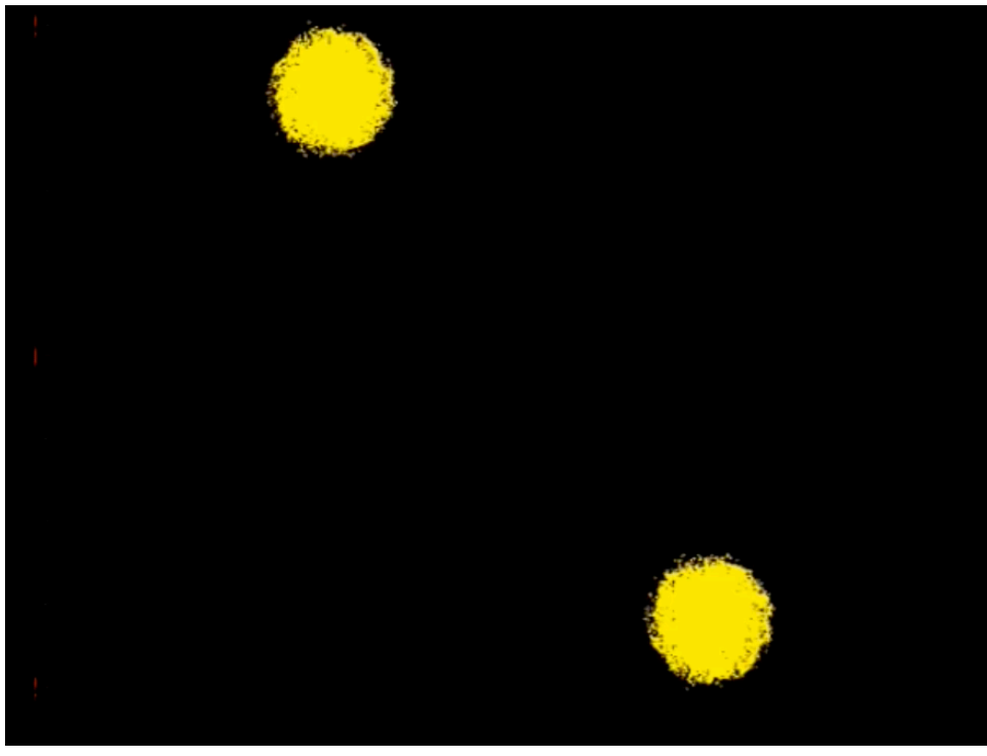
NagBody lectures: Tree methods



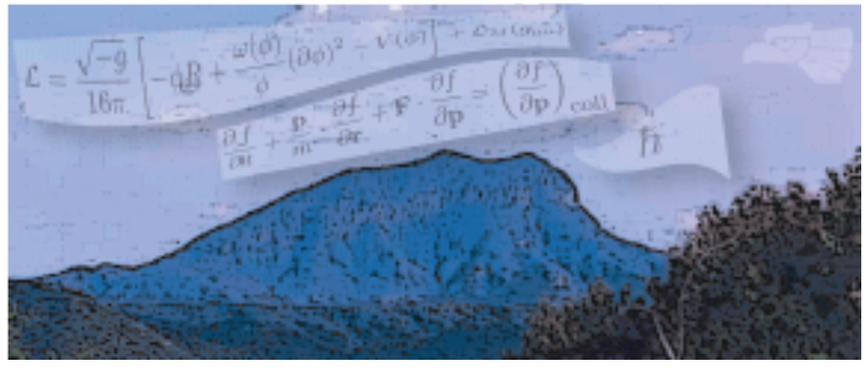
Mario Alberto Rodríguez-Meza

Instituto Nacional de Investigaciones Nucleares

Correo Electrónico: marioalberto.rodriguez@inin.gob.mx

http://bitbucket.org/rodriguezmeza

Seminario de investigación,
Departamento de Física,
Universidad de Guanajuato
3 de febrero al XX de junio de 2022
Sesiones virtuales (Zoom, Meet, etcétera)



References and material

- Cosmología numérica y estadística: NagBody kit (http://bitbucket.org/rodriguezmeza). Mario A. Rodríguez-Meza. And: https://github.com/rodriguezmeza/NagBody_pkg.git

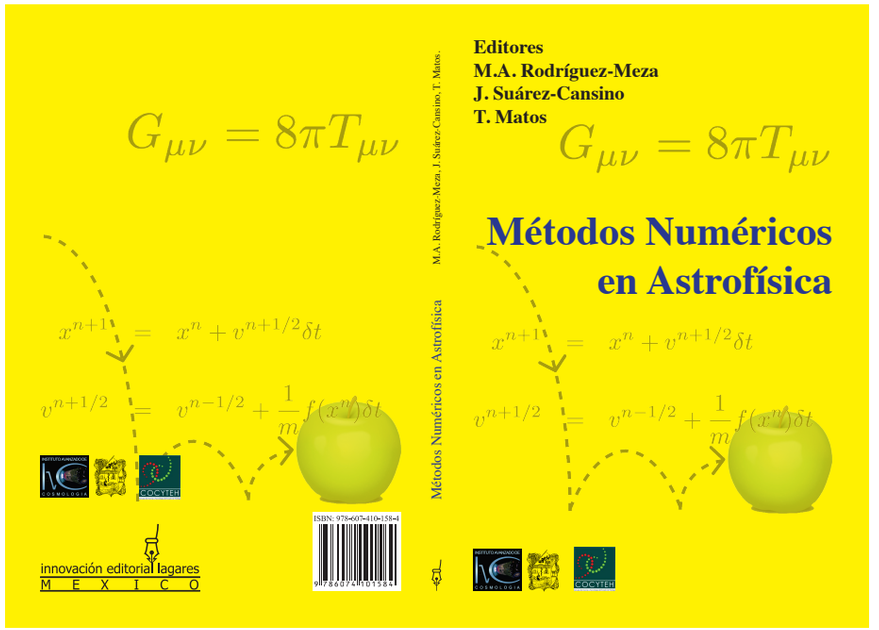
- Métodos numéricos en astrofísica, capítulo 1, Método de N-cuerpos en astrofísica. (https://www.researchgate.net/publication/316582859_Metodo_de_N-Cuerpos_en_Astrofisica)

- La estructura a gran escala del universo. Capítulo 22 en Travesuras cosmológicas de Einstein et al. https://www.researchgate.net/publication/316582400_La_estructura_a_gran_escala_del_universo_simulaciones_numericas

- https://www.researchgate.net/profile/Mario_Rodriguez-Meza

- https://www.researchgate.net/publication/314281416_Los_agujeros_negros_y_las_ondas_del_Dr_Einstein

- M.A. Rodriguez-Meza, Adv. Astron. 2012, 509682 (2012). arXiv: 1112.5201. (https://www.researchgate.net/publication/51967093_A_Scalar_Field_Dark_Matter_Model_and_Its_Role_in_the_Large-Scale_Structure_Formation_in_the_Universe)



Content:
Tree methods

- Concept of tree structure
- Three dimensional trees for N-body simulations
- Tree methods

Tree methods:
Tree concept

Examples:

