USER'S GUIDE

Leigh Korbel
University at Buffalo

May 5, 2017

Contents

1 Introduction

This package was created to visualize cosmological simulation data on a planetarium dome that is equipped with Sky-Skan's DigitalSky 2 software. The package is also capable of being visualized on a PC via Partiview software. DigitalSky 2 is proprietary commercial software, therefore planetarium end users must have the software and a license from Sky-Skan if they wish to utilize this package on their planetarium dome. Information can be found here:

`https://www.skyskan.com/products/ds`

Partiview is free, open-source software from the National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign. The Partiview executable is included with this package (please see the Partiview License at the end of this document), but PC end users can also download the software from the following location:

`http://virdir.ncsa.illinois.edu/partiview/`

The only requirement for the end user is to have either of the aforementioned software distributions installed on their planetarium/PC before downloading this package.

In addition to the software listed above, this package would not have been possible without GADGET-2 which is a publicly available code (under the GNU general public license) for cosmological N-body/SPH simulations on massively parallel computers with distributed memory. GADGET follows the evolution of a self-gravitating collisionless N-body system, and allows gas dynamics to be optionally included. For our purposes, GADGET is used to address the formation of large-scale structure in the universe. This code was used at the Center for Computational Research (CCR) in Buffalo, NY - a leading academic supercomputing facility that maintains a high-performance computing environment - to produce the data that is visualized in this package. For more information about GADGET, visit:

`http://wwwmpa.mpa-garching.mpg.de/gadget/`

This package is split into two components, a DigitalSky component and a Partiview component. The primary motivation behind this is the fact that DigitalSky interprets values differently than Partiview does (e.g. point size and luminosity values). Therefore, certain settings that work well in Partiview do not translate well to DigitalSky. In addition, some Partiview commands are not accepted by Digital Sky (e.g. jump, interest, clip, etc.). The differences are seen in the configure files, but both components can use the same data files (see Section 3). Digital Sky users will need to copy the 'data' subfolder of Partiview to their system as well as the configure files within the 'DigitalSky' folder.

In order to understand the basic operation of the package, a familiarity with the Partiview file formats is necessary. Planetarium operators familiar with DigitalSky or The Digital Universe Atlas (from Hayden Planetarium) will likely be familiar with the file formats used, even if they have never dealt with Partiview, and likewise should be familiar with the other elements (e.g. button scripts) necessary to get the package running successfully on their planetarium domes.

2 Simulations

The Lambda-Cold Dark Matter (ΛCDM) model is the standard theoretical paradigm for understanding cosmological structure formation in the universe. With initial conditions consisting of a nearly scale-free spectrum of Gaussian fluctuations as predicted by cosmic inflation, and with cosmological parameters that have been increasingly constrained by observations (e.g Planck observations of the cosmic microwave background), ΛCDM makes detailed predictions for the hierarchical gravitational growth of structure. The model assumes that structure grew from weak quantum density fluctuations present in the otherwise homogeneous and rapidly expanding primordial universe. These fluctuations are amplified by gravity, eventually turning into the rich structure that we observe today.

In a ΛCDM universe, the formation of structure is strongly affected by the presence and nature of cold dark matter. It clumps into "halos" which grow by the gravitational collapse and hierarchical aggregation of ever more massive systems. Galaxies form at the centers of these dark matter halos by the cooling and condensation of baryonic gas that eventually fragments into stars upon becoming sufficiently dense. Groups and clusters of galaxies form as halos and aggregate into larger systems. They are arranged in what is commonly known as the "cosmic web" - the large-scale pattern of galactic filaments and sheets - which are a nonlinear gravitational "sharpening" of the pattern already present in the Gaussian random field of initial fluctuations. While the initial, linear growth of density perturbations can be calculated analytically, the gravitational collapse of fluctuations and the subsequent hierarchical build-up of structure that is observable today is a highly nonlinear process which can be followed in detail only through direct numerical simulation.

Computer simulations have played an important role in cosmology since the computer science field began back in the 1960s and 1970s. Historically, these simulations were quite small (on the order of a few hundred or thousand particles) and focused on astrophysical problems such as the evolution of the solar system. With the steady improvement in computer hardware and computational algorithms the field has advanced to such a point where large simulations containing billions of particles are able to simulate a representative sample of the entire universe. Cosmological N-body simulations are indeed some of the largest simulations to be performed in all of computer science.

N-body simulations are a primary theoretical tool for physicists since they have been essential for understanding the nonlinear gravitational growth of structure in the universe, such as the aforementioned galaxy filaments and galaxy halos. They have been absolutely critical for studying the properties of dark matter halos in the ΛCDM cosmology. Without direct numerical simulation, the hierarchical buildup of three-dimensional structure would be largely inaccessible. Simulations are also an invaluable tool for analyzing galaxy surveys, for studying the evolution of clusters of galaxies, and for semi-analytical models of galaxy formation.

Successful large-scale simulations of the evolution of galaxies, with results consistent with what is actually seen by astronomers in the night sky, provide evidence that the theoretical underpinnings of the models employed (i.e. the supercomputer implementations of the standard ΛCDM model) are sound bases for understanding galactic dynamics and the history of the universe.

The dominant mass component in the ΛCDM model is the cold dark matter, which is assumed to consist of weakly interacting elementary particles that interact only gravitationally. The N-body simulations performed for this package consist either of dark matter particles only or a combination of dark matter particles and gas particles (i.e. baryonic matter). This collisionless dark matter/baryonic fluid is represented by a set of massive discrete point particles. The dark matter particles are on the order of 10^9 solar masses each. This representation is only a coarse approximation, and improving its reliability requires the use of as many particles in the simulation as possible, while at the same time, keeping the simulation computationally tractable. The simulations performed for this package consist of either 64^3 particles (262,144 total) or 128^3 particles (2,097,152 total), or when both dark matter and gas particles are used, twice those numbers. For research purposes, these would be considered small simulations, but for visualization purposes, our hands are tied due to several factors. The first issue is the rendering capabilities of our GPU. We found that both our PCs and the planetarium renderer were maxed out with the simulations containing 128^3 particles. It is possible that state-of-the-art equipment could handle 256^3 particles and above, but we are also faced with another limitation, and that is the fact that both Partiview and Digital Sky 2 are 32-bit software and thus their capabilities for handling large amounts of RAM are limited. It is possible to recompile the Partiview source code into a 64-bit version for those interested. In addition, Sky-Skan's newest edition of Digital Sky, named Dark Matter (set to be released in the summer of 2017), will be 64-bit. We found that the maximum size for data files that could be handled well by both Partiview and Digital Sky 2 was around 2.1 gigabytes. Therefore, we have chosen the largest data sets that could be rendered smoothly and be handled by the software we are using.

