

**Name:** Your Name Here

**Netid:** Your NetID Here

---

**Problem 1.** Consider the problem of solving a scalar nonlinear equation  $f(x) = 0$  where  $f : \mathbb{R} \rightarrow \mathbb{R}$  is a differentiable function. For a nonlinear problem there may be no solution or many solutions (e.g. a nontrivial quadratic equation might have 0, 1, or 2 solutions depending on the coefficients).

Recall Newton's method for solving such an equation: Given an initial guess  $x^{[0]}$ , iterate for  $\nu = 0, 1, \dots$  by setting  $x^{[\nu+1]} = x^{[\nu]} - f(x^{[\nu]})/f'(x^{[\nu]})$  until some convergence test is satisfied (see below), or until some maximum number of iterations is reached, since Newton's method doesn't always converge).

Newton's method is based on the Taylor series approximation

$$f(x) \approx f(x^{[\nu]}) + f'(x^{[\nu]})(x - x^{[\nu]}) \quad (1)$$

by setting the right hand side to zero and solving for  $x$  as our next approximation, i.e. we solve

$$0 = f(x^{[\nu]}) + f'(x^{[\nu]})\delta^{[\nu]} \quad (2)$$

and then set  $x^{[\nu+1]} = x^{[\nu]} + \delta^{[\nu]}$ .

- (a) Suppose we want to approximate  $\sqrt{a}$  for some  $a > 0$  using only basic arithmetic operations. This can be viewed as looking for a solution to  $f(x) = 0$  where  $f(x) = x^2 - a$ . Show that Newton's method reduces to:

$$x^{[\nu+1]} = \frac{1}{2} \left( x^{[\nu]} + a/x^{[\nu]} \right).$$

Note that if  $x^{[\nu]} = \sqrt{a}$  then so does  $a/x^{[\nu]}$ , while if one of these overestimates  $\sqrt{a}$  then the other underestimates and vice versa, so this makes sense.

- (b) Implement Newton's method for this problem in Matlab or Python and test it out for different starting values with  $a = 3$ .

As a convergence test, stop when  $\min(|\delta^{[\nu]}|, |f(x^{[\nu]}|)$  is below the tolerance **1e-14**. Note that these two quantities are related by (2) and since  $|f'(x^*)| = \mathcal{O}(1)$  it doesn't really matter which you check, but more generally if you search for a solution to a problem where  $|f'(x^*)|$  is either very small or very large, you cannot expect both of these quantities to be small in the presence of roundoff.

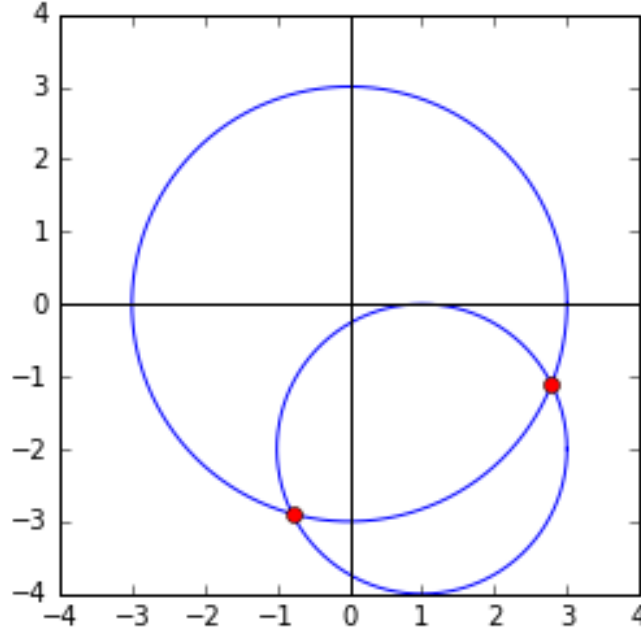
Illustrate the following with your experiments:

- You can converge to one of two different values depending on the starting guess.
- Once the value is sufficiently close to a root the convergence is *quadratic*, is  $|x^{[\nu+1]} - x^*| \approx C|x^{[\nu]} - x^*|^2$  for some constant  $C$ . This means the number of correct digits should roughly double each iteration. Print out 15 digits in your approximation each iteration to illustrate this.
- What happens if your initial guess is very close to 0?

---

**Problem 2.** Now consider a *nonlinear system* of equations  $f(x) = 0$  where  $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$ . For example, to find the intersection points of the two circles shown below we could solve

$$\begin{aligned}x_1^2 + x_2^2 &= 9, \\(x_1 - 1)^2 + (x_2 + 2)^2 &= 4\end{aligned}$$



Newton's method can be generalized based on (1), where now  $f'(x) \equiv J(x) \in \mathbb{R}^{m \times m}$  is the Jacobian matrix defined by

$$J_{ij} = \frac{\partial f_i(x)}{\partial x_j}$$

Now the equation (2) to solve in each iteration is a linear system of the form

$$J(x^{[\nu]})\delta^{[\nu]} = -f(x^{[\nu]}) \in \mathbb{R}^m$$

for the vector  $\delta^{[\nu]} \in \mathbb{R}^m$ , and then set  $x^{[\nu+1]} = x^{[\nu]} + \delta^{[\nu]}$ .