- Family tree
- The draw sheet of a tournament
- Roots of a plant

Binary tree:
branching factor=2

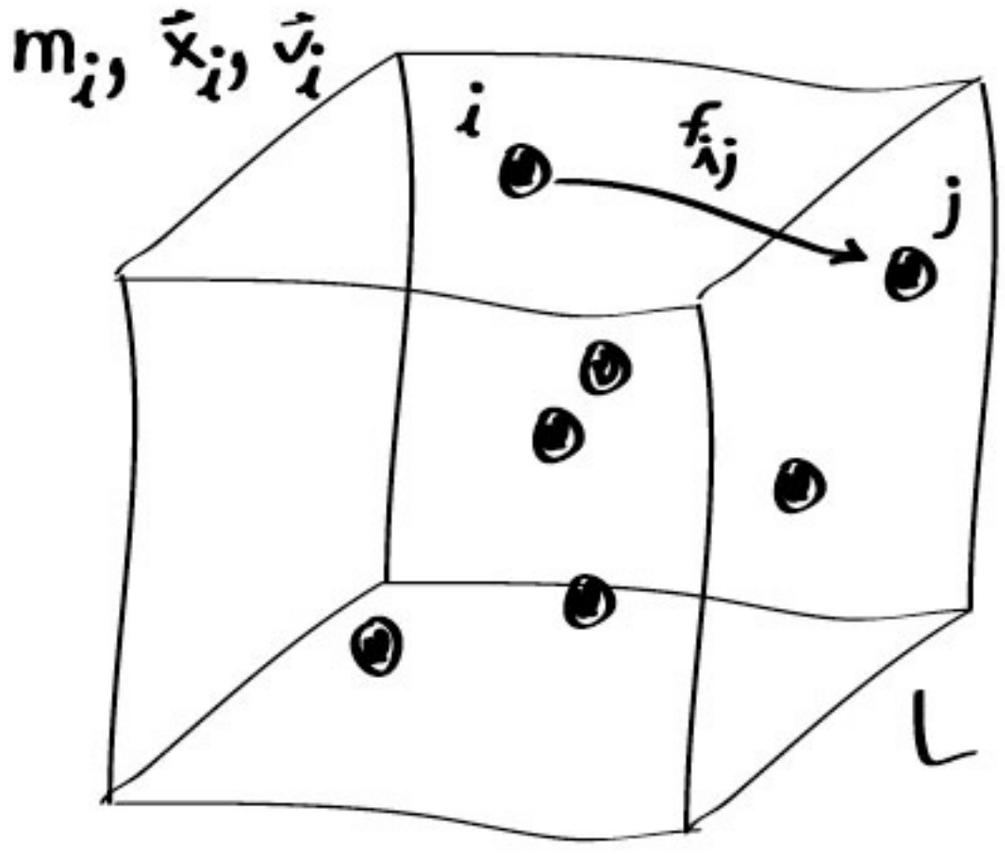
In computing:

- A tree consists of elements called *nodes* organized in a hierarchical arrangement.
- The top of the hierarchy is called the *root*. Nodes directly below the root are its *children*. They are *siblings*. And they have children also. Each node has one parent.
- The number of children a node may parent depends on the type of tree. This number is the *branching factor*.

Tree methods: Simulations

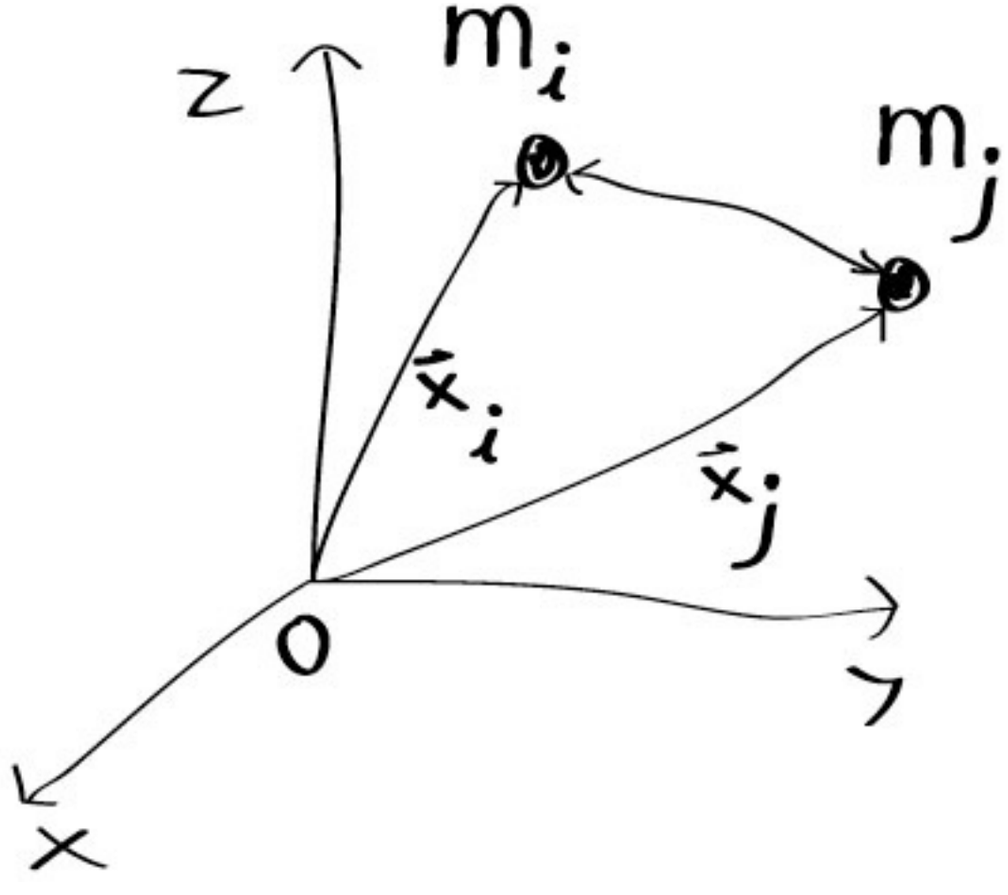
- The N-body problem.
- Define a simulation box.
- Put in there N particles, giving their positions, velocities and masses.
- Compute forces among them.

$$\mathbf{f}_{ij} = -\frac{G_N}{1+\alpha}\frac{m_j(\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^3} \times F_{SF}(|\mathbf{x}_i - \mathbf{x}_j|, \alpha, \lambda)$$



Tree methods:
Simulations: equations of motions

$$\ddot{\mathbf{x}}_i + 2\frac{\dot{a}}{a}\mathbf{x}_i = -\frac{1}{a^3}\frac{G_N}{1+\alpha}\sum_{\substack{j=1\\ j\neq i}}^{N}\frac{m_j(\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^3}F_{SF}(|\mathbf{x}_i - \mathbf{x}_j|, \alpha, \lambda)$$



Tree methods: Simulations

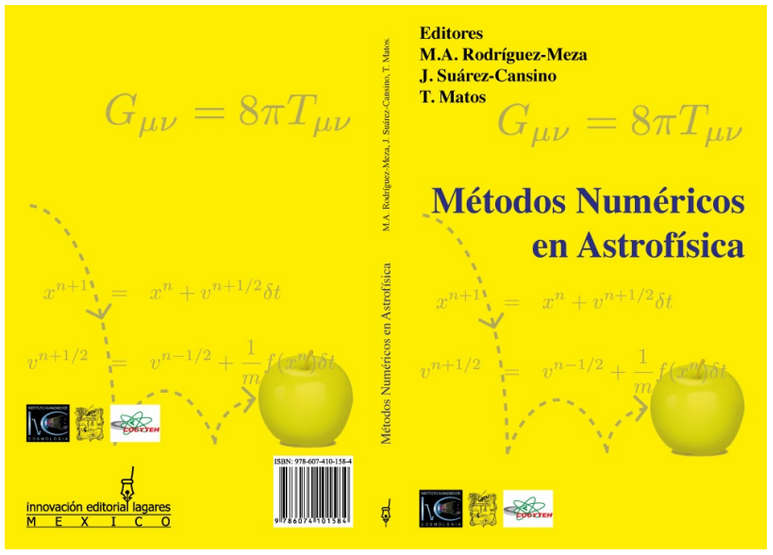
- Equations of motion:

$$\ddot{\mathbf{x}}_i + 2\frac{\dot{a}}{a}\mathbf{x}_i = -\frac{1}{a^3}\frac{G_N}{1+\alpha}\sum_{\substack{j=1\\ j\neq i}}^{N}\frac{m_j(\mathbf{x}_i-\mathbf{x}_j)}{|\mathbf{x}_i-\mathbf{x}_j|^3}F_{SF}(|\mathbf{x}_i-\mathbf{x}_j|,\alpha,\lambda)$$

- where:

$$F_{SF}(r,\alpha,\lambda) = 1+\alpha\left(1+\frac{r}{\lambda}\right)e^{-r/\lambda}$$

