Model G with fluid dynamics implemented in Tensorflow 2

Lumi Pakkanen

March 2020

1 Kinetic Reaction Equations

The kinetic scheme of Model G reads

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} G \tag{1}$$

$$G \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} X \tag{2}$$

$$B + X \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} Y + Z \tag{3}$$

$$2X + Y \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} 3X \tag{4}$$

$$X \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} \Omega \tag{5}$$

This is represented by to following differential equations:

$$\frac{\mathrm{d}G}{\mathrm{d}t} = R_G(G, X, Y) = -(k_{-1} + k_2)G + k_{-2}X + k_1 A \tag{6}$$

$$\begin{split}\frac{\mathrm{d}X}{\mathrm{d}t} &= R_X(G, X, Y) = k_2 G - (k_{-2} + k_3 B + k_5)X \\ &+ k_{-3}ZY - k_{-4}X^3 + k_4 X^2 Y + k_5 \Omega\end{split} \tag{7}$$

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = R_Y(G, X, Y) = k_3 BX - k_{-3}ZY + k_{-4}X^3 - k_4 X^2 Y \tag{8}$$

Because the equation is polynomial in G, X and Y we can easily solve it by developing the solution into a series

$$G(t) = \sum_{n=0}^{\infty} a_n t^n$$
$$X(t) = \sum_{n=0}^{\infty} b_n t^n$$
$$Y(t) = \sum_{n=0}^{\infty} c_n t^n$$

The time derivates become

$$G'(t) = \sum_{n=1}^{\infty} n a_n t^{n-1}$$
$$X'(t) = \sum_{n=1}^{\infty} n b_n t^{n-1}$$
$$Y'(t) = \sum_{n=1}^{\infty} n c_n t^{n-1}$$

Plugging this back into 6 we can recursively solve a_{n+1}, b_{n+1} and c_{n+1} in terms of a_n, b_n and c_n. When integrating the system numerically we limit ourselves to $n \leq 4$ and start from a known position

$$G(0) = a_0, X(0) = b_0, Y(0) = c_0 \tag{9}$$

and calculate a short time Δt ahead

$$G(\Delta t) = \sum_{n=0}^{4} a_n \Delta t^n$$
$$X(\Delta t) = \sum_{n=0}^{4} b_n \Delta t^n$$
$$Y(\Delta t) = \sum_{n=0}^{4} c_n \Delta t^n$$

These new values for G, X and Y will be used as the a_0, b_0 and c_0 for the next time step. Further numerical considerations include solving the steady state equations $R_J(G_0, X_0, Y_0) = 0, \forall J \in \{G, X, Y\}$ and working with so called field variables $\varphi_J = J - J_0, J \in \{G, X, Y\}$. This makes sure that we're always adding small numbers together when integrating. The original equation has low numerical precission due to how floating point numbers work. Another issue to consider is the non-dimensionalization of the equations to make sure that all calculations happen in a numerically stable regime. The resulting differential equations are still polynomial in φ_G, φ_X and φ_Y so it's possible to recursively solve for another set of coefficients a'_n, b'_n and c'_n that produce a numerically stable and accurate step integrator $J(t) \mapsto J(t + \Delta t), J \in \{G, X, Y\}$

2 Diffusion Equations

The previous equations can be considered to be the 0-dimensional version of Model G. We now add a spatial component with diffusion.

$$\begin{aligned}
\frac{\partial G}{\partial t} &= D_G \nabla^2 G + R_G(G, X, Y) \\
\frac{\partial X}{\partial t} &= D_X \nabla^2 X + R_X(G, X, Y) \\
\frac{\partial Y}{\partial t} &= D_Y \nabla^2 Y + R_Y(G, X, Y)
\end{aligned}$$

When integrating the system numerically we alternate between solving the diffusion equation and the reaction equation for short time steps Δt. This is simpler than trying to solve the whole equation at once. The diffusion part has an exact solution.

$$G(t, \mathbf{x}) = \exp(-D_G |\overline{\omega}|^2 t + \mathrm{i}\overline{\omega} \cdot \mathbf{x}) \tag{10}$$

$$X(t, \mathbf{x}) = \exp(-D_X |\overline{\omega}|^2 t + \mathrm{i}\overline{\omega} \cdot \mathbf{x}) \tag{11}$$

$$Y(t, \mathbf{x}) = \exp(-D_Y |\overline{\omega}|^2 t + \mathrm{i}\overline{\omega} \cdot \mathbf{x}) \tag{12}$$

Which describes a decaying complex plane wave advancing in the direction of $\overline{\omega}$. This is recognized as a component of a Fourier decomposition. If we're working with periodic boundary conditions we can use a Fast Fourier Transform to transform the numerical solution into a simple component-wise multiplication by the decay tensor.

$$\mathbf{\Gamma}_{J,\overline{\omega}} = \exp\left(-D_J \Delta t |\overline{\omega}|^2\right), J \in G, X, Y \tag{13}$$

3 Fluid Dynamics

We can augment the system further by introducing fluid dynamics.

