nylon 6,6

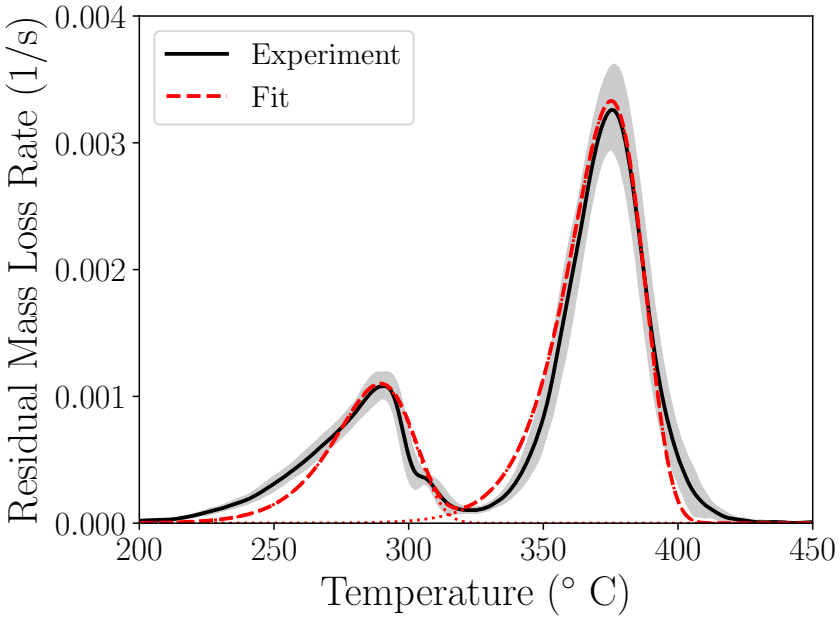




Kinetic Parameter	Reaction 1
T_p (K)	716.3
ΔT (K)	22.11
Δm	0.9754
ξ	0.03087
$\ln[A\ (s^{-1})]$	27.50
E (kJ/kmol)	192.9×10^3

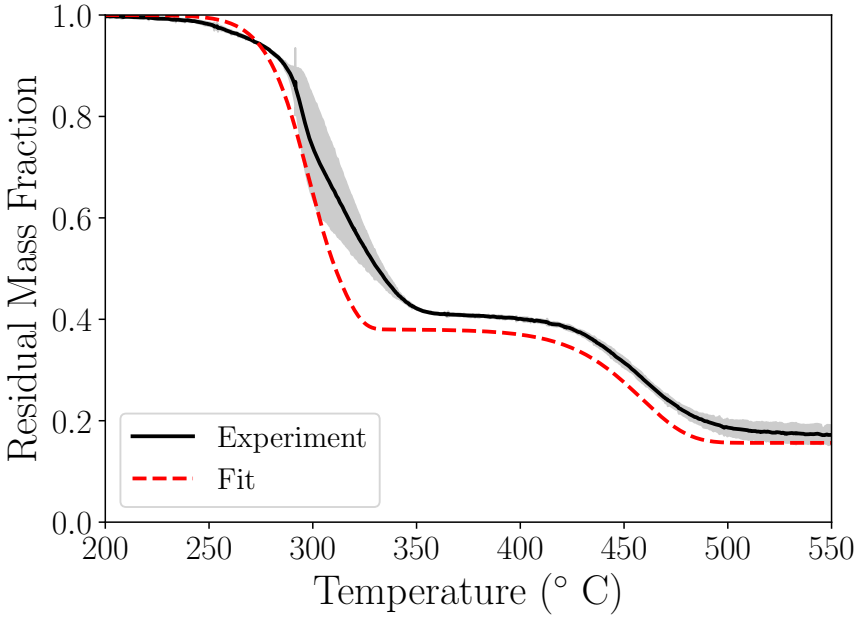
Validation: Polyurethane Foam

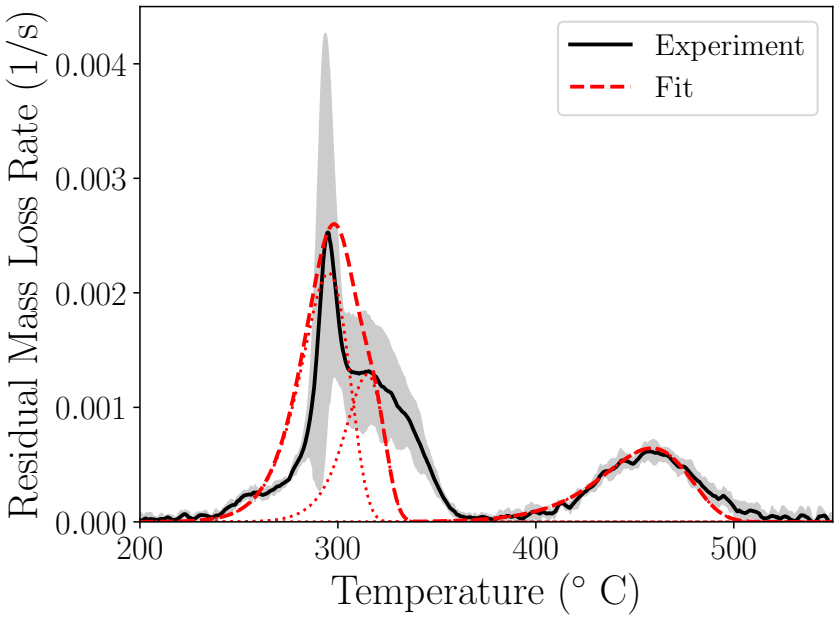




Kinetic Parameter	Reaction 1	Reaction 2
T_p (K)	562.7	648.5
ΔT (K)	14.50	13.69
Δm	0.2511	0.7280
ξ	0.02577	0.02112
$\ln[A\ (s^{-1})]$	34.34	42.95
E (kJ/kmol)	181.5×10^3	255.3×10^3