The simulation code GADGET-2 (GAlaxies with Dark matter and Gas intEracT) was used to produce the data that is visualized in this package. GADGET-2 was written by Volker Springel in standard ANSI C, and should run on all parallel platforms that support MPI (Message Passing Interface) for those that are interested in generating their own data. The code was run at the Center for Computational Research (CCR) in Buffalo, New York. This code is particularly suitable for simulations of cosmic structure formation. The simulation volume must be self-contained, therefore periodic boundary conditions must be used. The most natural geometry for a volume with periodic boundary conditions is a cube, thus the code is run using periodic boundary conditions on a cubic volume with a side length of 100 Mpc (326,156,600 light years). We simulate a ΛCDM cosmology, resulting in the following values for our parameters:

- $\Omega_m = 0.315$, the mass density of both ordinary (baryonic) matter, Ω_b, and dark matter, Ω_d.
- $\Omega_\lambda = 0.685$, the mass density of dark energy (i.e. the "cosmological constant").
- $\Omega_b = 0.0492$, the mass density of baryonic matter (required for gas simulation).
- $H_0 = 0.672$, the Hubble parameter - used to describe the expansion of the universe.

Note that the total density parameter is given by $\Omega_0 = \Omega_b + \Omega_d + \Omega_\lambda = 1$, which results in a flat universe. For visualizations involving time evolution, the first snapshot of the system is taken at redshift $z = 50$, which corresponds to 46.553 million years after the birth of the universe (13.832 billion years since the Big Bang) given the parameters used above.

GADGET-2 outputs snapshots of the state of the system at redshifts specified by the user. The number of snapshots used in the time-evolution visualizations were chosen based on the desire for the smoothest transitions from one time-step to the next under the constraint that the data file containing the particle information had to be small enough for either Partiview or DigitalSky 2 to process them (recall that both software packages are 32-bit). The visualizations involving time evolution use the x, y, and z positions of the particles (either dark matter or gas) at these specified snapshots and their velocity as a substitute for representing the density of the system. Since

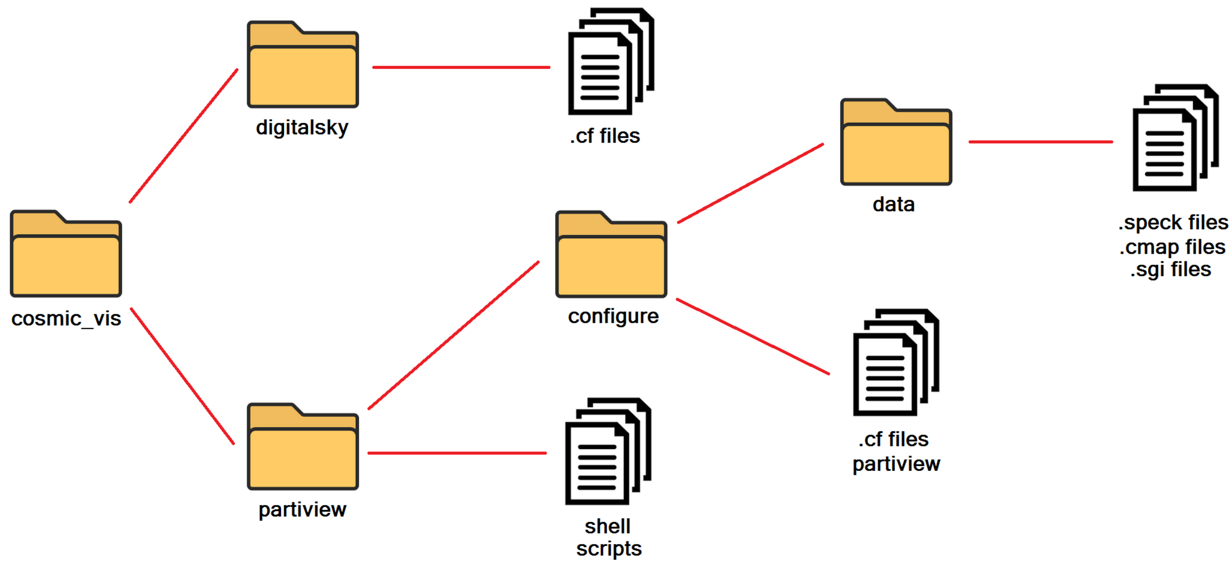

Figure 1: The directory structure of the package is shown here. DigitalSky 2 users must copy the 'digitalsky' folder to their systems as well as the 'data' subfolder located within the 'partiview' folder.

particles in denser regions will have higher velocities than those in sparser regions, the particle velocities, which are readily available from the GADGET-1 output, can be used in lieu of density for visualization purposes (this is explained in depth in the following Section 3). The static visualizations are the state of the system at $z = 0$ (the present). GADGET-2 uses a glass-like distribution of particles for its initial setup, and this is readily seen in the opening snapshots for the time evolution visualizations on the 64^3 particle data sets. GADGET-2 also requires a random seed used to create the initial conditions. For visualizations with two versions, a different random seed was used, which results in a different final configuration for the particles.

3 File Structure

The directory structure of the package is shown in Fig. 1. DigitalSky users need only copy two directories to their systems in order to use the package, the 'digitalsky' folder and its contents, as well as the 'data' subfolder located within the 'partiview' folder. The only file types that can be ignored by DigitalSky users are the shell scripts that can be executed by Partiview users based on the operating system they are using (Linux, Windows, Macintosh). There are several file types that this package deals with. A list of each can be found in Table 1. A description of each can be found in the subsections that follow. Note that despite the different file extensions, all of these files that follow (save for the shell scripts) are basic ASCII text files.

3.1 Start Scripts (.sh/.bat/.command) Files

Start scripts are for Partiview users and are basically shortcuts for executing Partiview in conjunction with a configure file. Users familiar with the command line can ignore these altogether and simply execute Partiview using the following format (without the square brackets):

File Name	Description
*.cf	Pre-loaded config. commands
*.speck	Formatted data file
*.cmap	Color map
*.sgi	Texture (image) file
*.sh	Linux start script
*.bat	Windows start script
*.command	Macintosh start script

Table 1: Different file types found within the package with short explanations of their function.

```
[path to partiview executable] [path to .cf file]
```

Windows users execute Partiview using `partiview`, while Linux and Mac users execute using `./partiview`, a difference of "`./`". Note that shell scripts are also run as executables and follow the same format. Each start script extension corresponds to the operating system (see Table 1 for reference).