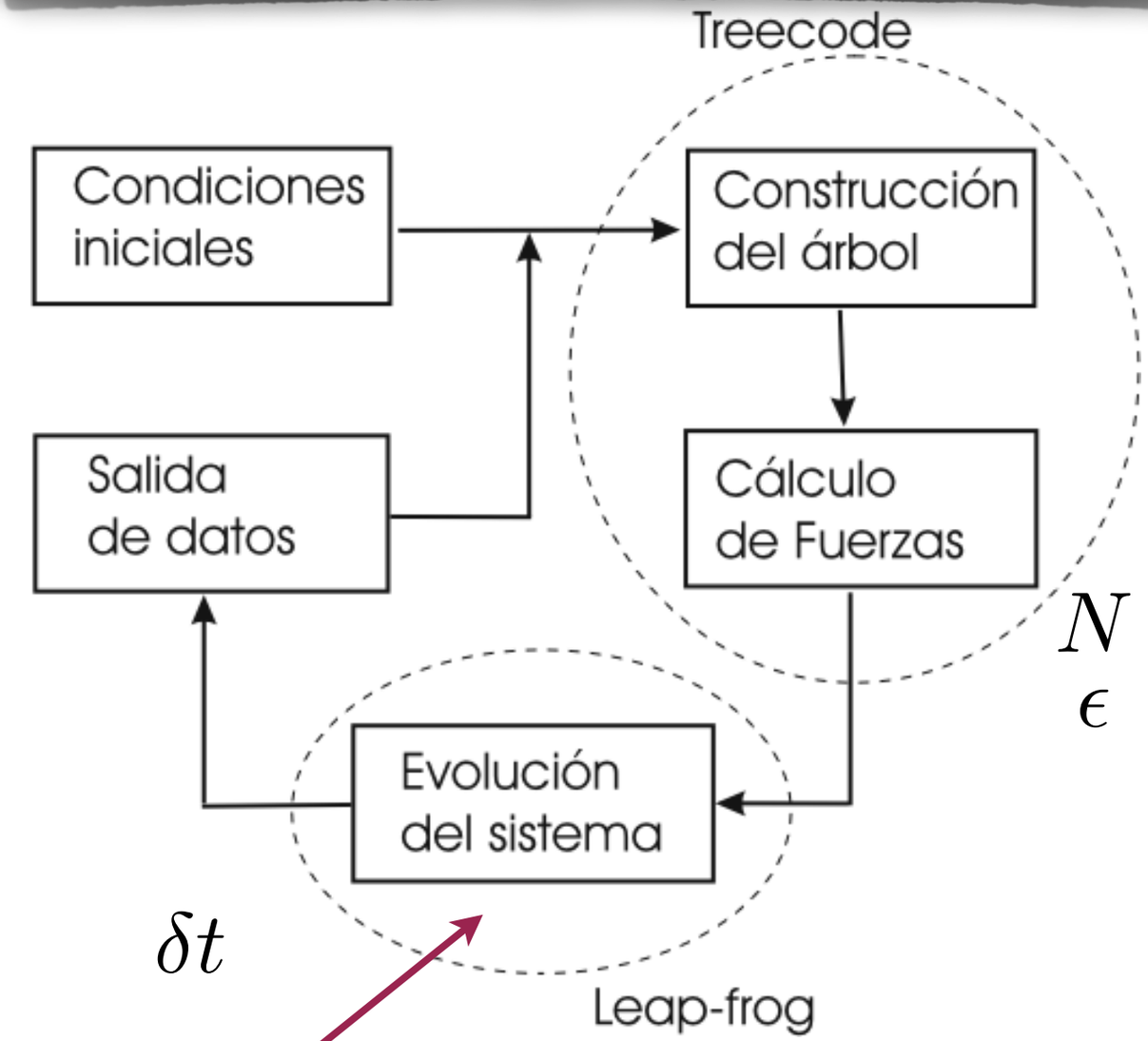
- Limits:



$$F_{SF}(r \ll \lambda,\alpha,\lambda) \approx 1+\alpha\left(1+\frac{r}{\lambda}\right)$$

