Assignment 2:
The Least-Squares Problem from Reaction Kinetics

Hand in a written report containing a short presentation of the problem, results, discussion, source code, and a print-out of the results from your MATLAB sessions. You must state which version of MATLAB you are using. Answer all questions and comment your results. If the report is incomplete, we will return it for completion before starting to grade it. You may work alone or in groups of two or three. All persons in a group should contribute to the solution and the report. Discussions between groups are encouraged. If you receive substantial help from another group, say so in the report. You are not allowed to copy solutions or computer codes from others.

The absolute deadline for submitting this year’s assignments is 2011-01-24. You will obtain a 1/2 bonus point on the final exam on 2010-12-13, if a correct solution for this assignment is handed in at the latest 2010-11-29.

The m-files enzyme.m and enzymeRhs.m, available for download in studentportalen and on the home page, are necessary for this assignment.

The Problem
In this exercise, we consider a parameter estimation problem from chemical kinetics. The underlying system (described more detailed in the appendix) concerns the formation of products (P) from substrates (S) using enzymes. In this exercise, we consider a system with the following reactions

\[
S \xrightarrow{\mu S} P, \quad P \xrightarrow{P} S, \quad S \xrightarrow{0.1} S,
\]

in which the constants \(\mu\) and \(\gamma\) are unknown, and where \(S \rightarrow S\) denotes that substrates (S) are formed without using any products (P). A chemist has done a series of experiments with different initial concentrations (at time \(t_0 = 0\)). At time \(T = 1\), the chemist has measured the concentration of S and P again. The results of the measurements are summarized in Table 1.

<table>
<thead>
<tr>
<th>Experiment 1</th>
<th>Experiment 2</th>
<th>Experiment 3</th>
<th>Experiment 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t_0 = 0)</td>
<td>(T = 1)</td>
<td>(t_0 = 0)</td>
<td>(T = 1)</td>
</tr>
<tr>
<td>(S)</td>
<td>2.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>(P)</td>
<td>1.0</td>
<td>2.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 1: The concentration of substrate S and product P at times \(t = 0\) and \(t = 1\) from four independent experiments.

Your task is to find the parameters \(\mu^*\) and \(\gamma^*\) that fit the experimentally found values for the concentration best in least-squares sense. This means the error between the concentrations predicted by the mathematical model (for the optimal parameters \(\mu^*\) and \(\gamma^*\)) and the ones measured should be as small as possible. First you have to model the optimization problem. Then you have to implement functions and gradients and use MATLAB’s Optimization Toolbox to solve the problem. You should also compare the performance of different algorithms and examine the influence of the initial guess. Below, you find a detailed description of tasks.
Tasks

1. Let \( y^T = ([S], [P]) \) and \( k^T = ([\mu], [\gamma]) \). Write down the system of ordinary differential equations that describes system (1). Note that you need to study the appendix. (Notation: \( \tfrac{dy}{dt} = f(y, k) \))

2. Discretize the ODE you found in Task 1 given initial data \( y(0) = y(t_0) \) using the explicit Euler scheme with a constant time step of \( \Delta t = 0.01 \). That is, write down the expression for \( y^{(n+1)} \), the concentrations after \( n + 1 \) time steps, given \( y^{(n)} \), the concentrations after \( n \) time steps. Here, \( n \) is an integer between 0 and \( N - 1 \), where \( N = T/\Delta t \).

3. Given rates \( k \), current state \( y^{(n)} \), and current Jacobian \( \nabla_k y^{(n)} \), derive an expression for Jacobian \( \nabla_k y^{(n+1)} \) by using the chain rule. Note that, since the initial concentration does not depend on the rates \( k \), we have that \( \nabla_k y^{(0)} = 0 \).

4. The file \texttt{enzymeRhs.m} contains the MATLAB function \texttt{enzymeRhs} that evaluates the function \( f(y, k) \) and computes the Jacobians \( \nabla_y f \) and \( \nabla_k f \). The file \texttt{enzyme.m} contains a skeleton for the MATLAB function \texttt{enzyme}. Edit the file \texttt{enzyme.m} by incorporating the expressions you derived in Tasks 2 and 3. That is, given rates \( k = k \) and initial concentrations \( y_0 = y^{(0)} \) the function \texttt{enzyme} should return \( y = y^{(N)} \), the concentration at time \( T \), as well as the Jacobian \( \partial y / \partial k \).