Validation: PVC



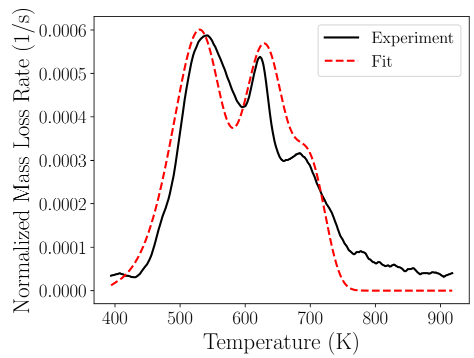


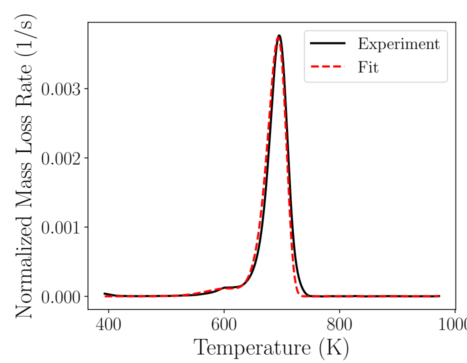
Kinetic Parameter	Reaction 1	Reaction 2	Reaction 3
T_p (K)	568.5	731.7	588.1
ΔT (K)	12.15	22.39	9.62
Δm	0.4200	0.2238	0.1999
ξ	0.02138	0.03060	0.01636
$\ln[A\ (s^{-1})]$	42.49	27.78	57.06
E (kJ/kmol)	221.1×10^3	198.8×10^3	298.8×10^3

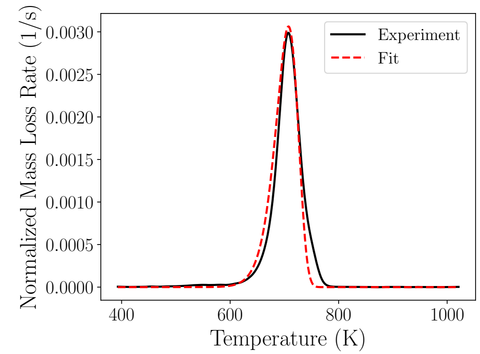
Validation Summary

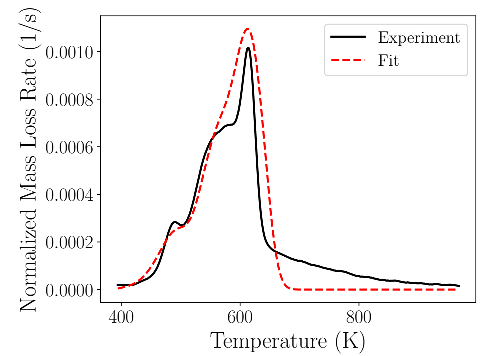
25 materials:

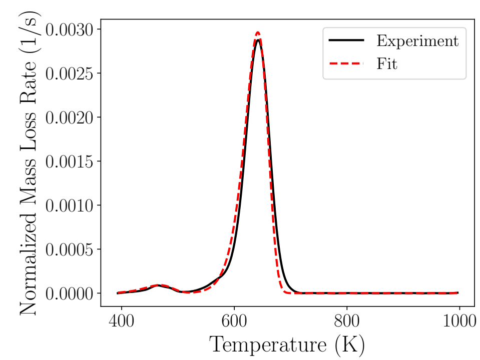
```
# list of all current TGA test series
tga_tests = [
    'ABS_Expt_TGA_N2_10K',
    'BigBerry-Leaf_Expt_TGA_N2_10K',
    'BigBerry-Stem_Expt_TGA_N2_10K',
    'Chamise-Leaf_Expt_TGA_N2_10K',
    'Chamise-Stem_Expt_TGA_N2_10K',
    'Chaparral-Leaf_Expt_TGA_N2_10K',
    'Chaparral-Stem_Expt_TGA_N2_10K',
    'DesertCeanothus-Leaf_Expt_TGA_N2_10K',
    'DesertCeanothus-Stem_Expt_TGA_N2_10K',
    'DouglasFir-Leaf_Expt_TGA_N2_10K',
    'HDPE_Expt_TGA_N2_10K',
    'HIPS_Expt_TGA_N2_10K',
    'Kydex_Expt_TGA_N2_10K',
    'LodgepolePine-Leaf_Expt_TGA_N2_10K',
    'LodgepolePine-Stem_Expt_TGA_N2_10K',
    'MaCFP_PMMA_Expt_TGA_N2_10K',
    'PolyIso1_Expt_TGA_N2_10K',
    'PolyIso2_Expt_TGA_N2_10K',
    'PolyIso05_Expt_TGA_N2_10K',
    'POMGF_Expt_TGA_N2_10K',
    'XPSgreen_Expt_TGA_N2_10K',
    'XPSpink_Expt_TGA_N2_10K'
]
```



