The code was tested using a module I wrote, called speed_analysis.py, where I set up a random nxn matrix and n-length vector. The relevant code snippet is as follows:

```
import numpy as np

dimension = int(input("Enter dimension of nxn matrix:"))
np.random.seed()
pb = np.random.permutation(dimension)
pA = np.random.randint(0, dimension*2, (dimension, dimension))
```

I then passed pA(matrix of shape n x n) and pb(vector of length n) to the parallel gaussian module.

For example, a user input of 64 would produce a matrix of 64 x 64 shape, and contain random numbers between 0 and 2*dimension.

10 iterations were performed for a set of data, and the average time taken was recorded.