f-25-jupyter-triadiag

May 4, 2021

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

[2]:
```
def house_plus(x):
    norm_x = np.linalg.norm(x)
    if norm_x == 0:
        v = np.zeros_like(x)
        v[0] = 1
        s = 0
        eps = 1
    else:
        u = x / np.linalg.norm(x)
        eps = -1 if u[0] >= 0 else +1
        s = 1 + np.abs(u[0])
        v = - eps * u
        v[0] += 1
        v /= s
    return v, s, eps, norm_x
```

[3]:
```
def tridiagonal_data(a):
    data = np.copy(a)
    if not np.allclose(a, a.T):
        raise np.linalg.LinAlgError(
            'In tridiagonal_data() input must ' +
            'be a symmetric matrix')
    n, _ = a.shape
    s = np.empty(n - 2)
    for j in range(n - 2):
        v, s[j], eps, norm = house_plus(data[j+1:, [j]])
        u = s[j] * (data[j+1:, j+1:] @ v)
        w = u - ((s[j]/2) * (u.T @ v)) * v
        v_wT = v @ w.T
        data[j+1, j] = eps * norm
        data[j, j+1] = data[j+1, j]
        data[j+1:, j+1:] -= v_wT + v_wT.T
        data[j+2:, [j]] = v[1:]
    return data, s
```

[4]:
```
def tridiagonal_qt(a):
    data, s = tridiagonal_data(a)
    n, _ = a.shape
    t = np.tril(np.triu(data, -1), 1)
    q = np.eye(n)
    for j in reversed(range(n-2)):
        x = data[j+2:, [j]]
        v = np.vstack([[1], x])
        q[j+1:, j+1:] -= s[j] * v @ (v.T @ q[j+1:, j+1:])
    return q, t
```

[5]:
```
rng = np.random.default_rng()
```

[6]:
```
n = 30
a = rng.normal(0.0, 5.0, (n, n))
a = (a + a.T)/2
a[:3, :3]
```

[6]:
```
array([[-3.17487956, -0.27000444, -2.12180906],
       [-0.27000444, -1.69949598, -2.78877896],
       [-2.12180906, -2.78877896,  5.02076562]])
```

[7]:
```
q, t = tridiagonal_qt(a)
```

[8]:
```
np.allclose(q.T @ q, np.eye(n), atol=2*np.finfo(float).eps)
```

[8]:
```
True
```

[9]:
```
np.allclose(q @ t @ q.T, a, atol= np.finfo(float).eps)
```

[9]:
```
True
```

[10]:
```
t[:4, :4]
```

[10]:
```
array([[-3.17487956, 13.96604787,  0.        ,  0.        ],
       [13.96604787, -7.22858264, 19.52594169,  0.        ],
       [ 0.        , 19.52594169, -3.65357184, 18.72767726],
       [ 0.        ,  0.        , 18.72767726,  2.87432424]])
```

[ ]:
```

```