Iteration and recursion

Question 1 **10 marks**

Consider the nonlinear equation

$$f(x) = \sin(\pi x) - x^2$$

(**a**) Can you "solve the equation by hand", i.e. express the solution x^* explicitly in terms of elementary functions and the constant π?

No, this is impossible. There is no such explicit expression for the root x^*.

(**b**) Show that the equation $f(x) = 0$ has the same solution(s) as $g(x) = x$ if

$$g(x) = \frac{1}{2\pi}\sin(\pi x) - \frac{1}{2\pi}x^2 + x$$

$$\begin{aligned} g(x) = x \quad &\Leftrightarrow \\ \frac{1}{2\pi}\sin(\pi x) - \frac{1}{2\pi}x^2 + x = x \quad &\Leftrightarrow \\ \frac{1}{2\pi}\sin(\pi x) - \frac{1}{2\pi}x^2 = 0 \quad &\Leftrightarrow \\ \sin(\pi x) - x^2 = f(x) = 0 \end{aligned}$$

In fact, we can show that for any initial point in $(0, 2]$ the sequence $x_k = g(x_{k-1})$ converges to a unique solution x^*.

(**c**) Write a function that computes this sequence. Your function should:

- Take for input the an initial point, a maximal number of iterations and a tolerance for $|x_k - x_{k-1}|$ (the estimated error) and for $|f(x_k)|$ (the residual).
- Print to the command window the list of iterates, stopping when either the maximal number of iterations is reached or the estimated error **and** the residual are below their respective tolerance.
- Output the approximate solution, its estimated error and its residual.

See codes on Blackboard.

(**d**) If you used a `for` or `while` loop in (**c**), program a function with the same functionality using recursion (i.e. no explicit loops). If you used recursion in (**c**) then program a function with the same functionality using loops (i.e. no recursion).
Name your functions `iteration.m` for the version with a loop and `recursion.m` for the version without.

See codes on Blackboard.

(**e**) What happens if you take $x_0 = 0$? What happens if you take $x_0 < 0$?

Since $x = 0$ is solution, the error and residual will be zero after one iteration. For negative x_0, the iterates diverge and approach $-\infty$.

Question 2 **5 marks**

(**a**) Write a function that implements the following pseudo-code:
Input: f, f', x, ϵ, N.
Output: x^*.

1. Repeat N times:

(a) Set $y_1 = x$.
(b) Take one Newton step, starting from y_1. Call the result y_2.
(c) Take one Newton step, starting from y_2. Call the result y_3.
(d) Set

$$x = y_1 - \frac{(y_2 - y_1)^2}{y_3 - 2y_2 + y_1}$$

(e) Display $|f(x)|$.
(f) If $|f(x)| < \epsilon$ print "converged!", break.

2. Ouput $x^* = x$.

This algorithm is called Steffensen's iteration.

See codes on Blackboard.

(**b**) Test your routine on the problem

$$\exp(-x^2 + x) - \frac{1}{2}x = 1.0836 \quad \text{(with initial guess } x = 1\text{)}$$

Show that Newton iteration does not converge quadratically, but your new iterative algorithm does.

Newton iteration gives this sequence of residuals (with the code from Blackboard in the folder *The unknown interest rate problem from lecture 4*):

```
it=1 x=6.109333e-01 err=3.890667e-01 res=1.207459e-01
it=2 x=4.564079e-01 err=1.545255e-01 res=3.021620e-02
it=3 x=3.785844e-01 err=7.782351e-02 res=7.656717e-03
it=4 x=3.388631e-01 err=3.972124e-02 res=1.916870e-03
it=5 x=3.190605e-01 err=1.980267e-02 res=4.619788e-04
it=6 x=3.098768e-01 err=9.183678e-03 res=9.759948e-05
it=7 x=3.065214e-01 err=3.355400e-03 res=1.291781e-05
it=8 x=3.059175e-01 err=6.038397e-04 res=4.171155e-07
it=9 x=3.058967e-01 err=2.084388e-05 res=4.967664e-10
No convergence!
```

which is obviously not quadratic convergence. Steffensen's iteration, however, gives

```
Iteration 0 x=0.354600 err=6.453999e-01 res=3.735449e-03
Iteration 1 x=0.300633 err=5.396688e-02 res=7.347056e-05
Iteration 2 x=0.307404 err=6.770361e-03 res=3.268170e-05
Iteration 3 x=0.305888 err=1.515694e-03 res=1.750423e-07
Iteration 4 x=0.305897 err=8.772445e-06 res=4.418688e-14
Iteration 5 x=0.305897 err=2.213341e-12 res=0.000000e+00
```

and this sequence approaches the quadratic convergence $\epsilon_k \approx \epsilon_{k-1}^2$.
The reason for the slow convergence of Newton's method is, of course, that its condition number $1/|f'(x^*)|$ is large. The derivative of the function is about 0.02 at the solution...