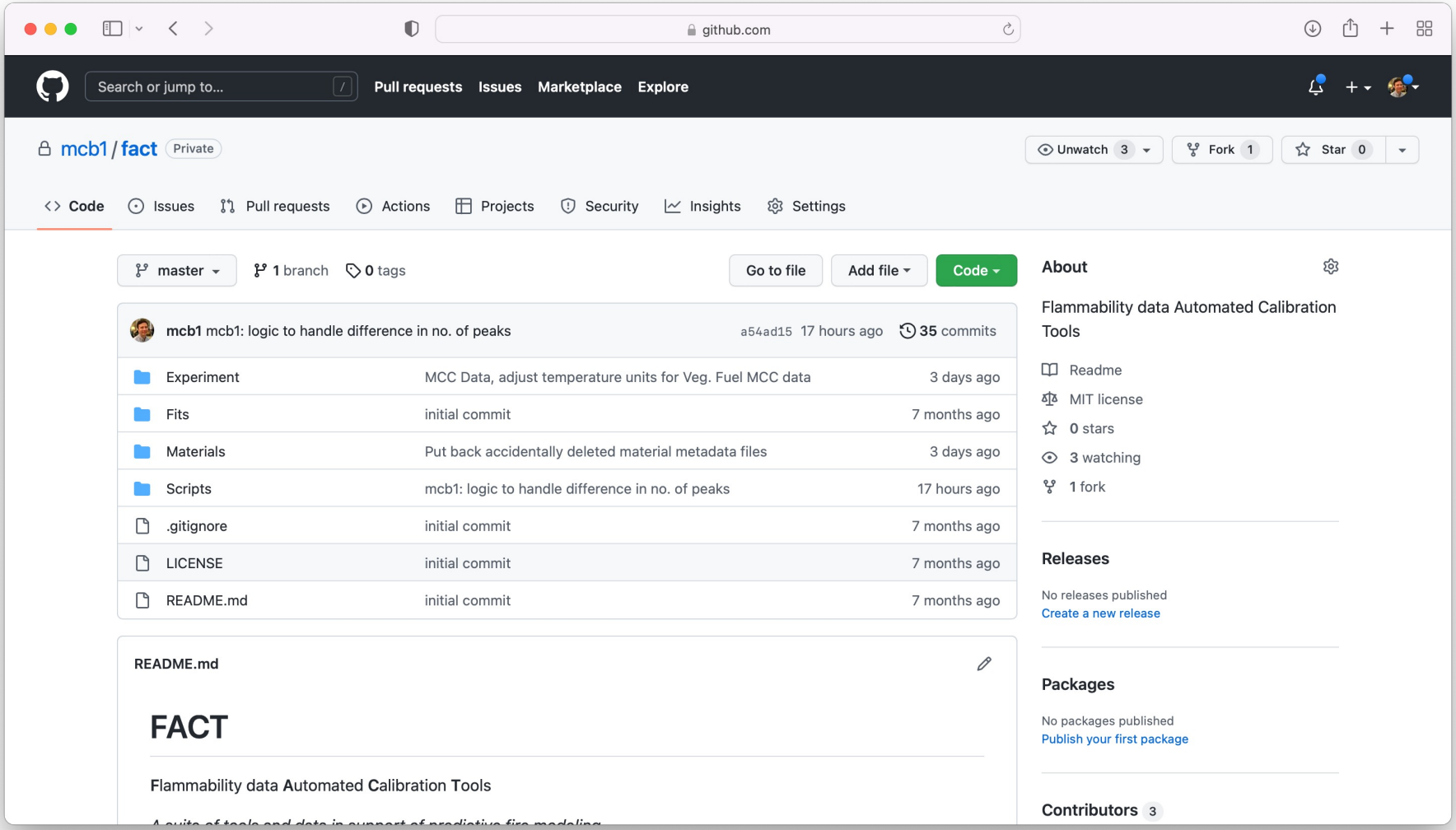




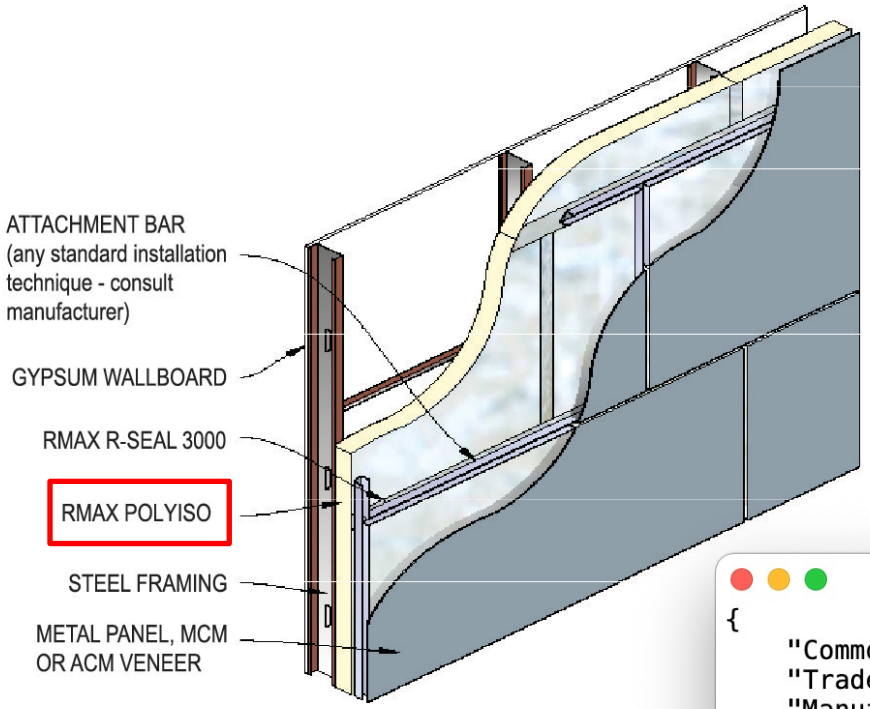




FACT: **F**lammability data **A**utomated **C**alibration **T**ools



Example: Polyisocyanurate Board Insulation



www.rmax.com

`fact/Materials/metadata/PolyIso2.json`

PolyIso2.json

```
{
    "Common Names": ["Poly(Isocyanurate) foam", "PolyIso foam"],
    "Trade Name": "Thermasheath",
    "Manufacturer": "RMAX",
    "Distributor": "Lowe's",
    "Date Acquired": "February 2020",
    "Description": "Yellow polyiso foam core bonded to Kraft-paper-
reinforced aluminum facers on each side; both sides have a reflective
surface; nominal thickness 2in. (50.5 to 51.7 mm)",
    "Categories": ["Synthetic Polymer", "Porous Foam", "Thermoset"]
}
```

Replicate TGA Data Files:

```
Time,Temperature,Mass
[s],[K],[mg]
0,300.772,5.64900711774
4.998,300.77222,5.65034706054
9.996,300.77115,5.64897096414
14.994,300.77005,5.64900768264
19.992,300.77194,5.64766717494
24.99,300.77516,5.64650687034
29.988,300.79185,5.64423088824
34.986,300.8111,5.64203116764
39.984,300.87397,5.63949985074
```

