Parallel implementation of IIF solver for stiff Advection-Reaction-Diffusion equations

Lokesh B Jogi

September 8, 2022

Internship Report

1.Introduction

Reaction-Diffusion-Advection systems can be modeled by the general equation

$$\frac{\partial u}{\partial t} = D\Delta u + f(u) \tag{1}$$

where $u \in \mathbb{R}^m$ represents concentration of m types of molecules or chemical species, $D \in \mathbb{R}^{m\times m}$ represents the diffusion matrix, and $f(u)$ represents reactions among different species. Δu represents the laplacian of u.The analytical solution for the above equation may or may not exist which inturn depends on the function $f(u)$. The Laplacian can be approximated with the help of Finite Difference Methods which is discussed in the next section. This approximation leads to an equation of the form

$$u_t = AU + F(U) \tag{2}$$

IIF is then applied to the equation to obtain a Numerical Solution to the given problem. The following sections discusses in detail about Finite Difference Approximation, some numerical methods to solve partial differential equations, IIF(Implicit Integration Factor) method.We also present some numerical examples to test the IIF method followed by results and conclusions.

2.Finite Difference Approximation

2.1 First Order Difference

A finite difference method proceeds by replacing the derivatives in the differential equations by finite difference approximations. The derivatives are replaced by algebraic equation which can be computed easily. Suppose the value of the function is known at the few discrete points, we would want to arrive at an expression which approximates the derivative.Let u be a function of x.Suppose $u(x), u(x+h), u(x-h)$ are known.We would want to approximate the derivative using the known values.
Using Taylor Series Expansion :

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2!}u''(x) + \frac{h^3}{3!}u'''(x) + O(h^4) \tag{3}$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2!}u''(x) - \frac{h^3}{3!}u'''(x) + O(h^4) \tag{4}$$

From (3) and (4) we get two one sided approximations of the derivative $u'(x)$

$$D_+u(x) = \frac{u(x+h) - u(x)}{h}$$

$$D_-u(x) = \frac{u(x) - u(x-h)}{h}$$

The above two expressions are called as the forward and backward difference respectively. These approximate the derivative with an error of $O(h)$. However taking average of the two equations gives us a better aprroximation of the derivative. This experession is known as central difference.

$$D_0 u(x) = \frac{u(x+h) - u(x-h)}{2h}$$

From Tayler Series expansion we can observe that the central difference has an error of $O(h^2)$.Therefore it provides a better approximate of the derivative than the one-sided ones. As more and more points are used to approximate the derivative the order of accuracy tends to increase.

2.2 Second Order Differences

Approximations to the second derivative $u''(x)$ can be obtained in an analogous manner. The standard second order centered approximation is given by

$$D^2 u(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} \tag{5}$$

This formula is obtained by repeatedly applying first order differences. We can view $D^2 u(x)$ as being a difference of first differences. This formula also has an error of $O(h^2)$. In this report,we will make use of this formula to discretize the second order spatial derivative. Approximations for higher order differences can be obtained similarly or by using method of undetermined coefficients.

3.Numerical Methods for Partial Differential Equations

Consider the heat equation with no heat generation term as given by

$$\frac{\partial u}{\partial t} = \alpha \Delta u \tag{6}$$

3.1 Explicit Method

Considering 1D case, let the time step chosen be Δt and the width between two nodes be Δx. Let u_i^j denote the value of u at the ith node and jth time step. Then using the finite difference approximation equation (6) can be written as

$$\frac{\partial u_i^j}{\partial t} = \alpha \left(\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2} \right) + O(\Delta x^2)$$

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \alpha \left(\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2} \right) + O(\Delta x^2) + O(\Delta t)$$

Let $\lambda = \dfrac{\alpha \Delta t}{\Delta x^2}$. We get

$$u_i^{j+1} = \lambda u_{i+1}^j + (1 - 2\lambda) u_i^j + \lambda u_{i-1}^j$$

Since the right hand side of the above equation is known, the value of u can be calculated explicitly at each time step. Hence this method is known as explicit time method. The explicit method is fast and inexpensive computationally. However this method isn't unconditionally stable and requires $\lambda \leq 0.5$ or else the method fails. Therefore grid size and time step must be chosen carefully. To avoid this complication the implicit method is used.

