Numerical Modeling of Incomplete Combustion in a Liquid Chemical Rocket Engine

Departments of Mathematics and Chemistry

Anton M. Hengst, Dr. Holly Hirst, Dr. Jefferson Bates

Appalachian State University

Table of contents

1. Motivation and Introduction
    - Background Work
2. Mathematical Tools
    - Legendre Transformations
    - Lagrange Multipliers
    - Multivariate Newton's Method
3. Derivation of Iteration System
4. Assessment and Conclusions

Intro

Why this project?

Reasons to pursue a model of a rocket engine:

- Numerical modeling is cool.

[4]

Why this project?

Reasons to pursue a model of a rocket engine:

- Numerical modeling is cool.
- Rocket engines are cool.

[4]

Why this project?

Reasons to pursue a model of a rocket engine:

- Numerical modeling is cool.
- Rocket engines are cool.
- Liquid rocket engines are very cool.

[4]

Why this project?

Reasons to pursue a model of a rocket engine:

- Numerical modeling is cool.
- Rocket engines are cool.
- Liquid rocket engines are very cool.
- Industry standard software isn't free.

[4]

Why this project?

Reasons to pursue a model of a rocket engine:

- Numerical modeling is cool.
- Rocket engines are cool.
- Liquid rocket engines are very cool.
- Industry standard software isn't free.
- A learning opportunity!

[4]

The Liquid Rocket Engine

A rocket engine accelerates a rocket by expelling a reaction mass in the opposite direction of the rocket's motion.

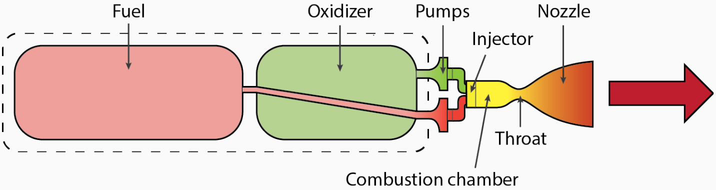


Figure 1: An idealized liquid rocket engine.

The reaction mass of a liquid rocket engine is the gaseous products of combustion. These gases are expelled by their expansion.

[6], Source: Anton Hengst (2020)

Variables of the Equation of State

Internal Energy U, total energy stored in a system, related to the first law of thermodynamics.

Entropy S, a measure of microscopic randomness, related to the second law of thermodynamics.

Volume V, the amount of space a system is allowed to occupy.

Composition N_i, the amount of each component (species) composing a system.

Temperature T, a measure of the energy of the system, determines the flow of heat.

Pressure P, characterizes kinetic energy contained in collisions between the system and its surroundings.

[2, 3]

Chemical Equilibrium

$$\text{Internal Energy} = U(S, V, \{N_i\})$$

$$\frac{\partial U}{\partial S} = T \qquad \frac{\partial U}{\partial V} = -P$$

Minimum-energy principle

$$dU = T\,dS - P\,dV + \sum_i \frac{\partial U}{\partial N_i}\,dN_i = 0$$

Heads Up

Take note especially of the expressions defining T and $-P$.

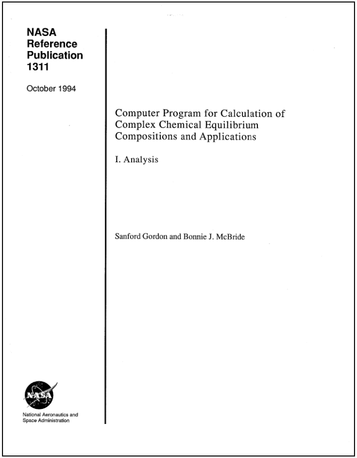
[2, 3]

Summer Work

Began with Gordon and McBride, 1994.

- First modern combustion modeling method.
- *Very* brief–notoriously so!
- Highly optimized.
- Had to work backwards through paper.

Halfway through semester–still unhappy with my understanding of the subject.