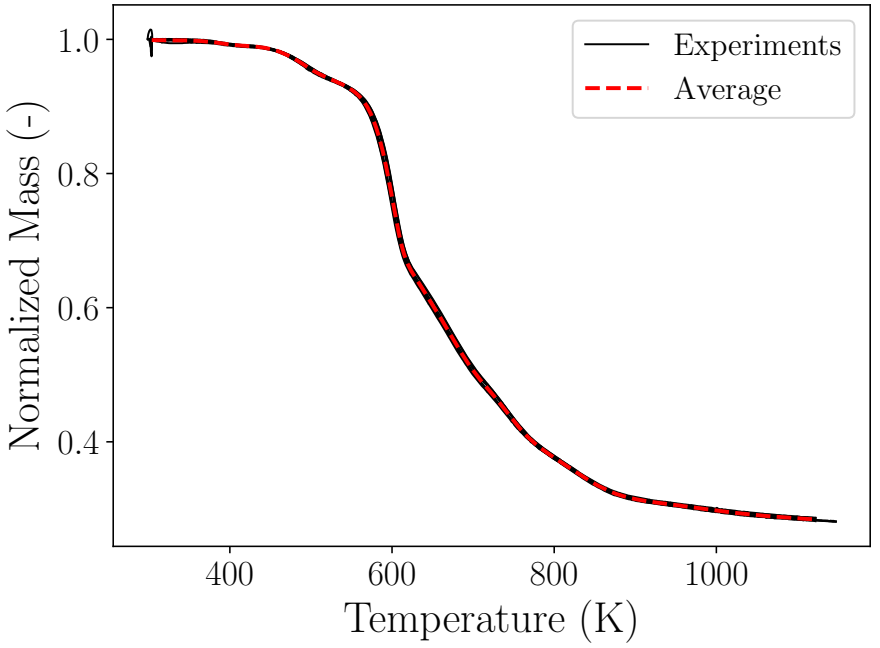
```
Time,Temperature,Mass
[s],[K],[mg]
0,303.653,4.53640016592
4.998,303.64491,4.53640016592
9.996,303.6319,4.53640016592
14.994,303.62205,4.53606994512
19.992,303.609,4.53472955712
24.99,303.60149,4.53220890192
29.988,303.591,4.52989826352
34.986,303.57958,4.52413981152
39.984,303.57796,4.51468270512
```

```
Time,Temperature,Mass
[s],[K],[mg]
0,298.818,4.88188150002
4.998,298.82011,4.88322139522
9.996,298.81941,4.88378928812
14.994,298.81898,4.88289179272
19.992,298.83343,4.88321602392
24.99,298.85833,4.88321700052
29.988,298.90316,4.88323457932
34.986,298.98718,4.88329171042
39.984,299.0697,4.88474342632
```

```
Time,Temperature,Mass
[s],[K],[mg]
0,301.008,4.23159800954
4.998,301.00631,4.23159800954
9.996,301.00344,4.23025805184
14.994,301.00297,4.23025805184
19.992,301.00235,4.23128787724
24.99,301.00844,4.22795131064
29.988,301.01631,4.22486945024
34.986,301.03529,4.22067695234
39.984,301.09055,4.21446457504
```

• • •

```
(base) Scripts $ python average_tga_series.py PolyIso2_Expt_TGA_N2_10K
============================================================

Reading data from:

   PolyIso2_Expt_TGA_N2_10K_r7.csv
   PolyIso2_Expt_TGA_N2_10K_r6.csv
   PolyIso2_Expt_TGA_N2_10K_r4.csv
   PolyIso2_Expt_TGA_N2_10K_r5.csv
   PolyIso2_Expt_TGA_N2_10K_r2.csv
   PolyIso2_Expt_TGA_N2_10K_r3.csv
   PolyIso2_Expt_TGA_N2_10K_r8.csv

Number of data sets in series =  7

============================================================
```

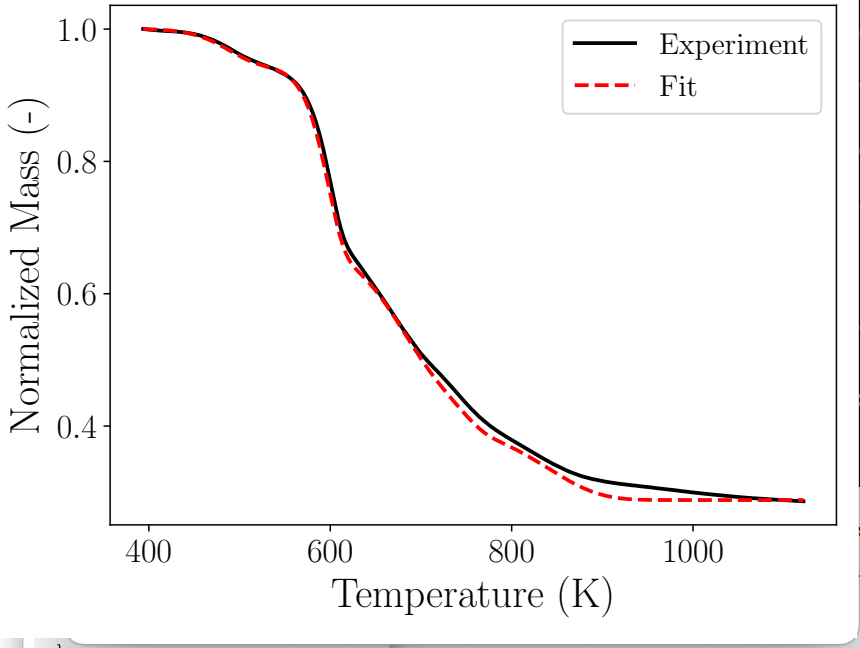


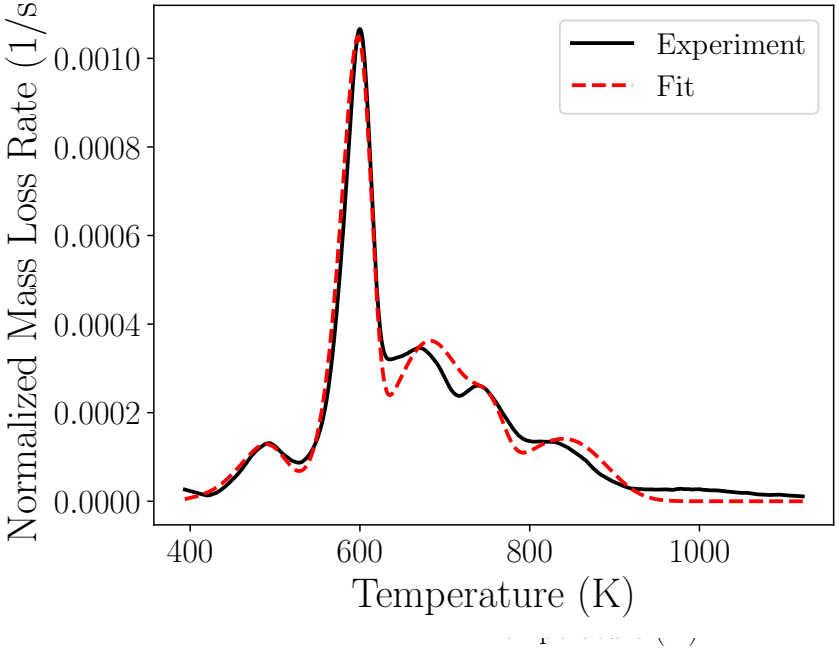
```
(base) Scripts $ python tga_fit.py PolyIso2_Expt_TGA_N2_10K_avg
```

