

Name: _____

Math 426 Numerical Analysis (Bueler)

Friday, 11 October 2024

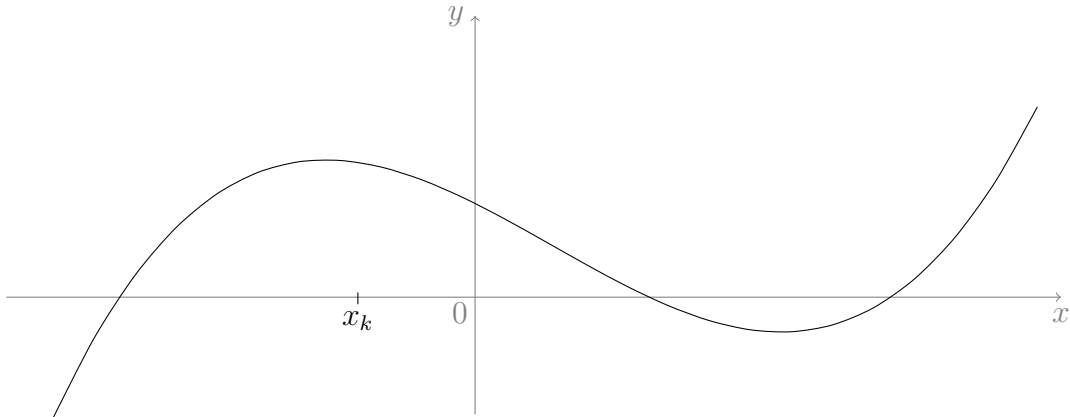
Midterm Exam

In class. No book, notes, electronics, or internet. 65 minutes. 100 points.

1. (a) (8 pts) We want to find a root of $f(x) = x^3 - 2x - 2$. Show that $a_0 = 0$ and $b_0 = 2$ is a bracket. Then apply two steps of the bisection method, reporting the new bracket at the end of each step. (*If you need these facts, you may use them: $f(0.5) = -2.875$ and $f(1.5) = -1.625$.*)

(b) (8 pts) For the same equation as in **(a)**, with $x_0 = 1$ and $x_1 = 2$, apply one step of the secant method, to compute x_2 .

- 2. (a) (5 pts)** A differentiable function $f(x)$ and an iterate x_k are shown on the axes below. Sketch, with appropriate labeling, how Newton's method determines the next iterate x_{k+1} .



- (b) (10 pts)** Suppose $\ell(x)$ is the linearization of $f(x)$ at x_k . Give a formula for $\ell(x)$ and then a formula for where it crosses the x -axis. Write the result as Newton's method in the box below.

$x_{k+1} =$

3. (15 pts) Solve the following system by Gauss elimination and back-substitution:

$$2x_1 + 3x_2 + 4x_3 = 6$$

$$-2x_1 - 2x_2 - 3x_3 = -5$$

$$4x_1 + 8x_2 + 8x_3 = 12$$

Show your steps in an organized way. It must be clear that you are following the algorithm we considered in class. (*Pivoting is not requested. The numbers are integers at every stage if you follow the algorithm. You do not need to use matrices.*)

4. (9 pts) Suppose we have used Gauss elimination with partial pivoting to factor some matrix A , so that $PA = LU$. Here P is a known permutation matrix, L is a known lower-triangular matrix, and U is a known upper-triangular matrix. Explain how to use this factorization to easily solve $A\mathbf{x} = \mathbf{b}$, assuming \mathbf{b} is also given, and identify what algorithms are needed.

5. (9 pts) The following Matlab code solves a unit lower diagonal linear system $L\mathbf{y} = \mathbf{b}$. That is, L is an n by n lower-triangular matrix, with ones on the diagonal, and \mathbf{b} is an n by 1 column vector. The output \mathbf{y} is also an n by 1 column vector. Please count **exactly how many floating point operations it does**, and give the answer as a function of n .

```
function y = lsolve(L, b)

n = length(b);           % get size
y = zeros(n,1);         % allocate space for answer
for i = 1:n              % count through the rows
    y(i) = b(i);         % start the numerator
    for j = 1:i-1
        y(i) = y(i) - L(i,j) * y(j); % use y(j) which are already known
    end
end
end
```

7. (a) (5 pts) Find all the fixed points x_* of $\varphi(x) = \frac{1}{4}x^2 + \frac{3}{4}$.

(b) (10 pts) Recall Theorem 4.5.1:

Theorem. Assume that $\varphi \in C^1$ and $|\varphi'(x)| < 1$ in some interval $[x_* - \delta, x_* + \delta]$ around a fixed point x_* of φ . If x_0 is in this interval then the fixed point iteration converges to x_* .

For the same function $\varphi(x)$ as in part (a), will the iteration

$$x_{k+1} = \varphi(x_k)$$

converge to each fixed point x_* for all x_0 near x_* ? Consider each x_* in turn.

8. (6 pts) Find the quadratic (degree two) Taylor polynomial for $f(x) = x^{1/3}$ using basepoint $a = 8$.

Extra Credit. (3 pts) Consider the Newton iteration to solve $f(x) = 0$. This iteration can be regarded as a fixed-point iteration. Do this, and then show using Theorem 4.5.1 that it is a fast-converging fixed point iteration. Clearly state what assumptions are needed so that it converges fast.

