For the archives: point-source a k-β model

Fredric Lam

November 8, 2018

1 On the echelon of models

This document describes a model incorporating k-β that sits somewhere about the Green's function model, and below a more detailed model that resolves space. That is, this model discretizes in time only, but incorporates more chemistry than a switch or instant energy dump.

Schematically, Figure 1 shows the place where this model would probably fit in the echelon of discrete source models (to my knowledge; maybe we have even more cool levels of modeling detail):
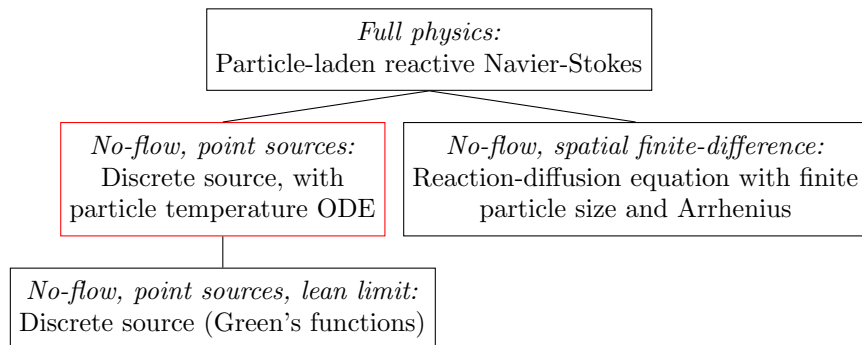

Figure 1: Tree schematic of models with descending level of detail, with model described in this document highlighted in red.

That is, this model sits on top of the Green's function model, and doesn't discretize in space. How useful/sensible this model is uncertain, since it doesn't seem very scalable (costly to add more particles and to increase simulation-time, i.e., $t_{\max}$). Nonetheless, it is here for completeness and to document this attempt.

2 Warm-up numerics

Let's test the waters using a one-dimensional toy/warm-up model to think about what math we need here:

$$\frac{\partial T}{\partial t} = \nabla^2 T + \delta(x - x_i)T(x, t) \tag{1}$$

is a modified heat equation (with point source dependent on local temperature T). It's not even clear whether it makes sense to multiply Dirac δ with T, but let's just assume everything works out fine.

This equation can be interpreted as the limit of a model with a smooth but increasingly narrow and tall functions that become δ. Thus, one can use a small region representing a particle, requiring some careful discretization of space.

Alternatively, one may try to solve this equation directly analogously to the Green's function method. That is, the equation with a predetermined source term:

$$\frac{\partial T}{\partial t} - \nabla^2 T = \delta(x - x_i)f(t) \tag{2}$$

lends itself to the analytic solution of

$$T^* = \int_0^t G(\tau)f(t - \tau)d\tau \tag{3}$$

with G the heat equation Green's function. If we're lucky, we can compute this analytically; this is the case for a boxcar f, i.e., the particle stays on for a fixed amount of time.

We can use this approach to solve for $T^($n + 1)$ at time t_{n+1} by approximating the convolution (really approximating the integral) with, say, the sum

$$T^{k+1} \approx \sum_{i=1}^{k} G(t_i)T_{k+1-i}\Delta t \tag{4}$$

with all the data for $T_i \approx T(t = t_i)$ before time t_{k+1}. This approach allows us to step forward the PDE (which look like Eqn. 1) while also stepping forward ODEs for the internal state of each particle. The disadvantage here is that we must keep track of the data for all time of each particle, which gets cumbersome; we also have no idea what the accuracy of this method is in T if we step in this fashion. Some comments are made about the mathematical difficulties here in Section 6.

We follow this approach in the following sections; if it works well enough, then we can get some results we can compare to experimental data.

3   Equation set

Assume particles with size insignificant relative to the length scale of diffusion (particle spacing).

3.1   Bulk Gas

Diffusion of heat through the gaseous medium with heat transfer from particles can be modeled as a heat equation with point sources (point sinks) representing heat transfer to the particle's internal temperature.

$$c_{\mathrm{g}}\rho_{\mathrm{g}}\frac{\partial T_{\mathrm{g}}}{\partial t} = \lambda_{\mathrm{g}}\nabla^2 T_{\mathrm{g}} + \sum_{j=1}^{N} \delta(x - x_j)\delta(y - y_j)h_{\mathrm{p}}A_{\mathrm{p,j}}(T_{\mathrm{p,j}} - T_{\mathrm{g}}) \tag{5}$$

with familiar parameters c_{g}, ρ_{g}, λ_{g}, and gas temperature T_{g}, heat transfer parameter h_{p} (feels like a cooking constant) and particle surface area A_{p}.

