

Hand in 7 - Eigenvalue-algorithms

In this lab you are going to try out the power-method

1. First you need a matrix of which you know the true eigenvalues. The following python-function constructs a 2x2-matrix which have the eigenvalues λ_a and λ_b . It uses that you can write a diagonalizable matrix as PDP^{-1} , where D has the eigenvalues on the diagonal.

```
def Create2by2Matrix(lambda_a, lambda_b):  
    P=np.array([[0.5, 0.5], [0.2, 0.8]])  
    D=np.array([[lambda_a, 0.0], [0.0, lambda_b]])  
    Pinv=inv(P)  
    return np.matmul(np.matmul(P,D),Pinv)
```

Here numpy was imported as `np` and `inv` as `from scipy.linalg import inv`. Try out this code.

2. Implement power iterations. *Hint* - wikipedia has quite a spoiler for the code on this. Try it out - do you get the right **dominant** eigenvalue? You can use a maximum number of iterations as a stopping criteria.
3. Experiment with letting λ_1 and λ_2 be very close to each other. What are the effects?

This should be included in this hand in: This is the only hand in of this final lab. Hand in your code and answer to the last question.