1. Given a curve $\mathcal{S} \subset \mathbb{R}^n$, the *curvature* of $\mathcal{S}$ at the point $\vec{p} \in \mathcal{S}$ is the magnitude of the acceleration when passing through $\vec{p}$ at unit speed (following the curve S). That is, if $\vec{r}(t)$ is an arc-length parameterization of S and $\vec{r}(t_0) = \vec{p}$, then the curvature of $\mathcal{S}$ at the point $\vec{p}$ would be $\|\vec{r}\,''(t_0)\|$.

   (a) Let $\mathcal{S}_r \subset \mathbb{R}^2$ be a circle of radius r centered at the origin. Compute the curvature of $\mathcal{S}_r$ at the point $\vec{p} = (r, 0)$.

   (b) The following points lie on the curve $\mathcal{C} \subset \mathbb{R}^2$.

x	y
0.7	0.49
0.8	0.64
0.9	0.81
1	1
1.1	1.21
1.2	1.44
1.3	1.69

   i. Suppose that $\vec{r}$ is an arc-length parameterization of $\mathcal{C}$ and that $\vec{r}(t_0) = (1, 1)$. Estimate $\vec{r}\,'(t_0)$. (*Hint: make sure your vector has the correct length!*)

   ii. Estimate the curvature of $\mathcal{C}$ at $(1, 1)$.

2. For this problem, we will be using MATLAB/OCTAVE, though not every part requires programming. Let $\mathcal{P}$ be the part of the parabola $1 - x^2$ that is above the x-axis and consider the following parameterizations of $\mathcal{P}$:

$$\vec{r}(t) = \begin{bmatrix} t \\ 1 - t^2 \end{bmatrix} \qquad \text{and} \qquad \vec{p}(t) = \begin{bmatrix} \sin(\pi t/2) \\ 1 - \sin^2(\pi t/2) \end{bmatrix}.$$

   (a) Find the domains of $\vec{r}$ and $\vec{p}$ such that they are indeed parameterizations of $\mathcal{P}$.

   (b) Plot a numerical estimate of the speed of $\vec{r}$ and $\vec{p}$ vs. time. You may find the following MATLAB/OCTAVE tips helpful:

   If you have a list `x=[1, 4, 9, 16, 25]`, for example, and you would like to get a list of the consecutive differences between entries in `x`, you can use the command

```
x(:, 2:length(x)) - x(:, 1:(length(x)-1))
```

   If you have a list `vecs` whose *columns* are vectors and you'd like to get a list containing the lengths of those vectors, you can use the command

```
sqrt(sum(vecs .* vecs, 1))
```

   The extra `1` in the `sum` command tells MATLAB/OCTAVE to sum along the columns (the command `sum(x, 2)` would sum along the rows).

   Hint: it will be worth your time to define MATLAB/OCTAVE *functions for $\vec{r}$ and $\vec{p}$ for use later in the problem. Also, make sure you understand why the example code above works before you use it.*

   (c) Plot the arc length of the path traversed by $\vec{r}$ and $\vec{p}$ with respect to time. For which t_0 do you expect $\text{arclen}_0^{t_0}(\vec{r}) = \text{arclen}_0^{t_0}(\vec{p})$? Explain. (*Hint: don't look for a formula for arc length! Use* MATLAB/OCTAVE *to create a list whose ith item is the arc length up to the ith time step.*)

   (d) Inverting functions given by formulas is hard, but inverting functions as a concept is easy—you just switch the x and y coordinates! In MATLAB/OCTAVE we have easy access to x and y coordinates, however we don't have access to *all* x and y coordinates. The solution is to estimate the points we don't have based on those we do. This process is called *interpolation*.

We will create an approximation of $f(x) = x^2$ and its inverse on the interval $[0, 10]$ using 11 regularly spaced points. Create two lists, `xs = 0:1:10` and `ys = xs .* xs`. We will define approximations to f and f^{-1} using the `interp1` command. Create two new functions with the following code:

```
fapprox = @(x) (interp1(xs, ys, x, 'spline'))
fiapprox = @(x) (interp1(ys, xs, x, 'spline'))
```

The `'spline'` argument tells MATLAB/OCTAVE to do a smooth approximation using polynomials rather than jaggedy approximation with lines or step functions.

i. Using at least 1001 equally spaced points in the interval $[0, 10]$, graph f and `fapprox` on the same plot. On a separate plot, graph f^{-1} and `fiapprox`. Where do the functions exactly match their approximations? Why? (*Hint:* `plot(x1s, y1s, x2s, y2s)` *can be used to plot two functions in the same graph.*)

ii. Using at least 1001 equally spaced points in the interval $[0, 10]$, plot `fiapprox` $\circ$ `fapprox` and `fapprox` $\circ$ `fiapprox`. If `fiapprox` and `fapprox` were perfect inverses of each other, what graph should you get? Why doesn't your graph look like that?

iii. Define two new functions `fgoodapprox` and `figoodapprox` using 51 equally spaced points between $[0, 10]$ as the basis for your approximations. Then, graph `figoodapprox` $\circ$ `fgoodapprox` and `fgoodapprox` $\circ$ `figoodapprox`. Is this closer to what you expected?

(e) Let $\vec{a}$ be the arc-length parameterization of $\mathcal{P}$ with $\vec{a}(0) = (-1, 0)$. Define $\vec{a}$ as a function in MATLAB/OCTAVE. On separate plots, plot $\|\vec{r}\,''(t)\|$, $\|\vec{p}\,''(t)\|$, and $\|\vec{a}\,''(t)\|$. You previously found values t_0 where $\vec{r}(t_0) = \vec{p}(t_0)$. At these values, should $\|\vec{r}\,''(t_0)\| = \|\vec{p}\,''(t_0)\|$? Explain.

(f) Let $c : \mathcal{P} \to \mathbb{R}$ be the function that takes a point on $\mathcal{P}$ and returns the curvature at that point. Use MATLAB/OCTAVE to plot $c \circ \vec{p}$, $c \circ \vec{r}$, and $c \circ \vec{a}$. You previously found values t_0 where $\vec{r}(t_0) = \vec{p}(t_0)$. At these values, should $c \circ r(t_0) = c \circ \vec{p}(t_0)$? Explain.