PolyIso2_props.json


```
{
    "A": {
        "Name": "Pre-exponential Factor",
        "Units": "s^-1",
        "Values": [
            117257.20357330737,
            8420315447744.545,
            29625.162033341265,
            371893548781.6816,
            7241.320706107077
        ]
    },
    "E": {
        "Name": "Activation Energy",
        "Units": "kJ kmol^-1",
        "Values": [
```

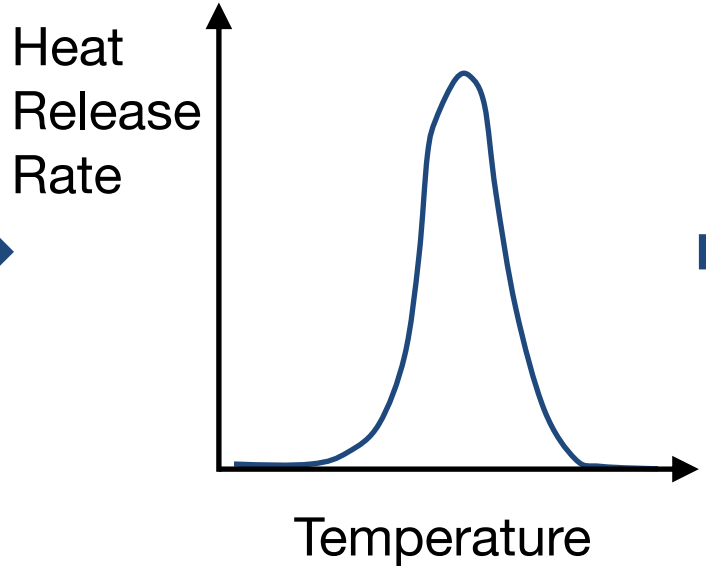

```
TGA/data/PolyIso2_Expt_TGA_N2_10K_avg.csv

42739
68e-05

 676.65 747.65 839.65]
7.36943641 42.71081662 23.66168942 57.03020686]
```





Analysis of MCC Data

How should we estimate individual reaction heats of combustion ($\Delta h_{c,i}$) from raw MCC data?



$\Delta h_{c,i}$

Single Reaction

$$\dot{Q}\left(T\right) = \dot{m}\left(T\right)\Delta h$$

1) Value at Peak Temperature: T_{p}

$$\Delta h = \frac{\dot{Q}\left(T_{\mathrm{p}}\right)}{\dot{m}\left(T_{\mathrm{p}}\right)}$$

2) Simple Average: N_{d} data points

$$\Delta h = \frac{1}{N_{\mathrm{d}}} \sum_{k=1}^{N_{\mathrm{d}}} \frac{\dot{Q}\left(T_k\right)}{\dot{m}\left(T_k\right)}$$

Multiple Reactions

$$\dot{Q}(T) = \sum_{i=1}^{N_\mathrm{r}} \dot{m}_i(T)\,\Delta h_i$$

1) Linear System: N_r equations, N_r unknowns

$$\sum_{i=1}^{N_\mathrm{r}} \dot{m}_i(T_{\mathrm{p},j})\,\Delta h_i = \dot{Q}(T_{\mathrm{p},j}),\ j = 1,\ldots,N_\mathrm{r}$$

2) Multiple Linear Regression: N_d data points

$$\sum_{i=1}^{N_\mathrm{r}} \dot{m}_i(T_k)\,\Delta h_i = \dot{Q}(T_k),\ k = 1,\ldots,N_\mathrm{d} > N_\mathrm{r}$$

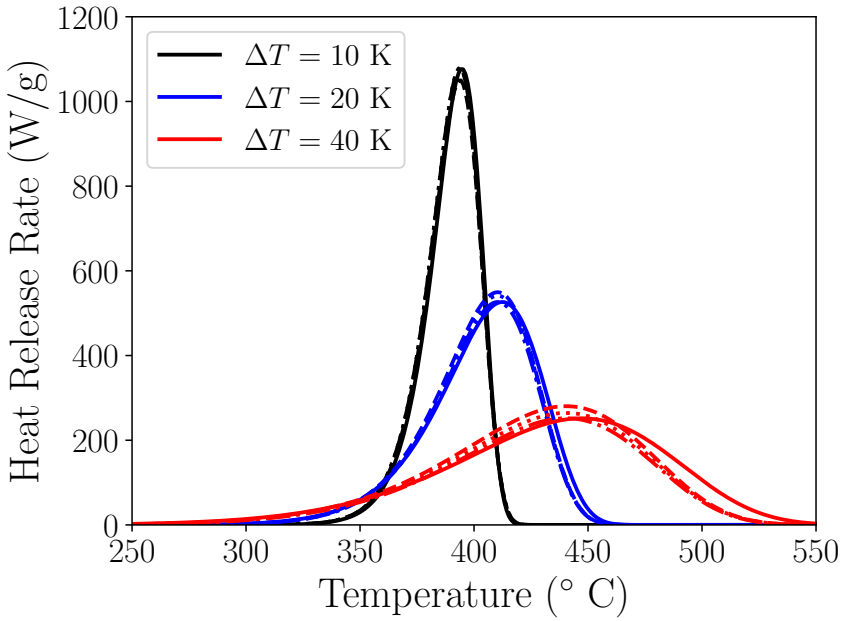
MCC Fit Verification

1. Assume **kinetic parameters** and **heats of combustion**
2. Generate TGA data
3. Use TGA fit algorithm to find kinetic parameters
4. Use TGA predictions and MCC data to find heats of combustion

Purpose:

1. Check implementation
2. Test validity of approximate solution

MCC: Single Reaction Verification



Solid: Simulated Data
Dash: Total HRR/mass
Dash-Dot: Peak Match
Dot: Simple Average

