Determining the intersection points of two planar parametric curves

We are given parametrizations of two curves, K and L, in the plane $\mathbb{R}^2$. Our task is to find all points of intersection of K and L. Let $\mathbf{p}(t)$ and $\mathbf{q}(t)$ be the parametrizations of curves K and L, defined on intervals $I = [a, b]$ and $J = [c, d]$, respectively. Find the points of intersection using the following procedure:

1. Divide the intervals I and J into subintervals of length $h > 0$, where h is sufficiently small.
2. Approximate K and L with polygonal chains determined by evaluating the parametrizations at subdivision points of I and J and find the intersections of the polygonal chains.
3. Use the points of intersection of the polygonal approximations as an initial guess for Newton's iteration, which will determine the actual intersection points of K and L more accurately.

For the Newton iteration you will need also the derivatives of the two parametrizations, $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$.

Task

1. To find all intersection points of curves parametrized by $\mathbf{p}$ and $\mathbf{q}$ means one needs to find all the solutions of the equation $\mathbf{p}(t) = \mathbf{q}(u)$. Suppose that $P'(x_i, y_i)$ is one of the intersection points of the polygonal chains K' and L'. What is a sensible choice for the values of parameters $t_i^{(0)}$ and $u_i^{(0)}$, such that $\mathbf{x}_i^{(0)} = [t_i^{(0)}, u_i^{(0)}]^\mathsf{T}$ is a good initial approximation for Newton's iteration? (That is: an $\mathbf{x}_i^{(0)}$ which converges to $\mathbf{x}_i = [t_i, u_i]$, such that $\mathbf{p}(t_i) = \mathbf{q}(u_i)$ is precisely the spatial vector of the intersection point P, closest to P'.)
2. Write the system of equations $\mathbf{p}(t) = \mathbf{q}(u)$ in a form suitable for Newton's iteration and express the corresponding Jacobian matrix using derivatives $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$.
3. Write an Octave/Matlab function which, for curves K and L, finds all their intersection points using the above method.

4. Pick two 'interesting' curves (with at least 3 intersection points) and include two pictures in your report: a drawing of both polygonal chains K' and L' with clearly marked intersection points, and an analogous drawing for the curves K and L.

Detailed instructions for the Octave/Matlab function

Write an Octave/Matlab function `intersectionOfCurves`, which computes the intersection points P of the curves K and L with given parametrizations. Function call should be of the form:

```
[P, Q] = intersectionOfCurves(p, pdot, intp, q, qdot, intq, h),
```

where:

- `p` (`q`) is a function handle, describing the first (second) plane curve,
- `pdot` (`qdot`) is a function handle, describing the derivative of the first (second) plane curve,
- `intp` (`intq`) is an interval, on which the first (second) curve is parametrized,
- `h` is the length of the subintervals dividing the intervals `intp` and `intq` (the last subinterval can be shorter),
- `P` is the list of the intersection points of K and L (a $2 \times m$ matrix),
- `Q` is the list of the intersection points of K' and L' (a $2 \times m$ matrix).

Example

For the curves K and L parametrized by

$$t \mapsto \begin{bmatrix} t \\ \cos t \end{bmatrix} \qquad \text{in} \qquad u \mapsto \begin{bmatrix} u \\ \sin u \end{bmatrix}$$

on the intervals $[-2\pi, 2\pi]$ and a step $h = 0.1$, we would use

```
p = @(t) [t; cos(t)];
pdot = @(t) [1; -sin(t)];
intp = [-2*pi, 2*pi];
q = @(u) [u; sin(u)];
qdot = @(u) [1; cos(u)];
intq = [-2*pi, 2*pi];
h = 0.1
[P, Q] = intersectionOfCurves(p, pdot, intp, q, qdot, intq, h)
```

Tests

The file with the function `intersectionOfCurves` should include at least one test for a pair of curves K and L, which should confirm the correctness of the return values of your function.

Submission

Use the online classroom to submit the following:

1. file **intersectionOfCurves.m** which should be well-commented and contain at least one test,

2. a report file **solution.pdf** which contains the necessary derivations and answers to questions.

While you can discuss solutions of the problems with your colleagues, the programs and report must be your own work. You can use all Octave/Matlab functions from problem sessions/lectures.