main

August 2, 2024

[ ]:

```python
import scipy.io
import numpy as np
import scipy.linalg
#    .mat
mat = scipy.io.loadmat('EQ.mat')

#    .mat
print(mat.keys())
```

```
dict_keys(['__header__', '__version__', '__globals__', 'None',
'__function_workspace__'])
```

[ ]:

```python
for i in mat.keys():
    print(str(i)+str(mat[i]))

# #      A     b
# A = mat[A_key]
# b = mat[b_key]
```

```
__header__b'MATLAB 5.0 MAT-file, Platform: PCWIN64, Created on: Fri Aug  2
11:10:48 2024'
__version__1.0
__globals__[]
None[(b'eqns', b'MCOS', b'sym', array([[3707764736],
        [         2],
        [         1],
        [         1],
        [         1],
        [         1]], dtype=uint32))           ]
__function_workspace__[[ 0  1 73 …  0  0  0]]
```

[ ]:

```python
# example
N = 3
A = np.array([range(N**2)]).reshape(N, N)
A = A - 1
b = np.array([range(N)])
b = b - 1
print(A)
print(b)
```

```
[[-1  0  1]
 [ 2  3  4]
 [ 5  6  7]]
[[-1  0  1]]
```

[ ]:

```
def solve_linear_system(A, b):
    b = b.flatten()
    #   A
    if A.shape[0] == A.shape[1]:
        try:
            #       Ax = b
            x = scipy.linalg.solve(A, b)
            print("    ", x)
        except np.linalg.LinAlgError as e:
            #
            print("         ", e)
    else:
        #        Ax = b
        x, residuals, rank, s = scipy.linalg.lstsq(A, b)
        print("     ", x)
        if residuals.size > 0 and np.all(residuals < 1e-10):
            print("     ")
        else:
            print("      ")
```

[ ]:

```
#
solve_linear_system(A, b)
```

```
     [ 0.375      0.58333333 -0.625     ]
```

```
/tmp/ipykernel_42612/3977183546.py:7: LinAlgWarning: Ill-conditioned matrix
(rcond=9.25186e-18): result may not be accurate.
  x = scipy.linalg.solve(A, b)
```