1. Stability can be considered in norm other than \( L^2 \). For instance, the \( \infty \)-norm of a doubly-infinite sequence is defined to be the supremum of absolute values of elements in the sequence. Show that if \( \sigma \leq 1/2 \), then the Euler method for \( u_t = u_{xx} \) is stable in the \( \infty \)-norm.

[Hint: In other words, show that the maximum absolute value in \( v^{n+1} \) does not exceed the maximum absolute value in \( v^n \) when \( \sigma \leq 1/2 \).]

2. In their original paper, CFL considered the following two-step explicit finite difference scheme for the full wave equation \( u_{tt} = u_{xx} \):

\[
v^{n+1}_j = 2v^n_j - v^{n-1}_j + \lambda^2(v^n_{j+1} - 2v^n_j + v^n_{j-1}).
\]

Rewrite this as an explicit one-step vector finite difference method. Then apply a Fourier transform to the vector formula (see 3.6 of the text if you don’t know how to do this) to determine the amplification factor \( G(\xi) \). Finally, determine upper bounds on the absolute values of the eigenvalues of \( G(\xi) \) assuming \( \lambda \leq 1 \).

3. Determine the order and analyze the stability of the backward Euler method for the heat equation:

\[
v^{n+1}_j = v^n_j + \sigma(v^{n+1}_{j+1} - 2v^{n+1}_j + v^{n+1}_{j-1}).
\]

4. Consider a chemical reaction taking place along a linear domain \([0, 1]\). The reaction is the same one as in PS2, except that the four species also diffuse as they react:

\[
\begin{align*}
\alpha_t &= c\alpha_{xx} - m_1\alpha\beta, \\
\beta_t &= c\beta_{xx} - m_1\alpha\beta, \\
\gamma_t &= c\gamma_{xx} + m_1\alpha\beta - m_2\gamma + m_3\delta, \\
\delta_t &= c\delta_{xx} + m_2\gamma - m_3\delta.
\end{align*}
\]

Use the same values for \( m_1, m_2, m_3 \) as in PS2. Use \( c = 0.05 \). Solve these equations in matlab using the “method of lines” (see the first two pages of 3.3 of the text). That is, first discretize in space only: Replace the space-derivatives in the above equations with difference approximations. (Use as boundary conditions \( \alpha_x(0, t) = \alpha_x(1, t) = 0 \) and similarly for \( \beta, \gamma, \delta \). Assume that the four functions \( \alpha, \beta, \gamma, \delta \) are continuous functions of \( t \), each of which is defined at each grid point to obtain a system of ODEs. Then apply \texttt{ode15s} to this system of ODEs. (If you do this correctly, there will be \( 4N \)
ODEs in the system, where \( N \) is the number of spatial grid points). Use as initial conditions \( \alpha(x, 0) = 1 \) for \( x \leq 0.5 \), \( \alpha(x, 0) = 0 \) for \( x > 0.5 \), \( \beta(x, 0) \) the complementary function, and \( \gamma(x, 0) = \delta(x, 0) = 0 \). Integrate to \( t = 6 \). Compare the performance of \texttt{ode15s} both with and without a user-specified Jacobian. Note that your user-specified Jacobian should be a matlab sparse matrix.

Hand in listings of your m-files, a few sentences of conclusions, and two interesting plots. In particular, make a surface plot of \( \gamma \) as a function of both \( x \) and \( t \). Use \texttt{surf} or \texttt{surfl} for this purpose. Use the form of these functions in which the x- and y-axis arguments are vectors, and the z-argument is a matrix.