Programming Assignment 2

Assigned: June 6
Due: June 20

Problem 1 (5 points)

The process that you considered in programming assignment 1, problem 2, is invertible if there are an odd number of points, though not if there are an even number of points. That is, if k is odd, then given a sequence of k points in the plane $\vec{u}_1 \dots \vec{u}_k$ you can find a sequence $\vec{v}_1 \dots \vec{v}_k$ such that

$\vec{u}_1$ is the midpoint of $\vec{v}_1$ and $\vec{v}_2$.
$\vec{u}_2$ is the midpoint of $\vec{v}_2$ and $\vec{v}_3$.
. . .
$\vec{u}_{k-1}$ is the midpoint of $\vec{v}_{k-1}$ and $\vec{v}_k$.
$\vec{u}_k$ is the midpoint of $\vec{v}_k$ and $\vec{v}_1$.

Write a MATLAB function `InvertMidpoints(Q)` which takes as argument a polygon `Q`, represented in the form used in programming assignment 1, and returns the polygon `P` such that each vertex in `Q` is the midpoint of a side in `P`.

Thus, for example, if `Q` is the array

$$\texttt{Q} = \begin{bmatrix} 1 & 3 & 1 & 2 & 3 \\ 4 & 4 & 3 & 6 & 3 \end{bmatrix}$$

then `InvertMidpoint(Q)` should return

$$\texttt{P} = \begin{bmatrix} 0 & 2 & 4 & -2 & 6 \\ 0 & 8 & 0 & 6 & 6 \end{bmatrix}$$

Hint: First, notice that the x and y coordinates in this problem are completely decoupled. That is, if we write $\vec{u}_i = \langle x_i, y_i \rangle$, and $\vec{v}_i = \langle a_i, b_i \rangle$, then we have

$x_1 = (a_1 + a_2)/2.$
$x_2 = (a_2 + a_3)/2.$
. . .
$x_{k-1} = (a_{k-1} + a_k)/2.$
$x_k = (a_k + a_1)/2.$

and the identical equations relates the y's to the b's. Thus, for example, for $k = 5$, the 5-dimensional vectors $\vec{x}, \vec{y}, \vec{a}, \vec{b}$ satisfy the equations $\vec{x} = C\vec{a}$, $\vec{y} = C\vec{b}$ where C is the matrix of coefficients,

$$C = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 0 & 1/2 \end{bmatrix}$$

So you can write the program by (a) creating the matrix C for size k; (b) solving the two systems of equations; (c) putting the two solutions together. (Do NOT write code to solve systems of linear equations; use the built in MATLAB solver.)

Problem 2

Note: This problem is long, but the code is actually *very* short; perhaps 20 lines in total.

Suppose that A and B are electrically charged objects, located at points $\mathbf{p}_A$ and $\mathbf{p}_B$ with charges

Q_A and Q_B. Then the force $\vec{F}_A(B)$ that B exerts on A is the vector

$$\vec{F}_A(B) = \frac{Q_A \cdot Q_B}{|\mathbf{p}_A - \mathbf{p}_B|^2} \cdot \frac{\mathbf{p}_A - \mathbf{p}_B}{|\mathbf{p}_A - \mathbf{p}_B|}$$

In the above product, the first factor is the magnitude of the force, which is the product of the charges divided by the distance squared; the second factor is the direction of the force, which is the direction from B to A.

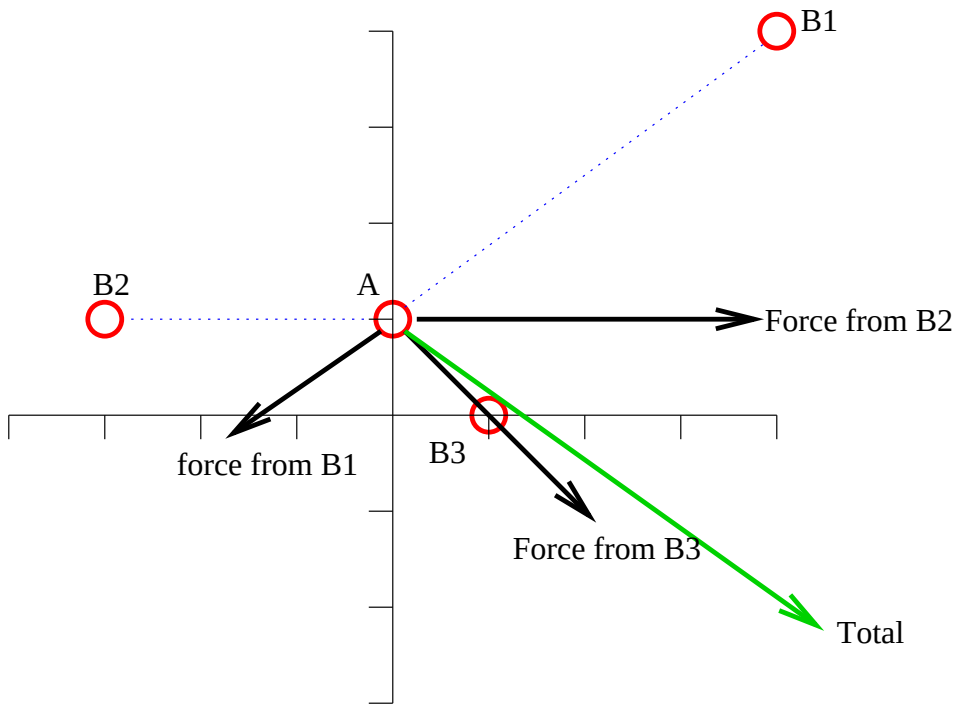
If there are several objects $B_1 \ldots B_k$ exerting a force on A, then the total force on A is the sum of the forces:

