Simulating dark matter in an expanding universe using a parallel multigrid method

Otto Hannuksela *Janne Lampilahti*

May 18, 2015

INTRODUCTION

Dark matter appears to be the dominant form of matter in the universe. While no direct observations of dark matter exist, it has been observed indirectly through its gravitational interaction with visible matter and radiation. The effect of dark matter on structure formation in the universe is the subject of an increasing scientific interest and a useful tool in the search for possible candidates of dark matter. Current progress in the investigation of structure formation is mainly driven by advances in computational methods and capabilities.

In this work we simulate dark matter, neglecting baryonic matter, by using a particle based method and solving the Poisson's equation of gravity in a comoving frame of reference. The software used is *Uintah*, a parallel partial differential equation solver on an adaptively refined multigrid.

THEORY

The gravitational potential ϕ is given by the Poisson's equation of gravity

$$\nabla^2\phi = 4\pi G\rho \tag{1}$$

where ρ is the mass density. In our case we have a set of massive particles. The corresponding force field can be solved from the gradient

$$\mathbf{F} = -\nabla\phi. \tag{2}$$

A new position and velocity for the particles after a time step dt can be solved from Newton's equation of motion.

$$\mathbf{v}(t+dt) = \int_t^{t+dt} \frac{\mathbf{F}(t')}{m} dt' + \mathbf{v}(t) \tag{3}$$

$$\mathbf{x}(t+dt) = \int_t^{t+dt} \mathbf{v}(t') dt' + \mathbf{x}(t) \tag{4}$$

METHODS

The simulation is set up with respect to an adaptively refined three dimensional grid that supports particles (Fig. 1).

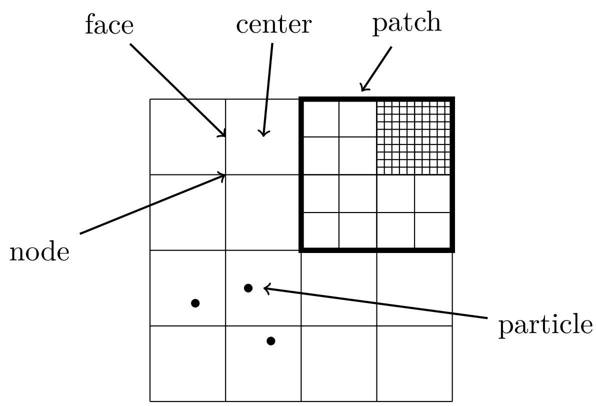

Figure 1. A two dimensional view of the grid with its various components and properties shown.

The overall algortihm is expressed in Algortihm 1.

Algortihm 1. Main program.

set boundary conditions
set initial $\mathbf{x}_p$, $\mathbf{v}_p$ and m_p for particles p
calculate initial ρ
set initial guess for ϕ
loop
    solve ϕ using SOR algorithm
    calculate $\mathbf{F} = -\nabla\phi$
    for every particle p **do**
        $\mathbf{v}_p(t + \Delta t) = (\mathbf{F}_p/m)\Delta t + \mathbf{v}_p(t)$
        $\mathbf{x}_p(t + \Delta t) = \mathbf{v}_p(t)\Delta t + \mathbf{x}_p(t)$
    end for
    calculate ρ
    $t \leftarrow t + \Delta t$
end loop

The Poisson's equation is solved with the successive over relaxation (SOR) algortihm (see Alg. 2). The potential ϕ is solved at each node. In the calculation of the potential gradient we obtain the potential values near the particles by interpolation and then calculate the gradient by numerical differentiation. For exmaple at the pth particle the gradient is calculated in the x direction as

$$(\nabla\phi)(x_p) = \frac{\phi(x_p + dx) - \phi(x_p - dx)}{2dx}, \tag{5}$$

where x_p is the particle's x coordinate and dx is a small distance.

The particle velocity and position are evolved over a small time step dt, assuming that the force remains constant. After this the particle masses are again interpolated back to the nodes to obtain a new mass density ρ.

Algortihm 2. Calculating potential ϕ using the SOR algorithm.

function SOR(ϕ, tolerance, max_iterations)
  for $n = 0, 1, \ldots$ max_iterations **do**
    error $\sigma \leftarrow 0$
    for every node $\phi_{i,j,k}$
      $\phi_{i,j,k}^{(n+1)} \leftarrow (1-\omega)\phi_{i,j,k}^{(n)} + \frac{\omega}{6}(\phi_{i+1,j,k}^{(n)} + \phi_{i-1,j,k}^{(n)} + \phi_{i,j+1,k}^{(n)} + \phi_{i,j-1,k}^{(n)} + \phi_{i,j,k+1}^{(n)} + \phi_{i,j,k-1}^{(n)} + h^3\rho_{i,j,k})$
      update σ
    end for
    if $\sigma \leq$ tolerance, **break**
  end for
  return ϕ
end function

IMPLEMENTATION

Uintah

Program

RESULTS

CONCLUSIONS