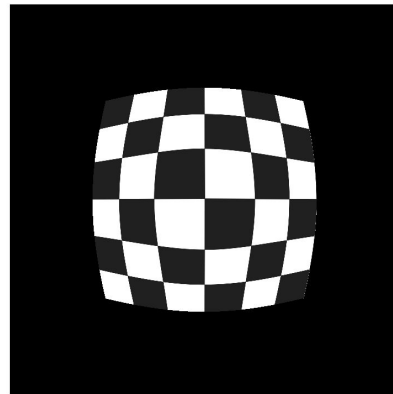
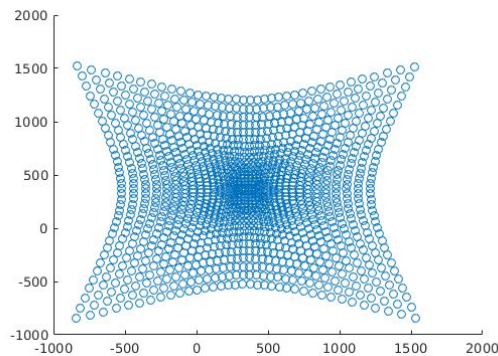


Barrel Distortion: Correction

- Radial distortion is typically defined as $x_d = x_u (k_1 + k_2 r + k_3 r^2 + k_4 r^3 + k_5 r^4)$. Whether it is barrel or pincushion depends on the values of coefficients k_2, k_3 , etc. Positive coefficients would push the radially far points farther as compared to the nearer ones, thus resulting in a pincushion distortion. The inverse, barrel distortion, would occur when the coefficients are negative. The scatter plot below shows an example of points distorted under positive coefficients.



- Interp2:** However, this changes when these set of points are subjected to MATLAB's interp2. Its internal normalizations convert these point sets into an image that looks like the one displayed above. This seemingly is in line with OpenCV's [documentation](#) on distortion under multiplicative model and their assumptions about the coefficients:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$$

$$y'' = y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$

$$\text{where } r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

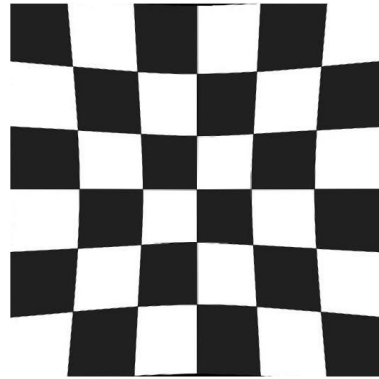
$$v = f_y * y'' + c_y$$

k_1, k_2, k_3, k_4, k_5 , and k_6 are radial distortion coefficients. p_1 and p_2 are tangential distortion coefficients. Higher-order coefficients are not considered in OpenCV.

The next figure shows two common types of radial distortion: barrel distortion (typically $k_1 > 0$ and pincushion distortion (typically $k_1 < 0$).

Further, the same behaviour persists on using MATLAB's maketform function to distort/undistort the input image.

3. **Input Image:** The question had a discrepancy in the input image provided in that it belonged to a lesser distorted model. On solving with the provided image, the output should have looked like this:



4. This is a manual negligence and should have been avoided at all costs and warrants an apology from our end. Of the submissions that we looked at, one of the groups has been able to find this solution. Though, they too, like many others remained in confusion due to the interp2's inversion.
5. The correct solution requires the following pseudo code:

```
for i = 1 to nSteps, do
    if i == 1, then
         $X_u = X_d$ 
    end
     $R = \text{L2norm}(X_u)$ 
     $\text{dist\_factor} = k_1 + k_2 R + k_3 R^2$ 
     $X_u = X_d / \text{dist\_factor}$ 
end
```

The complete code will be provided in the solutions. If used in conjunction with Interp2, the inputs and outputs should appear like this:

