HW 4

1: Crossing cubics  A Bezier curve of degree \( n \) is the curve traced out by

\[
f(t) = \sum_{i=0}^{n} p_i B^n_i(t), \quad t \in [0, 1]
\]

where the points \( p_i \in \mathbb{R}^2 \) are control points and the functions \( B^n_i(t) \) are the Bernstein polynomials

\[
B^n_i(t) = C^n_i (1-t)^{n-i} t^i, \quad C^n_i = \frac{n!}{i!(n-i)!}.
\]

A common type of Bezier curve in computer graphics is the cubic Bezier curve defined by four control points. Complete the following function to compute the intersection of two such curves

```matlab
function [x] = bezier_intersect(pf, pg)

% Attempt to compute the intersection of the cubic Bezier curves
% defined by the columns of pf and pg (each of dimension 2-by-4).
% Assume a unique intersection.

Illustrate with an example that your solution works correctly.
```

2: Funky fixed point  Argue that the iteration

\[
\begin{align*}
3x_{k+1} + 2y_{k+1} &= \cos(x_k) \\
2x_{k+1} + 4y_{k+1} &= \cos(y_k)
\end{align*}
\]

converges to a unique fixed point \((x_*, y_*)\), regardless of the initial point, and that \( \|e_{k+1}\|_2 < 0.7\|e_k\|_2 \), where \( e_k = (x_k - x_*, y_k - y_*) \). Starting from the point \((1, 1)\), draw a semi-logarithmic plot of the error versus \( k \) to illustrate the convergence.