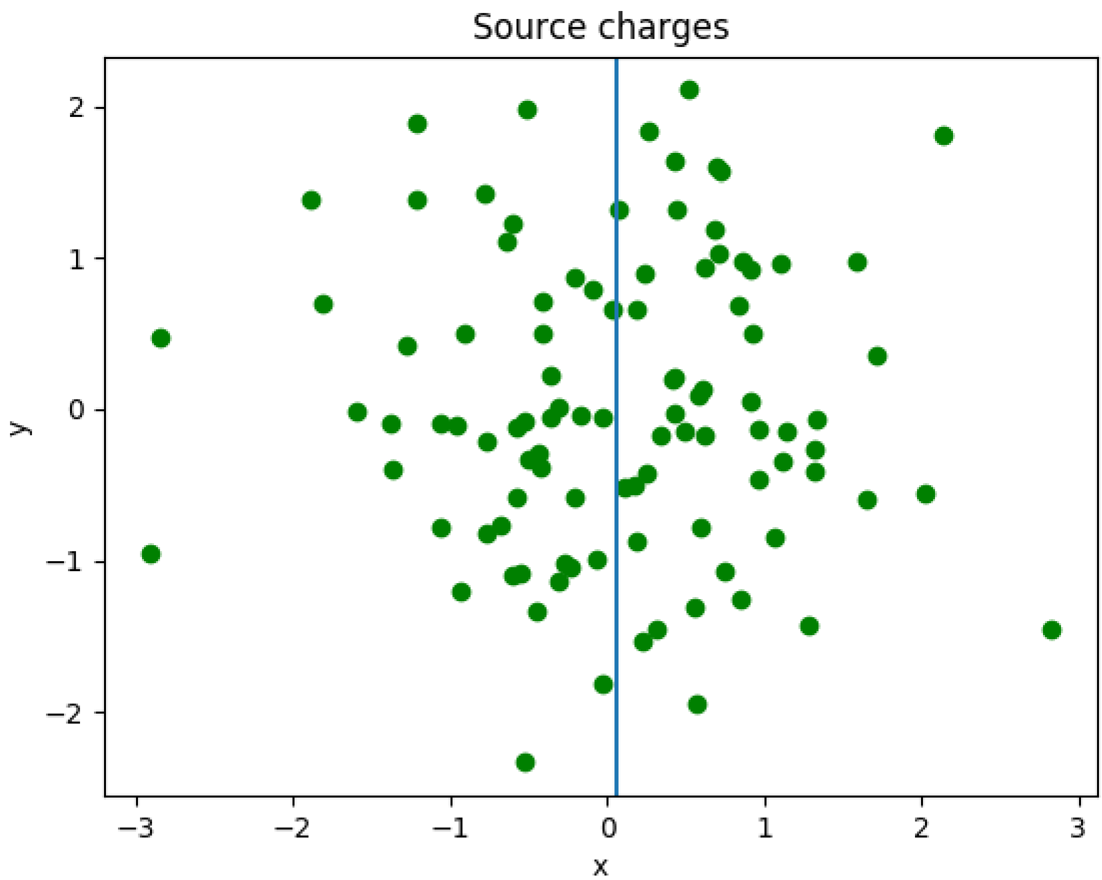
Semiseparable Matrices

10 points

The electric potential from a point charge, Q, at a distance r from a charge is given as

$$V = \frac{1}{4\pi\epsilon_0}\frac{Q}{r}.$$

Now, consider computing the potential from a collection of point charges, $q_1, q_2, \ldots, q_n$ at the locations of each of the point charges. Suppose that the point charges are numbered in such a way that particles $q_1, q_2, \ldots, q_j$ are left of the midpoint, and that $q_{j+1}, \ldots, q_n$ are to the right of the midpoint of the particles.



The interaction matrix comprised that computes the potential at $q_1, \ldots, q_n$ given a vector of charges $Q_1, \ldots, Q_n$ has a special structure known as a *semiseparable* (or SS for short). SS matrices show up in a variety of different applications from computing potentials, to PDE solvers, to multivariate statistics in the form of the covariance matrix.

