Third Programming Assignment, MECH510

Shayan Hoshyari
Student #: 81382153

November 2015

Contents

1 The Problem

In this assignment the incompressible, laminar, 2D energy equation:

$$T_t + \nabla \cdot (uT) = \frac{1}{\mathrm{RePr}} \nabla^2 T + \frac{\mathrm{Ec}}{\mathrm{Re}} (2u_x^2 + 2v_y^2 + (v_x + u_y)^2)$$

Will be solved for a given velocity field. The computational domain is a rectangular box and the velocity field is given by the fully-developed profile:

$$u(x, y) = 18 \frac{y}{H} (1 - \frac{y}{H})$$
$$v(x, y) = 0$$

Where H is the channel height. The temperature will be fixed at the inflow (different for each problem) and the walls (lower wall: $T = 0$, upper wall: $T = 1$), and the temperature will be considered fully developed at the outflow. Finally the governing dimensionless numbers are given as: Re=50, Pr=0.7, Ec=0.1. To ease the notation, two additional numbers will be defined as: $\chi = \frac{\mathrm{Ec}}{\mathrm{Re}}$ and $\kappa = \frac{1}{\mathrm{Re} \cdot \mathrm{Pr}}$.

2 Numerical Method

Solving the energy equation is divided into two main parts: space discretization and time discretization. To discretize the equation in space we use the finite volume method and the second order central interpolation scheme to transform the original PDE into a set of algebraic differential equations:

$$\frac{d\mathbf{T}}{dt} = \mathbf{FI}(\mathbf{T})$$

To solve this system we will use two different schemes. The first scheme is the explicit second order Runge Kutta method (subscript n means at time step n, and FI is the flux integral plus the source term):

$$\mathbf{T}_* = \mathbf{T}_n + \frac{\Delta t}{2}\mathbf{FI}(\mathbf{T}_n)$$
$$\mathbf{T}_{n+1} = \mathbf{T}_n + \Delta t \mathbf{FI}(\mathbf{T}_*)$$

To find the stability bound of this method we first recall the eigenvalues of the energy equation:

$$\lambda \Delta t = \nu(-I \sin\phi) + \frac{2\nu\kappa}{\Delta x}(1 - \cos\phi)$$

Where ν is the CFL=$u_{max}\Delta t/\Delta x$ number and u_{max} is found from the velocity profile to be 4.5. Now with the given value of κ we propose the following maximum stable CFL number for the RK2 scheme:

$$\nu_{max} = \begin{cases} \frac{\Delta x}{2\kappa} & \Delta x \leq 0.1 \\ 1.44 & \Delta x = 0.2 \\ \text{we do not care} & \text{otherwise} \end{cases}$$

In this assignment we will consider mesh sizes $\Delta x = 0.2$, $\Delta x = 0.1$, $\Delta x = 0.05$, $\ldots$, so we are not concerned with $0.1 < \Delta x < 0.2$. Figure 1 shows the RK2 stability region and the $\lambda\Delta t$ curves for different values of Δx while the CFL number is found using the proposed method. We can clearly see that all cases are stable.

The second method that we will consider is the Implicit Euler method:

$$\mathbf{T}_{n+1} = \mathbf{T}_n + \Delta t \mathbf{F}(\mathbf{T}_{n+1})$$

After linearization the this method reduces to the solution of the following system of equations:

$$\left(\frac{\mathbf{I}}{\Delta t} - \frac{\partial \mathbf{FI}_n}{\partial \mathbf{T}_n}\right)\delta\mathbf{T} = \mathbf{FI}_n$$

We will solve this equation using the approximate factorization method. Theoretically, this scheme should be stable for any CFL number.

To find the steady state solution we continue to advance the solution in time until the maximum change in the solution drops below 10^{-10}. The ratio $\frac{\Delta x}{\Delta y}$ is always kept equal to 2 in this assignment.

3 Results

In this section the results for each part of the assignment are presented.