NASA Reference Publication 1311

October 1994

Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications

I. Analysis

Sanford Gordon and Bonnie J. McBride

National Aeronautics and Space Administration

[5]

Tools

Legendre Transformations

The Legendre transformation changes the variables of a function to the associated partials.

$$f(x_1, \dots, x_n) \xrightarrow{\text{Leg.}} \theta\left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}\right)$$

How? The tangent line equation

$$f = df \cdot x + \theta$$

Gives us

$$\theta = f - df \cdot x$$

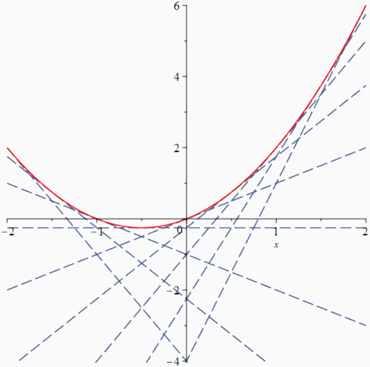


Figure 2: Tangent lines of $f(x) = x^2 + x$

Note

Unique partials are required to obtain a well-defined function.

Lagrange Multipliers

The method of Lagrange multipliers aids in finding extrema of a function subject to some constraints. Observe that:

- $\nabla f = \vec{0}$ at a global extremum.
- $\nabla f = \lambda \nabla c$ at a constrained extremum.
- $\nabla f = \sum_i \lambda_i \nabla c_i$ at a multiply-constrained extremum.

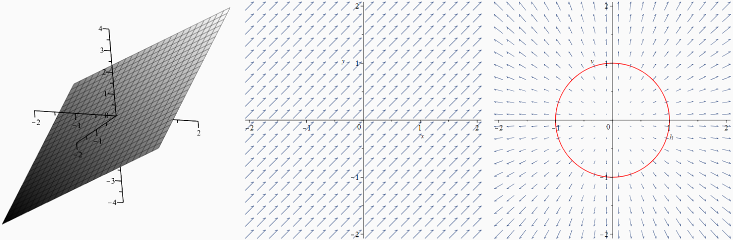


Figure 3: A function, $f(x, y) = x + y$, its gradient, and a constraint with the gradient of the function of which the constraint is a contour line.

Multivariate Newton's

Continuously differentiable multivariate functions $f(\vec{x})$ may be linearly approximated for small (but not necessarily infinitesimal) changes of $\vec{x}$ like so:

$$f(\vec{x}+\Delta\vec{x}) = f(\vec{x}) + \sum_n \frac{\partial f}{\partial x_n}\Delta x_n + O(\Delta\vec{x}^2)$$

If we seek a root,

$$f(\vec{x}+\Delta\vec{x}) = 0$$

Implying that

$$\sum_n \frac{\partial f}{\partial x_n}\Delta x_n \approx -f(\vec{x})$$

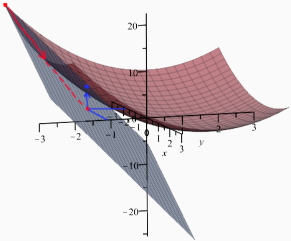


Figure 4: Multivariate Newton's rootfinding method descends most steeply along a tangent plane each iteration.

Derivation

Derivation, pt. 1

We transform internal energy to obtain *enthalphy*, H, and *Gibbs energy*, G. Minimizing the Gibbs, subject to constant T, P, is equivalent to minimizing U subject to constant S, V.

$$U(S, V, \{N_i\}) \xrightarrow{\text{Legendre}} H(S, V, \{N_i\}) \xrightarrow{\text{Legendre}} G(T, P, \{N_i\})$$

$$U = U \qquad H = U - PV \qquad G = H + TS$$

Internal Energy vs. Enthalpy

While it is difficult to measure U in a laboratory environment, it is feasible to acquire H data over a large range of temperatures.

[2, 3, 5]

Derivation, pt. 2

Let us say we have a system of i elements and j species, some of which may be gaseous, and some of which may not be (condensed).

