Mandatory Assignment 1

Jørgen D. Tyvand

September 14, 2015

Case:

We are to "calculate the velocity potential and the added mass forces, for a circle, ellipse, a square and a rectangle, moving laterally, and with rotation.".

Deriving the set of equations:

We start from the integral equation for a body with boundary C. We want to calculate the potential along the body, so the source point as well as the field points are located on the surface C. The integral equation in 2D for a source point on the surface can be written as

$$-\pi\phi(\mathbf{x}) + \int\limits_C \phi(\xi)\,\frac{\partial}{\partial n_\xi} lnr\,\mathrm{d}l_\xi = \int\limits_C lnr\,\frac{\partial\phi}{\partial n_\xi}\,\mathrm{d}l_\xi$$

Since we want to discretize the problem for N segments along the surface, the integrals can be assumed to be sums over the integrals, with $\phi(\xi)$ constant and $\frac{\partial\phi}{\partial n_{\mathbf{x}}}$ known on each segment :

$$-\pi\phi(\mathbf{x}) + \sum_{n=1}^{N}\phi_n(\xi)\int\limits_{C_n}\frac{\partial}{\partial n_\xi} lnr\,\mathrm{d}l_\xi = \sum_{n=1}^{N}\frac{\partial\phi_n}{\partial n_\xi}\int\limits_{C_n} lnr\,\mathrm{d}l_\xi$$

The integral on the left hand side can be simplified by introducing the complex variable $z = re^{i\theta} = x + iy$, and saying that $lnr = Re(lnz)$. We also have

$\frac{\partial}{\partial n_{\mathbf{x}}} = n_1\frac{\partial}{\partial x} + n_2\frac{\partial}{\partial y}$, giving

$$\int\limits_{C_n}\frac{\partial}{\partial n_\xi} lnr\,\mathrm{d}l_\xi = Re\int_A^B\left(n_1\frac{\partial}{\partial x} + n_2\frac{\partial}{\partial y}\right) lnz\,\mathrm{d}l_\xi$$

Using the following results and idientities

$\frac{\partial}{\partial x} lnz = \frac{\mathrm{d}}{\mathrm{d}z} lnz \frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}z} lnz = \frac{1}{z}$

$\frac{\partial}{\partial y} lnz = \frac{\mathrm{d}}{\mathrm{d}z} lnz \frac{\mathrm{d}z}{\mathrm{d}y} = i\frac{\mathrm{d}}{\mathrm{d}z} lnz = \frac{i}{z}$

$n_2 dl = dx, \qquad -n_1 dl = dy$

Inserted into the above integral

$$\int\limits_{C_n}\frac{\partial}{\partial n_\xi} lnr\,\mathrm{d}l_\xi = Re\int_A^B\left(\frac{n_1 + in_2}{z}\right)\mathrm{d}l_\xi = Re\int_A^B\frac{i^2dy + idx}{z}\mathrm{d}l_\xi = Rei\int_A^B\frac{\mathrm{d}z}{z} =$$

$$Re(i\,lnz)\big|_0^1 = Re(i(lnr + i\theta)\big|_0^1 = -(\theta_B - \theta_A)$$

We also define that $-(\theta_B - \theta_A) = \pi$ when $\mathbf{x_0}$ is on the segment.

For the right hand integral, I have used a simple trapezoidal rule method, giving

$\int\limits_{C_n} lnr \, \mathrm{d}l_{\mathbf{x}} = \frac{lnr_A + lnr_B}{2} dl$

The final simplified equation can then be written as

$$-\pi\phi(\mathbf{x}) + \sum_{n=1}^{N} \phi_n(\xi)(-(\theta_{B_n} - \theta_{A_n})) = \sum_{n=1}^{N} \frac{\partial \phi_n}{\partial n_\xi} \frac{lnr_A + lnr_B}{2} dl$$

Since we compute the values of ϕat each segment using the value at other segments, this gives rise to a set of N+1 coupled equations, which we must solve as a matrix/vector problem.

The numerical program:

The complete code of the numerical program is listed at the end of this document. Comments on the functionality of the individual parts of the program is given in the code.

The process in the program is to create N+1 points along the geometry of the body, where the first and last point will be the same. The midpoints of each segment are then defined to be the point petween each of the previously generated points, giving N midpoints (and segments). The X and Y locations of these points are stored, and used to calculate the length dl of the segment, as well as the normal vector and its components. The book gives $\frac{\partial \phi_i}{\partial n} = n_i$ for i = 1, 2, 3 and $\frac{\partial \phi_i}{\partial n} = (\mathbf{r}' \times \mathbf{n})_{i-3}$ for i = 4, 5, 6, so the normal component n_6 is calculated as $xn_2 - yn_1$ at each midpoint.

An empty N-by-N matrix is then created to hold the values of all the angles from the left hand side of the equation, as well as thre arrays holding the values for the trapeziodial integral in the three non-zero directions (1, 2 and 6). The program then runs a loop N times, each time calculating the distance from the current segment to each of the end points of all the other segments. This is then used to calculate the angle between the distance vectors, as well as the right hand side. These values are finally added to the matrix and the three arrays.

A summary of the results:

A running of the program for a circle of raduis 1, an ellipse with a = 1, b = 2, and a square with sides 2a is printed below:

```
Case: circle with radius 1
[[ 3.1405  0.      0.    ]
 [ 0.      3.1405  0.    ]
 [ 0.      0.      0.    ]]
Calculated in 4.933230 seconds

Case: ellipse with a = 2, b = 1
[[  3.1408   0.       0.    ]
 [  0.      12.5598   0.    ]
 [  0.       0.       3.5315]]
Calculated in 4.883140 seconds

Case: square with sides 2
[[ 4.7506  0.      0.    ]
 [ 0.      4.7506  0.    ]
 [ 0.      0.      0.723 ]]
Calculated in 4.918705 seconds
```

The wording of the assignment puzzled me a bit, as it asked for the added mass forces, as well as the cross-coupling added mass coeeficients. If I'm not fully mistaken, the cross-coupling coefficients for symmetric 2D bodies are all zero, as well as the coefficients m_{33}, m_{44} and m_{55}. The added mass force is defined as $F_j = -\dot{U} m_{ji}$, but we are not given a U, so i have assumed this to be of size unity. The added mass forces would then be as follows (for the cases listed above)

For circle: $F_1 = -3.1405 \qquad F_2 = -3.1405$
For ellipse: $F_1 = -3.1408 \qquad F_2 = -12.5598 \qquad F_6 = -3.5315$
For square: $F_1 = -4.7506 \qquad F_2 = -4.7506 \qquad F_6 = -0.723$