Implement this method to solve for both sets of intersection points shown by red dots in the plot above, by choosing different initial guesses. As a convergence test, stop when  $\min(\|\delta^{[\nu]}\|_\infty, \|f(x^{[\nu]})\|_\infty)$  is below the tolerance **1e-14**.

Check that you obtain quadratic convergence. Note that even though you don't have the exact solution to compare against, the vector  $\delta^{[\nu]}$  computed in each iteration is an approximation to the error, so print out  $\|\delta^{[\nu]}\|_\infty$  each iteration and observe that this decays quadratically. Also print out  $\|f(x^{[\nu+1]})\|_\infty$  each iteration.

(Note that if the matrix  $f'(x^{[\nu]})$  is singular at any iteration then Newton's method breaks down, as does the scalar algorithm if  $f'(x^{[\nu]}) = 0$ . If  $f'(x^*)$  is singular but is nonsingular for nearby  $x$  values, then the algorithm will generally still converge but might only be linearly convergent in this case, or perhaps even slower.)

---

**Problem 3.**

Exercise 24.1 in the book. For part (c) you might want to split both the eigenvalue and eigenvector into real and imaginary parts.

---

**Problem 4.** Exercise 24.2. For part (b) consider a deformation where the diagonals are left alone and the off-diagonals are all multiplied by a scalar  $\alpha$ , and then consider what happens to the eigenvalues of this matrix  $A_\alpha$  (and to the Gerschgorin circles) as  $\alpha$  varies from 0 to 1.

---

**Problem 5.**

- (a) The Gerschgorin disks discussed in the previous problem are sometimes called the “row disks” since the radius is given by the sum over other elements in the same row as the center  $a_{ii}$ . We can also define “column disks” based on the sum of the off-diagonals in the same column as  $a_{ii}$ . Explain why the same result as in Exercise 24.2 holds if row disks are replaced by column disks, even in the complex case (recall that the eigenvalues of  $A^*$  are the complex conjugates of the eigenvalues of  $A$ ).

- (b) Consider the matrix

$$A = \begin{bmatrix} 3 & 0 & -1 \\ 2 & -2 & i \\ 0 & 1 & 3i \end{bmatrix}$$

Sketch the Gerschgorin row disks and column disks and comment on which set of disks can be used to deduce that  $A$  is nonsingular.

---

**Problem 6.** Use the Gerschgorin theorem to show that if  $A$  is a hermitian matrix that is strictly column diagonally dominant, and if all the diagonal elements of  $A$  are positive,  $a_{ii} > 0$ , then  $A$  is hermitian positive definite.

---

**Problem 7.**

Let  $A \in \mathbb{C}^{m \times m}$  be any square matrix. Then the  $2m \times 2m$  matrix

$$B = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}$$

is hermitian.

Show that the eigenvalues of  $B$  are the  $2m$  values  $\pm\sigma_i$  where  $\sigma_i$  are the  $m$  singular values of  $A$ . Also display the factorization of  $B = Q\Lambda Q^*$  in block form, where

$$\Lambda = \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix}$$

and  $Q$  is a block  $2 \times 2$  unitary matrix. The blocks in  $Q$  can be written in terms of the matrices  $U$  and  $V$  from the SVD of  $A = U\Sigma V^*$ .

**Hint:** Consider  $B$  applied to the following vector, obtained from the  $j$ th columns of  $U$  and  $V$ :

$$\begin{bmatrix} v_j \\ u_j \end{bmatrix} \in \mathbb{C}^{2m}$$

to show that this gives one set of eigenvectors of  $B$ , and then find another set.

---

**Problem 8.** The Rayleigh Quotient iteration of Algorithm 27.3 can be used to approximate an eigenvalue and eigenvector of a matrix. It doesn't say how to choose  $v^{(0)}$  in general to converge towards a particular eigenvalue. One approach, if you know some approximation  $\mu$  to an eigenvalue, is to first take one or more steps of inverse iteration (Algorithm 27.2) with a fairly arbitrary  $v^{(0)}$ , and then use the result of that as an initial vector in Algorithm 27.3.

Consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- (a) Use the `eig` command in Matlab or Numpy to compute the “exact” eigenvalues.
- (b) Try using the approach described above, using a single step of inverse iteration (which amounts to using  $\mu$  in place of  $\lambda^{(0)}$  in the first iteration of 27.3) to see if you can converge to all three of the eigenvalues of  $A$ .

In particular, try using  $v^{(0)} = [1, 1, 1]^T$  and values of  $\mu = -10, 100$  and values close to 0. For the  $\lambda = 0$  eigenvalue you may find you need to use a different initial  $v^{(0)}$ .

Can you observe the expected cubic convergence?