3.2 Implicit Method

In this method, instead of making use of the values at the current time step we make use of the values at current and the next time step, i.e

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \alpha \left(\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{\Delta x^2} \right) + O(\Delta x^2) + O(\Delta t)$$

We get

$$-\lambda u_{i+1}^{j+1} + (1 + 2\lambda) u_i^{j+1} - \lambda u_{i-1}^{j+1} = u_i^j$$

Here the right hand side is known and the left hand side is modified as $A\mathbf{u_i^{j+1}}$ where A is a tridiagonal matrix and $\mathbf{u_i^{j+1}}$ is a column vector. This leads to solving the matrix equation $A\mathbf{u_i^{j+1}} = \mathbf{u_i^j} + \mathbf{b}$ where $\mathbf{b}$ represents the boundary conditions.

$$A = \begin{pmatrix} 1 + 2\lambda & -\lambda & & 0 \\ -\lambda & \ddots & \ddots & \\ & \ddots & \ddots & -\lambda \\ 0 & & -\lambda & 1 + 2\lambda \end{pmatrix}.$$

At each time step we need to solve this matrix equation. Since the matrix is tridiagonal, we can use tridiagonal solvers to solve the equation. Even though the computational cost is higher than explicit scheme this method is unconditionally stable and provides better stability region than explicit method. Hence this method is widely used.

3.3 Crank Nicolson Method

This method makes use of both implicit and explicit scheme. It takes the average of both the implicit and explicit methods. The advantage of this method is that the error is second order both with respect to space and time. This is also an unconditionally stable method making it a very strong numerical method for solving stiff equations.

There are other numerical methods which make use of implicit and explicit methods and have their own advantages and disadvantages. However they are not discussed in this paper.

4. Implicit Integration Factor

Rewriting equation (1) as described earlier.

$$\frac{\partial u}{\partial t} = D\Delta u + f(u) \tag{7}$$

This equation has two terms. One relating to diffusion and the other relating to reaction. If explicit scheme alone is used, the time step becomes a huge constraint as there is stiffness from both diffusion term and the reaction term. Since the diffusion term is linear it can be treated explicitly and exactly whereas the reaction term is approximated implicitly. This helps to resolve stiff response from both the terms. In this section we derive the IIF scheme using a scalar equation which is a modified version of (7) and later can be generalised.

$$u_t = cu + f(u) \quad , t > 0, u(0) = u_0 \tag{8}$$

where c is a constant representing the diffusion, and f is a function representing the reaction.
Multiplying e^{-ct} to the above equation and integrating over one time step, we get

$$u(t_{n+1}) = e^{c\Delta t}u(t_n) + e^{c\Delta t}\int_0^{\tau} e^{-c\tau} f(u(t_n + \tau))\, d\tau \tag{9}$$

The critical step in constructing IIF scheme is the choice of approximating the integrand in the above expression. Defining $p(\tau) = e^{-c\tau} f(u(t_n + \tau))$
To construct a scheme of rth order truncation error, we approximate $p(\tau)$ with an $(r-1)$th order Lagrange polynomial, $p(\tau)$, with interpolation points at $t_{n+1}, t_n, ..., t_{n+2-r}$:

$$p(\tau) = \sum_{i=-1}^{r-2} e^{ic\Delta t} f(u_{n-i}) \prod_{j=-1, j\neq i}^{r-2} \frac{\tau + j\Delta t}{(j-i)\Delta t}$$

Using second order lagrange polynomial interpolation we get

$$p(\tau) = \frac{1}{\Delta t}[f(u_n) + e^{-c\Delta t} f(u_{n+1})\,\tau]$$

Using the above approximation to solve (9) we obtain the IIF scheme

$$u_{n+1} = e^{c\Delta t}(u_n + \frac{\Delta t}{2} f(u_n)) + \frac{\Delta t}{2} f(u_{n+1})$$

The second order method is unconditionally stable. Hence there is no limitations for choosing the time step. Also the term $e^{c\Delta t}$ is a constant and can be pre-calculated and stored. We can also observe that the exponential term and the reaction term is decoupled which reduces the computational cost when compared to other standard schemes. At each time step for the IIF , a system with m unknowns needs to be solved at each spatial grid point . The $Nm \times m$ systems are independent of each other with every system of the same structure. The same can be generalised when c is a matrix. In this paper, we use a scaling and squaring algorithm with a Pade´ approximation to approximate the exponential of the matrix and is implemented in Matlab with the function name"expm". The system of Non-Linear equations can be solved by applying Newton's Method at each point.