3.2 Configure (.cf) Files

The configure files contain the pre-loaded commands necessary to interpret and visualize the data. As mentioned in Section 1, DigitalSky accepts some Partiview commands, but not all of them. For the instances in which DigitalSky does not accept Partiview commands, there exist complementary commands within DigitalSky that accomplish the same tasks. These need to be executed within DigitalSky, as explained in Section 6. For this reason, there exist separate `.cf` files for DigitalSky and Partiview. For details regarding Partiview commands, see Section 4.

In the case where both dark matter and gas particles are used, each particle data set is contained in its own data file (a `.speck` file, see below), and that file is assigned to a group within the configure file. Object 'g1' is the dark matter particles (assigned as 'dm') and object 'g2' is the gas particles (assigned as 'gas'). Each group can be toggled on/off within either Partiview or DigitalSky.

3.3 Speck (.speck) Files

The speck files contain the data from the simulations. At the top of each file are Partiview commands that tell Partiview what each data column corresponds to. These commands are also found in the configure files for DigitalSky since that software wants to see them there, whereas I found that Partiview (in the case where there are multiple groups each having their own speck file) wanted them in the speck files. The fact that the commands are repeated in the speck files (in the case of using the speck files for DigitalSky) is not an issue.

The data itself consists of five columns. In some instances there may be a sixth column that corresponds to the mass of the particle. This can be ignored since it is not used. The first three columns correspond to the x, y, and z positions of the particle. The fourth column is just the number 1, which corresponds to the only texture being used (if there were multiple textures, they would each be given a corresponding number). The fifth column corresponds to the color that the particle is assigned (see subsection 3.4 below for reference). In the case of time evolution, each snapshot is assigned a 'datatime' number, starting from zero, which Partiview/DigitalSky use for time-stepping.

1	2	3	4	5	6	7	8	9	10	11
0000FF	0909FF	1212FF	1C1CFF	2525FF	2E2EFF	3838FF	4141FF	4B4BFF	5454FF	5D5DFF
12	13	14	15	16	17	18	19	20	21	22
6767FF	7070FF	7979FF	8383FF	8C8CFF	9696FF	9F9FFF	A8A8FF	B2B2FF	BBBBFF	C4C4FF
23	24	25								
CECEFF	D7D7FF	E1E1FF								

BLUE GRADIENT

1	2	3	4	5	6	7	8	9	10	11
FF3F00	FF4609	FF4D12	FF541C	FF5B25	FF622E	FF6938	FF7041	FF774B	FF7E54	FF855D
12	13	14	15	16	17	18	19	20	21	22
FF8C67	FF9470	FF9B79	FFA283	FFA98C	FFB096	FFB79F	FFBEA8	FFC5B2	FFCCBB	FFD3C4
23	24	25								
FFDACE	FFE1D7	FFE9E1								

DARK ORANGE GRADIENT

1	2	3	4	5	6	7	8	9	10	11
3FFF00	46FF09	4DFF12	54FF1C	5BFF25	62FF2E	69FF38	70FF41	77FF4B	7EFF54	85FF5D
12	13	14	15	16	17	18	19	20	21	22
8CFF67	94FF70	9BFF79	A2FF83	A9FF8C	B0FF96	B7FF9F	BEFFA8	C5FFB2	CCFFBB	D3FFC4
23	24	25								
DAFFCE	E1FFD7	E9FFE1								

NEON GREEN GRADIENT

1	2	3	4	5	6	7	8	9	10	11
FF7F00	FF8309	FF8812	FF8D1C	FF9125	FF962E	FF9B38	FF9F41	FFA44B	FFA954	FFAE5D
12	13	14	15	16	17	18	19	20	21	22
FFB267	FFB770	FFBC79	FFC083	FFC58C	FFCA96	FFCF9F	FFD3A8	FFD8B2	FFDDBB	FFE1C4
23	24	25								
FFE6CE	FFEBD7	FFF0E1								

ORANGE GRADIENT

1	2	3	4	5	6	7	8	9	10	11
4900E5	4E07E6	530EE7	5916E8	5E1DE9	6325EA	692CEB	6E33EC	743BED	7942EE	7E4AEF
12	13	14	15	16	17	18	19	20	21	22
8451F0	8959F2	8E60F3	9467F4	996FF5	9F76F6	A47EF7	A985F8	AF8CF9	B494FA	B99BFB
23	24	25								
BFA3FC	C4AAFD	CAB2FF								

PURPLE GRADIENT

1	2	3	4	5	6	7	8	9	10	11
FF0000	FF0909	FF1212	FF1C1C	FF2525	FF2E2E	FF3838	FF4141	FF4B4B	FF5454	FF5D5D
12	13	14	15	16	17	18	19	20	21	22
FF6767	FF7070	FF7979	FF8383	FF8C8C	FF9696	FF9F9F	FFA8A8	FFB2B2	FFBBBB	FFC4C4
23	24	25								
FFCECE	FFD7D7	FFE1E1								

RED GRADIENT

Figure 2: There are seven color maps available for use with the data sets. Pictured here are the six color-to-white gradients available. Not pictured is the "rainbow" gradient, which transitions from red to pink to purple to blue.

