CP.2.3.4, Sauer3

Let A be the n-by-n matrix with entries

$$a_{ij} = \sqrt{(i-j)^2 + \frac{n}{10}}.$$

Define $x = (1, \ldots, 1)^T$ and $b = Ax$. For $n = 100$, 200, 300, 400, and 500, use the python program from Computer Problem 2.1.1 or Numpy's `numpy.linalg.solve` command to compute `xc`, the double precision computed solution. Calculate the infinity norm of the forward error for each solution. Find the five error magnification factors of the problems $Ax = b$, and compare with the corresponding condition numbers.

Hint: Note that this $a_{ij} = \sqrt{(i-j)^2 + \frac{n}{10}}$ formula is the same with 0-base indexing (Python) or 1-base indexing (Sauer and Matlab). Here is a code snippet to generate A with $n = 5$.

```
from math import sqrt
n = 5
A = np.zeros( [ n, n ], dtype=float )
for i in range(0,n):
  for j in range(0,n):
      A[i,j] = sqrt( ( i - j )**2 + n / 10. )
print(A)
```

```
[[0.70710678 1.22474487 2.12132034 3.082207   4.0620192 ]
 [1.22474487 0.70710678 1.22474487 2.12132034 3.082207  ]
 [2.12132034 1.22474487 0.70710678 1.22474487 2.12132034]
 [3.082207   2.12132034 1.22474487 0.70710678 1.22474487]
 [4.0620192  3.082207   2.12132034 1.22474487 0.70710678]]
```