$$\vec{F}_A(\{B_1 \ldots B_k\}) = \sum_{i=1}^{k} \vec{F}_A(B_i)$$

If the charge on A and the position of all the charges is fixed, then the net force is a linear function of vector of charges $\langle \vec{Q} = Q_1 \ldots Q_k \rangle$.

For instance, in two dimensions, we could have the following situation, illustrated in the picture.

Object	Location	Charge	$\lvert\mathbf{p}_A - \mathbf{p}_B\rvert$	Magnitude of $\vec{F}_A(B)$			$\vec{F}_A(B)$		
A	$\langle 0, 1\rangle$	1		—			—		
B_1	$\langle 4, 4\rangle$	50	5	$50/25$	$=$	2	$2 \cdot \langle -4, -3\rangle/5$	$=$	$\langle -1.60, -1.20\rangle$
B_2	$\langle 1, 0\rangle$	-6	$\sqrt{2}$	$-6/2$	$=$	-3	$-3 \cdot \langle -1, 1\rangle/\sqrt{2}$	$=$	$\langle 2.12, -2.12\rangle$
B_3	$\langle -3, 1\rangle$	36	3	$36/9$	$=$	4	$4 \cdot \langle 3, 0\rangle/3$	$=$	$\langle 4.00,\ 0.00\rangle$
Total									$\langle 4.52, -3.32\rangle$



Problem 2.A (2.5 points)

Write a function `function F = ForceMatrix(PA,PB)` where

- `PA` is a 2-dimensional column vector of the coordinates of object A of charge 1.
- `PB` is a $2 \times k$ matrix, where the ith column, `PB[:,i]` is the coordinates of object B_i.
- `F`, the value returned, is the $2 \times k$ matrix with the property that for any vector of charges $\vec{\mathtt{Q}}$, the value $\mathtt{F} \cdot \vec{\mathtt{Q}}$ is the net force on A.

For instance, in the above example, we could call

```
> PA = [0;1];
> PB = [4,1,-3; 4,0,1];
> F = ForceMatrix(PA,PB)
F =
    -0.0320    -0.3536   0.1111
    -0.0240     0.3536   0
> QB = [50; -6; 36];
> F*QB
ans =
    4.5213
   -3.3213
```

Problem 2.B: 0.5 points

Write the following two functions: `function TF = TotalForce(PA,PB,QB)` and `C = PossibleCharge(PA,PB,TF)`. In both of these `PA, PB` are the same as in problem 1. In `TotalForce`, the input `QB` is a column vector of the charges on B and the value returned `TF` is the total force on A, a column vector. In `PossibleCharge`, `TF` is the total force as a column vector and the value returned `C` is a possible charge vector that would give rise to that force. If $k > 2$ then there are multiple possible answers but your code only has to return one of these. For example, using the same values of `PA,PB,QB` we could write,

```
> TF = TotalForce(PA,PB,QB)
TF =
    4.5213
   -3.3213

> C = PossibleCharge(PA,PB,TF)
         0
   -9.3941
   10.8000

>> TotalForce(PA,PB,C)
ans =
    4.5213
   -3.3213
```

Having done problem 2.A, each of these functions should consist of one quite simple line of MATLAB. The code for `TotalForce` should always work, unless A is at the same position as one of the B_i's. If all the points, are collinear, then there may not exist any solution for `PossibleCharge`.

Problem 2.C: 2 points

Suppose as before there are k fixed charges $B_1 \dots B_k$ in the plane. You know the locations, but not the value of the charges, and you want to find out the value of the charges. A way to do this is as follows: You take an object A with charge 1, you put it at various points in the plane, and you measure the net force on it.

Write a function `function C = FindCharges(PA,PB,TF)` where

- `PA` is a $2 \times w$ matrix, where the ith column, `PA[:,i]` is the coordinates of the ith placement of the test charge A. The dimension w is the number of different placements you try.
- `PB` is the locations of the charges $B_1 \dots B_k$, as above.
- `F` is a $2 \times q$ matrix, where the ith column `F[:,i]` is the total force on A in its ith placement.
- The value returned `C` is the k-dimensional column vector of charges on the B_i.

Hint: Look up the Matlab `reshape` function.

For instance, in the above example, we could call

```
> PA = [0,2;1,0];
> PB = [4,1,-3; 4,0,1];
> TF(:,1) = TotalForce(PA(:,1),PB,QB);
> TF(:,2) = TotalForce(PA(:,2),PB,QB);
> C = FindCharges(PA,PB,TF)
C =
   50.0000
   -6.0000
   36.0000
```