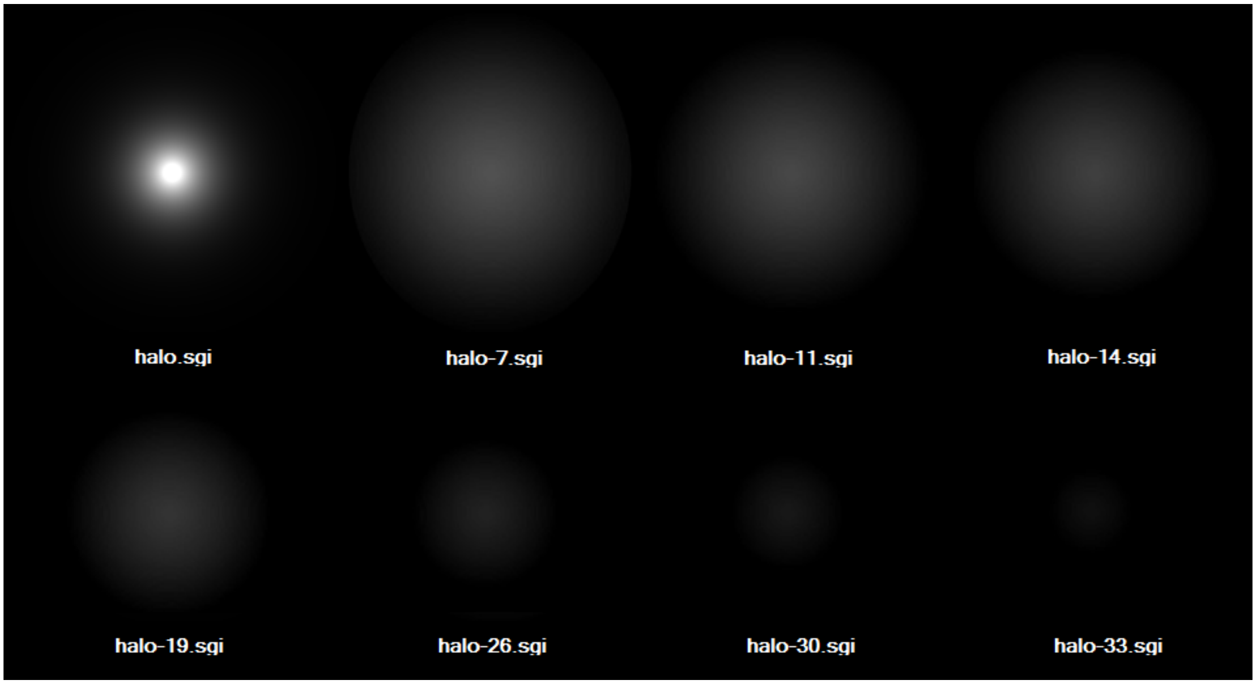

Figure 3: There are two different default halos used as textures for the particles. The brighter halo, 'halo-7.sgi,' is used in the 64^3 data sets, whereas the dimmer halo, 'halo-14.sgi,' is used in the 128^3 data sets. These different halos were employed so that the luminosity settings in the configure files would remain consistent. Each simulation used the same box size of 100 megaparsecs, which results in a much denser and brighter result for the data set with more particles. There are several other options, as shown above, which the user is free to experiment with.

3.4 Color Map (.cmap) Files

As mentioned at the end of Section 2, the velocities of the particles were used to represent densities within the simulation. The velocity was calculated via the equation $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$, and the logarithm of the velocity was used to determine a color value (an integer) to assign to each particle. The color gradients (see Fig. 2) consist of 25 possible colors, each from a chosen color transitioning to white (save for the 'rainbow' gradient). The maximum velocity reached by a particle was determined and the range was divided into 25 subgroups, with each group being assigned a number from 1-25 based on their velocity. The lowest velocity particles were assigned the solid color, whereas the highest velocity particles were assigned a mostly white version of that color. The fifth column of the speck file corresponds to the numbers used to assign colors from the color map.

The default color gradients are blue-to-white for dark matter and red-to-white for gas, whose overlap results in an appealing purple-to-white colored gradient (see Fig. 4 for reference). The color maps are user configurable and easy to create (open one of the color map files for a link to a useful page for doing so).

3.5 Texture (.sgi) Files

These are simply image files. For our purposes, the only images used are halos that replace pixels for representing each particle. There are two default halos used in this package: 'halo-7.sgi' is the default halo for the 64^3 particle simulations and 'halo-14.sgi' is the default halo used for the 128^3 particle simulations. These different halos were employed so that the luminosity settings in the configure files would remain consistent. Each simulation used the same box size of 100 megaparsecs, which results in a much denser and brighter result for the data set with more particles. An image of each, as well as other options, are shown in Fig. 3.

3.6 Simulations & Their Respective Files

Now that the user has an understanding of each file type and their contents, I will correlate each simulation with its respective `.cf` and `.speck` file. The following list is for dark matter only simulations. Note that there are no stationary versions of this variety since groups can be toggled on and off in both Partiview and DigitalSky, thus a stationary dark matter and gas visualization simply needs the gas particle group turned off for a stationary dark matter only visualization:

1. `dm_evolve_64_version1.cf` - This is a simulation of 64^3 particles evolving from redshift $z = 50$ to $z = 0$ using 50 snapshots. The corresponding speck file is `ver1_evolve_64.speck`.

2. `dm_evolve_64_version2.cf` - The same as above with a different random seed resulting in a different final particle configuration. The corresponding speck file is `ver2_evolve_64.speck`. Note that this random seed is only used in the second versions of the dark matter only visualizations here. For the first versions of these visualizations, and all of the dark matter and gas simulations, the same random seed is used.

3. `dm_evolve_128_version1.cf` - This is a simulation of 128^3 particles evolving from redshift $z = 50$ to $z = 0$ using 25 snapshots. The corresponding speck file is `ver1_evolve_128.speck`.

4. `dm_evolve_128_version2.cf` - The same as above with a different random seed resulting in a different final particle configuration. The corresponding speck file is `ver2_evolve_128.speck`.

The following list is for dark matter and gas simulations. The first two are stationary simulations at redshift $z = 0$ and the final two are time evolutions simulations from redshift $z = 50$ to $z = 0$.

1. `dmandgas_64.cf` - The corresponding speck files for this visualization are `dm_64.speck` and `gas_64.speck`.

2. `dmandgas_128.cf` - The corresponding speck files for this visualization are `dm_128.speck` and `gas_128.speck`.

3. `dmandgas_evolve_64.cf` - This simulation consists of 2×64^3 particles using 25 snapshots. The corresponding speck files for this visualization are `dm_evolve_64.speck` and `gas_evolve_64.speck`.

4. `dmandgas_evolve_128.cf` - This simulation consists of 2×128^3 particles using 12 snapshots. The corresponding speck files for this visualization are `dm_evolve_128.speck` and `gas_evolve_128.speck`.

