UCDs for simulations

Laurie D Shaw and Nicholas A Walton

June 2004

1 Introduction

This is a brief document outlining some of the UCDs that would be required to describe a simulation dataset. It is based upon the first proposal for a simulation data model. Two classes of UCD are required, those that would be used to identify the origin and purpose of the dataset itself and therefore would only describe 0-1 quantities in the dataset (i.e. 'header' metadata), and those that would describe columns of data. This document is not complete and only aims to identify some of the more obvious metadata requirements and provoke further discussion

2 Data Structure

It is clear that a new top-level field (in the manner of 'phys', 'em', etc) is required which would contain UCDs that are unique to simulated datasets. We propose that this is named 'sim'. The three key questions that can be asked about a simulation are

1. What is being simulated?
2. How is it being simulated?
3. Initial conditions or parameters assumed?

These question are in some way in reverse order with regards to an actual simulation being executed in real time, the initial conditions are set and are evolved through some technique to hopefully produce the objects we intend to simulate. This gives us immediately five sub-fields of sim - obj(ect), alg(orithm), param(eter), theory and res(ources). Or, in UCD1+ notation (see http://cdsweb.u-strasbg.fr/UCD/),

- sim.obj
- sim.alg
- sim.param
- sim.theory
- sim.res

One of the most basic queries an astronomer may make is to look for simulations of a particular type of object (star, galaxy, black hole, etc). The 'object' field allows us to identify both the physical entities in the simulation and the structures that they are meant to form. To use the example of an nbody dark matter simulation, the building blocks are the particles themselves. These, through calculated gravitational interactions, eventually group into dark matter halos and subhalos - the objects that ultimately we want to simulate. The associated UCDs would take the following form -

- sim.obj.particle
- sim.obj.halo
- sim.obj.subhalo
- sim.obj.gal(axy)
- sim.obj.star

and would be used to as a tag to indicate the object to which a physical parameter is describing. For example, the mass of a halo would be: phys.mass;sim.obj.halo.

Of course, not all simulations work by pushing particles around - some may define grids or finite elements in order to evolve fluid equations or analyse the propogation of waves through some medium. However, in most astrophysical simulations the aim is to create or evolve some 'observable'. It is these, or their discrete constituents that obj attempts to describe. There are of course different forms that the particles in a simulation could take - they may be infitissemal, smoothed, non-ineracting, of different masses or have other differing properties (dark matter, baryons). But these properties define the method that is to be used to evolve the simulation and would be described in the 'algorithm' sub-field.

The algorithm field aims to describe the approach that has been taken in order to evolve the system in question and how it has been invoked. Examples are nbody (in this case assumed to mean infinitessimal particles), SPH (particles with size and volume) hydrodynamic (grid based?) simulations and finite element analyses(lattice based??). Each of the techniques employ a specific algorithm, normally designed to perform the calculations in an as accurate but efficient manner possible. Returning to the nbody dark matter simulation, examples would be tree-particle-mesh or particle particle - particle mesh (adaptive grid) routines or just a direct force calculation. Hence new UCDs would be

- sim.alg.sph
- sim.alg.hydro
- sim.alg.nbody.tpm
- sim.alg.nbody.pppm
- etc,

The UCDs built using the algorithm sub-field would normally only be required once (in the header) of a particular simulation dataset, they wouldn't be able to provide any information about the actual physics assumed and approximations used in the simulation - e.g. whether gravitational interactions were calculated using the equations of general relativity or just the newtonian approximation. It is likely that this kind of detailed, qualitative, information can not be conveyed using UCDs at all and that at this stage a URL to an associated paper may be required. Nevertheless, we leave open the possibility of having a 'theory' subfield, in case it may be required.

The next sub-field that we outlined above is 'parameter'. This sub-field is intended to contain quantative technical information regarding the simulation. Examples are the number particles, maximum and minimum grid spacings, timestep, the softening length (all of which are analogous to resolution), the initial and final simulation 'times' (or redshifts) and the simulation box size . Also important are the physical parameters used in a simulation - the cosmological parameters, σ_8 (densities in 8mpc spheres) and so on. But these parameters are also critical in qualifying some observations, e.g. measuring w in the equation of state or indeed measuring any of the cosmological parameters. The hubble constant is also assumed to be a particular value for the purpose of analysing observational data or deriving another physical parameter. Therefore we also propose a 'param' subfield of the phys field, so that the assumed values of any physical parameters can be displayed.