From physical experimentation, we know that:

$$h = \sum_j h_j^\circ n_j \qquad s = \sum_j s_j n_j \qquad s_j = \begin{cases} s_j^\circ & j \text{ cond.} \\ s_j^\circ - R \ln \left(P \cdot \frac{n_j}{n} \right) & j \text{ gas.} \end{cases}$$

Substitution obtains our fundamental expression for the Gibbs.

$$g = \sum_j^{\text{cond.}} \left(h_j^\circ - T s_j^\circ \right) n_j + \sum_j^{\text{gas.}} \left(h_j^\circ - T s_j^\circ + RT \ln \left(P \cdot \frac{n_j}{n} \right) \right) n_j$$

Here, lowercase thermodynamic symbols are the function specific to mass. ° indicates a value at the standard pressure of 1 bar. n is the total number of specific mols of gas in the system.

[2, 3, 5]

Derivation, pt. 3

Our system is subject to constant-temperature, constant-pressure, and atomic conservation constraints. Can be expressed as:

$$T = T_0, \text{ or the function } t = 0, \text{ where } t = T - T_0$$

$$P = P_0, \text{ or the function } p = 0, \text{ where } p = P - P_0$$

And the set of i atomic conservation constraints

$$\sum_j a_{ij} n_j = b_{0,i}, \text{ or the functions } e_i = 0 \text{ where } e_i = \sum_j a_{ij} n_j - b_{0,i}$$

Here, a_{ij} is the number of atoms of element i in a molecule of species j and $b_{0,j}$ is the initial specific number of atoms j.

From this, we know we have the Lagrange equations:

$$t = 0 \qquad p = 0 \qquad e_i = 0, \ \forall i$$

[5]

Derivation, pt. 4

To complete the Lagrange system, we must additionally find

$$\nabla g = \sum_i \lambda_i \nabla e_i + \lambda_t \nabla t + \lambda_p \nabla p = \begin{cases} \frac{\partial g}{\partial T} = \sum_i \lambda_i \frac{\partial e_i}{\partial T} + \lambda_t \frac{\partial t}{\partial T} + \lambda_p \frac{\partial p}{\partial T} \\ \frac{\partial g}{\partial P} = \sum_i \lambda_i \frac{\partial e_i}{\partial P} + \lambda_t \frac{\partial t}{\partial P} + \lambda_p \frac{\partial p}{\partial P} \\ \frac{\partial g}{\partial n_j} = \sum_i \lambda_i \frac{\partial e_i}{\partial n_j} + \lambda_t \frac{\partial t}{\partial n_j} + \lambda_p \frac{\partial p}{\partial n_j}, \ \forall j \end{cases}$$

This gives us the final system:

$$\begin{cases} T - T_0 = 0 \\ P - P_0 = 0 \\ \sum_j a_{ij} n_j - b_{0,i} = 0, \ \forall i \end{cases} \quad \& \quad \begin{cases} nR - \sum_j s_j n_j = \lambda_t \\ \frac{nRT}{P} = \lambda_p \\ h_j^\circ - T s_j^\circ = \sum_i \lambda_i a_{ij}, \ \forall \text{ cond. } j \\ h_j^\circ - T s_j^\circ + RT \ln\left(P \cdot \frac{n_j}{n}\right) = \sum_i \lambda_i a_{ij}, \ \forall \text{ gas. } j \end{cases}$$

Derivation, pt. 5

With a bit of rearranging, we can express this as a set of functions of which we seek the root. (These functions are named with a * to keep them distinct from previous functions.)

$$\begin{cases} T^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = T - T_0 \\ P^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = P - P_0 \\ E_i^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = \sum_j a_{ij} n_j - b_{0,i}, \ \forall \text{ elements } i \\ t^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = nR - \sum_j s_j n_j - \lambda_t \\ p^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = \frac{nRT}{P} - \lambda_p \\ C_j^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = h_j^\circ - Ts_j^\circ - \sum_i \lambda_i a_{ij}, \ \forall \text{ cond. } j \\ G_j^*(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p) = h_j^\circ - Ts_j^\circ + RT \ln\left(P \cdot \frac{n_j}{n}\right) - \sum_i \lambda_i a_{ij}, \ \forall \text{ gas. } j \end{cases}$$

Note

T^* and P^* should always be equal to 0. t^* and p^* are similarly worthless to us, as we have no need to find λ_t or λ_p.

