Computational Linear Algebra
Mastery Project

Tudor Trita Trita
CID 01199397

January 13, 2020

This is my own work unless stated otherwise.

1 Structure & Values of A

We can write the linear system described in the question as a matrix-vector system $\dot{v} = -Av$, where $A \in \mathbb{R}^{(N^2,\ N^2)}$, and v is as described in the question, with $\dot{v}$ being the time derivative of v.

We recognise that Equation (2) given in the question is the 5-point stencil and the matrix associated with it, A is written in the following way. Writing the structure of the matrix as a block-matrix, where blank spaces denote zeroes in the matrix, then

$$A = \frac{1}{\Delta x^2}\begin{pmatrix} B & -I_N & & \\ -I_N & \ddots & \ddots & \\ & \ddots & \ddots & -I_N \\ & & -I_N & B \end{pmatrix}$$

where I_N is the NxN identity matrix and B is a tridiagonal matrix with format

$$B = \begin{pmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{pmatrix}$$

2 Code for Building A

The code for building A can be found in the file *code.py* inside the function *buildingA*.

3 Implementation of Exponentiation of A

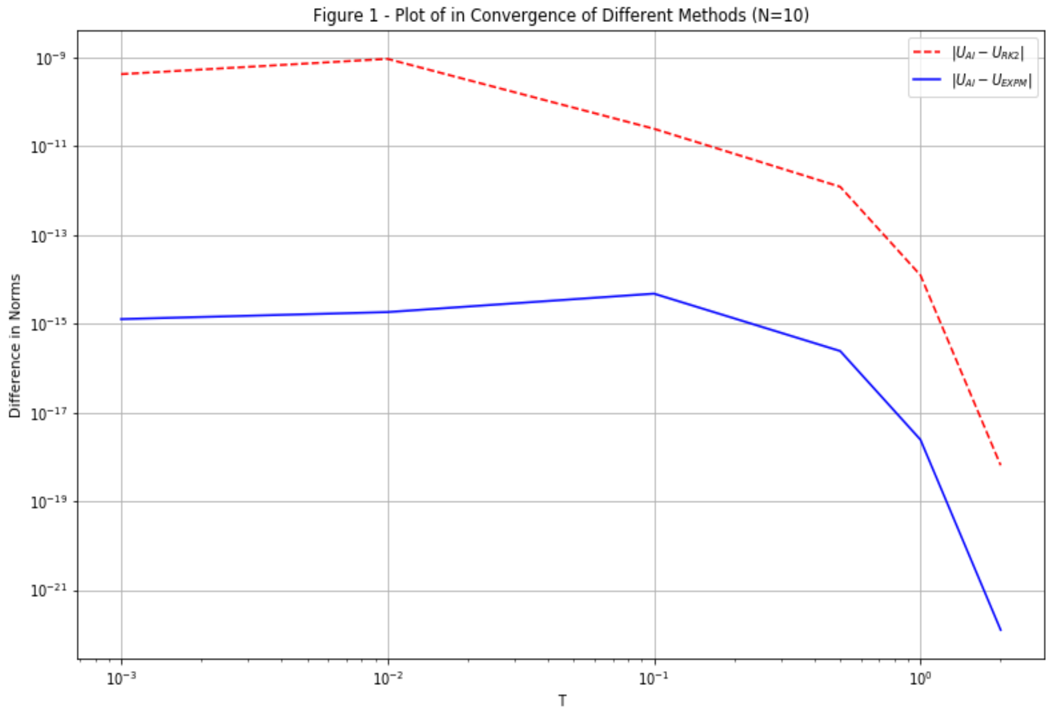
The implementation of the Arnoldi Iteration can be found inside the `arnoldi_iteration` function. To compute solutions to the Heat equation, we have a different function that solves the equation. This is locate inside of the `heat_equation_arnoldi` function.

4 Computing Numerical Solution of the Heat Equation

I have chosen to investigate the following initial condition:

$$u_0(x, y) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} - \frac{(y-\mu)^2}{2\sigma^2}\right)$$

We now validate the implementation by comparing this method with an explicit Runge-Kutta 2nd order method as well as the direct implementation of Scipy's `scipy.sparse.linalg.expm` sparse matrix exponential:



As we can see, the norm computed is very close to both RK2 and EXPM methods, and so this succesfully valildates my implementation.