# PyBolt: A PyCube Realistic Lightning and Electric Discharge Simulator

William Baker, Peter Brinkmann, George Francis

August 5, 2004

**Abstract**

PyBolt is a realtime distributed graphics program that uses Python, Syzygy, and SWIG bindings. PyBolt uses the Master/Slave framework of Syzygy, and PyBolt also uses the numarray numerical python package for its linear algebra operations (see calcltn3D.py).

PyBolt simulates lightning and electric discharge. Users can place charges in the scene, and then trigger electrical discharge to occur. PyBolt determines how the electrical discharge should occur, and how the potential should change in the scene as a result of the discharge.

The mathematical model used to simulate lightning come from the paper "A Discrete Model for the Lightning Discharge" by William W. Hager This paper was published in the "Journal of Computational Physics" 144, 137-150 (1998).

## 1 How to Run PyBolt

### 1.1 After Copying Files from CVS

After copying the PyBolt files from CVS, there is still one step that needs to be done before running the program. If you plan on using the default settings, all that needs to be done is to type the following at the command prompt:

```
[..]$ python calc_laplace.py
```

This will create a file named "inv_laplace12" which contains the laplacian matrix that is used when the Cube's potential is represented by a 12x12x12 matrix.

If a different level of resolution is being used inside the Cube, then the laplacian needs to be calculated with a size other that 12. To do this, you need to use calc_laplace.py in Python's interactive mode:

```
[...]$ python
>>> import calc_laplace.py
>>> calc_laplace.calc(size)
```

This will precompute a laplacian to be used for for the specified size.

## 1.2 Starting PyBolt

To start PyBolt on a console:

```
[...]$ python bolt.py
```

To run PyBolt in the Cube (with North as the Master):

```
[...]$ dex pycube bolt.py
```

To run PyBolt on the Cube's console (normally used for testing purposes):

```
[...]$ dex vfire bolt.py
```

To run PyBolt in the Cube (with the console Fire as the Master): NOTE: This is not recommended!! Fire is not on the gigabit network. Recommended usage is to run PyBolt with North as the Master.

```
[...]$ dex firecube bolt.py
```

## 1.3 Using PyBolt

### 1.3.1 User Modes and Button Control

Once PyBolt is started, a title screen will appear that will persist until all the files that need to be loaded have finished. After the title screen disappears a cursor will appear.

Once the cursor has appeared, all of the input will be using the wand, and the 6 buttons on the wand. The input has two modes, one of these modes is Zeus mode, and the other is Cleanup Mode. Zeus is the default mode, and is used primarily for placing charges and triggering lightning. Cleanup mode is used primarily for deleting charges that have been placed in the scene. *Important:* Cleanup mode can only be entered if charges exist. This is done to prevent users from trying to delete charges that don't exist. In fact, if while in Cleanup mode all of the charges are deleted, the program will automatically return to Zeus mode.

*Wand Controls: Zeus Mode*

A Button: Press and hold while moving the wand to draw a charge. Release the button when the charge is at the desired size.

B Button: Switch between positive and negative charges (the cursor color will indicate which mode is the current mode, Magenta=Postive, Cyan=Negative)

C Button: Press once to intiate strike of lightning. Press again to make the lightning disappear. If no discharge occurs, cursor will temporarily be green.

X Button: Toggle on and off the potential markers inside the Cube.

Y Button: Reset the potential.

Z Button: Switch to Cleanup mode (if there is at least one charge drawn)

*Wand Controls: Cleanup Mode*

A Button: Cycle through the charges in one direction.

B Button: Cycle through the charge in the other direction.

C Button: Toggle on and off solid textured charges.

X Button: Delete selected (yellow) charge.

Y Button: Delete all charges.

Z Button: Switch to Zeus mode.

### 1.3.2 Color Guide

*Cursor Colors:*

Magenta: Ready to draw a positive charge.

Cyan: Ready to draw a negative charge.

Green: After trying to cause a lightning strike, it has been determined that there will not be any electric discharges.

Orange: The potential has just been reset.

White: User is in Cleanup mode.

*Charge Colors:*

Magenta: Positive charge.

Cyan: Negative charge.

Green: After trying to cause a lightning strike, it has been determined that there will not be any electric discharges.

Orange: The potential has just been reset.

Yellow: User is in Cleanup mode, and this charge is currently selected.

*Potential Marker Colors:*

Red/Orange: Positive potential.

Green: Negative potential.

Yellow: Zero Potential.