Derivation, pt. 6

Each function f can have its root iteratively found by solving:

$$\frac{\partial f}{\partial T}\Delta T + \frac{\partial f}{\partial P}\Delta P + \sum_j \frac{\partial f}{\partial n_j}\Delta n_j + \sum_i \frac{\partial f}{\partial \lambda_i}\Delta \lambda_i + \frac{\partial f}{\partial \lambda_t}\Delta \lambda_t + \frac{\partial f}{\partial \lambda_p}\Delta \lambda_p$$

$$= -f(T, P, \{n_j\}, \{\lambda_i\}, \lambda_t, \lambda_p)$$

And then adding each of the Δ values to each variable.

To solve our system, we can arrange the partials in a matrix and solve for a simultaneous solution to all equations. In the interest of both space and time, we will not show the massive matrix this derivation produces.

Final System

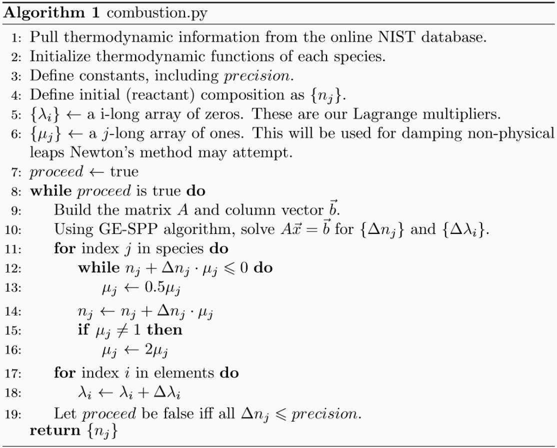
After trimming trivial components of the system, we obtain the following reduced iteration matrix equation.

$$
\begin{array}{cc}
\begin{array}{ccc}
\overbrace{\hphantom{a_{ij}}}^{j\ (\text{cond.})} & \overbrace{\hphantom{a_{ij}}}^{j\ (\text{gas.})} & \overbrace{\hphantom{a_{ij}}}^{i}
\end{array} & \\
\left[\begin{array}{ccc}
a_{ij} & a_{ij} & 0 \\
0 & \begin{cases} \frac{RT}{n_j} - \frac{RT}{n} & \text{on diag.} \\ -\frac{RT}{n} & \text{off-diag.} \end{cases} & -a_{ij} \\
0 & 0 & -a_{ij}
\end{array}\right] &
\begin{array}{l}
\}\ i \\
\Big\}\ j\ (\text{gas.}) \\
\}\ j\ (\text{cond.})
\end{array}
\end{array}
\cdot
\left[\begin{array}{c}
\Delta n_j \\
\Delta n_j \\
\Delta \lambda_i
\end{array}\right]
\begin{array}{l}
\}\ j\ (\text{cond.}) \\
\}\ j\ (\text{gas.}) \\
\}\ i
\end{array}
$$

$$
= \left[\begin{array}{c}
b_{0,i} - \sum_j a_{ij} n_j \\
\sum_i \lambda_i a_{ij} - h_j^\circ + T s_j^\circ - RT \ln\left(P \cdot \frac{n_j}{n}\right) \\
\sum_i \lambda_i a_{ij} - h_j^\circ + T s_j^\circ
\end{array}\right]
\begin{array}{l}
\}\ i \\
\Big\}\ j\ (\text{gas.}) \\
\}\ j\ (\text{cond.})
\end{array}
$$

