Project: To be continued

Recall the case of a circular elastic membrane pinned at $z = 1$ and subjected to an electrostatic force applied on the plane $z = 0$. If we exploit the circular symmetry, we can reduce the problem to a BVP in the disk radius r:

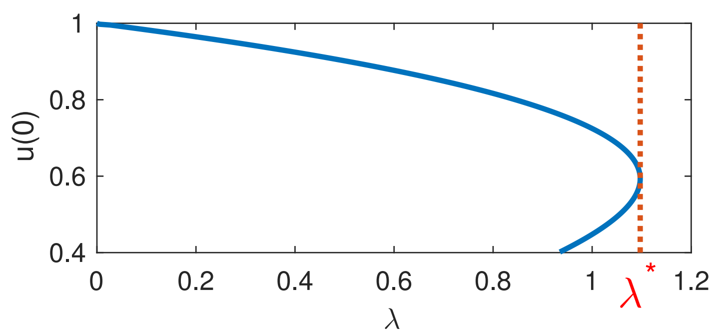
$$u''(r) + \frac{1}{r}u'(r) = \frac{\lambda}{u^2}, \qquad u'(0) = 0, \quad u(1) = 1. \tag{1}$$

This has been used as a model for an actuator device in a microelectromechanical system (MEMS). Because we would have some headaches with division by zero in this ODE, we multiply through by r to get

$$ru''(r) + u'(r) = \frac{r\lambda}{u^2}. \tag{2}$$

The behavior of this system as a function of the field strength λ is quite interesting. Clearly $u \equiv 1$, $\lambda = 0$ is a trivial solution. As λ increases from zero, the membrane begins to deflect downward toward zero, with the greatest amount of deflection (i.e., smallest value of u) at the center point $r = 0$. The amount of deflection continues to decrease until, at some critical value λ^*, no solution exists. However, for some values $\lambda < \lambda^*$ there are at least two valid solutions. It's not at all clear how to figure out good initial guesses that will drive a Newton-type iteration to find the others.

When the BVP is discretized, we use $n+1$ unknown values of u and $n+1$ equations to specify them. But if we also think of λ as a parameter, we have $n+1$ equations in $n+2$ variables. This defines a curve or path in $(n+2)$-dimensional space—just like $x^2 + y^2 = 1$ defines a curve in the plane. The situation can be visualized by this graph:



(This picture is *not* quantitatively accurate for your results!)

What we really want to do is follow along the path, a technique known as *continuation*. There are many ways this can be done, but this particular problem gives us a fairly easy option. The key is to observe in the figure above that λ is a poor choice for parameterizing the path, because it fails the "vertical line test." We are fortunate that in this problem, the value $u(0)$ is monotonic along the path, so it will serve as the parameter. To be specific, we solve a discretized form of (2) for the $n+2$ unknowns $\lambda, u_0, u_1, \ldots, u_n$. As has been our practice, the ODE is discretized at the $n-1$ interior nodes only. To this are added the *three* supplemental conditions

$$u'(0) = 0, \qquad u(1) = 1, \qquad u(0) = 1 - s, \tag{3}$$

where $0 \le s < 1$ is a value under your control. You start at $s = 0$ with the trivial solution mentioned above, then gradually increase s to get a sequence $s_1, s_2, \ldots$, corresponding to solutions $(\lambda_1, \boldsymbol{u}_1), (\lambda_2, \boldsymbol{u}_2)\ldots$, which are points along the path. When solving the nonlinear equations for $s = s_{k+1}$, the starting guess should be $(\lambda_k, \boldsymbol{u}_k)$, which is hopefully near the next point.

Project assignment

The heart of the assignment is a function

```
[u,lambda] = mems_solve(u0,lam0,s)
```

that solves the ODE with the three side conditions in (3). Here `lam0` is an initial guess for λ, and the $(n+1)$-vector `u0` is an initial guess for $\boldsymbol{u}$. You should model this function after `bvp`, but your function does not need to solve a general problem, just the specific one at hand. Like `bvp`, your function will call `levenberg` on a subfunction named `residual`, but your version of `residual` evaluates $n+2$ quantities for values of the $n+2$ variables in a given vector $\boldsymbol{z} = \begin{bmatrix} \lambda \\ \boldsymbol{u} \end{bmatrix}$. Once `mems_solve` is working, you can call it within a loop to implement the steady increase of s.

Your objectives are:

1. Find λ^* to at least two significant digits. This means taking small enough steps along the curve as well as giving evidence that the answer has converged as a function of the discretization size n.

2. Find a second turning point, $\lambda^\dagger < \lambda^*$, to two digits. Past this turning point, λ begins to increase again as a function of s.

3. Make a plot like the one above for $0 \le s \le 0.99$.

4. On one graph, plot $u(r)$ against r for $s = 0.1, 0.3, 0.5, 0.7, 0.9, 0.95$.

5. How accurately can you find λ^*? Think about how to get the most out the discretization accuracy and how to use something more sophisticated than just taking super-small steps in s. Style points matter here!

Submit a PDF report describing your results and providing any necessary supporting information. Keep it concise and to the point—for instance, don't describe multiple attempts at the last item above, just your best one. Don't forget to label your plots! Also include your `mems_solve.m` and the code to reproduce the results for items 1–3 above.