5. The parameter estimation problem can be formulated as a least-squares problem, that is
\[
\min_k \frac{1}{2} \sum_{i=1}^{8} r_i(k)^2. \tag{2}
\]
Find expressions for the residuals \( r_i(k) \) as well as their gradients. Write a MATLAB function \texttt{enzymeLeastSquares} that given rates \( k \) returns vector \( (r_1(k), r_2(k), \ldots, r_8(k))^T \) and matrix
\[
\begin{pmatrix}
\frac{\partial r_1}{\partial k_1} & \frac{\partial r_1}{\partial k_2} & \cdots & \frac{\partial r_1}{\partial k_8} \\
\frac{\partial r_2}{\partial k_1} & \frac{\partial r_2}{\partial k_2} & \cdots & \frac{\partial r_2}{\partial k_8} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial r_8}{\partial k_1} & \frac{\partial r_8}{\partial k_2} & \cdots & \frac{\partial r_8}{\partial k_8}
\end{pmatrix}^T.
\]

6. Solve problem (2) with the Levenberg–Marquardt method by invoking the routine \texttt{lsqnonlin} from MATLAB’s Optimization Toolbox. Use the following syntax to call \texttt{lsqnonlin}
\[
\text{options} = \text{optimset}('\texttt{Jacobian}', 'on', '\texttt{Display}', 'testing', 'Algorithm', 'levenberg-marquardt');
\]
\[
k = \text{lsqnonlin}(@\text{enzymeLeastSquares}, k0, [], [], options);
\]
where \( k_0 \) is your initial guess. The options above specify that gradients are available, and that we want to invoke the Levenberg–Marquardt method. The \( @ \) in the first argument of \texttt{lsqnonlin} indicates a function handler, and tells \texttt{lsqnonlin} to use the MATLAB function \texttt{enzymeLeastSquares} to evaluate function values and gradients. Use \( k_0 = (1, 1)^T \) as initial guess and record carefully the number of iterations and function evaluations. Using the option ‘\texttt{Display}', 'testing', you can read off the values from the function counter.
7. The routine \texttt{fminunc} in MATLAB’s Optimization Toolbox solves general minimization problems without constraints. To use this routine, you need to write a function, \texttt{enzymeGeneral}, that returns the scalar
\[
\frac{1}{2} \sum_{i=1}^{8} r_i(k)^2, \quad \text{and the vector} \quad \sum_{i=1}^{8} r_i(k) \nabla_k r_i(k). \tag{3}
\]
To solve problem (2) with \texttt{fminunc}, use the following syntax:
\begin{verbatim}
options = optimset('GradObj','on','Hessian','off',
                    'LargeScale','off','Display','testing');
a = fminunc(@enzymeGeneral,a0,options);
\end{verbatim}
The options above specify that gradients but no Hessians are available. Furthermore, we use the medium-scale algorithm (a BFGS quasi-Newton algorithm with line search). Again, use \( k_0 = (1,1)^T \) as initial guess and record carefully the number of iterations and function evaluations.

8. Compare the performance of the two algorithms that you have tested. Explain your observations!

9. To study how the initial guess influences the performance, rerun the experiments from Tasks 6 and 7 using initial guess \( k_0 = (10,10)^T \). Record the number of iterations and function evaluations for all your experiments. Compare your results with what you recorded for the old initial guess.

\textbf{Appendix: Introduction to Reaction Kinetics}

A quantitative interpretation of chemical reaction systems is frequently based on the construction and analysis of kinetic mechanisms or models. These models are based on the idea of breaking up the overall reaction into a number of components or \textit{elementary steps}. The elementary steps of how an enzyme (\( E \)) binds substrates (\( S \)) and turns them into products (\( P \)) using a reactant (\( R \)) are
\[
E + S \xrightarrow{k_1[E][S]} ES, \quad ES \xrightarrow{k_2[ES]} E + S \quad ES + R \xrightarrow{k_3[R][ES]} E + P. \tag{4}
\]
By the law of mass action, the \textit{rate} (the number of reactions per time unit) of an elementary step is given by the product of the concentrations of the participating species. For the first reaction in (4), the rate is given by \( k_1[E][S] \), where \( [E] \) is the concentration of the enzyme, \( [S] \) is the concentration of the substrate, and the constant \( k_1 \) is called the \textit{reaction rate coefficient} or the \textit{reaction rate constant}.\footnote{Typically, rate constants are independent of concentration but depend on the temperature of the reacting mixture.} Thus for our current system, \( k_1[E][S] \) mol of \( ES \) are formed each second. We can form an ordinary differential equation describing this evolution of the concentration of the included species. Let us now consider the change of concentration of the enzyme per time unit. Summing up the contributions from the three elementary steps listed in (4), we arrive at the following equation
\[
\frac{d[E]}{dt} = -k_1[E][S] + k_2[ES] + k_3[R][ES].
\]
Note that the term that removes the enzymes depends on the concentration \([E]\). Thus if the concentration \([E]\) is zero then the derivative \(d[E]/dt\) is non-negative ensuring that the concentration stays non-negative.

Assuming that the concentration of substrate bound enzyme \(ES\) and the reactant \(R\) are (essentially) constant (Michaelis–Menten kinetics), the concentration of enzyme \(E\) will also be constant due to conservation. Under the above assumptions, the essence of the reaction of the above system may be written as the single step

\[
S \xrightarrow{\mu[S]} P,
\]

where \(\mu\) and \(\gamma\) are constants.