Homework 2: MATLAB Programming and Debugging

Assigned on 9/17/2015 and Due on 10/1/2015

In the textbook, the Newton's method is provided for evaluating square root, $\sqrt{x}$, via an iterative procedure, i.e., $r_k = \frac{1}{2}\left(r_{k-1} + \frac{x}{r_{k-1}}\right)$.

Write a newtsqrt function to compute the square root of a number using this algorithm. Your function should encapsulate the code in the text book. Initialize your code as follows:

```
r = x;
rold = 2*r;
```

Q2-1 (5 points): The function should have the following inputs: (1) *x*, (2) the tolerance, *delta*, and (3) the maximum number of iteration, *maxit*. You should use the value of nargin at the beginning of your function file to find out how many input arguments were supplied. If a user of your code does not supply all the inputs, use the default value of 5.0e-6 for *delta* and 25 for *maxit*. Refer to demoArgs.m file about the use of nargin and nargout.

Q2-2 (5 points): In addition, your code should have the following:

(1) A prologue including summary, synopsis, inputs, and outputs of your code.
(2) Comments on important operations of your code.
(3) Diagnostic error message. For example, what if the input *x* is negative due to a typo and your calculation of *r* becomes negative?

Q2-3 (5 points):

Using the default tolerance and maximum number of iterations, run your code to compute the square root of *x* = 4, 400, and 4e-16. Since you know the exact solutions, you should be able to evaluate whether your code is correct.

Q2-4 (5 points): If your code does not provide correct solution, run your code again using the default maximum number of iterations but using smaller tolerance 5e-9, 5e-12, 5e-16, and 5e-24 to see if you can have the correct solution.

If the problem still cannot be solved, run your code again using the default tolerance but larger maximum number of iteration, say, 30, to see if the problem can be resolved.

At this moment, you may be able to identify the solution to the problem. Explain your finding.

Q2-5 (5 points): Redesign your code to solve this problem. For example, if the default parameter is not good, you should provide an error message to the users so that they can specify appropriate input values.

Q2-6 (5 points): Write a newtsqrtFor function to compute the square root of a number with the Newton's algorithm and using "for … end" instead of "while … end". How do you cause the for loop to terminate when the convergence criteria is met before the maximum number of iteration is performed?

Bonus Question: (20 points)
Study the MATLAB function *spy* by typing "edit spy". Write an m-file function, newspy(A), that
(1) plots red for negative values in the matrix
(2) plots blue for positive values in the matrix
(3) plots green for zeros values in the matrix
Build a matrix of your own and plot it using the newspy(A) function.