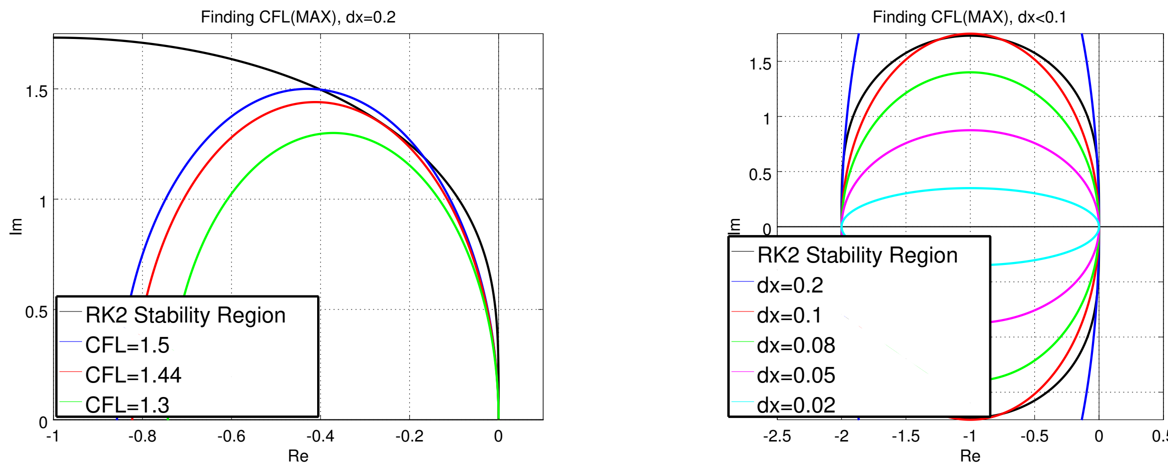

Figure 1: The RK2 stability curve and the $\lambda\Delta t$ curves for the energy equation. The left figure shows the eigenvalue curves when $\Delta x = 0.2$ for different CFL numbers. It is clear that CFL=1.44 is stable in this case. The right figure shows the eigenvalue curves for different Δx values when the CFL number is set to $\frac{\Delta x}{2\kappa}$, suggesting that CFL=$\frac{\Delta x}{2\kappa}$ is stable for $\Delta x \leq 0.1$

3.1 Flux Integral and Source Term Verification

In this section, the method of manufactured solutions will be used to verify the correctness of the flux integral and source term codes. The computational domain is defined as the region: $[0 \quad 1] \times [0 \quad 1]$. The average control volume values (including ghost cells) of temperature and velocities will be set using the following equations:

$$T(x,y) = \cos(\pi x) sin(\pi y)$$
$$u(x,y) = y\sin(\pi x)$$
$$v(x,y) = x\cos(\pi y)$$

Then the values S_{ij} and FI_{ij} will be calculated and compared with their exact values which are found using the midpoint integration rule and the exact values given below:

$$FI = -\frac{1}{A}\int_A \left[\pi y \cos(2\pi x)\sin(\pi y) + \pi x \cos(\pi x)\cos(2\pi y) + 2\kappa\pi^2\cos(\pi x)\sin(\pi y)\right] dA$$

$$S = -\frac{\chi}{A}\int_A \left[2(\pi y\cos(\pi x))^2 + 2(\pi x \sin(\pi y))^2 + (\sin(\pi x) + \cos(\pi y))^2\right] dA$$

The error norms are shown in Figure 2. With their slopes close to two, the correctness of the code is verified.

3.2 Stability and Accuracy Check

In this section the accuracy of the explicit RK2 and the implicit solver will be verified.

The energy equation is solved on a 5×1 channel, with the initial condition of $T_0(x,y) = y$. The boundary conditions, and the velocity field have their default values mentioned in Section 1, while the input temperature profile is set to its fully developed value:

$$T(0,y,t) = y + \frac{27}{4}\mathrm{Pr}\cdot\mathrm{Ec}\left[1 - (1-2y)^4\right]$$