$$F_{SF}(r \gg \lambda,\alpha,\lambda) \approx 1$$

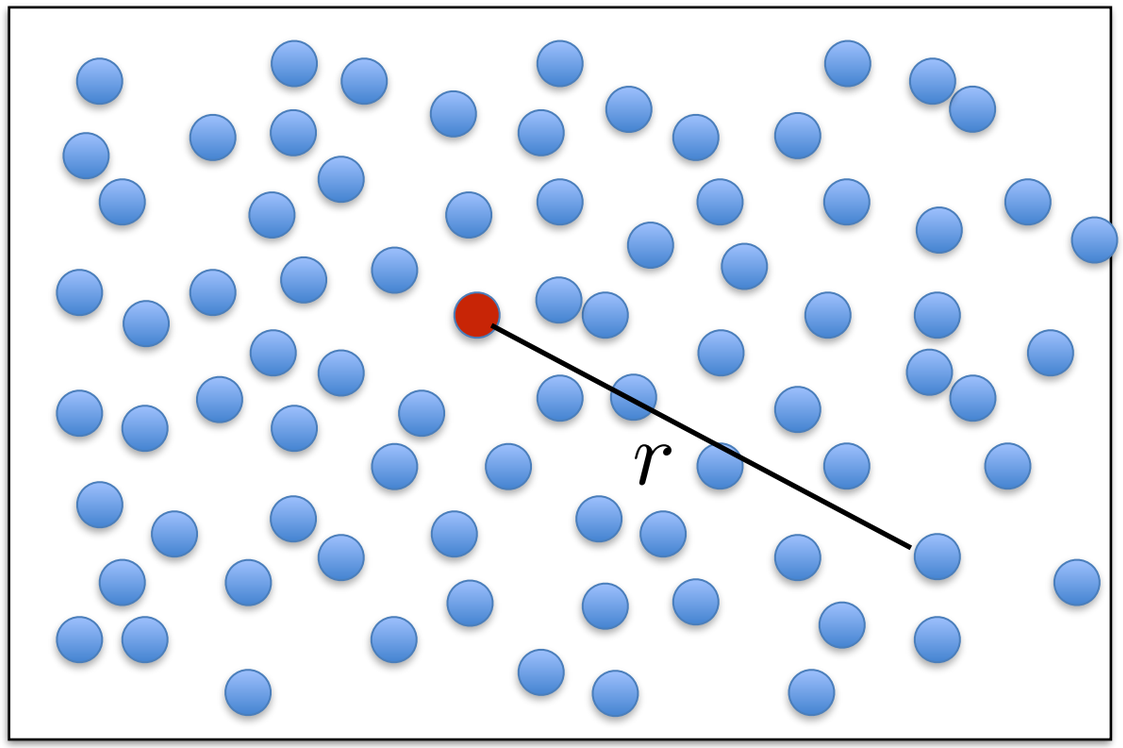
Tree methods:
N-body general strategy



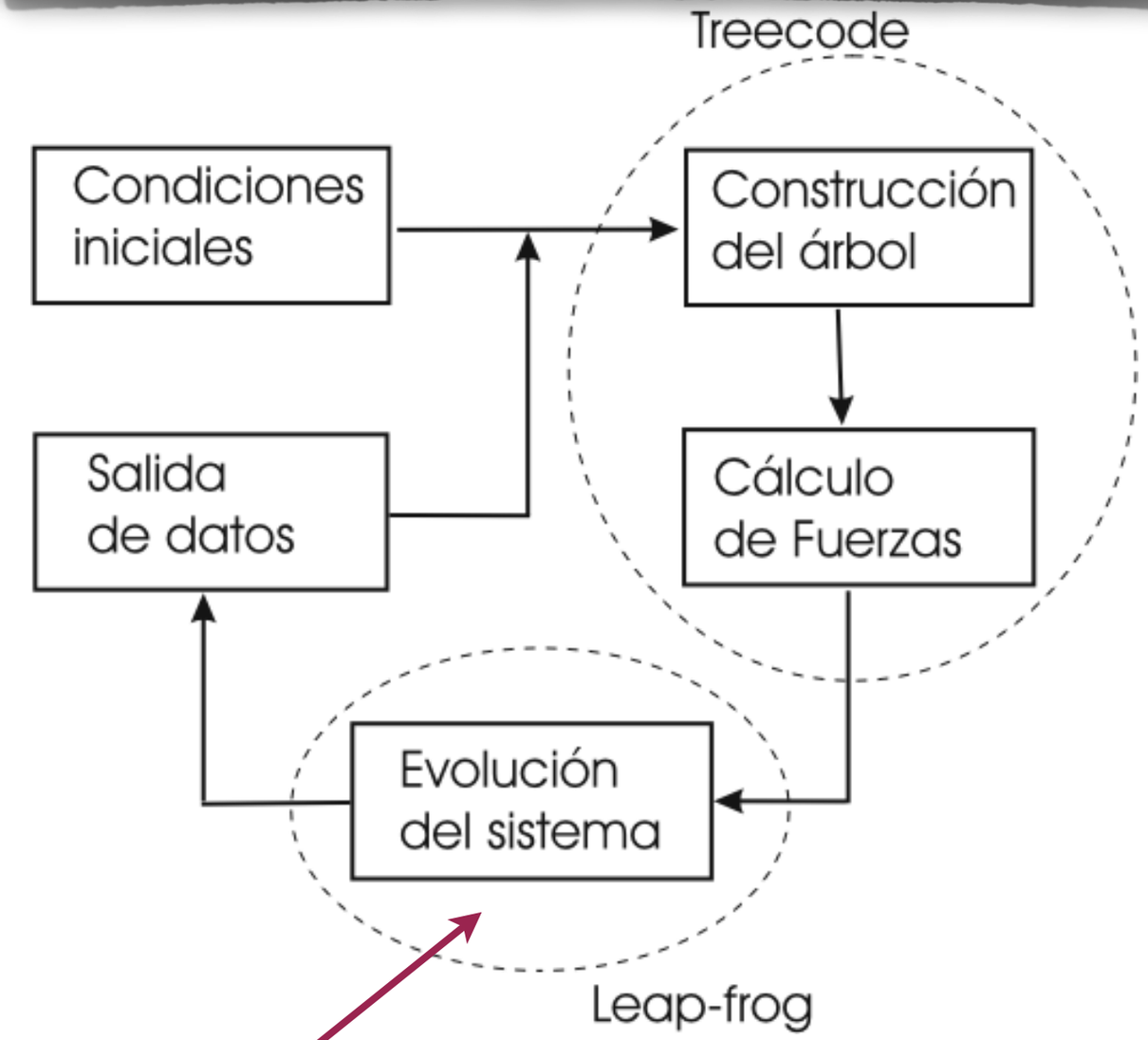
$$\Phi_N = -\frac{Gm}{\sqrt{r^2+\epsilon^2}}$$

- Particle-Particle (PP)
- Particle-Mesh (PM)
- Particle-Particle/Particle-Mesh (P3M)
- Particle-Adaptive Mesh (AMR)
- TreeCode (TC)
- TreeCode/Particle-Mesh (TreePM)

$$\nabla^2\Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r})$$



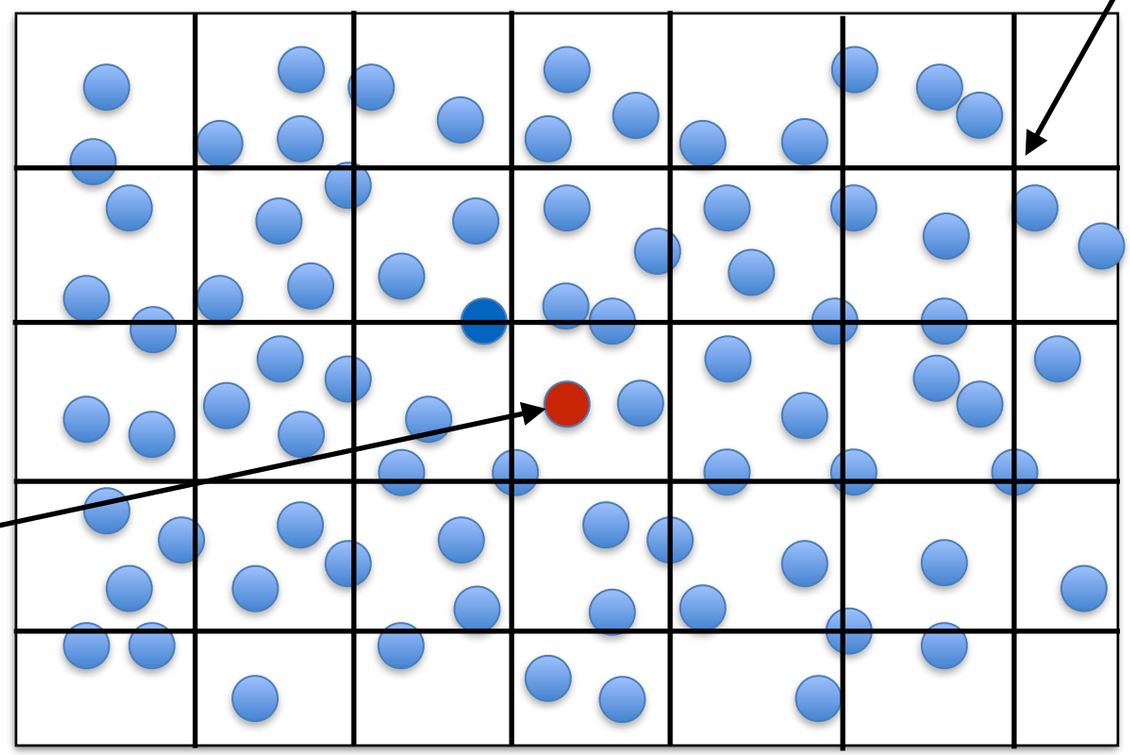
Tree methods:
N-body general strategy



- Particle-Particle (PP)
- Particle-Mesh (PM)
- Particle-Particle/Particle-Mesh (P3M)
- Particle-Adaptive Mesh (AMR)
- TreeCode (TC)
- TreeCode/Particle-Mesh (TreePM)