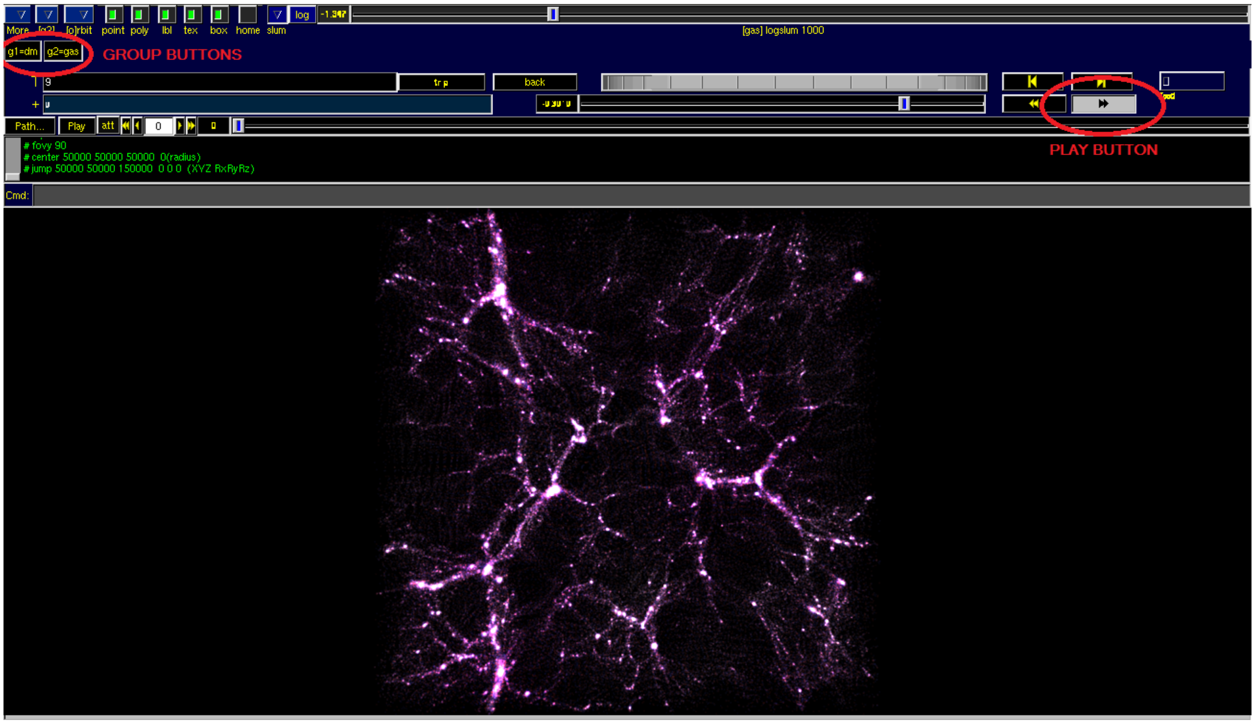

Figure 4: The Partiview user interface. Of particular importance are the group buttons (circled on the left), which are used to toggle on/off either dark matter (dm) or gas for visualizations with both, and the play button (circled on the right), which is used to play through the time steps in visualizations with time evolution.

4 Partiview Commands

The interested user can also visit the following webpage for an extensive list of Partiview commands with comprehensive explanations for each:

`http://virdir.ncsa.illinois.edu/partiview/doc/partiview-4.html`

For Partiview commands that are used within our `.cf` and `.speck` files, see the comments within those files after each command for an explanation therein.

5 Using Package in Partiview

Using the package in Partiview is very simple, since all of the work is already done via the commands in the configure files (which contain the pre-loaded configuration commands) and speck files. Users can enter new commands via the command line (see Fig. 4, listed as Cmd:), but this isn't necessary as all settings have been optimized for our purposes. Note that the buttons for toggling the groups on/off are on the left, and in the case of time evolution visualizations, the play button is on the right.

Moving through the data is done either by flying, orbiting, rotating, or translating. Rotating is done by pressing and holding the left mouse button and scrolling within the viewing window. Flying is done by pressing and holding the right mouse button and moving forward or backwards (quickly) and then releasing. To use the orbit or translate option, it must be selected by clicking the button in the upper left corner. Rotating rotates about the point of interest (origin chose in configure file) and translating moves in a direction parallel to the display. To stop any motion, simply click the left mouse button.

In addition, the interface has options to toggle on/off points, polygons, labels, textures, and boxes (labels and boxes are not used in this package), and the option to adjust the luminosity via the slider. Also, as seen in Fig. 4, the buttons to turn groups on/off are on the left, and the play button (note that the time control window only shows for time evolution visualizations) is on the right.

The default global viewing options are set such that the point of interest (i.e. the origin) is in the center of the simulation and the view begins outside of the simulation box (via a jump command). There are several other options that are commented out within the configure files. One option is to begin in the center of the simulation box and the other is the same, except in this case the point of interest is no longer the center of the box. This is due to the fact that flight scales in Partiview are linear near the point of interest and logarithmic away from it. This results in slow speeds near the center and fast speeds away (so long as the center is the point of interest). The user will have to give their mouse a workout if they wish to fly forward/backward in the former scenario. In the latter scenario, this is no longer the case and the user can fly through the data quickly (note that orbiting is affected negatively in this case though, since orbiting always occurs around the point of interest).

Note that for the time evolution visualizations, the default speeds are listed within the corresponding speck files and were chosen based on the number of snapshots used. The user can edit the evaluation speed by changing the command in the speck file, or simply adjust the speed by using the time control buttons on the right (these are the two buttons above the play forward/backward buttons circled in the figure). These buttons increase/decrease the time by $(0.1 \times speed)$ data time units.

If there is any confusion regarding the Partiview user interface, please see the Partiview User Guide available at:

`www.amnh.org/content/download/54255/811110/file/Partiview%2520Users%2520Guide.pdf`

6 Using Package with DigitalSky 2

Planetarium operators looking to use this package with DigitalSky on their planetarium domes likely already have a working familiarity with the software. Of particular importance for our purposes is replacing the Partiview commands that DigitalSky does not accept with DigitalSky commands that reproduce them. This is best accomplished by end-user created button scripts. Once the button scripts are created, they can be used for any of the time evolution simulations. The particular commands that we wish to reproduce from Partiview are:

- **speed [num]** - For time-dependent data, advance datatime by num many time units per wall-clock second.
- **interest [X Y Z]** - Set point of interest. This is the center of rotation in orbit and rotate modes. And, in orbit mode, translation speed is proportional to the viewer's distance from this point.