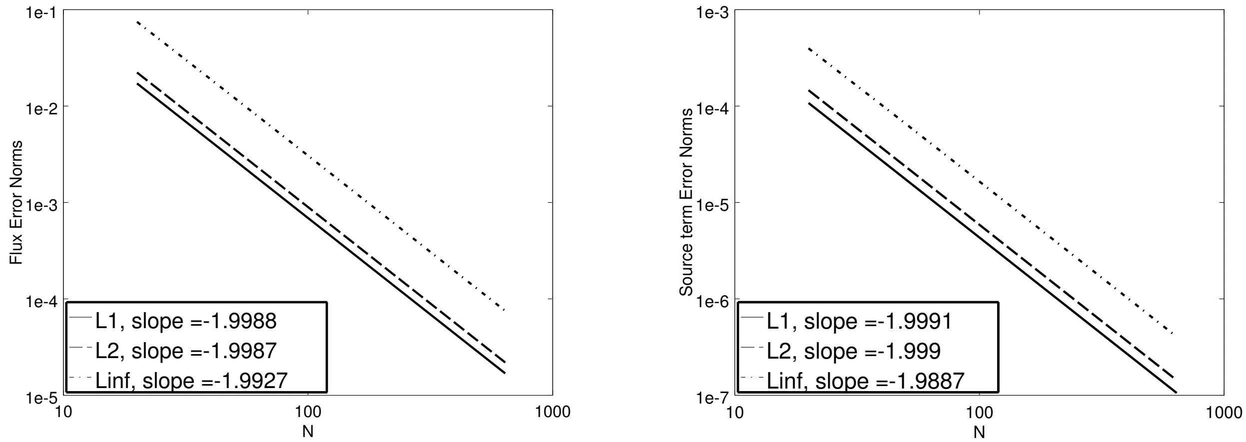

Figure 2: The norms of the source term and flux integral error. The horizontal axis shows the number of control volumes in each direction and the slopes are found using least square curve fitting.

The PDE is solved to the steady state on a series of refined meshes ($N_x = 25, 50, 100, 200, 400$) and the error norms of the steady state solution are shown in Figure 3. N_x and N_y are the number of CV's in the x and y directions respectively. It is clearly seen that the error curves have slopes close to two, so the solutions are truly second order.

3.3 Efficiency Test

In this section the efficiency of the explicit and implicit methods and the effect of time step selection for the implicit method are studied. We will provide further information about the run time, time step and the number of iterations of the test cases solved in Section 3.2.

For the implicit solver, I considered two cases. In the first case I just used the well performing time step of $\Delta t = 0.1$ for all the meshes. In the second case I tried to use the largest time step possible. I realized that when the time step is increased too much the run time gets ridiculously long. I tried to choose as large time steps as my patience to wait for the solution allowed. It was interesting that the solution did not get unbounded in any case, it just did not converge (or converged really slowly).

For the explicit solver, the time step dictated by the stability analysis with a safety factor of 80% performed well in all cases except $N_x = 50$. In this case I had to use a safety factor of 62%. The time steps used for each case are shown in Table 1 and Figure 4 shows the run time and number of iterations for each case. Three interesting remarks can be made about these results:

- Increasing the time step of the Implicit Euler scheme more than a certain limit makes the performance even worse than the explicit RK2 scheme, while the stability analysis guarantees stability for any Δt. This inconsistency can be explained by the fact that the stability analysis assumes no source terms, periodic boundary conditions, a one-dimensional problem and constant speed. Violating these conditions might be the reason for this inconsistency. Alternatively, the approximate factorization method for the solution of the equation $(\frac{\mathbf{I}}{\Delta t} - \frac{\partial \mathbf{FI}}{\partial \mathbf{T}})\delta \mathbf{T} = \mathbf{FI}$ with a too big time step reverts to a Jacobi like ADI method. The ADI method is known to perform well for diagonally dominant matrices. The presence of the $\frac{\mathbf{I}}{\Delta t}$

