Problem 1 (PDE Toolbox)
On the last exercise sheet we like to “play” around with the PDE Toolbox provided by Matlab to get some understanding of how a Finite Element approximation of a PDE looks like.

We approximate Poisson’s equation (an elliptic PDE)
\[-\Delta u = f\]
on the domain \(\Omega\).

Because Matlab is expecting a function of type
\[-\nabla \cdot (c\nabla u) + au = f\]
we first need to set \(c\) and \(a\) to zero. Then we can define the right hand side of the equation and set \(f = 1\).

The mesh can be defined using the \([p,e,t] = \text{initmesh}(g)\) function. Where \(p\), \(e\) and \(t\) define the points, edges and triangles, respectively. \(g\) is either a geometry matrix (see documentation for details) or the name of a geometry file provided by Matlab. The mesh can be refined by the function \(\text{refinemesh}\) and plotted by \(\text{pdemesh}\).

After a mesh is generated we can assemble the stiffness matrix and solve Poisson’s equation using the Finite Element Method by using the function
\[u = \text{assempde}(b,p,e,t,c,a,f);\]
\(p\), \(e\), \(t\), \(c\), \(a\) and \(f\) are defined as above. \(b\) defines the boundary conditions. For this problem we can, again, use functions provided by Matlab, e.g. \(\text{circleb1}\) defines the Dirichlet boundary condition
\[u = 0 \quad \text{on} \quad \Gamma\]
for the \(\text{circleg}\) geometry.

Finally we can derive the exact solution of the Poisson problem by
\[\text{exact} = (1-p(1,:)^2-p(2,:)^2)/4;\]