Numerical Method

National Cheng Kung University Department of Engineering Science

Instructor: Chi-Hua Yu

LAB 6

Programming, Due 17:00, Tuesday, April 11th , 2023

注意事項

1. Lab 的繳交期限為星期二**(4/11) 17:00 p.m.**。
2. Lab 的分數分配:Lab 分數 100%。
3. 請儘量於 Lab 時段完成練習, 完成後請找助教檢查, 檢查後即可離開。
4. 檔名規定:檔名錯誤將記為 0 分
   i. Lab: 請用 學號**_LabNumber** 為檔名做一個資料夾**(e.g.,N96091350_Lab6)**, 將 **ipynb** 檔放入資料夾, 壓縮後上傳至課程網站**(e.g., N96091350_Lab6.zip)**。
   ii. Bonus: 請用 學號**_bonus** 為檔名做一個資料夾**(e.g.,N96091350_bonus)**, 將 **ipynb** 檔放入資料夾, 壓縮後上傳至課程網站**(e.g., N96091350_bonus.zip)**。
5. **Code** 中需有註解。
6. 未完成者可於下周一 **(4/17) 09:00 a.m.** 前上傳至 Moodle, 惟補交的分數將乘以0.8計, 超過期限後不予補交。
7. Bonus 需於下周一 **(4/17) 09:00 a.m.** 前上傳至 Moodle, 不予補交。
8. 準時繳交者, 請交至「Lab6 準時繳交區」;補交者, 請交至「Lab6 補交區」;bonus 請繳交至「bonus 繳交區」。

請勿抄襲, 抄襲者與被抄襲者本次作業皆**0**分計算

---

1. **(100%)** Please download the template file `Inverse_Power_Method.ipynb`. Write a Python program to find the smallest eigenvalue and the associated eigenvector by using the inverse power method. You can use `[1, 1]` as the initial vector x to start the iteration. In each iteration is usually normalized, which will make the largest element in the absolute vector equal to 1. Normalization will provide the smallest eigenvalue and its corresponding eigenvector at the same time.

   First iteration:

$$A^{-1}x_0 = c_1\frac{1}{\lambda_1}\left[v_1 + \frac{c_2}{c_1}\frac{\lambda_2}{\lambda_1}v_2 + \ldots + \frac{c_n}{c_1}\frac{\lambda_n}{\lambda_1}v_n\right] = c_1\frac{1}{\lambda_1}x_1$$

   Second iteration:

$$A^{-1}x_1 = \frac{1}{\lambda_1}\left[v_1 + \frac{c_2}{c_1}\frac{\lambda_2^2}{\lambda_1^2}v_2 + \ldots + \frac{c_n}{c_1}\frac{\lambda_n^2}{\lambda_1^2}v_n\right] = \frac{1}{\lambda_1}x_2$$

   □

   k[th] iteration:

$$A^{-1}x_{k-1} = \frac{1}{\lambda_1}\left[v_1 + \frac{c_2}{c_1}\frac{\lambda_2^k}{\lambda_1^k}v_2 + \ldots + \frac{c_n}{c_1}\frac{\lambda_n^k}{\lambda_1^k}v_n\right] = \frac{1}{\lambda_1}x_k$$

$$A^{-1}x_{k-1} \sim \frac{1}{\lambda_1} v_1$$

Below is the running example

Sample 1

```
x = np.array([1, 1])
a = np.array([[0, 2],[2, 3]])

The Minimum Eigenvalue: -1.0
Eigenvector: [ 1.  -0.5]
```

Sample 2

```
x = np.array([1, 1, 1])
a = np.array([[1, 5, 2],[2, 4, 3],[2, 1, 6]])

The Minimum Eigenvalue: -0.7958315233127196
Eigenvector: [ 1.         -0.25654736 -0.25654736]
```