So now we also have -

- sim.param.softlength
- sim.param.timestep
- sim.param.boxsize

and -

- phys.param.hubble
- phys.param.omega_lambda
- phys.param.omega_m
- phys.param.sigma8

and so on.

Finally, it would also be usefull to have information regarding the resources, or hardware, that was used. This would be contained in the resource sub-field and would contain such information as the number of processors, the amount processor time taken to complete the simulation and the total amount of diskspace the simulation requires. UCD examples are:

- sim.resource.processors;stat.num
- sim.resource.memory

3 Application to quantities in a simulation dataset

Many of the currently defined UCDs were determined by going through the vizier catalogue and attempting to allocate one for each quantity contained in the catalogue. We now repeat this exercise by attempting to define UCDs for each quantity in a simulation dataset. To continue with the example given above we use the typical output of an nbody dark matter simulation, in which a large number of particles have been distributed in a simulation box at high redshift and then allowed to evolve through gravitational interactions alone untill a time equivalent to zero redshift has been reached.

Self-bound structures (halos) and sub-structures (sub-halos) are then identified in the final simulation box, and some of their intrinsic properties are calculated. It was found during this exercise that several further 'atoms' (UCD1+ jargon) were required to be added to some of the field already defined in the UCD1+ proposal.

It is assumed that information in the 'header' of the file would indicate that the data is part of an nbody dark matter only simulation, using a TPM alogirthm, with x number of particles, in a y^3 volume simuluation box, softening length z, maximum timestep t, etc, etc. The raw data of the simulation is that particles posistion, velocity and local density. We have constructed the following UCDs for these quantities:

- particle positions - pos;sim.obj.particle;sim.coordsys.cart.x(or y or z)
- particle velocity components - phys.vel;sim.obj.particle;sim.coordsys.cart
- particle local density - phys.density;sim.obj.particle

The following are properties derived from the halos and subhalos that the particles form:

- virial radius - phys.size.radius.viral
- peak density position - pos;phys.density;stat.max;sim.coordsys.cart.x(or y or z)
- halo id number - meta.id
- centre of mass position - pos;phys.mass.com;sim.coordsys.cart.x(or y or z)
- number of subhalos - meta.number;sim.obj.halo.subhalo
- mass - phys.mass
- number of particles - meta.number;sim.obj.particle
- rms velocity - phys.vel;stat.rms
- fraction of mass in substructure (subhalos) - phys.mass.fraction;sim.obj.halo.subhalo
- total potential energy - phys.enery.pot;sim.obj.halo
- maximum circular velocity - phys.vel.circ;stat.max

We found during this exercise that it was necessary to define a new field - sim.coordsys - to describe the coordinate components for many of the quantities in the dataset. These would include -

- sim.coordsys.cart (cartesian)
- sim.coordsys.sph (spherical)

In the future coordsys could be replace by a term that adequately encompasses the metric (minkowski, scharzschild, etc) employed as well.

4 summary

To summarise - listed below are all the new UCD words that we propose be added to the standard set to enable the adequate description of simulation datasets. Those particular to some simulations have only been listed in general terms. There are also clearly many more to be added.

- sim
- sim.obj
- sim.obj.gal
- sim.obj.gal.spiral
- sim.obj.gal.elliptical
- sim.obj.star
- sim.obj.halo
- sim.obj.halo.subhalo

- sim.alg
- sim.alg.nbody.tpm
- sim.alg.nobdy.pppm
- sim.alg.sph
- sim.alg.hydro

- sim.param
- sim.param.softlength
- sim.param.timestep
- sim.param.boxsize

- sim.theory

- sim.resource
- sim.resource.processors
- sim.resource.memory

- sim.coordsys
- sim.coordsys.cart.x

- sim.coordsys.cart.y
- sim.coordsys.cart.z
- sim.coordsys.sph.th
- sim.coordsys.sph.ph
- sim.coordsys.sph.r

Additions to the ’phys’ field

- phys.param
- phys.param.hubble
- phys.param.omega_lambda
- phys.param.omega_m
- phys.param.sigma8
- phys.size.radius.viral
- phys.enery.pot

An addition to the ’stat’ field

- stat.rms