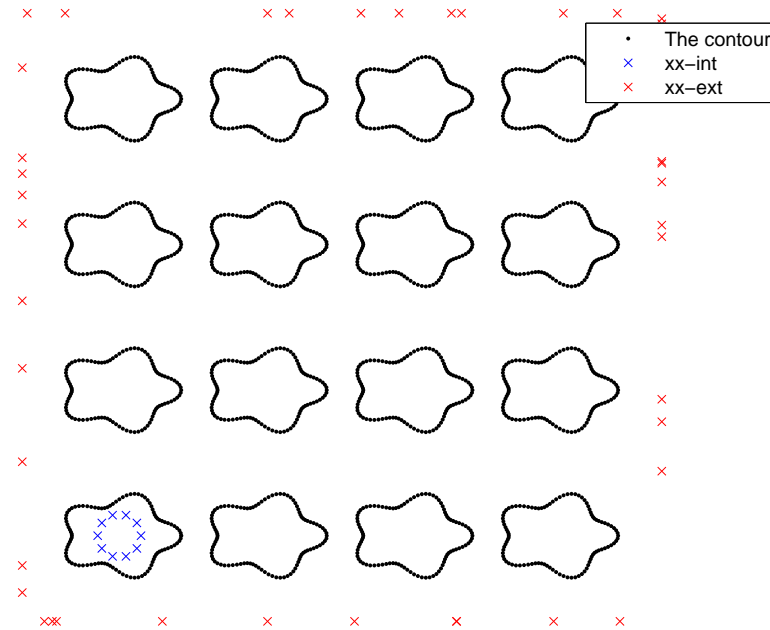


## APPM4720/5720 — Homework 7 — preliminary version

Consider a multi-component contour  $\Gamma$  as shown in black in the figure below:



The contour has  $N_{\text{star}}$  points per star, and there are  $N_{\text{side}} \times N_{\text{side}}$  stars. (In the figure show,  $N_{\text{star}} = 100$  and  $N_{\text{side}} = 4$ ). In other words, the total number of points is

$$N = N_{\text{side}}^2 N_{\text{star}}.$$

A precise definition of the contour is given in the file `main_hw07`.

Let  $\Omega$  denote the domain *exterior* to  $\Gamma$  and consider the Neumann problem:

$$\text{(BVP)} \quad \begin{cases} -\Delta u(x) = 0, & x \in \Omega, \\ u_n(x) = g(x), & x \in \Gamma. \end{cases}$$

Your task is to convert (BVP) to a Boundary Integral Equation and to solve this integral equation numerically using a Krylov method.

**Problem 7.1:** Try to determine via computational experiments how the condition number of your discretized BIE depends on  $N_{\text{star}}$  and  $N_{\text{side}}$ . For this part of the problem, consider only  $N_{\text{star}}$  and  $N_{\text{side}}$  that are small enough that you can compute the SVD of the coefficient matrix via the Matlab `svd` command. You may also want to experiment with the `cond` command.

**Problem 7.2:** Solve the linear system you obtain upon discretization using two different methods:

- (1) The Matlab backslash command. The complexity is  $O(N^3)$  complexity.
- (2) GMRES. The complexity is  $O(N_{\text{iter}} \times N^2)$  where  $N_{\text{iter}}$  denotes the number of iterations.

Provide a plot of  $N_{\text{iter}}$  versus  $N$  for a couple of different values of  $N_{\text{star}}$ . Does GMRES ever outperform the backslash command for problems sizes that are small enough to fit in your memory?

**Problem 7.3 (extra credit):** Repeat problem 7.2, but now include an option where the FMM is used to accelerate the matrix-vector multiply. You may use the FMM code posted on the webpage for the class, but note that you will need to write a wrapper! The complexity of the resulting method should be  $O(N_{\text{iter}} \times N)$ . Does your code seem to achieve this?

**Problem 7.4 (extra credit):** Develop tools for plotting the solution to (BVP). You could create “quiver” diagrams of the gradient of  $u$ , or 3D plots of  $u$  in  $\Omega$ . A particularly simple thing to do would be to just compute  $u|_{\Gamma}$  and use the `plot3` command to plot  $u$  on the boundaries.

Now create some “interesting” problems, plot the solution, and submit your plot along with a brief description of the problem you are solving.