GrayScott with ETD-RK4

1 Equations

The GrayScott reaction diffusion system reads :

$$\begin{aligned} \frac{\partial u}{dt} &= D_u \nabla^2 u - uv^2 + F(1-u) \\ \frac{\partial v}{dt} &= D_v \nabla^2 v + uv^2 - (F+k)v \end{aligned} \tag{1}$$

where $u[m,n](t)$ and $v[m,n](t)$ denote the concentration of two chemical species over a two dimensional layout discrete layout.

Simulating this system could be straightforwardly done with the Forward Euler method. However, it turns out that the system has very slow dynamics and can quickly diverge if one uses a large time step Δt. Recently, [4, 2] proposed a method for simulating stiff differential equations which is based simulating the system in the spectral domain with exponential euler and runge kutta 4 and a numerical stabilisation procedure for evaluating terms like $(e^z - 1)/z$. In this document, we specify their algorithm in the case of the GrayScott reaction diffusion system. The code implementing their algorithm in the case of GrayScott is available online [3].

2 GrayScott with ETD-RK4

2.1 GrayScott in the spectral domain

The first step is to formulate the system of equations (1) in the spectral domain. We consider the spatial domain to be discrete and bounded of size $d \times d$ with N points over each dimension. We denote by capital letters the

discrete fourier transforms :

$$
\begin{aligned}
U[k,l] &= TF(u)[k,l] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u[m,n]e^{(-2\pi i\frac{km}{d})}e^{(-2\pi i\frac{ln}{d})} \\
u[m,n] &= TF^{-1}(U)[m,n] = \frac{1}{d^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} U[k,l]e^{(2\pi i\frac{km}{d})}e^{(2\pi i\frac{ln}{d})} \\
V[k,l] &= TF(u)[k,l] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} v[m,n]e^{(-2\pi i\frac{km}{d})}e^{(-2\pi i\frac{ln}{d})} \\
v[m,n] &= TF^{-1}(V)[m,n] = \frac{1}{d^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} V[k,l]e^{(2\pi i\frac{km}{d})}e^{(2\pi i\frac{ln}{d})}
\end{aligned}
$$

where we follow the convention of Numpy[1] where the direct transforms are unscaled and the inverse transforms are scaled.

2.2 Integration with exponential Euler/Runge Kutta 4

2.3 Numerical stability

As shown in [4], there is a numerical stability issue when evaluating the exponential terms that appear when applying exponential euler. The authors suggest a more stable numerical procedure which relies on the Cauchy integral.

References

[1] Numpy fft documentation. `https://docs.scipy.org/doc/numpy-1.13.0/reference/routines.fft.html#module-numpy.fft`, 2017.

[2] S.M. Cox and P.C. Matthews. Exponential Time Differencing for Stiff Systems. *Journal of Computational Physics*, 176(2):430–455, 2002.

[3] Jeremy Fix. Grayscott. `https://github.com/jeremyfix/GrayScott`, 2017.

[4] Aly-Khan Kassam and Lloyd N. Trefethen. Fourth-Order Time-Stepping for Stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.