5 Numerical Tests on a reaction diffusion system using IIF :

5.1) Considering a linear 1 Dimensional problem defined as follows;

$$u_t = du_{xx} - au + v, \qquad 0 < x < \frac{\pi}{2}$$
$$v_t = dv_{xx} - bv, \qquad 0 < x < \frac{\pi}{2}$$
$$u_x|_{x=0} = 0, \qquad v_x|_{x=0} = 0,$$
$$u_{x=\frac{\pi}{2}} = 0, \qquad v_{x=\frac{\pi}{2}} = 0$$

The exact solution for the above system of equations is given by
$u(x,t) = (e^{-(a+d)t} + e^{-(b+d)t})\cos(x)$,

$v(x,t) = (a-b)e^{-(b+d)t}\cos(x)$.

A second order central difference is used for the approximation of u_{xx} . Let $dx = \frac{0.5\pi}{N+1}$ where $N+2$ is the total number of mesh points in $[0, \pi/2]$, the semi-discretization of the above equation is in the form of (2) . The matrix A is a tridiagonal matrix , $A = \frac{d}{\Delta x^2} C$, where C is a tri-diagonal matrix with -2 at the diagonal entry and 1 at the off-diagonal entry, except for the two-non-zero entries in the first row due to the no-flux boundary condition. Two different sets of parameters were chosen to test the method. $N = 575$ and $a = 100$ is chosen in both cases. In the first case (Case 1) $b = 1, d = 10^{-3}$, in the second case (Case 2) $b = 10^{-2}, d = 1$. The results obtained after computation are as follows.

Δt	L^∞-Case 1	L^∞-Case 2
4×10^{-2}	4.85×10^{-3}	3.42×10^{-4}
2×10^{-2}	1.21×10^{-3}	8.79×10^{-5}
1×10^{-2}	3.03×10^{-3}	2.23×10^{-5}
5×10^{-3}	7.58×10^{-5}	5.67×10^{-6}

5.2) We consider the following linear reaction–diffusion equation on a rectangle $\Omega = [0, 2\pi]^2$

$$\frac{\partial u}{\partial t} = 0.2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + 0.1u$$

Boundary Conditions are considered to be periodic . The exact solution for this problem is

$$u = e^{-0.1t} \left(\cos(x) + \cos(y) \right)$$

The initial condition is determined by the exact solution. The final computation time is t = 1. The time step is proportional to the spatial grid size, here we choose $dt = 1/Nx$. The L^∞ error is measured by difference between the numerical solution and the exact solution. The results obtained are as follows

$N_x \times N_y$	$\Delta t = \frac{1}{N_x}$	L^∞
32×32	0.0313	1.16×10^{-3}
64×64	0.0156	2.91×10^{-4}
128×128	0.0078	7.27×10^{-5}
256×256	0.0039	1.82×10^{-5}

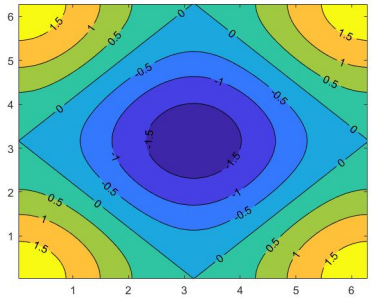

Figure 1: Analytical Solution

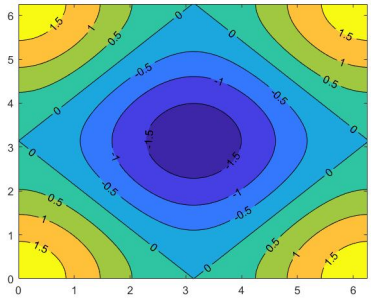

Figure 2: Numerical Solution

5.3) CIMA MODEL :

