M3/4/5N9 Computational Linear Algebra
Mastery project (20% of total mark) Due January 12th 2020 (must submit on Blackboard)

Prof Colin Cotter

Autumn Term 2019

Reminders

1. Be sure to follow all project guidelines (separate document on Blackboard).
2. Write your code in Python. You may use the `numpy`, `scipy` and `matplotlib` Python modules to complete the tasks unless otherwise indicated.
3. In terms of code, marks will be given for: clear, accessible, readable, re-useable code that makes sensible use of `numpy` array operations, and is well-organised into suitable files.
4. In terms of reporting, marks will be given for: clear, appropriate description that demonstrates understanding of the material.

This section is about exponential methods for the heat equation

$$u_t = u_{xx} + u_{yy}, \quad u(0,y,t) = u(1,y,t) = u(x,0,t) = u(x,1,t) = 0. \tag{1}$$

As in Project 2, we will consider an exponential method to compute the solution at time T for some large T, given the solution at time 0.

To do this, we will make a finite difference approximation, to get

$$\dot{u}_{i,j} = \frac{-4u_{i,j} + u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{\Delta x^2}, \quad i,j = 1,\ldots,N, \tag{2}$$

where we take $u_{0,j}(t) = u_{N+1,j}(t) = u_{i,0}(t) = u_{i,N+1}(t) = 0$.

1. Equation (2) is a linear ODE for the $u_{i,j}$ values and hence can be expressed as a matrix-vector ODE system
$$\dot{v} = -Av, \tag{3}$$
where $v = (u_{1,1}, u_{2,1}, \ldots, u_{N,1}, u_{1,2}, u_{2,2}, \ldots, u_{N,2}, \ldots, u_{1,N}, \ldots, u_{N,N})^T$. Describe the structure and values of A.
2. Write code to construct matrix A in CSR format, using `scipy.sparse.csr_matrix`.
3. The paper Gallopoulos and Saad (1992) describes an algorithm for computing the exponential of a large sparse matrix using Arnoldi iteration. Write code to implement this exponentiation for A. You may use the function `scipy.linalg.expm` to evaluate the exponential of the dense matrices H^k, but should only use sparse matrix operations in the Arnoldi construction.
4. Use this method to compute numerical solutions of the heat equation with your own chosen initial conditions, and validate the solution using Runge-Kutta time integration.

References

Gallopoulos, E., Saad, Y., 1992. Efficient solution of parabolic equations by Krylov approximation methods. SIAM Journal on Scientific and Statistical Computing 13 (5), 1236–1264.