In this problem, you will solve a linear system with an SS matrix of the following form using the Sherman-Morrison formula

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

In general, semiseparable matrices have diagonal blocks which have full rank and off-diagonal blocks which have (approximately) low rank. In this particular problem, we assume that the SS matrix has off-diagonal blocks with rank 1 (i.e. an outer product of two vectors). In this particular problem, A_{11} is a $n \times n$ matrix, and A_{22} is a $m \times m$ matrix. x_1 and b_1 are lenght n vectors. x_2 and b_2 are length m vectors. $A_{12} = \boldsymbol{u}\boldsymbol{v}^T$ where $\boldsymbol{u}$ is a length n vector and $\boldsymbol{v}$ is a length m vector. Further, $A_{21} = \boldsymbol{w}\boldsymbol{p}^T$, where $\boldsymbol{w}$ is a length m vector and $\boldsymbol{p}$ is a length n vector.

What do I need to do?

1. Using Gaussian-Elimination on the block-structure of the matrix and applying the Sherman-Morrison formula, solve the linear system for `x_1` and `x_2`.
2. Solving a semiseparable system in this fashion has some perfomance benefits. In a comment block, please describe the expected cost, in terms of n and m, for solving the entire system without taking advantage of the semiseparable structure of the matrix, such as by calling `la.solve`.
3. In another comment block, specify the expected cost, in terms of n and m, for solving the system by taking advantage of the semiseparable structure of the matrix.

NOTE: You may assume that the cost of solving a generic $p \times p$ system is $cost \approx Cp^3$, where C is some constant that does not depend on the problem size. (You may also find a detailed operation count for LU in our textbook.) You need not specify C, you may simply use it as a known constant. For your cost analysis, please only consider the leading order terms, e.g., you may simplify $2nm^2 + 3n^2m + n^3 + m^3 + 457nm + 1589n$ as $2nm^2 + 3n^2m + n^3 + m^3$.

INPUT:

- `A11` : A numpy array of the top left $n \times n$ block of A.
- `A11_inv` : A numpy array of the inverse of the A_{11} block of A.
- `A22` : A numpy array of the bottom right $m \times m$ block of A.
- `A22_inv` : A numpy array of the inverse of the A_{22} block of A.
- `u` , `p` : numpy arrays of size n.
- `v` , `w` : numpy arrays of size m.
- `b` : A numpy array of size $n + m$ with the right-hand side of the linear system.

OUTPUT:

- `x1` : A numpy array of size n containing the first n entries of the solution.
- `x2` : A numpy array of size m containing the last m entries of the solution.

Problem set-up code (click to view)

Starter code (click to view)

Answer*

```
#2: B1 = b1 - A12A22-1*b2
B1 =  b1 - np.dot(u,np.dot(v.T,np.dot(A22_inv,b2)))

#3: yy = A11-1 * B
yy = np.dot(A11_inv,B1)
```

```

#4: x1 = y + ((pT y)/(1 - pT z))z
denom = 1 - np.dot(p.T, zz)
pty = np.dot(p.T,yy)
x1 = yy +  np.dot(pty/denom,zz)

'''
To solve x1 x2 directly we may need to get the inverse of A(n+m)*(n+m).
So that would be O((n+m)^3). Then we multiply A^-1 with b which give us O
The total leading term would be O(n^3+m^3+3nm^2+3n^2m)
'''
'''
If we take advantage of the semiseperarable matrix, the above four steps
#1:m^2+m+n+n^2
#2:m^2+m+n
#3:n^2
#4:n+n+m+n
```

Press F9 to toggle full-screen mode. Set editor mode in user profile (/profile/).

Overall grade

The overall grade is 100%.

Autograder feedback

The autograder assigned 9/9 points.

Your answer is correct.

Here is some feedback on your code:

- 'x1' looks good
- 'x2' looks good
- Execution time: 0.3 s -- Time limit: 10.0 s

Your code ran on relate-02.cs.illinois.edu.

Human feedback

The human grader assigned 1/1 points.

The following code is a valid answer:

```
import numpy as np
#split b into b1 and b2
n = len(u)
m = len(v)
b1 = b[:n]
b2 = b[n:]

b2 -= np.inner(p,A11_inv@b1)*w
z=A22_inv@w
y=A22_inv@b2
my_v = np.inner(p,A11_inv@u)*v
x2 = y+(np.inner(my_v,y)/(1-np.inner(my_v,z)))*z
x1 = b1 - np.dot(v,x2) * u
x1 = A11_inv@x1
```