Chapitre 9

Annexes

9.1 Aide-mémoire des fonctions rencontrées dans les librairies Python

NumPy

```
import numpy as np
np.linspace(start, stop, num=50, endpoint=True, retstep=False)
np.logspace(start, stop, num=50, endpoint=True)
np.arange(start, stop, step)
np.zeros(shape, dtype=float)
np.ones(shape, dtype=None)
np.empty(shape, dtype=float)
np.array(object, dtype=None)
np.empty_like(a)
np.zeros_like(a)
np.ones_like(a)
ndarray.ndim
ndarray.shape
ndarray.dtype
np.eye(N)
np.reshape(a, newshape)
np.dot(a, b)
a.T
a.transpose()
np.linalg.det(a)
np.linalg.inv(a)
np.linalg.eig(a)
np.random.rand(N)
np.meshgrid(x,y,indexing='ij')
np.gradient(f)
```

```
np.loadtxt(fname, comments='##', skiprows=0, usecols=None, unpack=False)

np.savetxt(fname, X, fmt='\%.18e', delimiter=' ', newline='\n ',
header=' ', footer=' ', comments='\#')
```

SciPy

```
from scipy import constants
from scipy import optimize
from scipy import misc
from scipy import integrate

constants.c
constants.m_e
constants.g
constants.physical constants[``speed of light in vacuum'']
constants.physical constants[``electron mass'']
constants.physical constants[``standard acceleration of gravity'']
optimize.curve_fit(f, xdata, ydata)
optimize.bisect(f, a, b, xtol=2e-12, maxiter=100,full_output=False)

optimize.newton(func, x0, fprime=None, tol=1.48e-08,maxiter=50,
fprime2=None, full_output=False)
optimize.fsolve(func, x0, xtol=1.49012e-08)
misc.derivative(func, x0, dx=1.0, n=1, order=3)
integrate.quad(f, a, b)
integrate.odeint(f, y0, t, tfirst=True)
```

Matplotlib

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

plt.figure(num=None, figsize=None)
plt.plot(x, y)
plt.errorbar(x, y, yerr=None, xerr=None)
plt.scatter(x, y)
plt.quiver(x,y,u,v)
fig.suptitle(str)
ax = plt.subplot(m,n,a)
ax = plt.subplot() (ou ax = fig.gca() )
ax = plt.subplot(projection='3d') (ou ax = fig.gca(projection='3d')
ax.plot(x, y)
ax.barh(y, width)
ax.contour(x,y,z,levels)
ax.plot_surface(x,y,z)
ax.quiver(x,y,u,v)
ax.streamplot(x, y, u, v, linewidth=1, density = 2,
```

```
ax.set_title(str)
ax.set_yticks(labels)
ax.set_xlabel(xlabel)
ax.set_ylabel(ylabel)
ax.set_zlabel(zlabel)
ax.imshow(x)
plt.fill_between(x, y1, y2)
plt.clabel(cs)
plt.axis('equal')
plt.axis('scaled')
plt.grid()
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.legend()
plt.title(label)
plt.savefig(fname)
plt.imshow(x)
plt.show()
mpimg.imread(fname)
mpimg.imsave(fname, arr)
```