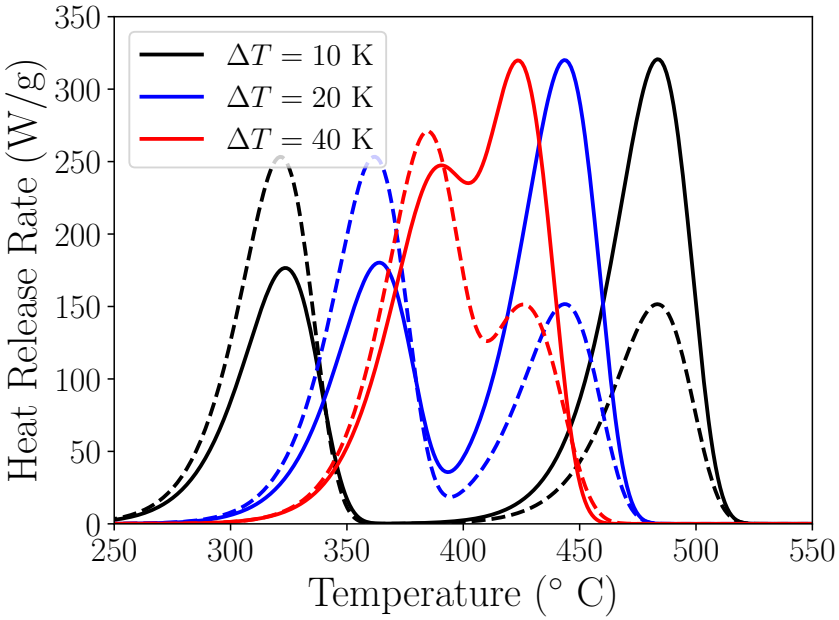
- TGA Data
  - 10 K/min
  - $T_{\mathrm{p}} = 650$ K
- MCC Data
  - 60 K/min
  - $\Delta h = 30$ kJ/g

MCC: Single Reaction Verification

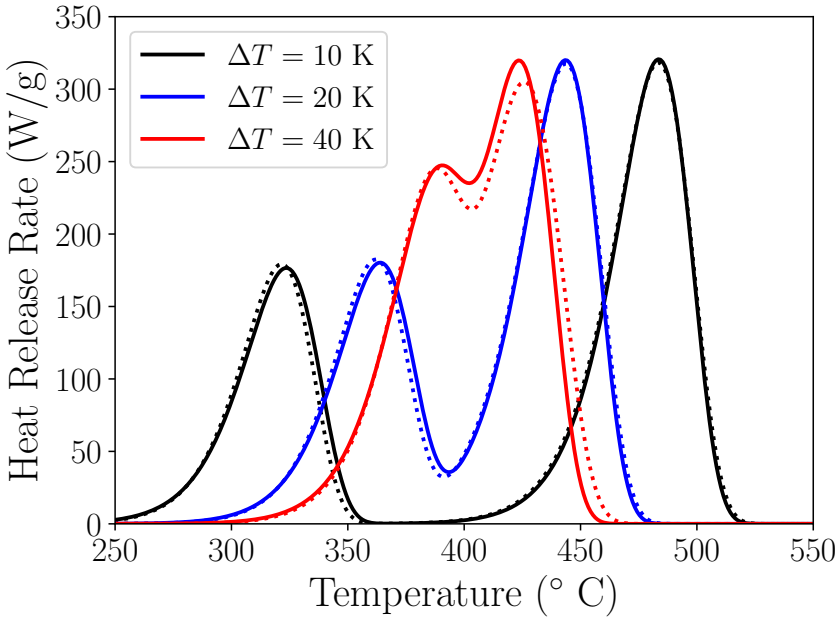
Scenario	Δh (kJ/g) (Total HR/mass)	Method 1: Peak Ratio Δh (kJ/g)	Method 2: Simple Average Δh (kJ/g)
ΔT = 10 K	30.026	30.792	30.849
ΔT = 20 K	29.997	28.731	29.556
ΔT = 40 K	29.997	26.949	28.291

- Total HR/mass not applicable to multiple reaction
- Method 2 performs better than Method 1

MCC: Two Reactions Verification



Evenly distributed heat release



Method 2: Multiple linear regression

MCC: Two Reactions Verification

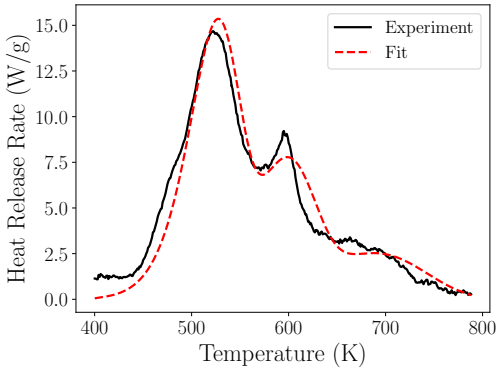
		$-\Delta h_1$ (kJ/g)	$-\Delta h_2$ (kJ/g)	$-\Delta h_{total}$ (kJ/g)
Specified Value		15	45	21
ΔT = 10 K	Simple Integration	--	--	20.998
	Linear Regression	14.854	44.085	21.021
ΔT = 20 K	Simple Integration	--	--	20.998
	Linear Regression	14.953	43.973	21.061
ΔT = 40 K	Simple Integration	--	--	20.998
	Linear Regression	15.415	42.304	21.501

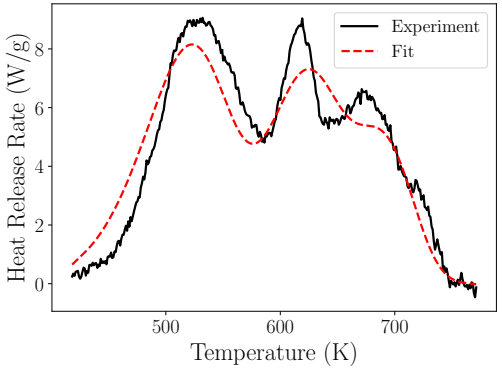
- Accuracy decreases with “broader” reactions
- Future work: correct individual reaction values to force match of total value

