Notizen zu h5py

1 Was ist das?

h5py ist ein Python-Modul, es möglich macht, HDF5-Dateien in Python zu lesen und zu schreiben. HDF5 ist "ein Dateisystem für Daten".

In eine HDF5-Datei können datasets and groups geschrieben werden. Datasets sind "homogene, reguläre Arrays von Daten, vergleichbar einem NumPy-Array". Groups sind Container für Datasets und andere Gruppen.

HDF5 empfiehlt sich gegenüber Textdateien, weil das binäre Format weniger Platz benötigt. Trotzdem ist HDF5 standardisiert.

2 Dateien öffnen oder erzeugen

```
f = h5py.File('myfile.hdf5', 'w')
```

Erlaubte Modi sind:

r	Nur lesen, Datei muss exisitieren
r+	Lesen/schreiben, Datei muss exisitieren
w	Datei erzeugen, falls existiert, überschreiben (!)
w-	Datei erzeugen, Fehler falls Datei exisitiert
a	Falls Datei exisitiert: Lesen/Schreiben; sonst: Datei erzeugen (Standard)

3 Inhalt des Dateisystems auslesen

```
items = f.items()
```

gibt eine Liste aller Einträge in `f` zurück.

4 Daten lesen

Existiert das Dataset `bla`, so kann man dieses über

```
dat = f["bla"]
```

an `dat` binden. Die Werte in `f["bla"]` bekommt man über

```
dat[()]
```

5 Dataset erzeugen, Daten zuweisen, Daten lesen

5.1 Dataset erzeugen

```
ds = f.create_dataset(name, shape=None, dtype=None, data=None)
```

erzeugt ein neues Dataset. Entweder `data` oder `shape` müssen angegeben werden. Alternativ geht eine dictionary-artige Syntax:

```
f["some_name"] = some_data
```

5.2 Daten zuweisen

Zuweisen geht dann beispielsweise über

```
ds = f.create_dataset(name, shape=(200,200))
ds[5,:] = rand(200)
```

oder wie oben beschrieben mit

```
f["name"] = rand(200)
```

5.3 Daten lesen

Datasets unterstützen NumPy-artige Syntax, folgende Anweisungen sind erlaubt:

```
print(ds[::2, :])
print(ds[...])
```

6 Gruppen

Gruppen erlauben Struktur:

```
/    # root

    Gruppe1
        Daten1
        Daten2

    Gruppe 2
        MehrDaten1
        MehrDaten2
```

6.1 Gruppen erzeugen

```
gruppe = f.create_group("GruppenName")
```

Eine Gruppe kann Untergruppen enthalten. Namen und Parent auslesen:

```
print(gruppe.name)
print(gruppe.parent)
```

6.2 Daten in einer Group erzeugen

```
ds = gruppe.create_dataset("MehrDaten", data=array)
print(ds.name) # u"GruppenName/MehrDaten"
```

Zugriff ist möglich etwa über `data = file_obj["GruppenName/MehrDaten"]`.

7 Attribute

Zu Datasets oder Gruppen können Attribute hinterlegt werden (man denke an Parameter für eine Rechnung oder gar eine Commit-ID für den Code!):

```
ds.attr["parameter1"] = 0.0
gr.attr["commitID"] = "1f3fd71320f970e2275da196f8df2e417f620ed6"
```

8 Weitere Themen

8.1 Kompression von Datasets

Datasets können komprimiert werden:

```
s = f.create_dataset('Daten', shape=(2000,2000), compression='gzip',
                     compression_opts=4)
```

`compression_opts` gibt dabei an, wie stark komprimiert werden soll. Mögliche Kompressionsverfahren sind `gzip`, `lzf` (schnell) und `szip`.

Beachte: Nachträglich kann ein hdf5-File beispielweise mit dem Tool `h5repack` komprimiert werden, etwa so:

```
h5repack -v -f GZIP=4 file.hdf5
```

8.2 Paralles HDF5/MPI

Paralleles HDF5 wird von `h5py` nicht unterstützt.

8.3 Datasets vergrößern

Ein Dataset kann wachsen:

```
ds = f.create_dataset("MyDataset", (10, 1024), maxshape=(None, 1024))
print(ds.shape) # 10x1024
ds.resize(20, axis=0)
print(ds.shape) # 20x1024
```

8.4 Das kürzest-mögliche Tutorial

Von User DyneTrekk aus #scipy auf FreeNode:

```
h5py.File('foo.h5')['folder/dataset'] = np.linspace(0, 10)
```

9 Links und Quellen

- http://h5py.alfven.org/docs/intro/quick.html
- http://code.google.com/p/h5py/wiki/HowTo