Analogously, diffusion of oxidizer through the gas with consumption, say proportional to the local concentration, can be modeled with

$$\frac{\partial C_{\mathrm{g}}}{\partial t} = D\nabla^2 C_{\mathrm{g}} - \sum_{j=1}^{N} \delta(x - x_j)\delta(y - y_j)k_{\mathrm{eff},j}C_{\mathrm{g}}A_{\mathrm{p},j} \tag{6}$$

with constant diffusivity D, and oxidizer concentration[1] C_{g}.

Note that ρ_{g} is the density expressed in the dimension of the simulation (e.g., for two-dimensional domain we talk about mass per area). Area A_{p} is just the surface area in three dimensions.

---

[1]Only the gas has an oxidizer concentration, while both gas and particle have temperature not necessarily the same.

3.2 Particle

Writing conservation of energy on particle i and balancing internal thermal energy, reaction, and heat loss to the gas, we have

$$c_{\text{s}} \frac{d(m_{\text{p},i} T_{\text{p},i})}{dt} = q k_{\text{eff},i} C_{\text{g}}(\mathbf{x}_i) A_{\text{p,i}} - h_{\text{p},i} A_{\text{p},i} (T_{\text{p},i} - T_{\text{g}}(\mathbf{x}_i)) \tag{7}$$

and from the fuel mass equation for particle i, we have

$$\frac{dm_{\text{p},i}}{dt} = -k_{\text{eff},i} C_{\text{g}}(\mathbf{x}_i) A_{\text{p},i} \tag{8}$$

3.3 Reaction rates

Assuming that the profile of oxidizer near the particle surface equilibrates essentially instantly, following M. Soo et al.,

$$k_{\text{eff},i} = \frac{k_i \beta_{\text{p},i}}{k_i + \beta_{\text{p},i}} \tag{9}$$

where

$$k_i = \kappa \exp\left(-\frac{T_{\text{a}}}{T_{\text{p},i}}\right); \beta_{\text{p},i} = \frac{D}{r_{\text{p},i}} \tag{10}$$

4 Solution Method

The solution method is motivated by our above practice problem, and relies only on discretizing time; each particle acts as pulses of heat and oxidizer transfer with magnitude depending on the state at the previous time. That is, the contribution of each particle can be represented as a train of Green's function.

Equivalently, a convolution of a function of the history of each particle's temperature and oxidizer with the Green's function precisely gives the solution T and C. The computation of this convolution can be performed numerically, which amounts to summing up the train of Green's function.

In steps, the computation is as follows:

Step 1: Compute coefficients for each particle i.

$$\beta_{\text{p},i} = D/r_{\text{p},i}$$

$$k_j = \kappa \exp\left(-\frac{T_{\text{a}}}{T_{\text{p},i}}\right)$$

$$k_{\text{eff},i} = \frac{k_i \beta_{\text{p},i}}{k_i + \beta_{\text{p},i}}$$

$$A_{\text{p},i},\ r_{\text{p},i} \text{ from } m_{\text{p},i},\ \rho_{\text{s}}$$

Step 2: Compute gas temperature and oxidizer at each particle location due to diffusion through the medium.

Define the shorthand for the n-dimensional (free-space, heat equation) Green's function as

$$G_{j,\alpha}(\mathbf{x}, t; \alpha) := \frac{1}{(4\pi\alpha t)^{n/2}} \exp\left(-\frac{||\mathbf{x} - \mathbf{x}_j||^2}{4\alpha t}\right)$$

This function sends the information about the gas temperature at $\mathbf{x}_j$ to any other point in space.

Each particle j exerts the following temperature influence at any location $\mathbf{x}$:

$$T_{\text{g}}^{j} = \frac{1}{c_{\text{g}}\rho_{\text{g}}} \int_{0}^{t} G_{j,\frac{\lambda_{\text{g}}}{c_{\text{g}}\rho_{\text{g}}}}(t-\tau)h_{\text{p}}A_{\text{p},j}(\tau)(T_{\text{p},j}(\tau)-T_{\text{g}}(\mathbf{x}_j,\tau))d\tau$$

and the following oxidizer concentration influence:

$$C_{\text{g}}^{j} = \int_{0}^{t} G_{j,D}(t-\tau)k_{\text{eff},j}(\tau)C_{\text{g}}(\mathbf{x}_j,\tau)A_{\text{p},j}(\tau)d\tau$$

Then any location (with or without a particle) has gas temperature due to the contribution of all particles, plus the ambient (uniform) initial condition (i.e., any background temperature) given by

$$T_{\text{g}}(\mathbf{x},t) = T_{\text{ambient}} + \sum_{j=1}^{N} T_{\text{g}}^{j}(\mathbf{x},t)$$

and corresponding oxidizer concentration given by

$$C_{\text{g}}(\mathbf{x},t) = C_{\text{ambient}} - \sum_{j=1}^{N} C_{\text{g}}^{j}(\mathbf{x},t)$$

Step 3: Timestep particle properties.

