In the previous part of this lab assignment, you saw how the stiffness of the ODE system affected the step-size of the numerical method. You also saw that explicit methods are unstable unless a very small step-size is chosen. But how do the built-in Matlab ODE solvers choose step-sizes? They choose step-sizes automatically so that a certain accuracy is obtained. Do stiff ODEs lead to stability problems for these solvers as well? In this part of the lab assignment, you will solve a real problem in chemistry where the system is stiff in some cases. You will compare \texttt{ode45}, which is based on an explicit method, with \texttt{ode15s}, which is based on an implicit method.

1 Background

At a certain time \( t = 0 \), three chemicals A, B and C are mixed in a flask. A reaction takes place and the concentrations of these chemicals will change in time. Your task is to simulate this reaction.

The concentrations of A, B and C at time \( t \) are denoted \( \alpha(t), \beta(t), \gamma(t) \). The mathematical model of the reaction is a system of ordinary differential equations:

\[
\begin{align*}
\alpha'(t) &= s(\beta - \alpha\beta + \alpha - q\alpha^2) \\
\beta'(t) &= s^{-1}(-\beta - \alpha\beta + \gamma) \\
\gamma'(t) &= w(\alpha - \gamma)
\end{align*}
\]

The initial concentrations are \( \alpha(0) = 30.0 \), \( \beta(0) = 1.0 \), \( \gamma(0) = 30.0 \). The parameters \( s, q, \) and \( w \) are constants. Values of these are given below.
What to do

Your task is to compute concentrations from time $t = 0$ to 10 seconds using Matlab and to plot the result.

a) As before, begin by writing a Matlab function that defines the right-hand side of the system

b) When this is completed, type the command that solves the differential equation by using the built-in Matlab solver. The program should be designed in such a way that it will be easy to change the values of the parameters $s$, $q$, and $w$.
Begin with $s = 1$, $q = 1$, and $w = 0.1610$. Use the built-in ODE solver \texttt{ode45}. Plot the solution.

c) Change the constants $s$, $q$ and $w$ to $s = 77.27$, $q = 8.375 \times 10^{-6}$ ($w$ same as before). This will turn the system to a stiff problem. Run your program again and check what happens.
(If it takes too long (= some minutes) to obtain a solution, you can interrupt the computation in Matlab with \texttt{ctrl-c}).

d) Then change the ODE solver from \texttt{ode45} to \texttt{ode15s}. Run the program and compare with c). What is the difference?
Study the plots from experiment b) and compare with the plots from the present experiment. What happens to the solution curves when the problem becomes stiff?

Some hints

As a starting point, you can use an example from the previous lab assignment (Ordinary Differential Equations, Part 1). Use the program \texttt{PredatorPreySim} and the function \texttt{PredatorPreyODE} as a template. Remember that the right-hand side of the ODE system should be defined in a separate function, compare with \texttt{PredatorPreyODE}. It is this function that tells Matlab what equation to solve, and the function will be called by \texttt{ode45} and \texttt{ode15s} (without your notice). The function must have (at least) two input parameters and one output parameter. The structure of the function header is \texttt{yout = odeRHS(t, y)} and the independent variable \texttt{t} must be included as an input parameter even if it is not used in the right-hand side. The input parameter \texttt{y} is a vector containing $\alpha(t)$, $\beta(t)$ and $\gamma(t)$, i.e.,

$$y = [\alpha, \beta, \gamma]^T$$
To think about

> What difficulties arise when solving stiff differential equations in Matlab with *ode45*?
> Compare with Explicit Euler in the previous part of this lab assignment. Do you get the same oscillating solution curves for a stiff ODE when using *ode45*?
> The solution curves for stiff problems differ from those for non-stiff problems. How do the solution curves change when the problem becomes increasingly stiff?