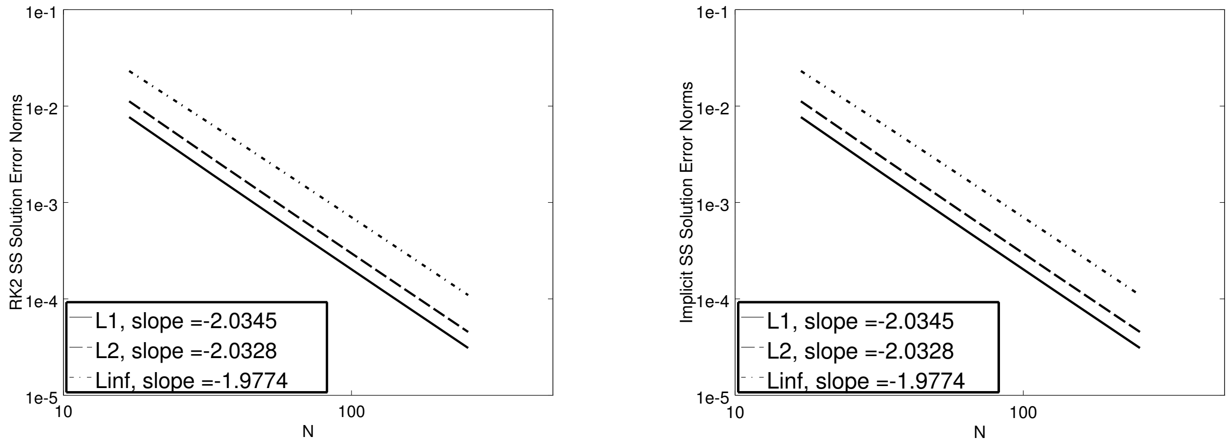

Figure 3: The almost identical error norms of the steady state solution when the steady state temperature profile is used as the inlet boundary condition. As the domain is not a square, the horizontal axis shows the average value of $\bar{N} = \sqrt{N_x N_y}$. The implicit solutions are obtained by using a time step of 0.1, while the explicit solver used the maximum stable time step obtained from the stability analysis and a safety factor of 80%.

Table 1: The time steps for the efficiency test cases

Mesh Size	25×10	50×20	100×40	200×80	400×160
RK2	0.050	0.0250	0.008	0.002	0.0005
Implicit with huge Δt	1000	1000	100	10	5

term is in fact making the equation diagonally dominant. When we increase Δt too much we will loose the diagonal dominancy, which results in poor convergence.

- RK2 time step limit increases from $N_x = 25$ to $N_x = 50$. This can be explained by Figure 1. When $N_x = 25$, i.e., $\Delta x = 0.1$ the upper part of the $\lambda \Delta t$ ellipse causes the instability while for $\Delta x \geq 0.2$ the left side of the $\lambda \Delta t$ ellipse causes the instability. This change in the behavior of the $\lambda \Delta t$ curve is the reason for the reverse behavior of maximum stable time step.

- The superiority of the implicit method becomes more evident for fine meshes.

3.4 The Real Problem

In this section the inflow boundary condition is changed to $T(0, y, t) = y/H$, where H is the height of the channel. Our goal here is to find (a) the fully-developed temperature gradient and (b) the thermal development length of the channel. To achieve this goal I will use a rather brute force method: I will increase the length of the channel to the value of 120!, so that I am sure that my outflow boundary conditions are correct. Then I will study the flow inside this long channel. The reason to choose this length is that the outlet temperature profile for a length of $L = 60$ looked similar to the analytic fully developed profile to me, so I chose a safety factor of two to come up with $L = 120$.