Each particle mass is updated by ODE

$$\frac{dm_{\text{p},i}}{dt} = -k_{\text{eff},i}(t)C_{\text{g}}(\mathbf{x}_i,t)A_{\text{p},i}(t)$$

and finally particle temperature is updated by ODE (after some manipulation of equations)

$$\frac{dT_{\text{p},i}}{dt} = \frac{A_{\text{p},i}(t)}{c_{\text{s}}m_{\text{p},i}(t)}\left[k_{\text{eff},i}(t)C_{\text{g}}(\mathbf{x}_i,t)(c_{\text{s}}T_{\text{p},i}(t)+q) - h_{\text{p}}(T_{\text{p},i}(t)-T_{\text{g}}(\mathbf{x}_i,t))\right]$$

.

4.1 Useful equations

$$\frac{\partial A_{\text{p}}}{\partial m_{\text{p}}} = \frac{2}{3m_{\text{p}}}A_{\text{p}} \tag{11}$$

$$\frac{\partial k_{\text{eff}}}{\partial m_{\text{p}}} = -\frac{1}{3}\left(\frac{4\pi\rho_{\text{s}}}{3}\right)^{\frac{1}{3}} D\frac{k^2}{(k+\beta)^2}m_{\text{p}}^{-\frac{4}{3}} \tag{12}$$

$$\frac{\partial k_{\text{eff}}}{\partial T_{\text{p}}} = \frac{\beta^2}{(k+\beta)^2}\kappa\exp\left(-\frac{T_a}{T_p}\right)\frac{T_a}{T_p^2} \tag{13}$$

Jacobian of f:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial m_{\text{p}}} & \frac{\partial f_1}{\partial T_{\text{p}}} \\ \frac{\partial f_2}{\partial m_{\text{p}}} & \frac{\partial f_2}{\partial T_{\text{p}}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial m_{\text{p}}} & \frac{\partial f_1}{\partial T_{\text{p}}} \\ \frac{\partial f_2}{\partial m_{\text{p}}} & \frac{\partial f_2}{\partial T_{\text{p}}} \end{bmatrix} \tag{14}$$

5 Sample Results

The attached code generates the following results. Figure 2 shows the flame speed as a function of fuel concentration for various activation temperatures. The model is dimensional.

Figure 3 is a (less useful) debugging figure, probing various variables for one particular run.
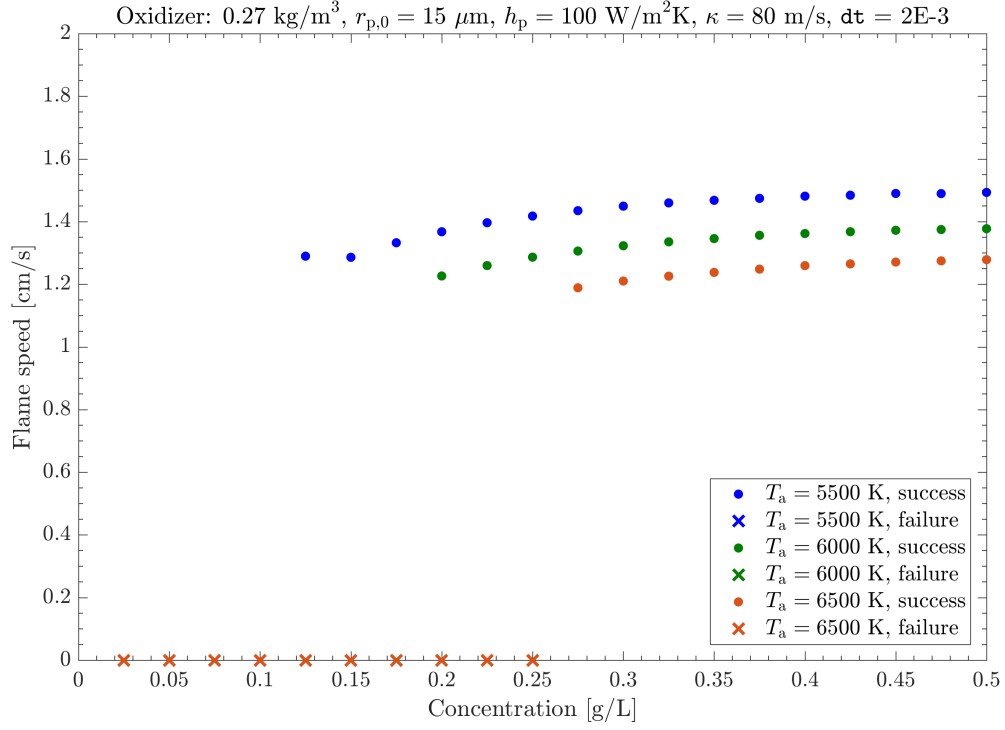
Figure 2: Flame speed as a function of oxygen concentration for various activation temperature parameters. Other parameters are specified at the top of the plot box. X-markers (stacked one atop another) denote failure to propagate.
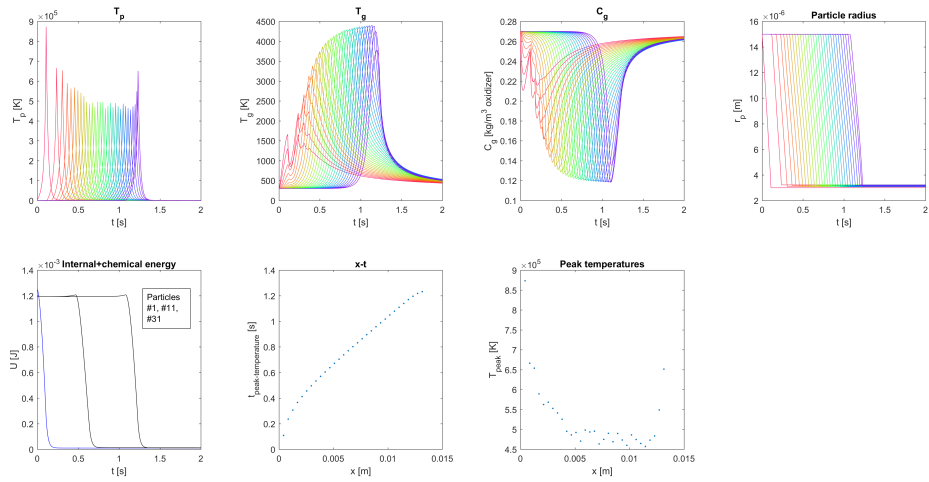