Implementation

We implement the solution in Python 3 by the following:



Algorithm 1 combustion.py

```
Pull thermodynamic information from the online NIST database.
Initialize thermodynamic functions of each species.
Define constants, including *precision*.
Define initial (reactant) composition as $\{n_j\}$.
$\{\lambda_i\} \leftarrow$ a i-long array of zeros. These are our Lagrange multipliers.
$\{\mu_j\} \leftarrow$ a $j$-long array of ones. This will be used for damping non-physical leaps Newton's method may attempt.
*proceed* ← true
while *proceed* is true do
    Build the matrix $A$ and column vector $\vec{b}$.
    Using GE-SPP algorithm, solve $A\vec{x} = \vec{b}$ for $\{\Delta n_j\}$ and $\{\Delta\lambda_i\}$.
    for index $j$ in species do
        while $n_j + \Delta n_j \cdot \mu_j \leqslant 0$ do
            $\mu_j \leftarrow 0.5\mu_j$
        $n_j \leftarrow n_j + \Delta n_j \cdot \mu_j$
        if $\mu_j \neq 1$ then
            $\mu_j \leftarrow 2\mu_j$
    for index $i$ in elements do
        $\lambda_i \leftarrow \lambda_i + \Delta\lambda_i$
    Let *proceed* be false iff all $\Delta n_j \leqslant$ *precision*.
return $\{n_j\}$
```

[1]

Conclusions

Comparison with CEA

	CEA	Our Method
H_2	4.9476000E-01	4.9506127E-01
N_2	2.5040000E-01	2.5040559E-01
CO	2.4750000E-01	2.4767425E-01
H_2O	3.1579000E-03	2.9432857E-03
CH_4	1.7431000E-03	1.6540116E-03
HCN	1.5955000E-03	1.5125876E-03
CO_2	3.4484000E-04	3.2292458E-04
NH_3	2.1157000E-04	2.1523088E-04
H	1.4791000E-04	1.4791430E-04
HNC	8.0486000E-05	

Table 1: Hot, high pressure CHON system. 2000 K, 60 bar. 1 mols N_2O; 1 mols CH_4. Results in mol fractions. Results truncated for size.

[5]

Assessment vs. CEA

The majority of variation appears to be from our NIST-sourced data being more incomplete than the database CEA draws from.

Missing data: HCNO and most cyclic hydrocarbons

Worse data: e.g. HCN (hydrogen cyanide) is present. HNC (hydrogen isocyanide) is not.

While the missing species are minor, their absence still slightly affects calculations for other major species.

[1]

Summary

From first thermodynamic principles (equation of state), we:

1. Derive an expression for the **Gibbs energy** with respect to **enthalphy and entropy of each species**.
2. Re-express the function as a system of T, P, composition $(\{n_j\})$, and **Lagrange multipliers**, including expressions for **isothermal**, **isobaric** and **elemental constraints**.
3. Produce a matrix expression of the **simultaneous multivariate Newton iteration equation** for all the equations in our system.
4. Obtain thermodynamic data from the **National Institute of Standards and Technology** (NIST) and solve the system in Python.

Questions?

References I

- *NIST Standard Reference Database Number 69.*

- H. B. Callen.
  Thermodynamics and an Introduction to Thermostatistics.
  American Association of Physics Teachers, 1998.

- Y. A. Cengel and M. A. Boles.
  Thermodynamics: An Engineering Approach 6th Editon (SI Units).
  The McGraw-Hill Companies, Inc., New York, 2007.

- J. D. Clark.
  Ignition!: an informal history of liquid rocket propellants.
  Rutgers University Press, 1972.

- S. Gordon and B. J. McBride.
  Computer program for calculation of complex chemical equilibrium compositions and applications. part 1: Analysis.
  1994.
- G. P. Sutton and O. Biblarz.
  Rocket propulsion elements.
  John Wiley & Sons, 2016.