$$\frac{\partial G}{\partial t} + (\mathbf{u} \cdot \nabla)G = D_G \nabla^2 G + R_G(G, X, Y) - G\nabla \cdot \mathbf{u} \tag{14}$$

$$\frac{\partial X}{\partial t} + (\mathbf{u} \cdot \nabla)X = D_X \nabla^2 X + R_X(G, X, Y) - X\nabla \cdot \mathbf{u} \tag{15}$$

$$\frac{\partial Y}{\partial t} + (\mathbf{u} \cdot \nabla)Y = D_Y \nabla^2 Y + R_Y(G, X, Y) - Y\nabla \cdot \mathbf{u} \tag{16}$$

$$\tag{17}$$

It is assumed that the concentrations of A,B,Z and Ω are held constant so that they follow along with the fluid, but do not cause additional density gradients that would couple back to the fluid flow. Here $\mathbf{u}$ is a velocity field that obeys the Navier-Stokes equations for a compressible fluid

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -c_s^2 \nabla \ln \rho + \nu \left(\nabla^2 \mathbf{u} + \frac{1}{3}\nabla(\nabla \cdot \mathbf{u}) + 2\mathbf{S} \cdot \nabla \ln \rho \right) \tag{18}$$

where $-c_s^2 \nabla \ln \rho$ is the pressure term in the isothermal case and c_s is the constant speed of "sound" in ether. ν is is the kinematic viscosity coefficient, which is assumed constant and $\mathbf{S}$ is the traceless rate-of-strain tensor:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_j} - \frac{2}{3}\delta_{ij} \nabla \cdot \mathbf{u} \right) \tag{19}$$

The total density ρ is a sum of the individual concentration densities

$$\rho = \rho_0 + \alpha_G G + \alpha_X X + \alpha_Y Y \tag{20}$$

We note that the internal viscosity and shear equation

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \left(\nabla^2 \mathbf{u} + \frac{1}{3}\nabla(\nabla \cdot \mathbf{u}) \right) \tag{21}$$

Has the solution (with some abuse of notation)

$$u_j(t, \mathbf{x}) = \exp\left(-\nu \left(|\overline{\omega}|^2 + \frac{1}{3}\overline{\omega}_j(\overline{1} \cdot \overline{\omega}) \right) + \mathrm{i}\overline{\omega}_j \cdot \mathbf{x} \right) \tag{22}$$

giving rise to the Fourier domain decay tensor

$$\mathbf{\Theta}_{j,\overline{\omega}} = \exp\left(-\nu \Delta t \left(|\overline{\omega}|^2 + \frac{1}{3}\overline{\omega}_j(\overline{1} \cdot \overline{\omega}) \right) \right) \tag{23}$$

for component-wise multiplication. Because we're already working in the Fourier domain due to utilizing 10 and 23 instead of using a local stencil we can calculate partial derivatives globally. For example partial differentation with respect to the x direction becomes.

$$\frac{\partial G}{\partial x} = \mathcal{F}^{-1}\left(\mathrm{i}\omega_x \mathcal{F}(G)\right) \tag{24}$$

Where the ω_x is a tensor corresponding to the wave numbers in the x direction and the multiplication is done componentwise. It's good to note that many FFT libraries abuse the fact the frequencies wrap around so care must be taken when constructing ω_x to make sure that frequencies above the Nyquist bound are treated as negative frequencies. The remaining parts of 14 and 18 are integrated in terms of these very accurate global spatial derivatives. Unfortunately it still gives us only $O(\Delta t)$ accuracy in the t direction.

4 Algorithm

Putting it all together we first derive the centered polynomial reaction integrator using a computer algebra system and further compile it with all constants curried in

$$I_R(\Delta t, A, B, Z, \Omega, k_1, k_{-1}...)(\varphi_G, \varphi_X, \varphi_Y)|_{t=t_0} \mapsto (\varphi_G, \varphi_X, \varphi_Y)\,|_{t=t_0+\Delta t} \tag{25}$$

Working in a spatial volume (L_x, L_y, L_z) divided into an equidistant grid of (N_x, N_y, N_z) points we then construct the decay tensor for diffusion in the Fourier domain.

$$\omega \leftarrow meshgrid(range(-N_j/2, N_j/2) \cdot 2\pi/L_j \text{ for } j \in (x,y,z)) \tag{26}$$
$$\Gamma_J \leftarrow \exp\left(-\Delta t D_J(\omega_x^2 + \omega_y^2 + \omega_z^2)\right), J \in (G, X, Y) \tag{27}$$

The reaction diffusion integration over one time step Δt now becomes

$$\varphi_G, \varphi_X, \varphi_Y \leftarrow I_R(\varphi_G, \varphi_X, \varphi_Y) \tag{28}$$
$$J \leftarrow \mathcal{F}^{-1}\left(\Gamma_J \mathcal{F}(J)\right), J \in (G, X, Y) \tag{29}$$

TODO: Fluid dynamics algorithm