Lengyel and Epstein proposed a two-variable kinetic mechanism for CIMA reaction. In this skeleton version, iodide and chlorite play respectively the roles of the activator and the inhibitor:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + \frac{1}{\sigma}\left(a - u - 4\frac{uv}{1+u^2}\right) \tag{10}$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + b\left(u - \frac{uv}{1+u^2}\right) \tag{11}$$

where $D_u = \frac{1}{\sigma}, D_v = d, d = 1.07$,and $\sigma = 50$. We will solve the CIMA model on two-dimensional case . The domains on 2D are chosen as $\Omega = [0, 100]^2$. In our computation, we choose the mesh as 64×64 . Random initial concentration distributions of both species are used. Boundary conditions are considered to be periodic. Since the problem is non-linear we have to use Newton's Method at each node at each time step. The margin of error in Newton's Method is set to 10^{-7}.The Turing pattern needs a long computation time to appear. Here we set the final computation time as t = 10,000. Different patterns are obtained by selecting three sets of values of parameters a, b .The first set $(a = 8.8, b = 0.09)$ leads to an H_0 hexagon pattern as shown in Fig. 3. The second set $(a = 10, b = 0.16)$ gives rise to a stripe pattern (see Fig. 4). The third set $(a = 12, b = 0.39)$ generates an H_π hexagon pattern plotted in Fig. 5 .

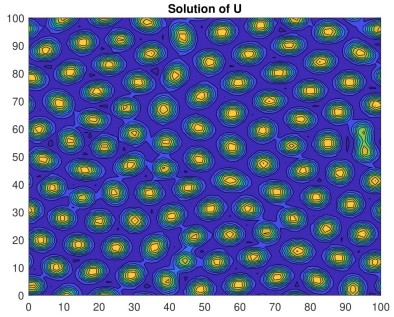

Figure 3: a=8.8,b=0.09

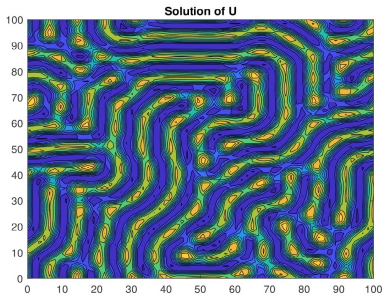

Figure 4: a=8.8,b=0.09

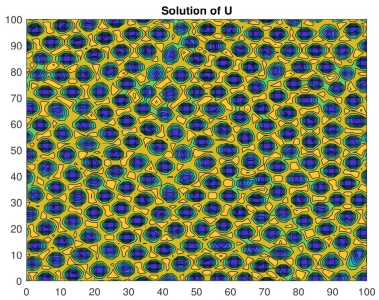

Figure 5: a=12,b=0.39

6. Conclusions

From the above examples, we can conclude that the IIF method is accurate, robust and a very efficient method. It is second order accurate in space and time. It removes the restriction offered by diffusion term by solving it explicitly and exactly and also solves the reaction term by implicit methods offering a greater stability range. In fact, the second order IIF is unconditionally stable. Also the decoupling of the exponential matrix and the non-linear reaction term reduces the computional cost hence enhancing the efficiency. Therefore this method is more advantageous than many other numerical methods.

7. Acknowledgements

I would like to thank National Supercomputing Mission and Prof.Rupesh Nasre for providing me this internship opportunity. A very special thanks to Prof. Amar Gaonkar and Prof. Amlan Barua for providing all the material and guiding me throughout this internship. It was very exciting to explore and learn various numerical methods. I look forward to explore more on other numerical methods for RDA equations in future.

8. References

1. Efficient semi-implicit schemes for stiff systems , Qing Nie , Yong-Tao Zhang, Rui Zhao , Journal of Comp. Physics

2. A compact finite difference method for reaction–diffusion problems using compact integration factor methods in high spatial dimensions, Rongpei Zhang, Zheng Wang , Jia Liu and Luoman Liu

3. Operator splitting implicit integration factor methods for stiff reaction–diffusion–advection systems, Su Zhao , Jeremy Ovadia , Xinfeng Liu , Yong-Tao Zhang , Qing Nie , Journal of Comp. Physics

4. Finite Difference Methods for Differential Equations, Randall J. LeVeque