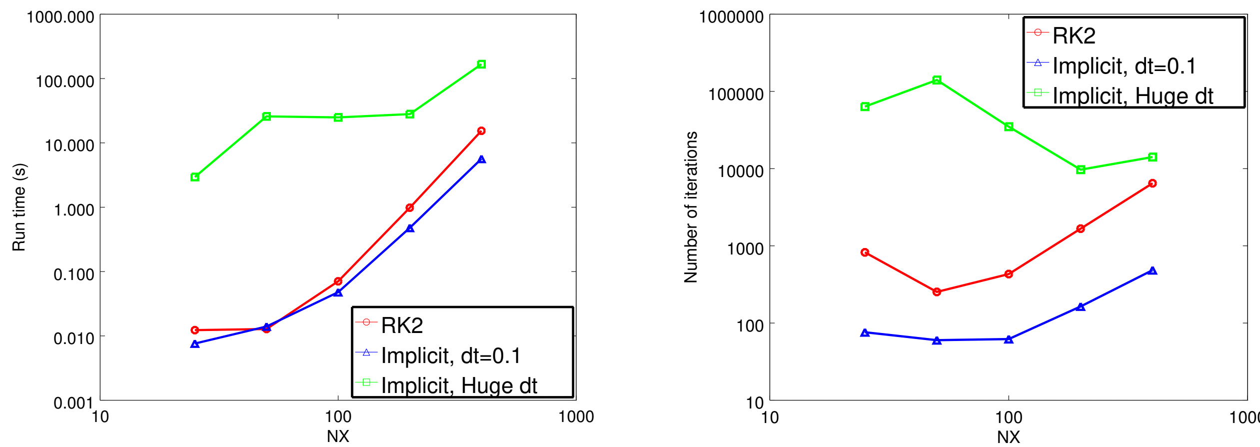

Figure 4: The number of iterations and the run time versus mesh size. Although the explicit and implicit methods perform almost the same for small meshes, the former performs significantly more efficiently than the latter for fine meshes, when a reasonable time step is chosen. For huge time steps the performance of the implicit method degrades immensely.

Firstly, I will plot the derivative of the temperature at the bottom wall as a function of x. Using this derivative I will define the fully developed length as the length at which the change in the derivative becomes relatively small. Mathematically:

$$T_x(x_{i+1},0) - T_x(x_i,0) < 10^{-3} T_x(x_i,0)\Delta x \quad x_i > L_d$$

Where L_d is the development length.

I will use this definition to find the ratio of development length to channel height for three cases (a) Ec=0.1, $H = 1$, $T^* = 1$ (b) Ec=0.1, $H = 2$, $T^* = 1$ (c) Ec=0.2, $H = 1$, $T^* = 2$. Where H is the channel height and T^* is the upper wall temperature. It is expected that all three cases result in the same ${}^{L_d}/_{H}$.

The value of temperature gradient at the bottom wall is shown in Figure 5 for each case. My definition results in the following values for ${}^{L_d}/_{H}$ for cases (a), (b) and (c) respectively: 44, 51, 50 (a mesh size of 4800×80 is used for all cases). These values are close to each other although they are not identical. This difference may be due to the fact that the definition of development length is somehow approximate and we should consider some tolerance for it.

Now I return to case (a) to evaluate the steady state wall temperature gradient (Note that we know the analytic value of this property is 4.78). As the development length was found to be less than the length of the channel, the value of this gradient at the end of the channel ($x = 120$) is definitely the fully developed value, so I will consider the wall temperature gradient at the last cell as the fully developed temperature gradient and use Richardson Extrapolation to investigate grid convergence.

I will show the fully developed wall temperature gradient by p and the order of convergence by r. The values p_1, p_2 and p_3 are found for mesh sizes of 4800×80, 2400×40 and 1200×20 respectively:

$$p_1 = 4.770261, \quad p_2 = 4.777353, \quad p_3 = 4.779126$$

Now the order of convergence will be:

$$r = \log_2 \left(\frac{p_1 - p_2}{p_2 - p_3} \right) = 2.00$$

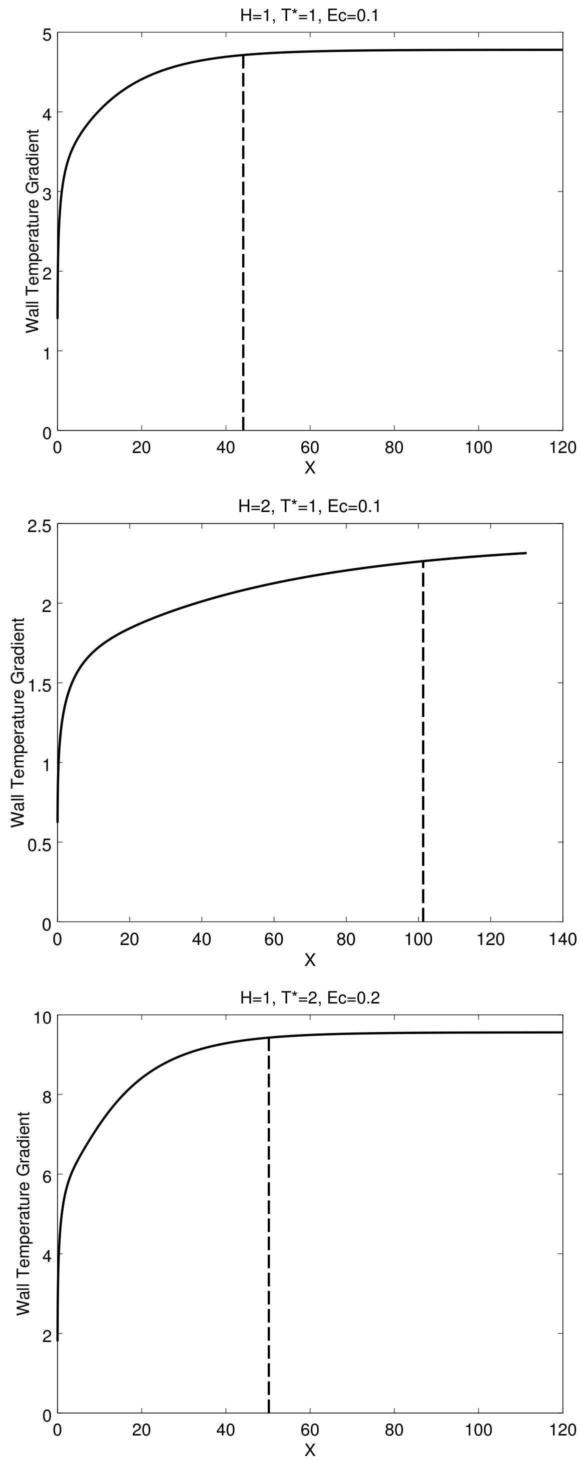

Figure 5: The bottom wall temperature gradient versus x, for the real problem cases (a), (b) and (c). The dashes show the location of development length.

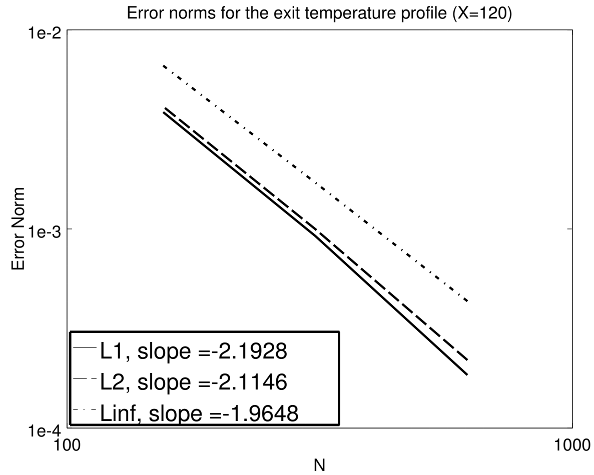

Figure 6: The error norms for the temperature profile at outlet ($x = 120$) for the real problem, case (a). It is clearly seen that the results are second order accurate.

And the extrapolated value for wall temperature gradient is:

$$p_{ext} = \frac{2^r p_3 - p_2}{2^r - 1} = 4.780$$

We can find both the approximate and exact error for this value:

$$e_{approximate} = \frac{p - p_{ext}}{p_{ext}} \times 100 = 0.012\%$$

$$e_{exact} = \frac{p - 4.78}{4.78} \times 100 = 0.000\%$$

Alternatively, the norm of the difference between the outlet temperature profile and the exact fully developed profile can be calculated and plotted versus the average mesh size $\sqrt{N_x N_y}$. This plot is shown in Figure 6. It is clear that the outlet temperature profile is second order accurate.

4 Conclusion

In this assignment the incompressible energy equation was solved using the finite volume method. The fluxes were evaluated using the central approximation scheme while the temporal discretization was done using the RK2 and the Implicit Euler methods.

Firstly, the numerical method was verified to be second order accurate. Then the performance of the explicit and implicit schemes were compared. It was shown that the implicit scheme performs more efficiently than the RK2 explicit scheme provided that a reasonable time step is chosen. Choosing a huge time step for the implicit scheme might not cause instability, but it increases the run time immensely.

Finally, the development of the temperature profile in a channel was studied and the temperature gradient at the bottom wall was approximated. The approximation was found to be second order accurate. The ratio of the development length to channel height was also found to be around 50.