$$\nabla^2 \Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r})$$

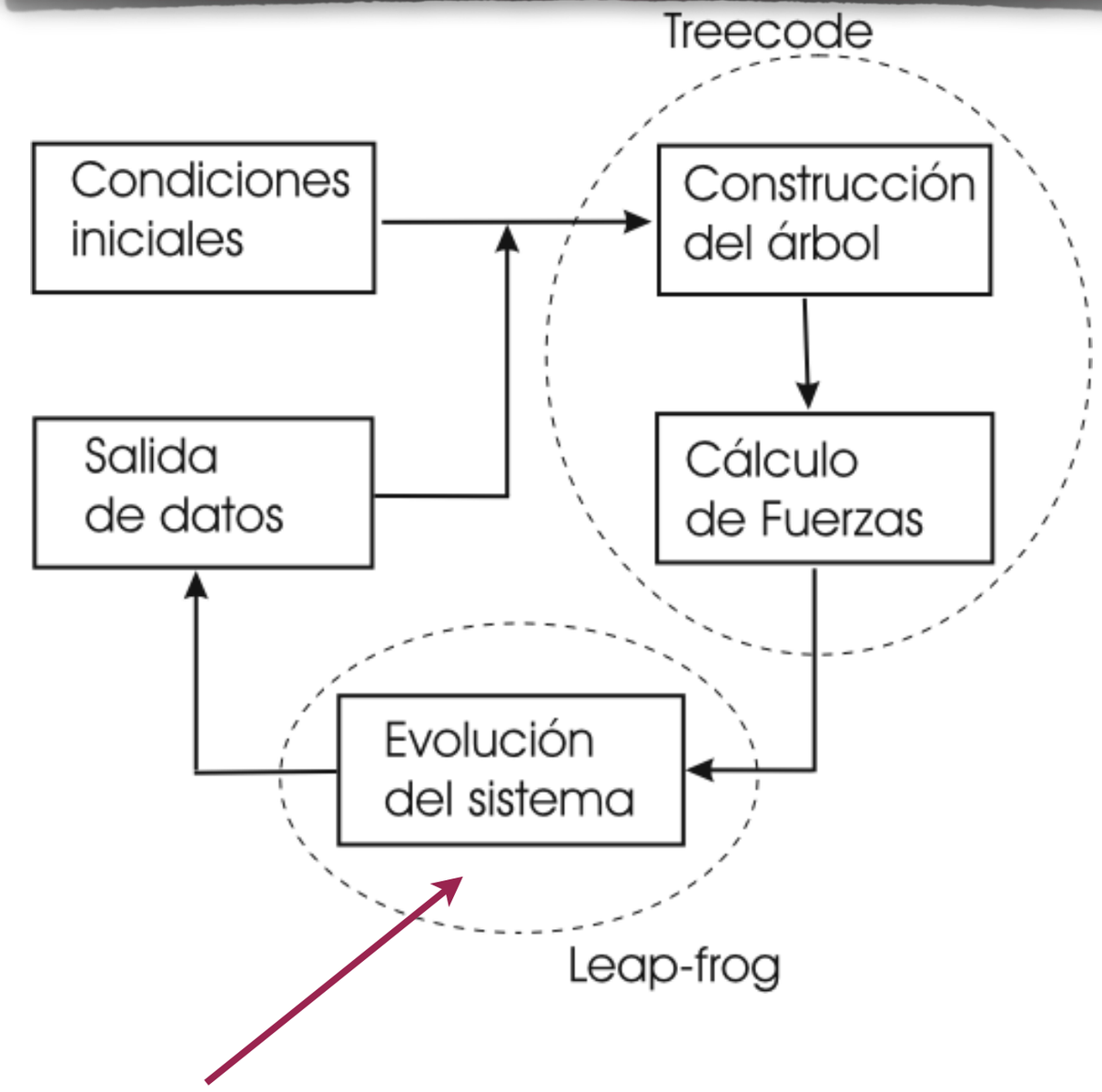
$\Phi_{i,j}$ $\rho_{i,j}$



$$\frac{d\mathbf{V}}{dt} = \mathbf{F}_V$$

$$\rho_i = \sum_j \rho_j W(r_{ij}, h)$$

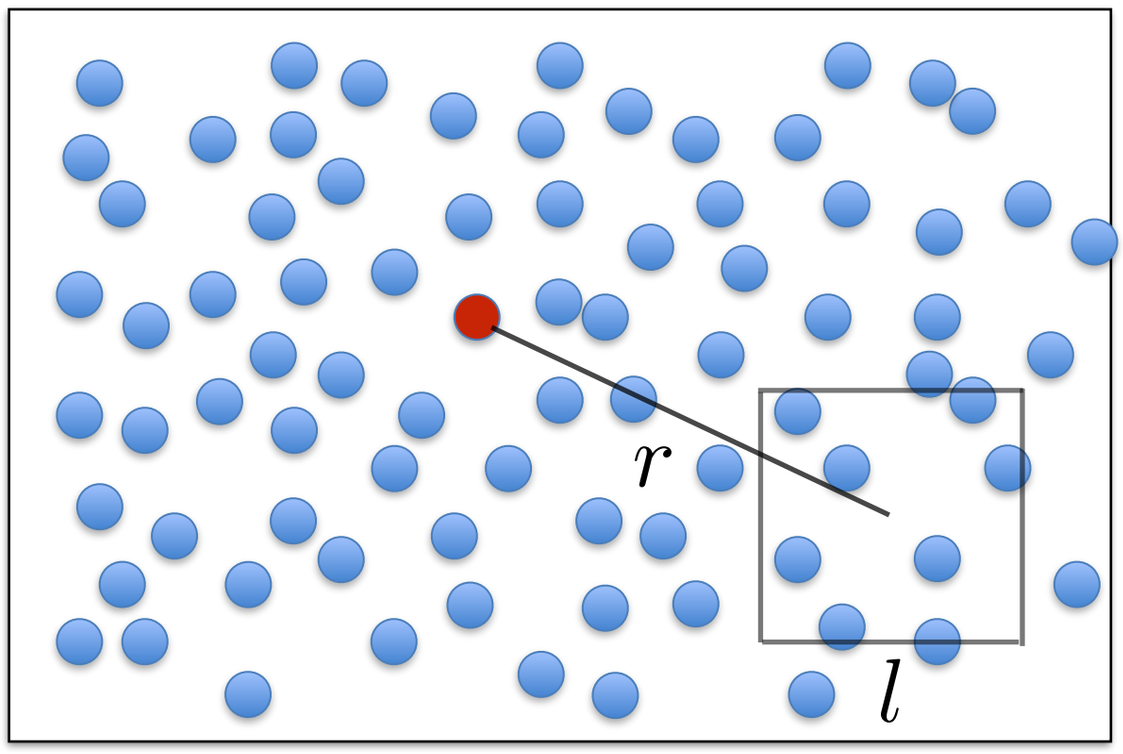
Tree methods:
N-body general strategy



$$\frac{d\mathbf{V}}{dt} = \mathbf{F}_V$$

- Particle-Particle (PP)
- Particle-Mesh (PM)
- Particle-Particle/Particle-Mesh (P3M)
- Particle-Adaptive Mesh (AMR)
- TreeCode (TC)
- TreeCode/Particle-Mesh (TreePM)

$$\nabla^2 \Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r})$$



$$r > \frac{l}{\theta}$$

Tree methods
Octagonal trees

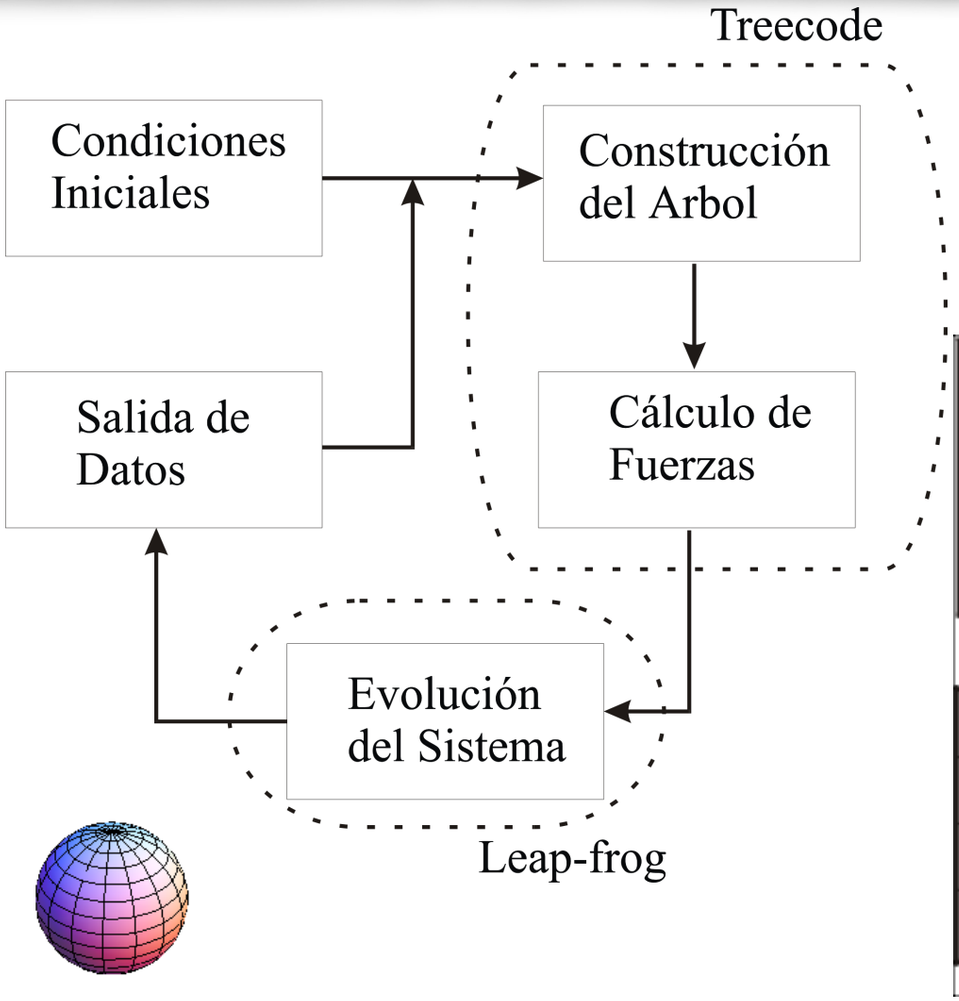
- Simulations are in 3D.
- Divide the simulations box in eight smaller boxes of equal size.

