

How to use: UFGet Python implementation

The Python code is provided in four files, and I will discuss each file separately. Most users will only need to read the first file.

1. The majority of the functionality is provided in the file `UFget.py`, which has 3 functions of note:
 - (a) `refresh()`: Forces a refresh of the index file.
 - (b) `UFgetAll()`: Downloads and extracts the entirety of the UFSMC.
 - (c) `UFget(index)`: Downloads the index's corresponding matrix in the UFSMC and returns a `Matrix` object representing that matrix. Alternately, can be given the file location of the matrix instead of the index and it will download that. If you would like to specify the folder where files are downloaded into, there is a global variable `UFfolder` that may be tweaked in the source code that holds that information.

It defines the class `Matrix`, which has the following data members:

- (a) `name`: The file location of the matrix from the directory of `UFget.py`
- (b) `title`: The name of the matrix, as given in the file.
- (c) `id`: The ID number of the matrix in the UF Sparse Matrix Collection (UFSMC).
- (d) `date`: The date of the matrix, as given in the UFSMC.
- (e) `author`: The authors of the matrix.
- (f) `ed`: The editors of the matrix, as given in the UFSMC.
- (g) `kind`: The kind of matrix stored, as given by UFSMC.
- (h) `nrows`: The number of rows in the matrix.
- (i) `ncols`: The number of columns in the matrix.
- (j) `nnz`: The number of non-zero entries in the matrix.
- (k) `dims`: A string which is simply `row + ' by ' + col`
- (l) `A`: A dictionary holding the actual matrix. In order to access the value at `[row, col]`, we use `A[row,col]`, which will return the value.

Note that if the matrix is binary, the values will be stored as 1s or 0s. If it is real, it will store the values as a float. If it is a complex-valued matrix, it will store the values as a python complex number.

In order to access the actual values of the matrix, we use code like this:

```
import UFget
ourMatrix = UFget.UFget(1)
ourMatrix.A[1,1] #The value at row 1, column 1 of the matrix
ourMatrix.A[1,2] #The value at row 1, column 2 of the matrix
```

The matrix class also has built-in conversions from a Matrix into a String that displays the appropriate meta-data.

2. The second file is indexGet.py, which contains only one method, get-Info. Because of a lack of availability of the index file outside of Matlab, this is the only operating system-dependent piece of code in the entire library. The following code will refresh the

index:

```
import indexGet
indexGet.getInfo()
```

3. getIndex.py is a very simple script to update the index. In order to update the index, simply run `python -i getIndex.py`.
4. UFget_example.py is a script that tests to make sure loading the index works properly and loads the first 10 matrices in to ensure that works as well. To use this, simply run `python -i UFget_example.py`.