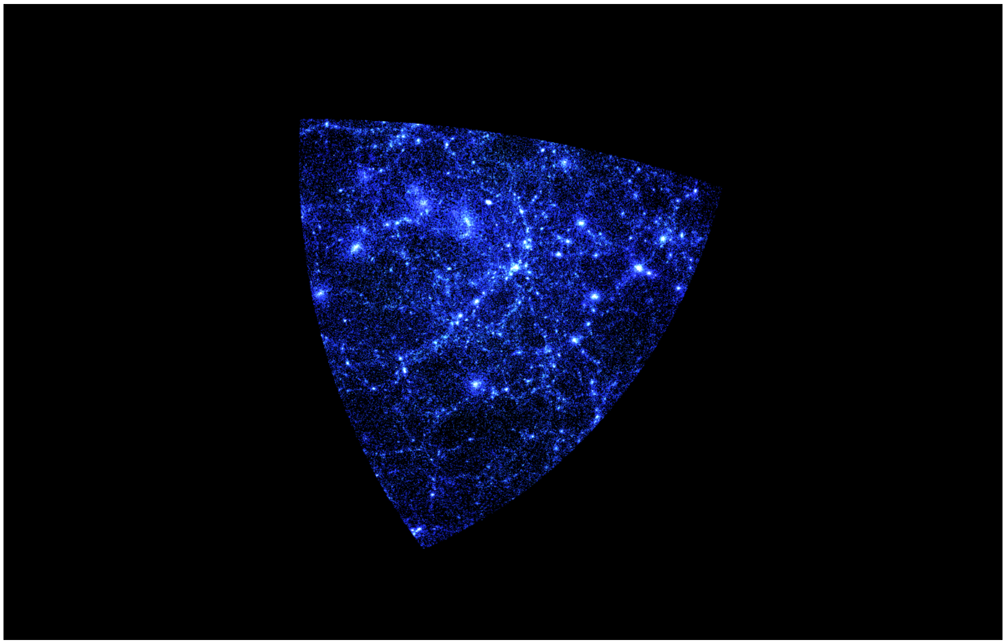

Figure 5: Upon loading the data into DigitalSky, the view is as above. We are outside of the simulation box with a corner view that makes the box look like a rounded triangle. Using button scripts (or simply navigating) we can adjust the view to look cubic (as with Partiview), or move to the center of the simulation box.

- **jump [X Y Z] [Rx Ry Rz]** - Get or set the current position (X Y Z) and/or viewing (Rx Ry Rz) angle.

The first is replaced by the `Run` command in DigitalSky. The other two are replaced by `Scene Camera Locate` commands. Before we begin, note that there is a command commented out in each of the DigitalSky configure files, `eval prescale 10`, which can be used to make the textures become more evident. This in turn increases the luminosity of the particles, so if this is used, the user will likely have to decrease the luminosity using the 'slum' slider within the Dataset Viewer window (see Fig. 7).

First, the data needs to be loaded into DigitalSky. Be sure to place the configure files in the same folder as the other files (speck files, color maps, and textures) coming from the 'data' folder. Assign a group name for all of the configure files (e.g. our group name is KINNEYSIMS, as can be seen in the following images). Click on the 'Plug-Ins' pull down menu on the button panel screen (see Fig. 6) and select 'Dataset Viewer.' This opens up the Dataset Viewer window (see Fig. 7), from which the simulation of choice can be loaded from the group. Make sure that 'dm' (and 'gas' when applicable) are both check marked as on. You should see something similar to what is shown in Fig. 5.

Now we need to create button scripts to adjust the view, or more importantly, for time evolution visualizations, get the time steps to run. In Fig. 6, we see a standard page of DigitalSky scripted buttons. Let's focus on getting the `Run` command working first. On the button panel screen, right-click on an empty button and select 'new script,' which opens up the DigitalSky Script Editor

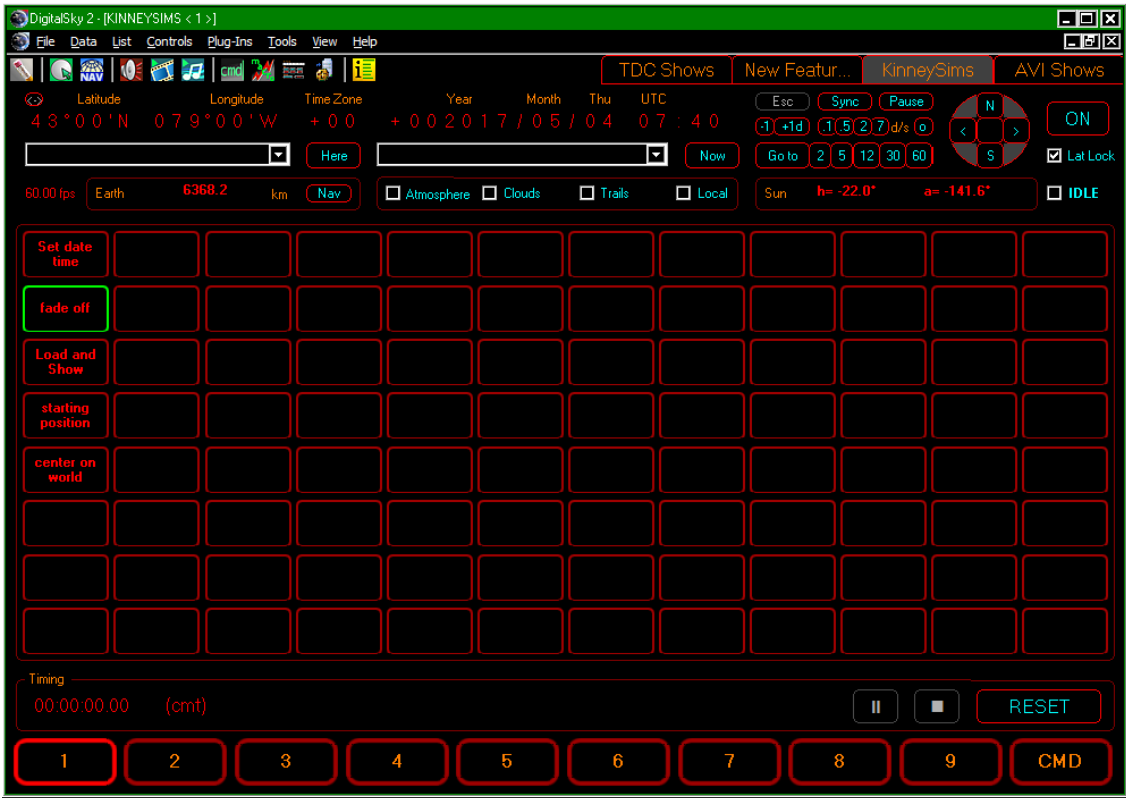

Figure 6: A standard page of DigitalSky scripted buttons. The button in green is the button we will script for running the time evolution in the visualization. It is currently in its third stage, 'fade off.'

window. Give the button a title (e.g. Run Time), and then copy the following script into the Script Assistant (very similar to what is seen in Fig. 8):

```
    Scene YOURGROUPNAME g1 on
    Buttontext "start evolution"
STOP
    Scene YOURGROUPNAME Run -1 1
    Buttontext "stop evolution"
STOP
    Scene YOURGROUPNAME Run -1 0
    Buttontext "fade off"
STOP
    Scene YOURGROUPNAME g1 off
```

This would work for time evolution on a dark matter only simulation. We would also need to create a duplicate command for g2 (gas) if we wanted both groups to run. This creates a button with three sections that has programmed pauses until the operator clicks the button again to continue. At each pause the new button text appears. For the `Run` command, the first number `-1` tells DigitalSky to run continuously (otherwise, put in a value in seconds), and the second number `1` tells DigitalSky to progress one time step every second (for one step every half second we would