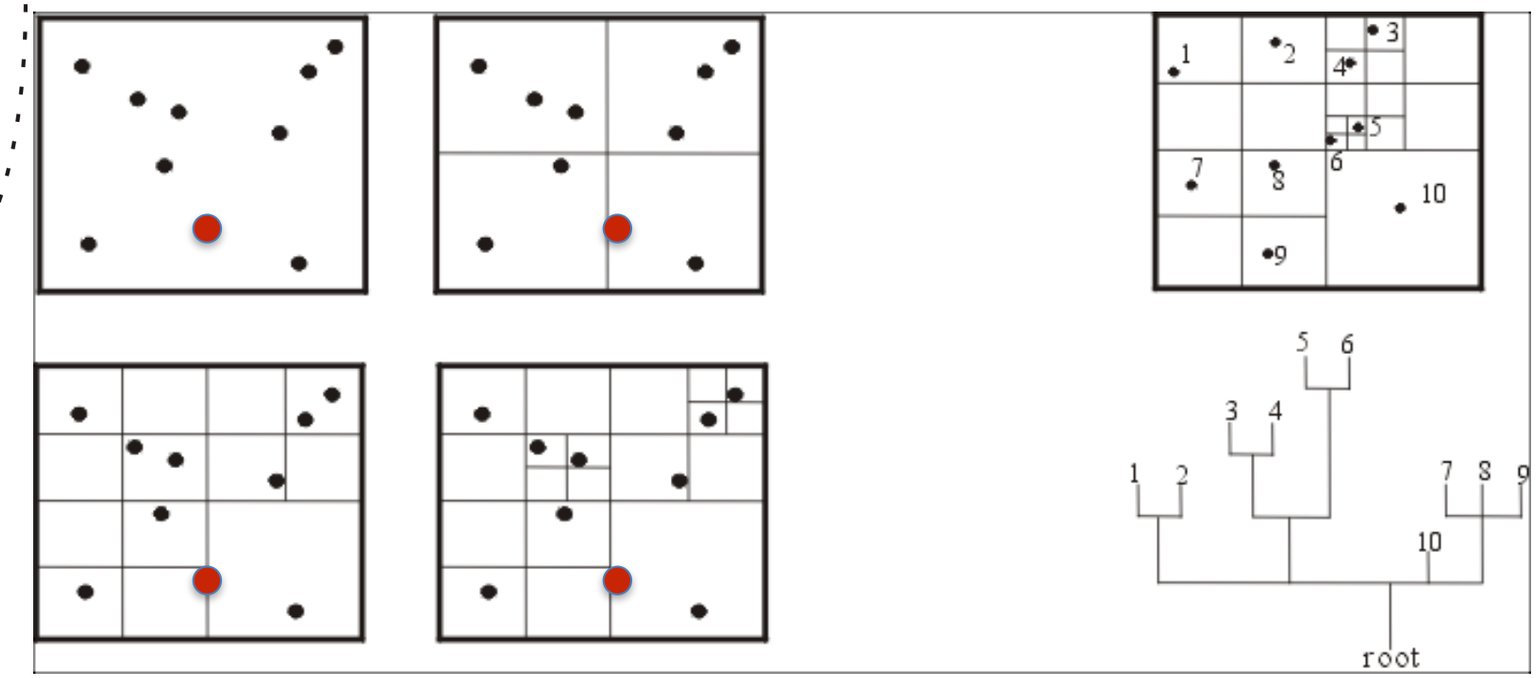
Tree methods: Particle methods

Tree method

(Multipole expansion of the gravitational potential)

$$\psi(\mathbf{r}) = \frac{m}{r} + \frac{1}{2}\sum_{i,j} Q_{ij}\frac{x_i x_j}{r^5} + \ldots$$

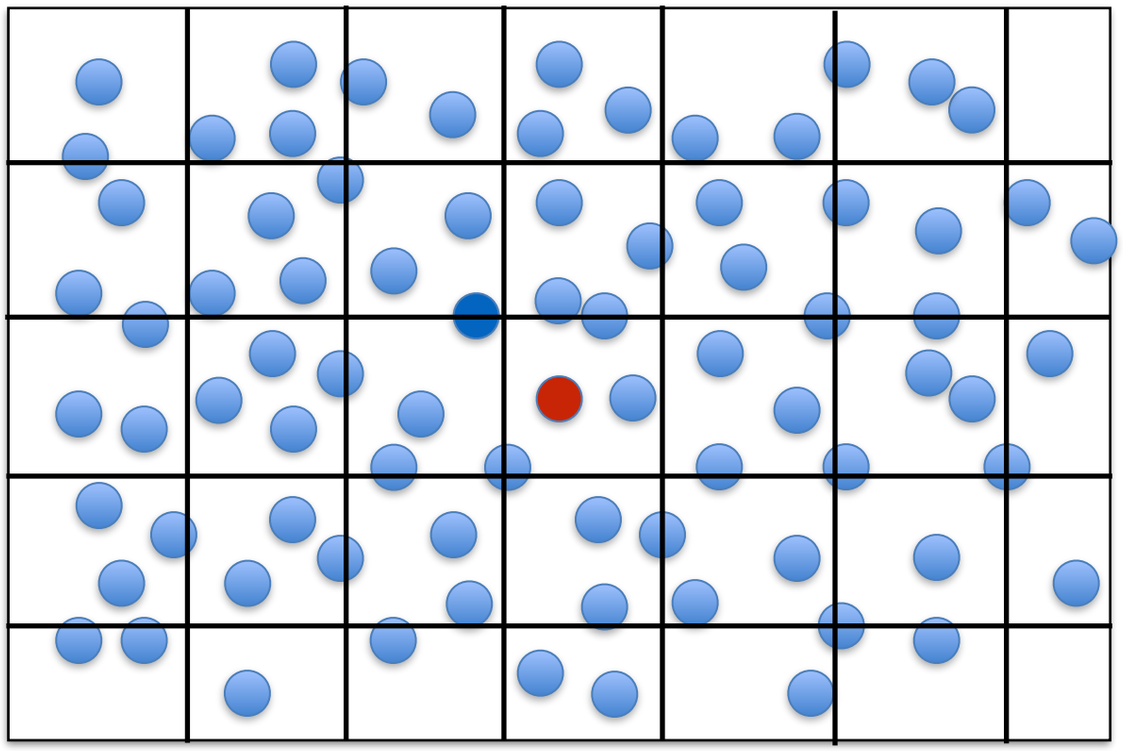




$$r > \frac{l}{\theta}$$

Tree methods:
N-body general strategy

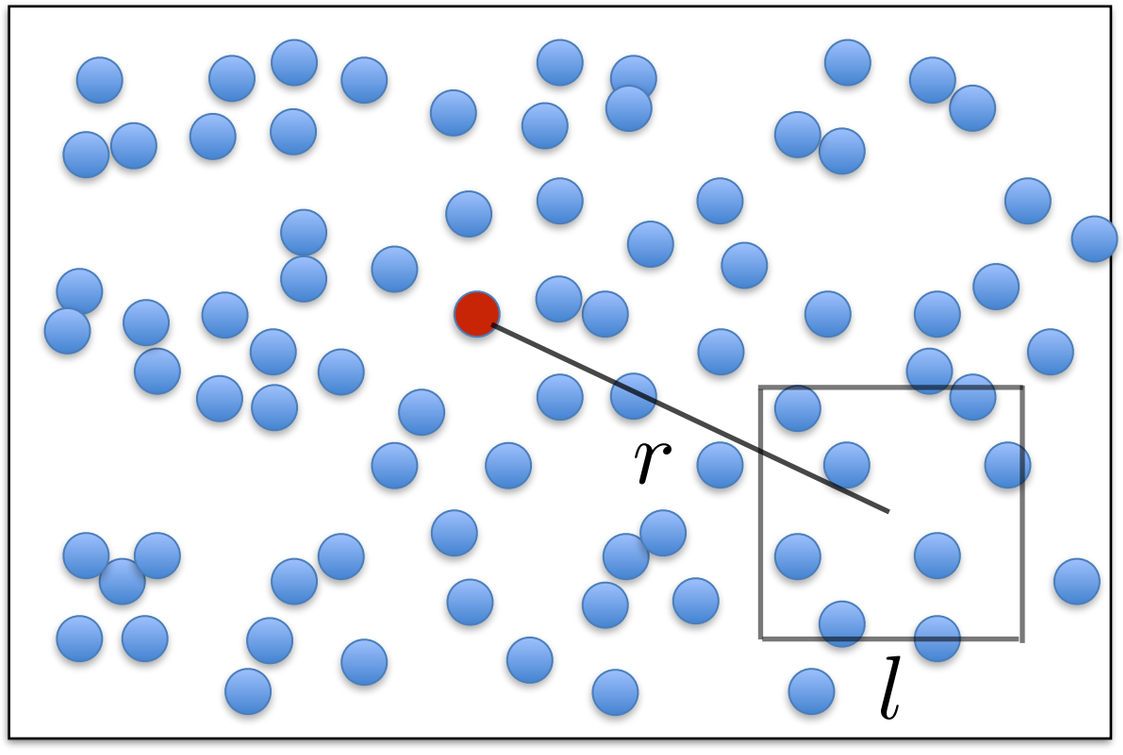
- Particle-Mesh (PM)



