Take-home Final exam. Due May 18, 10 AM

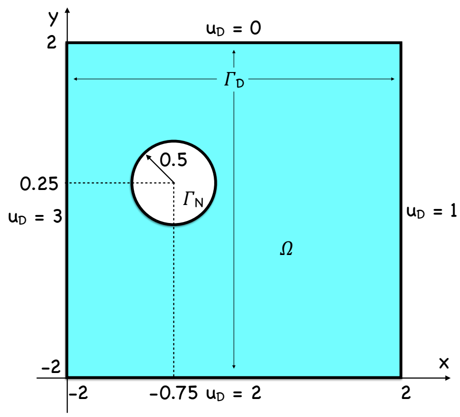
1. Consider the following Boundary Value Problem (BVP) in 2D:

$$\Delta u = 0, \quad (x,y) \in \Omega, \tag{1}$$

$$\left.\frac{\partial u}{\partial \hat{n}}\right|_{\Gamma_N} = 0, \tag{2}$$

$$u|_{\Gamma_D} = u_D, \tag{3}$$

where the domain Ω, its boundary $\partial\Omega = \Gamma_N \cup \Gamma_D$, and the Dirichlet boundary condition function u_D are shown in the figure below.



Set up a linear system of algebraic equations for the FEM solution of this problem and solve it using an iterative method studied in AMSC/CMSC661 (you need to write the solver yourself). Use `tol = 1e-12` for your linear solver. Make your program plot the following figures:

- with the computed solution (use `trisurf`);
- with the residual plotted versus the iteration number for the linear solver. Set `'YScale'` logarithmic.

Submit a SINGLE(!) .m file with your code. Your .m file should include all functions called except for, possibly, the one for triangulation if it is `mesh2d`. *Note, if you use* `mesh2d`*, your input vector with boundary points should not contain repeated points* ☺.

2. Consider the following Initial and Boundary Value Problem (IBVP) in 2D:

$$u_t = \Delta u + 1, \quad (x,y) \in \Omega = \{(x,y) \in \mathbb{R}^2 \mid 1 < r < 2\}, \tag{4}$$
$$u|_{t=0} = r + \cos(\phi), \tag{5}$$
$$u|_{r=1} = u|_{r=2} = 0, \tag{6}$$

where r and ϕ are the polar coordinates. Solve this problem using the finite element method and a scheme based on the trapezoidal rule:

$$u_{n+1} = u_n + \tfrac{1}{2}\Delta t\,(\Delta u_{n+1} + \Delta u_n) + \Delta t.$$

(a) Derive equations for the weak and the FEM solutions of the IBVP (4)-(6) analogous to Eq. (13) and the two unnumbered equations right below it in Section 9 on page 127 in Remarks around 50 lines of Matlab: short finite element implementation. Use time step `dt = 0.01`.

(b) Make your program plot the following figures:

- with the computed solution at $t = 0.1$ (use `trisurf`);
- with the computed solution at $t = 1$ (use `trisurf`);
- with the computed solution at time $t = 1$ as a function of r. You can do it e.g., as follows:

```
u = U(:,N+1); % N+1 corresponds to t=1.
r = sqrt(coordinates(:,1).^2 + coordinates(:,2).^2);
[rsort,isort] = sort(r,'ascend');
usort = u(isort);
plot(rsort,usort,'Linewidth',2);
```

At $t = 1$, the function u will virtually reach the stationary solution $\Delta u + 1 = 0$ satisfying the BC (6). This stationary solution can be found exactly:

$$u(r) = \frac{1 - r^2}{4} + \frac{3\log(r)}{4\log 2}. \tag{7}$$

Plot the graph of the exact stationary solution (7) in the same figure.

Submit a SINGLE(!) .m file with your code and a .pdf file with the requested equations for FEM. Your .m file should include all functions called except for, possibly, the one for triangulation if it is `mesh2d`. *Note, if you use* `mesh2d`*, your input vector with boundary points should not contain repeated points ☺.*

3. (a) Write a matlab code that builds the Daubechies D4 scaling function ϕ by a recursive procedure.

   (b) and constructs the D4 wavelet ψ.

   The D4 scaling function ϕ and the wavelet ψ are supported on the interval $[0, 3]$. The low-pass filter coefficients are

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}},$$

   the high-pass filter coefficients are given by $b_k = (-1)^k h_{3-k}$, $k = 0, 1, 2, 3$.

   The recursive procedure for building ϕ can be devised e.g. as follows. Start with $\phi(0)\phi(3)$. Find $\phi(1)$ and $\phi(2)$ as explained in the notes by V. Balan and C.Condea. Then, at the recursion level p, ϕ is found on the set $\mathcal{D}_p \backslash \mathcal{D}_{p-1}$, where

$$\mathcal{D}_p = \{k * 2^{-p} \mid 1 \le k \le 3 * 2^p - 1\}, \quad p = 1, \dots, p_{\max},$$

   using the formulas

$$\phi(x) = \sqrt{2} \sum_k h_k \phi(2x - k), \quad \psi_k = \sqrt{2} \sum_k b_k \phi(2x - k).$$

   Use the maximal recursion level $p_{\max} = 12$.

   Normalize ϕ and ψ so that

$$\int_{\infty}^{\infty} |\phi|^2 dx = \int_{\infty}^{\infty} |\psi|^2 dx = 1.$$

   Make your code plot graphs of ϕ and ψ on the same figure.

   Submit a SINGLE(!) .m file with your code.