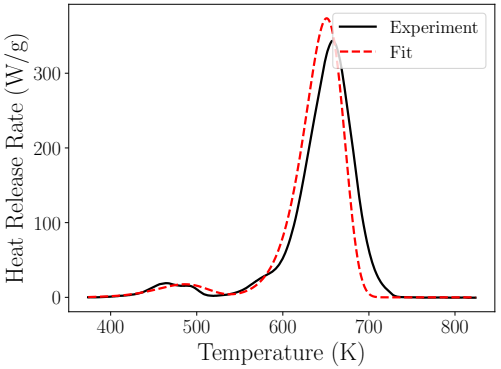
MCC Validation

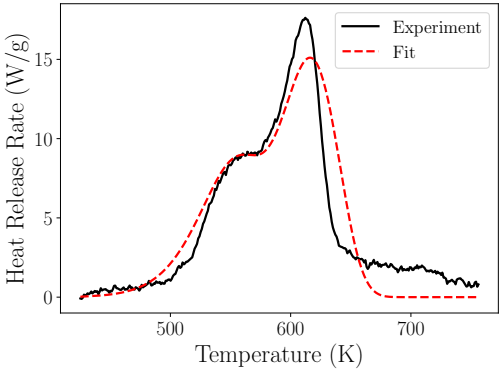
12 materials:

```
# list of all current MCC test series
mcc_tests = [
    'BigBerry-Leaf_Expt_MCC_N2_10K',
    'BigBerry-Stem_Expt_MCC_N2_10K',
    'Chamise-Leaf_Expt_MCC_N2_10K',
    'Chamise-Stem_Expt_MCC_N2_10K',
    'Chaparral-Leaf_Expt_MCC_N2_10K',
    'Chaparral-Stem_Expt_MCC_N2_10K',
    'DesertCeanothus-Leaf_Expt_MCC_N2_10K',
    'DesertCeanothus-Stem_Expt_MCC_N2_10K',
    'DouglasFir-Leaf_Expt_MCC_N2_10K',
    'LodgepolePine-Leaf_Expt_MCC_N2_10K',
    'LodgepolePine-Stem_Expt_MCC_N2_10K',
    'MaCFP_PMMA_Expt_MCC_N2_60K',
]
```









Example: Lodgepole Pine Stems

fact/Materials/metadata/LodgepolePine-Stem.json

LodgepolePine-Stem.json

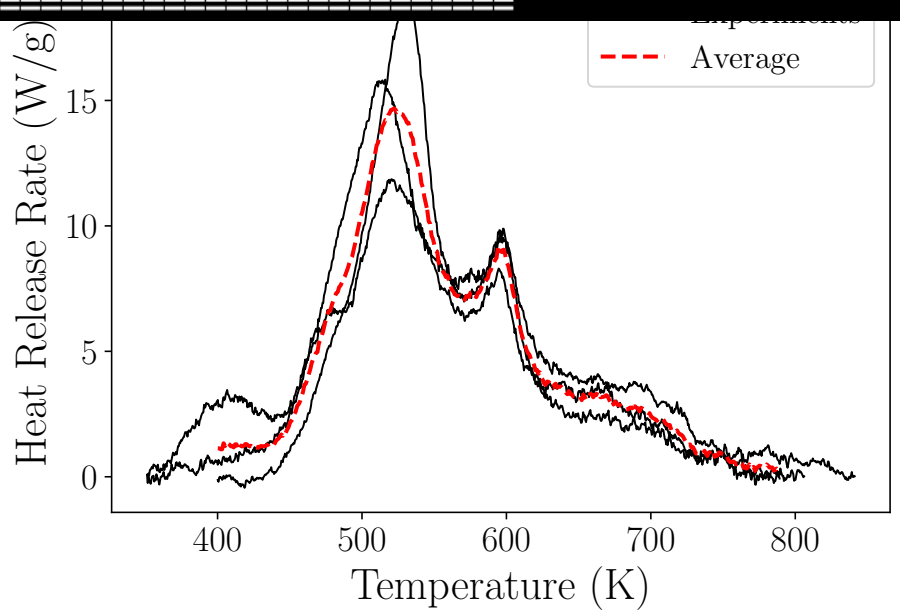

```
{
    "Common Names": "Lodgepole Pine Leaf",
    "Trade Name": "Pinus Contorta",
    "Manufacturer": "Natural Fuel",
    "Distributor": "Rocky Mountain Research Station (Missoula, Montana)",
    "Date Acquired": "May-July 2017",
    "Description": "Thin (<0.75 mm) flat slices of small branches picked
from a series of randomly selected individual Lodgepole Pine trees",
    "Categories": "Vegetation"
}
```

Replicate MCC Data Files:

```
(base) Scripts $ python average_mcc_series.py LodgepolePine-Stem_Expt_MCC_N2_10K
================================================================

 Reading data from:

    LodgepolePine-Stem_Expt_MCC_N2_10K_1.csv
    LodgepolePine-Stem_Expt_MCC_N2_10K_2.csv
    LodgepolePine-Stem_Expt_MCC_N2_10K_3.csv

 Number of data sets in series =  3

================================================================
```

```
Time,Temperature,HRR
[s],[K],[W/g]
3,400.15,-0.13
5.8,400.65,-0.2
8.7,401.15,-0.03
11.9,401.65,-0
14.8,402.15,-0.05
17.7,402.65,-0.03
21,403.15,-0.12
23.8,403.65,-0.09
26.9,404.15,0.04
```

```
Time,Temperature,HRR
[s],[K],[W/g]
0.5,352.15,0.09
6.5,352.65,-0.12
11.6,353.15,-0.21
17.1,353.65,-0.28
22.3,354.15,-0.12
27.2,354.65,0.01
32.3,355.15,-0
37.4,355.65,0.04
42.2,356.15,-0.19
```