¿Which is the best method?

$$\nabla^2 \Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r})$$

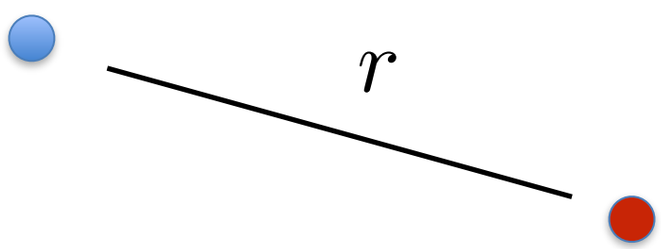
- TreeCode (TC)



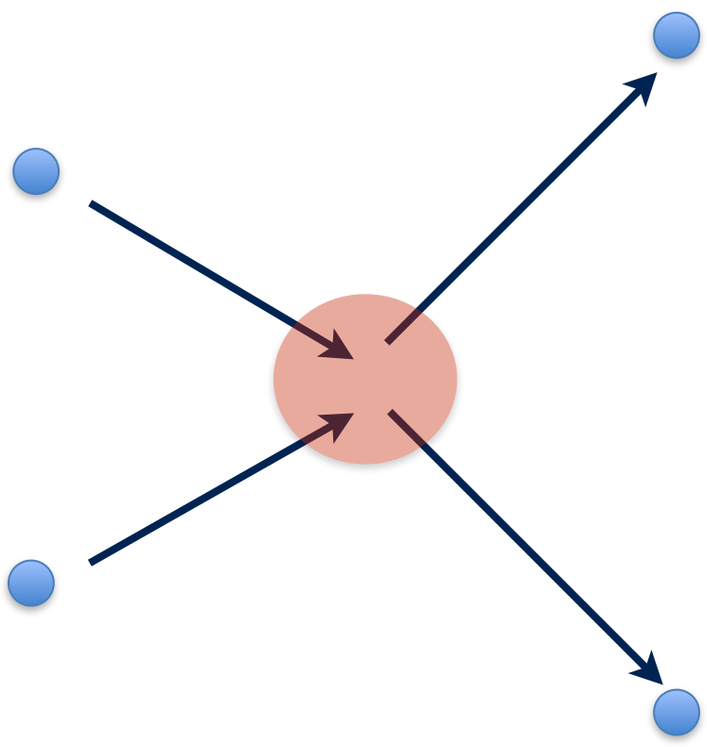
Tree methods:
N-body general strategy

Numerical relaxation

$$\nabla^2 \Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r}) \xrightarrow{\rho(\mathbf{r}) = \delta(\mathbf{r})} \Phi_N = -\frac{Gm}{r}$$



Before After



$$\frac{\partial f}{\partial} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{r}} + m\nabla\Phi_N \cdot \frac{\partial f}{\partial \mathbf{v}} = 0$$

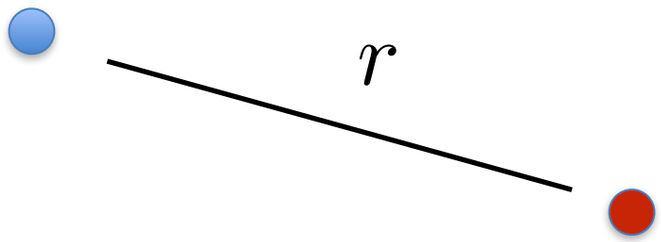
¿?

$$\frac{\partial f}{\partial} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{r}} + m\nabla\Phi_N \cdot \frac{\partial f}{\partial \mathbf{v}} = \left(\frac{\partial f}{\partial t}\right)_{col}$$

Tree methods:
N-body general strategy

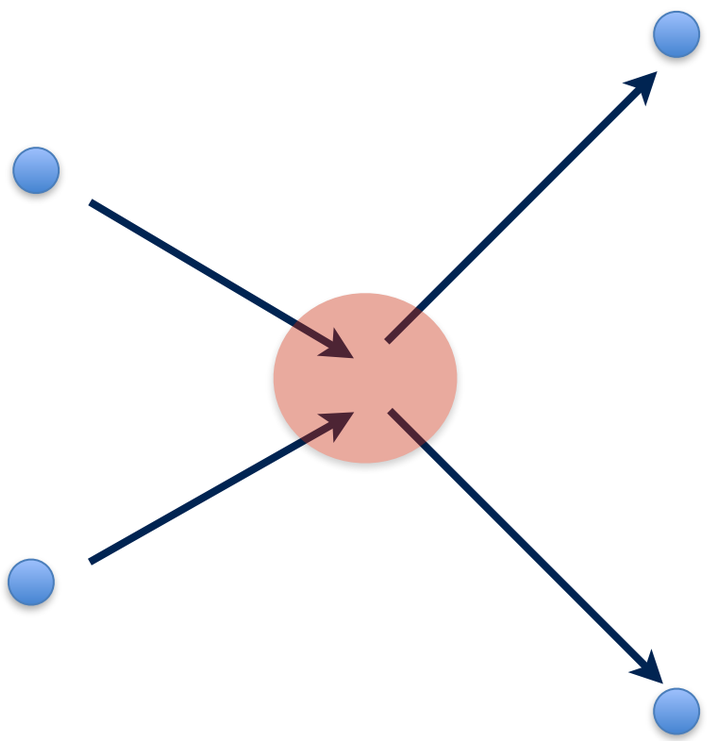
Numerical relaxation

$$\nabla^2 \Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r}) \qquad \rho(\mathbf{r}) = \delta(\mathbf{r}) \qquad \Phi_N = -\frac{Gm}{r}$$



Before

After



$$\frac{\partial f}{\partial} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{r}} + m\nabla\Phi_N \cdot \frac{\partial f}{\partial \mathbf{v}} = 0$$

¿?

$$\frac{\partial f}{\partial} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{r}} + m\nabla\Phi_N \cdot \frac{\partial f}{\partial \mathbf{v}} = \left(\frac{\partial f}{\partial t}\right)_{col}$$

Tree methods:
N-body general strategy

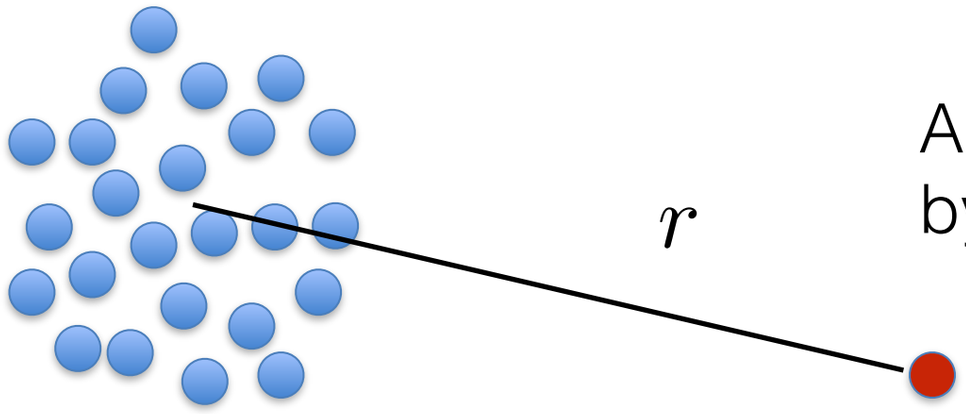
Numerical relaxation

$$\nabla^2 \Phi_N(\mathbf{r}) = 4\pi\rho(\mathbf{r})$$

$$\Phi_N = -\frac{Gm}{r}$$

A cluster of particles

Is a particle in the simulation



Artificial relaxation is avoided by using:

$$\rho_P = -\frac{3m}{4\pi\epsilon^3}\left(1+\frac{r^2}{\epsilon^2}\right)^{-5/2} \longrightarrow \Phi_N = -\frac{Gm}{\sqrt{r^2+\epsilon^2}}$$

