Intersection of two implicit surfaces

A surface in $\mathbb{R}^3$ can be described as a solution set of an equation $f(\mathbf{x}) = 0$, where $\mathbf{x} = [x_1, x_2, x_3]^\mathsf{T} \in \mathbb{R}^3$, and f is a function of three variables. Suppose we are given two surfaces given by $f_1(\mathbf{x}) = 0$ and $f_2(\mathbf{x}) = 0$. The intersection of these two surfaces is the solution set of the nonlinear system

$$\begin{aligned} f_1(\mathbf{x}) &= 0, \\ f_2(\mathbf{x}) &= 0. \end{aligned}$$

If f_1 and f_2 are smooth functions and some additional conditions are satisfied the intersection of these two surfaces is a smooth curve K. The objective is to find this curve.

Construction of the curve K

View equations $f_1(\mathbf{x}) = 0$ and $f_2(\mathbf{x}) = 0$ as equations of level sets of f_1 and f_2. The curve K is the intersection of these two level sets. Gradients of f_1 and f_2 are orthogonal to K at each point of K. In other words, the vector $(\operatorname{grad} f_1) \times (\operatorname{grad} f_2)$ is tangent to K. Set

$$\mathbf{F}(\mathbf{x}) = \frac{(\operatorname{grad} f_1(\mathbf{x})) \times (\operatorname{grad} f_2(\mathbf{x}))}{\|(\operatorname{grad} f_1(\mathbf{x})) \times (\operatorname{grad} f_2(\mathbf{x}))\|}.$$

This is a vector-valued function $\mathbf{F}\colon \mathbb{R}^3 \to \mathbb{R}^3$. The curve K can then be understood as a solution to the differential equation

$$\begin{aligned} \dot{\mathbf{x}}(t) &= F(\mathbf{x}) \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned} \tag{1}$$

where $\mathbf{x}_0$ is an initial point that lies in the intersection of both surfaces, i.e. $f_1(\mathbf{x}_0) = 0$ and $f_2(\mathbf{x}_0) = 0$. Since the vector product in $\mathbf{F}$ in normed, the solution to (1) is even parametrised according to its natural prarameter.

We can solve (1) using numerical methods for solving differential equations, say Eulers or Runge-Kutta methods.

Depending on the accuracy of the chosen method and the step size h we are using, the curve we get may eventually noticeably deviate from the intersection of the surfaces $f_1(\mathbf{x}) = 0$ and $f_2(\mathbf{x}) = 0$ due to numerical errors (this may happen even after just one step, if we are using Eulers method or the step size h is too large). We can correct this behaviour by continually checking if the current point $\mathbf{y}$ on the curve is still close enough to K and, if needed, move to a point on K that is 'close' to K. Considering that the values $f_1(\mathbf{y})$ and $f_2(\mathbf{y})$ are themselves good measures of how close $\mathbf{y}$ is to the surfaces, we can decide to make a correction whenever the quantity

$$d = \max\{|f_1(\mathbf{y}), |f_2(\mathbf{y})|\} \tag{2}$$

exceeds a given value $\varepsilon > 0$.

In the case $\mathbf{y}$ is too far away from the intersection and denoting $\mathbf{v} = \mathbf{F}(\mathbf{y})$, then $\mathbf{v}\cdot\mathbf{x} = \mathbf{v}\cdot\mathbf{y}$ is the equation of a plane, which is 'close' to a plane normal to K. We can use $\mathbf{y}$ to obtain $\mathbf{x}_1$, which actually lies on K by solving the system of nonlinear equations (in the unknown $\mathbf{x}$)

$$\begin{aligned} f_1(\mathbf{x}) &= 0, \\ f_2(\mathbf{x}) &= 0, \\ \mathbf{v}\cdot\mathbf{x} - \mathbf{v}\cdot\mathbf{y} &= 0. \end{aligned} \tag{3}$$

We will obtain the solution $\mathbf{x}_1$ of this system by using the Newton's iteration with initial guess $\mathbf{y}$. Expectation is, of course, that $\mathbf{x}_1$ is close $\mathbf{y}$. (If it is not, our choice of h was too large.)

With $\mathbf{x}_1$ on K obtained we continue with the numerical solving of (1).

A minor problem regarding the sensibility of this construction: In practice the starting point $\mathbf{x}_0 \in K$ is not known, we only know an approximation $\mathbf{y}$ for $\mathbf{x}_0$. A quick remedy: Solve the system (3) with this approximation, get $\mathbf{x}_0 \in K$, and use the method described above.

Task

1. Write down the vector-valued function $\mathbf{G}\colon \mathbb{R}^3 \to \mathbb{R}^3$ and its Jacobi matrix $J\mathbf{G}$ corresponding to the system (3). (Both can be expressed using f_i, $\operatorname{grad} f_i$, $\mathbf{y}$, and $\mathbf{v}$.)

2. Write an `octave` function

```
X = presekPloskevEuler(f1, gradf1, f2, gradf2, X0, h, n, tol, maxit, epsilon)
X = presekPloskevRK4(f1, gradf1, f2, gradf2, X0, h, n, tol, maxit, epsilon),
```

   which given:

   - functions $f_1, f_2\colon \mathbb{R}^3 \to \mathbb{R}$, functions of a vector argument $\mathbf{x} \in \mathbb{R}^3$,
   - gradients $\operatorname{grad} f_1, \operatorname{grad} f_2\colon \mathbb{R}^3 \to \mathbb{R}^3$, vector-valued functions $\mathbb{R}^3 \to \mathbb{R}^3$,
   - approximation `X0` for the initial point on the curve (this has to be 'adjusted' using the system (3) first),
   - step length `h`,
   - the number `n` of consecutive points on the curve to be constructed,
   - the tolerance `tol` for Newton's iteration, and
   - maximum allowed number of iteration steps `maxit` for Newton's iteration
   - maximum allowed distance (in the sense of (2)) from the intersection `epsilon`.

   returns a $3 \times (n+1)$ matrix `X` containing the points on the curve K as columns, ie. $X = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n]$. The first function uses Eulers method and the second uses the Runge-Kutta method of the 4th order. *Stick to specifications!*

3. Compare how often on average the functions *presekPloskevEuler* and *presekPloskevRK4* perform the correction described above for a few examples of surfaces and choices of step size h.

Submission

Use the online classroom to submit the following:

1. file **presekPloskev.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 creation. You can use all `octave` functions from problem sessions.