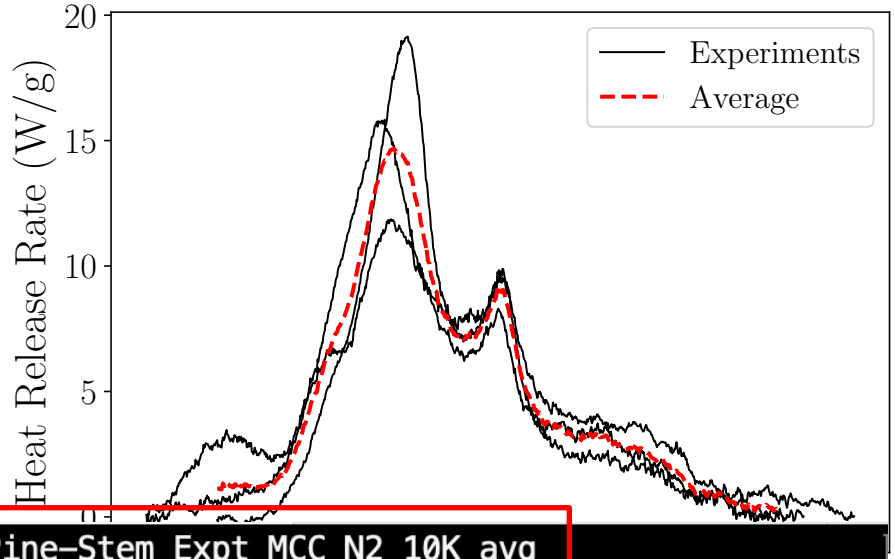


Pyrolysis Kinetics from TGA

LodgepolePine-Stem_props.json

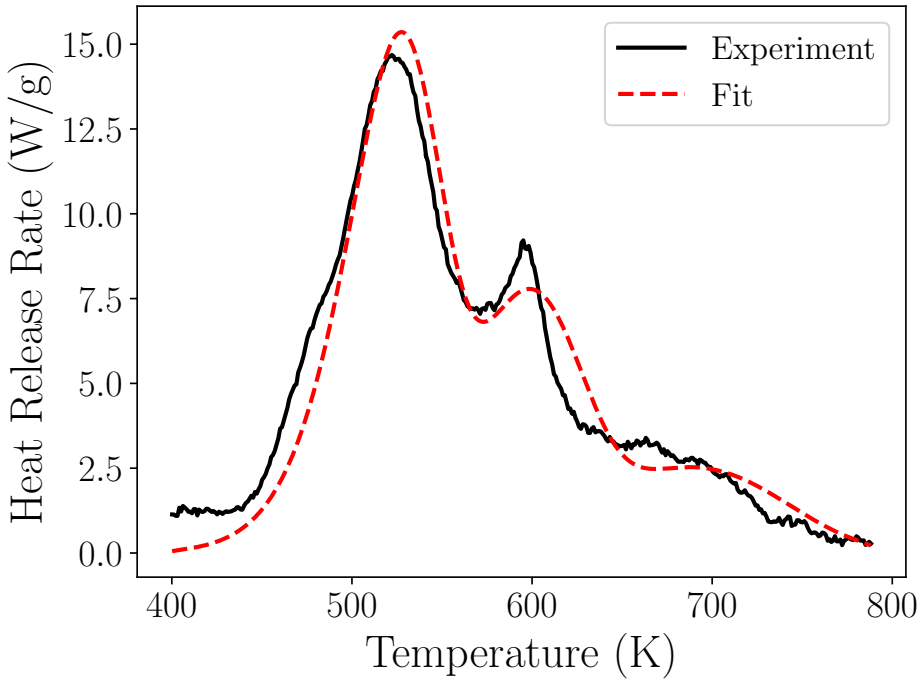

```
{
    "A": {
        "Name": "Pre-exponential Factor",
        "Units": "s^-1",
        "Values": [
            4895170.775500201,
            235982.41393676633,
            241.86226531218318
        ]
    },
    "E": {
        "Name": "Activation Energy",
        "Units": "kJ kmol^-1",
        "Values": [
            90468.95723804967,
            88696.41047329734,
            65820.96613844081
        ]
    },
```


Average MCC Data




```
(base) Scripts $ python mcc_fit.py LodgepolePine-Stem_Expt_MCC_N2_10K_avg
------------------------------------------------------------

 Fitting MCC data from file:  ../Experiment/MCC/data/LodgepolePine-Stem_Expt_MCC_N2_10K_avg
.csv
   at heating rate of  10.0  K/min.

------------------------------------------------------------

Results:
  RMS HRR Error     =  0.9460753645395685
  dh_c_tot (J/g)    =  11054.7026255
  dh_c (J/g)        =  [19534.18875694 11315.38066012 11669.52953085]

============================================================
```

```
(base) Scripts $ python mcc_fit.py LodgepolePine-Stem_Expt_MCC_N2_10K_avg
============================================================

 Fitting MCC data from file:  ../Experiment/MCC/data/LodgepolePine-Stem_Expt_MCC_N2_10K_avg
.csv
   at heating rate of  10.0  K/min.

------------------------------------------------------------

Results:
  RMS HRR Error     =  0.9460753645395685
  dh_c_tot (J/g)    =  11054.7026255
  dh_c (J/g)        =  [19534.18875694 11315.38066012 11669.52953085]

============================================================
```



LodgepolePine-Stem_props.json


```
{
    "A": {
        "Name": "Pre-exponential Factor",
        "Units": "s^-1",
        "Values": [
            4895170.775500201,
            235982.41393676633,
            241.86226531218318
        ]
    },
    "E": {
        "Name": "Activation Energy",
        "Units": "kJ kmol^-1",
        "Values": [
            90468.95723804967,
            88696.41047329734,
            65820.96613844081
        ]
    },
    "dmu": {
        "Name": "Normalized Mass of Reaction",
        "Units": " - ",
        "Values": [
            [
                0.2755842737690579,
                0.3073389550813397,
                0.18798788151838927
            ]
        ]
    },
    "dh_c": {
        "Name": "Heat of Combustion",
        "Units": "J g^-1",
        "Values": [
            19534.188756943473,
            11315.380660123741,
            11669.52953084592
        ]
    }
}
```

Summary

- **Predicting** fire growth requires material properties
- To obtain material properties:
  - Small-scale tests (TGA, MCC, DSC, etc.)
  - **Calibration algorithms**
- Calibration algorithms are presented for obtaining
  - pyrolysis kinetic parameters from TGA data
  - individual reaction heats of combustions from MCC data
- Algorithms performs well for
  - Manufactured solution verification cases
  - Multiple reaction materials with well-separated reaction peaks
- Algorithms implemented in **FACT:** **https://github.com/mcb1/fact/**