

# jacobi

May 18, 2018

# Jacobi by Iliya Sabzevari

The Jacobi algorithm is an iterative method to solve a diagonally dominant linear system of equations. Wikipedia does a great outline of the method and I would recommend anyone interested in how it works start there.

```
In [1]: import numpy as np

def jacobi(H_func, diag_H, b):
    d = len(diag_H)
    xguess = np.random.rand(d)
    while True:
        R_x = H_func(xguess) - diag_H*xguess
        A = b - R_x
        x = A/diag_H
        if np.linalg.norm(x-xguess) < 1e-8:
            break
        xguess = x
    return x
```

The function was tested with a two dimensional example from the wikipedia page.

```
In [2]: #!/usr/bin/env python
import numpy as np
import jacobi as j

b = np.array([11,13],dtype = np.float64)
xguess = np.array([1,1],dtype = np.float64)
A = np.array([[2,1],[5,7]],dtype = np.float64)

def Hessian(v):
    return np.dot(A,v)

x = j.jacobi(Hessian,np.diag(A),b)

print(x)

[ 7.11111111 -3.22222222]
```