Contents

1 Code-Listings

Code-listings are wrapped with an `mdframe` which enables page-breaking. The Code-listing title must be set by renewing the `codeTitle` command.

The Cluster Visualize Function

```
1  def visualize(self, i):
2      """
3      Shows a snapshot of the current state of the
           system at time t_i.  Particles are scaled by
           size.
4      """
5      fig = plt.figure()
6      ax = fig.add_subplot(111, projection='3d')
7      ax.set_xlim3d([-50, 50])
8      ax.set_ylim3d([-50, 50])
9      ax.set_zlim3d([-50, 50])
10     ax.view_init(elev=11, azim=360*float(i)/self.
           n_steps)
11     # scatter plots each body
12     for body in self.cb_list:
13         ax.scatter(body.r[i,0], body.r[i,1], body.r[i
               ,2], s=body.radius, c=body.radius
               **4/70**4, cmap='autumn')
14     plt.axis('off')
15     plt.savefig('tmp_%04d.png' % i)
16     plt.close()
```

Listing 1: Cluster.py

Does it do C++?

```cpp
void main(int argc, char* argv[]) {
    std::cout << "This is a test" << std::endl;
    return 0;
}
```

Listing 2: Testing.cpp

2 Tables

Run-time	Performance	Results
11s	93%	6/10
4s	76%	8/10
6s	82%	8/10
7s	78%	10/10

Table 1: A test-table

N	Result	Absolute error	Time [sec]
5	0.2642	0.0714	0.0004
10	0.0719	0.0463	0.0267
15	0.2390	0.0366	1.7411
25	0.1958	0.0030	6.6549

Table 2: Another test-table

3 Proclamations

Theorem 3.1 [Euclid]
For every prime p, there is a prime $p' > p$. In particular, the list of primes,

$$2, 3, 5, 7, \dots$$

is infinite.

Lemma 3.2 [Aubin-Lions]
Let X_0, X and X_1 be three Banach spaces with $X_0 \subseteq X \subseteq X_1$. Suppose that X_0 is compactly embedded in X and that X is continuously embedded in X_1. For $1 \leq p, q \leq \infty$, let

$$W = \{u \in L^p([0,T]; X_0) \mid \dot{u} \in L^q([0,T]; X_1)\}.$$

1. If $p < \infty$, then the embedding of W into $L^p([0,T]; X)$ is compact.
2. If $p = \infty$ and $q > 1$, then the embedding of W into $C([0,T]; X)$ is compact.

Emphasis is done using `emphcolor`, which can be redefined using `definecolor`. As you can see, compactly embedded is emphasised using color and not shape.