Figure 3: Debug dashboard containing some useful checks for a particular run of the model.

6   More on numerics of PDE

This is a technical section describing more how and why behind the numerical methods. Consider again the temperature-only PDE

$$\frac{\partial T}{\partial t} - \nabla^2 T = \sum_{i=1}^{N_{\text{particles}}} \delta(x - x_i) T(x, t) \tag{15}$$

where we try to solve by approximating the convolution

$$T^* = \int_0^t \sum_{i=1}^{N_{\text{particles}}} G(\mathbf{x} - \mathbf{x_i}, \tau) T(\mathbf{x}, t - \tau) d\tau \tag{16}$$

which was inspired by the same strategy for an equation with an explicit right-hand side, i.e., without T.

6.1   Did the method converge?

A convergence to a similar finite-difference model was obtained at quite roughly $\mathcal{O}(\Delta t^{\frac{1}{2}})$. Convergence is also obtained for Arrhenius. Finite-difference modeling was performed using a smoothed boxcar function that approaches a discrete Dirac delta. Adiabatic walls are used with domain length $L = 10$, and four particles at $x = 2, 4, 6, 8$.

6.2   Why the strange one-half convergence?

A difficulty with this model is the mix of PDE and ODE, especially when we decide to write our PDE as an integral equation, i.e., Eqn. 16.

   The integral equation is in a sense something we want to solve globally, while ODEs with initial values are something we want to solve by timestepping. By globally we mean that we want $T(t = t^*)$ to be determined as a function of $T(t < t^*)$ at previous times, so it makes sense that we should be able to update the $T(t < t^*)$ at previous times to get a better answer at time t^*. In fact, if we only had to deal with this integral in the wild, we would probably try to use a fixed point iteration (see Banach fixed point theorem), which proceeds as follows:

1. Pick some reasonable T_0.

2. Define the operator A by $AT = \int_0^t K(\mathbf{x}, t, \tau) T(\mathbf{x}, \tau) d\tau$ where K here is the sum of Green's functions in Eqn. 16.

3. For the desired interval $[0, t]$, fixing $\mathbf{x}$, iterate $T_{k+1}(\mathbf{x}, t), = AT_k(\mathbf{x}, t)$ until (hopefully) T_{k+1} is converged to T^*. The converged $T^*(\mathbf{x}, t)$ is then a function of t at a fixed $\mathbf{x}$.

   Then in this case, we are allowing T to change globally (in t), and the accuracy of the converged result depends on the Δt with which we approximate the integral.

   What's stopping us from going ahead and doing this is the ODE part of the model, for which it is more natural to use a timestepping strategy. We thus set in stone the previously computed gas temperatures T_{g} so we don't have to recompute the ODEs. The price we pay is we can't get the full convergence order on T from the PDE.