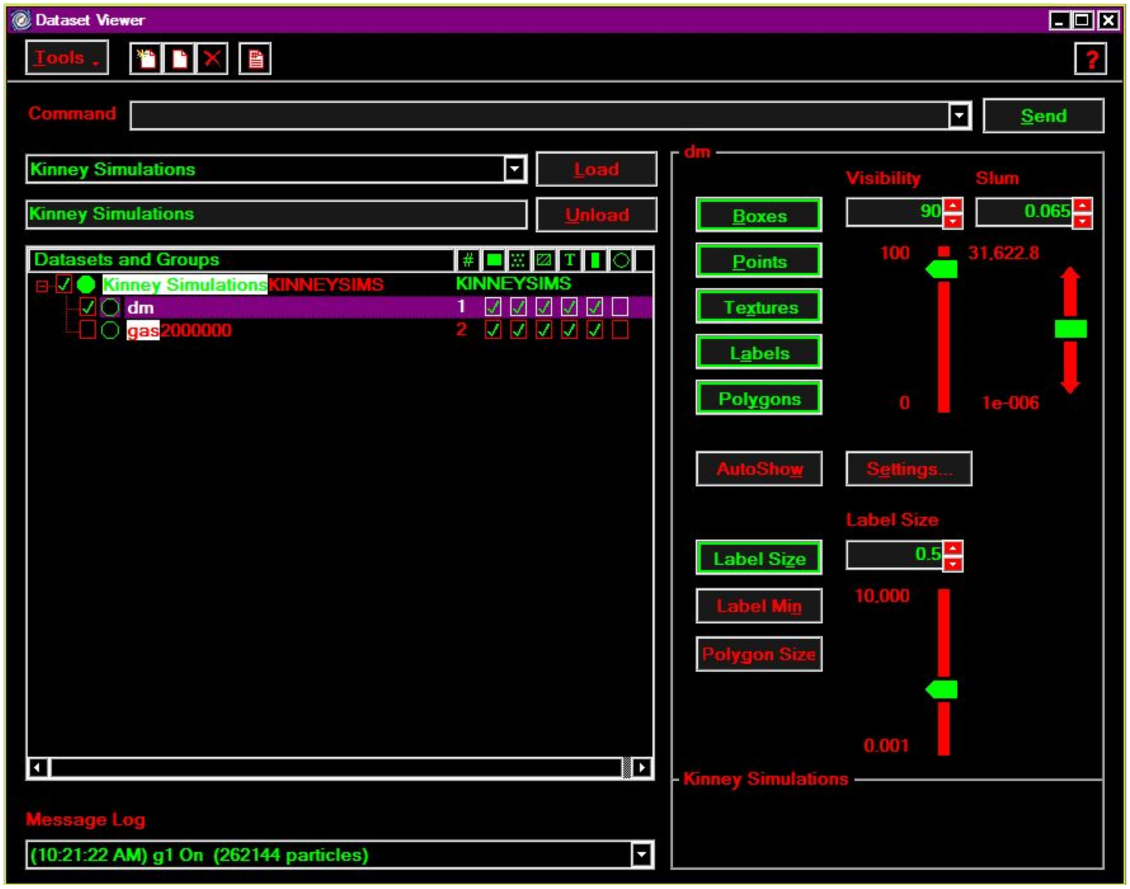

Figure 7: Once the data is loaded into the Dataset Viewer, the user can toggle on/off the groups (dark matter and gas), and adjust the luminosity using the slum slider to the right.

use 2 here). Recall that in Partiview we adjust the evaluation speed for the differing number of snapshots. Here the user is free to simply edit the button script to a speed of their choosing.

When writing button scripts there are two types of comments that can be made. The first type appears in yellow text and is preceded by a semicolon. This is known as a verbose comment and this text will appear on the bottom of the button panel window. An example use of a verbose comment in this instance would be ; dm must be on in order to run the time evolution typed on the first line, which would display on the button screen as a reminder (here red is used in lieu of yellow in order to be able to read the text). The other type of comment appears in gray and is preceded by an apostrophe. These are standard bookkeeping comments. An example here would be using 'dark matter after to run command to show that the dm group is running from that line. For reference, button scripts are saved as `*.sct` files in the 'Button' folder of DigitalSky.

Now that we have time evolution, the only thing left to do is create button scripts to replace our point of `interest` (i.e. origin) and `jump` commands. This is done by using `Scene Camera Locate` commands. A `Scene Camera Locate` command is followed by a series of numbers, usually x, y, and z position, and then followed by a time (e.g. +1, which would indicate to DigitalSky to wait one second before proceeding to the next command). The time delay is essential in most cases since each

