Assignment 1: Metropolis-Hastings

I expect a small paper (or pdf) report from you on this assignment that consists of:

1) An answer to all questions, item by item, including plots and your discussions.
2) Your Matlab program. No need to streamline it, just to see you haven't made serious errors.
3) Nothing more!

Standard Metropolis-Hastings

To understand the Metropolis-Hastings sampler better better, and especially the way it can be used in data assimilation, you have to write a Matlab program to estimate some statistics of a posterior pdf. It is a continuation of the problem you worked on for the Gibbs sampler, but now with a nonlinear observation operator. You will test and compare two standard Metropolis-Hastings with the Hybrid Monte-Carlo method.

The posterior pdf is constructed from a **100-dimensional** Gaussian prior with mean $x0 = 0$ and covariance

$$P = \begin{pmatrix} 1.0 & 0.5 & 0.25 & 0 & ... & 0 \\ 0.5 & 1.0 & 0.5 & 0.25 & ... & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & ... & 0 & 0.25 & 0.5 & 1 \end{pmatrix}$$

and a likelihood consisting of a Gaussian-distributed observation of the square of the first component of state vector x, so $y = x_1^2$, with observation error variance $R = 0.5$.

1) Derive an expression for the posterior pdf using Bayes Theorem. Ignore the normalisation factor.
2) Why can the normalisation factor be ignored?
3) Argue that the posterior pdf is bimodal. (Hint, the observation does not constrain the sign of x_1).
4) Implement the standard MH sampling scheme, using as start value a draw from $N(0, 0.01)$ for each component and $y = 2$. Take for the proposal $q(x^n|x^{n-1}) = N(x^{n-1}, 0.01)$ for each component. Show that the proposal is symmetric, and use that in your code for the acceptance criterion.

   Initialise the random number generation to obtain reproducible results by including

   rand('state',100)

   and

   randn('state',100)

   as first lines in your program.

5) Run the sampler for 100 steps. Is the acceptance rate acceptable?

6) Plot the first two components as function of time, and do the same with their incremental sum, so with $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^i$ for N growing from 1 to 100. Also plot the two first components against each other to visualise the motion in the x_1, x_2 plane. Has the chain converged?

7) One of the problems with M-H are the small moves. Try to increase the step size my changing q to e.g. $N(x^{n-1}, 0.1)$. What happens to the acceptance rate? For what value of the width of the proposal density is the acceptance rate above 0.2? Has the convergence rate increased?

8) Using $q = N(x^{n-1}, 0.04)$ for each component, increase the number of steps to 1000. Convergence? Increase to 10,000 to see what has been missed.

Hybrid Monte-Carlo

To improve the exploration of the posterior pdf implement the Hybrid Monte-Carlo scheme.

1) To implement that scheme first derive the gradient of the posterior pdf to the state vector.

2) Extend your Matlab program with the hybrid scheme that for each step: draws a random number from $N(0, 0.01)$ to perturb the momentum vector, draws a random number from $U(0, 1)$ to decide if the time step for the leap-frog scheme is positive or negative, runs the leap-frog scheme with this time step, evaluates the Hamiltonian, calculates the acceptance criterion and accept or reject the new move.

3) Choose as initial velocity $v^0 = 0$, the leap-frog time step as 0.1, and the number of leap-frog steps as 10, for 100 steps. What can you say about the acceptance rate? Why is it so high?

4) Produce the same plots as for the standard M-H scheme above. What can you say about the exploration of state space?

5) Do we have convergence on the mean? Increase the number of steps to 1000, and to 10,000. What is your conclusion on the convergence rate? Include the discussion of the standard M-H in your discussion.

6) Can you think of methods to improve the convergence? (I know of no standard answer.)