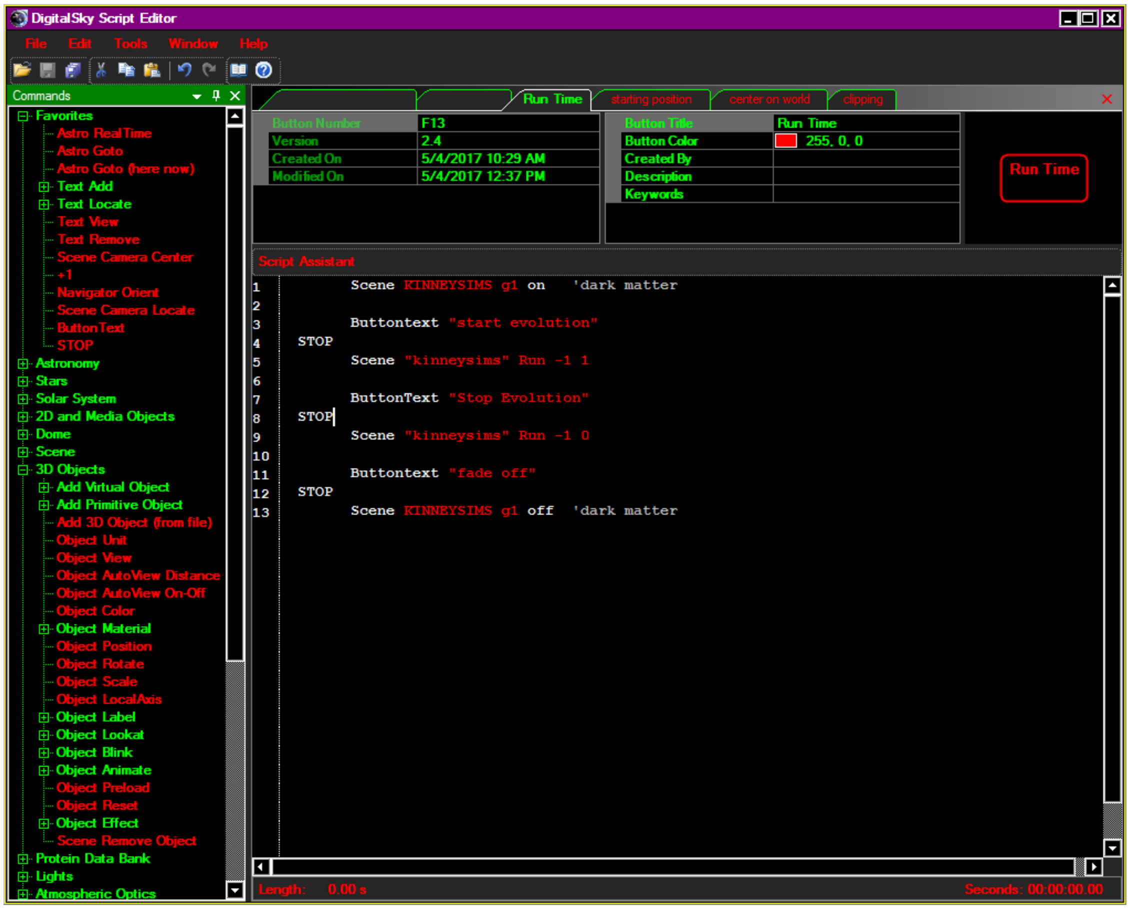

Figure 8: Using the DigitalSky Script Editor, we can program Button Scripts into the Button Panel shown in Fig. 6.

script line in a button operates in $1/20^{\text{th}}$ of a second. It needs to be preceded by a `Scene Camera Center` command which specifies the origin. So we can see that `Scene Camera Center` replaces the `interest` command in Partiview and `Scene Camera Locate` replaces the `jump` command.

The user can locate a point within the data set that interests them by flying through it in the same manner as Partiview using the Navigator window as shown in Fig. 9 (note that the 'nav' button in the button window may need to be clicked first). Once a point of interest is found, the camera position can be captured using the Navigator (see button circled in image). Once captured, paste the position in a button script. The button script for a `Scene Camera Locate` command looks like this:

```
Scene Camera Locate [2:2:2] 2.338521e+024 2.503214e+023 -8.666787e+023 -83.859114
-52.951966 -59.913856
```

The term in square brackets denotes `[accelerate:constant velocity:decelerate]`, thus the camera will arrive at the location in six seconds by accelerating for two seconds, traveling at a constant velocity for another two, and then decelerating for another two seconds. The next three numbers are the x, y, and z location (in meters) and the final three numbers are the x, y, and z camera rotation.

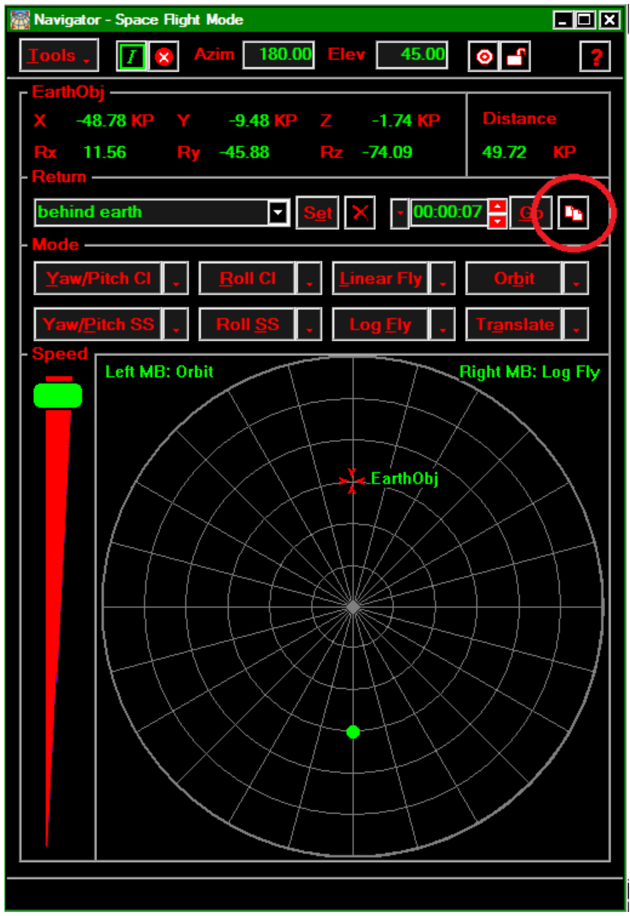

Figure 9: The Navigator window is used to move within the dataset, in a similar fashion to what is done in Partiview. Here the Scene Camera Center is Earth. This should be changed to World (center of Sun). In addition, the button circled is used to create a camera capture of the camera location for use with a Scene Camera Locate button script.

Thus a button script that would take the place of the `interest` and `jump` commands would look something like this:

```
    Scene Camera Center world
+4
    Scene Camera Locate [2:6:2] 1.543e+24 1.543e+24 1.543e+24 0 0 0
```

Users are free to experiment with settings that they find the most appealing to them. With that, you should be all set to visualize simulated cosmological data on your planetarium dome.

Acknowledgments

Special thanks go to:

Dr. William Kinney, Professor, Physics, University at Buffalo - Masters Project Advisor
Dr. Salvatore Rappoccio, Professor, Physics, University at Buffalo - Masters Project Advisor
Mark Percy, Williamsville Space Lab Planetarium Director
Martin Ratcliffe, Director of Professional Development, Sky-Skan Inc.
Center for Computational Research (CCR)
GADGET2 code - Volker Springel, Max-Planck Institute for Astronomy
Partiview software - Stuart Levy, National Center for Supercomputing Applications (NCSA)

This package is the result of my masters project required for an M.S. in physics at the University at Buffalo. It specifically came about due to the collaboration between my advisor, Dr. Will Kinney, a cosmologist at the University at Buffalo, and Mark Percy, the planetarium director at the Williamsville Space Lab Planetarium. I would also like to thank Martin Ratcliffe, a planetarium astronomer working for SkySkan Inc., who was very helpful with getting the Partiview data to interface with DigitalSky 2.

7 Partiview License

Copyright 2002 NCSA, University of Illinois
Urbana-Champaign, All rights reserved.

Developed by:
Stuart Levy, NCSA Virtual Director Group
University of Illinois Urbana-Champaign
`http://niri.ncsa.uiuc.edu/partiview/`

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- Neither the names of NCSA, University of Illinois Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED 'AS IS,' WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

This is an instance of the Illinois Open Source License, as certified by the Open Source Initiative in March 2002.

See: `http://www.otm.uiuc.edu/faculty/forms/opensource.asp`