Tree methods:
Take into account:

- Consistency
- Convergence
- Stability

- Evolution solver:
  Leapfrog
  (symplectic)

Guidelines:

- Energy, linear and angular momentum conservation.
- Phase space volumen conservation

Tree methods: Virial concept

1.1 Leyes de Newton

$$\mathbf{F} = m\,\mathbf{a}$$

1.2 Ecuaciones de Hamilton

$$\dot{q}_i = \frac{\partial H}{\partial p_i} \quad , \qquad \dot{p}_i = -\frac{\partial H}{\partial q_i}$$

1.3 Teorema del virial

$$\langle 2K \rangle + \langle \sum_i \mathbf{F}_i \cdot \mathbf{r}_i \rangle = \frac{1}{\tau}\left[G(\tau) - G(0)\right]$$

donde

$$G = \sum_i \mathbf{p}_i \cdot \mathbf{r}_i$$

Para un potencial

$$V = a r^{n+1}$$

El teorema del virial es

$$\langle K \rangle = \frac{n+1}{2}\langle V \rangle$$

y cuando $n = -2$ que es la gravedad estándar

$$\langle K \rangle = -\frac{1}{2}\langle V \rangle$$

$$\Phi_N = -\frac{Gm}{\sqrt{r^2 + \epsilon^2}}$$

Tree methods:
Flow diagram: Template

Programming start
template06

- main.c
  - mainloop.c

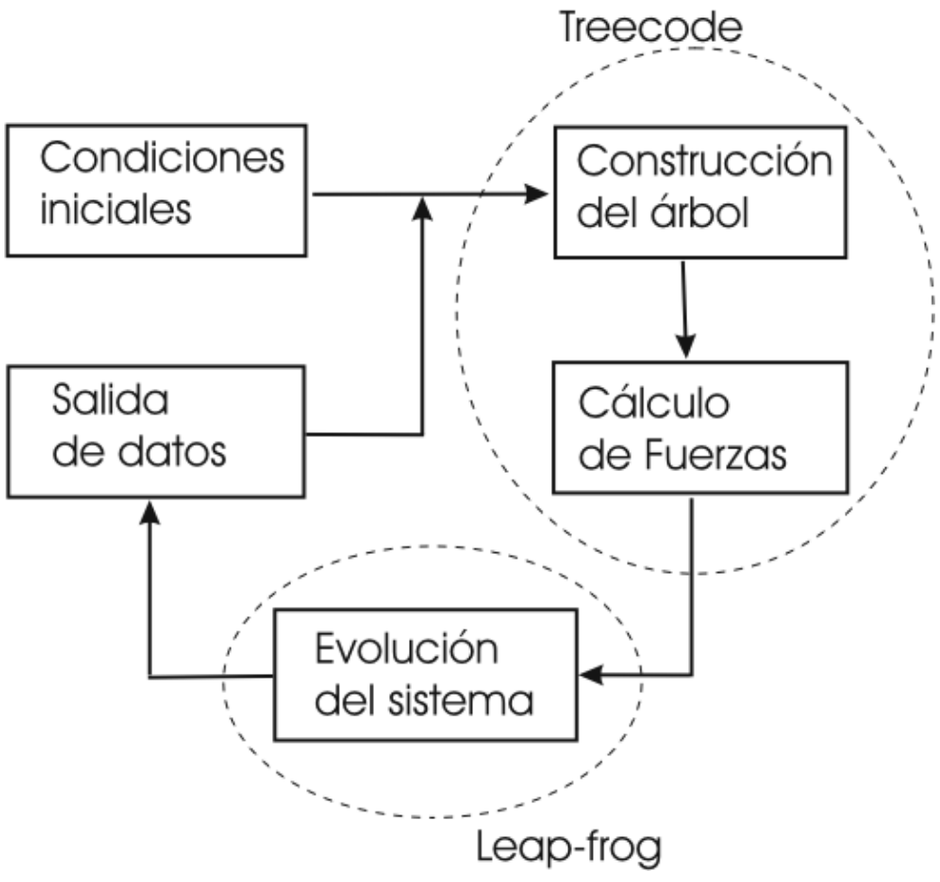
- cmdline_defs.h
- data_structure.h
- globaldefs.h
- protodefs.h

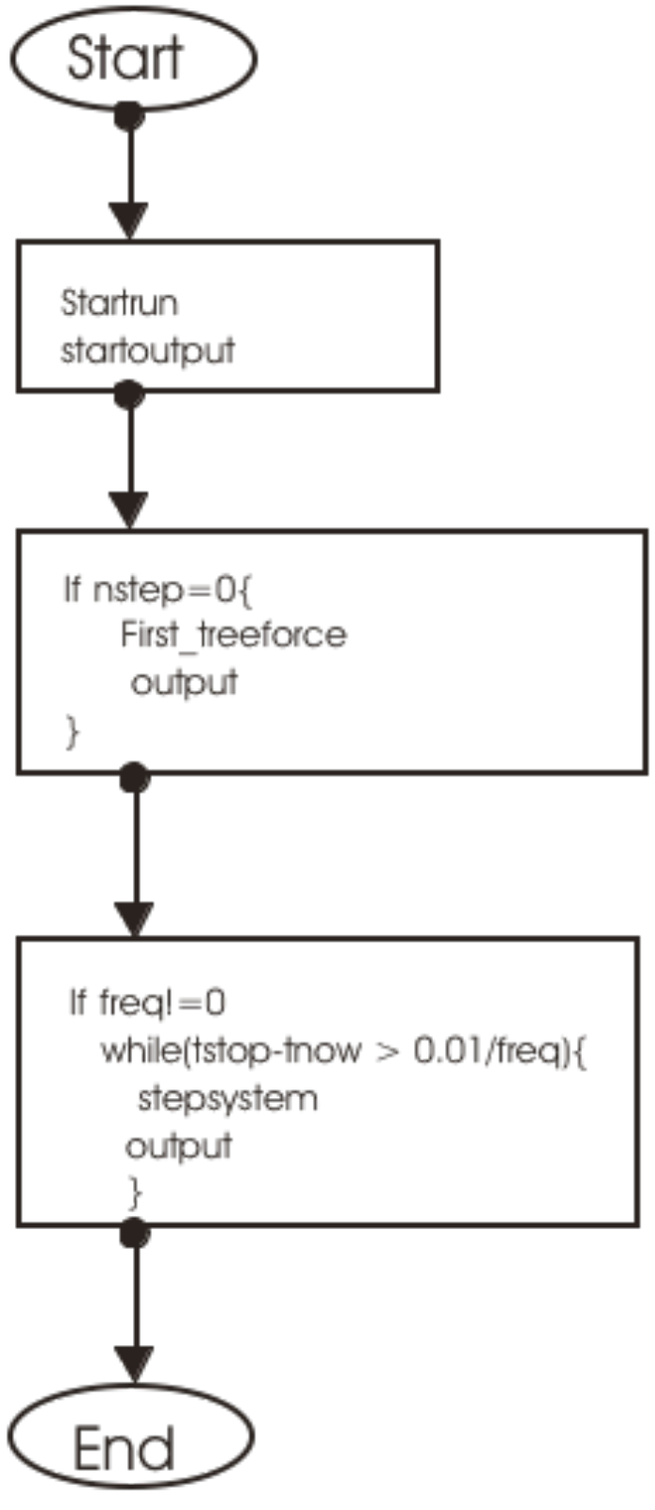
- template.c
- template_io.c
- startrun.c

- models.c
- models.h

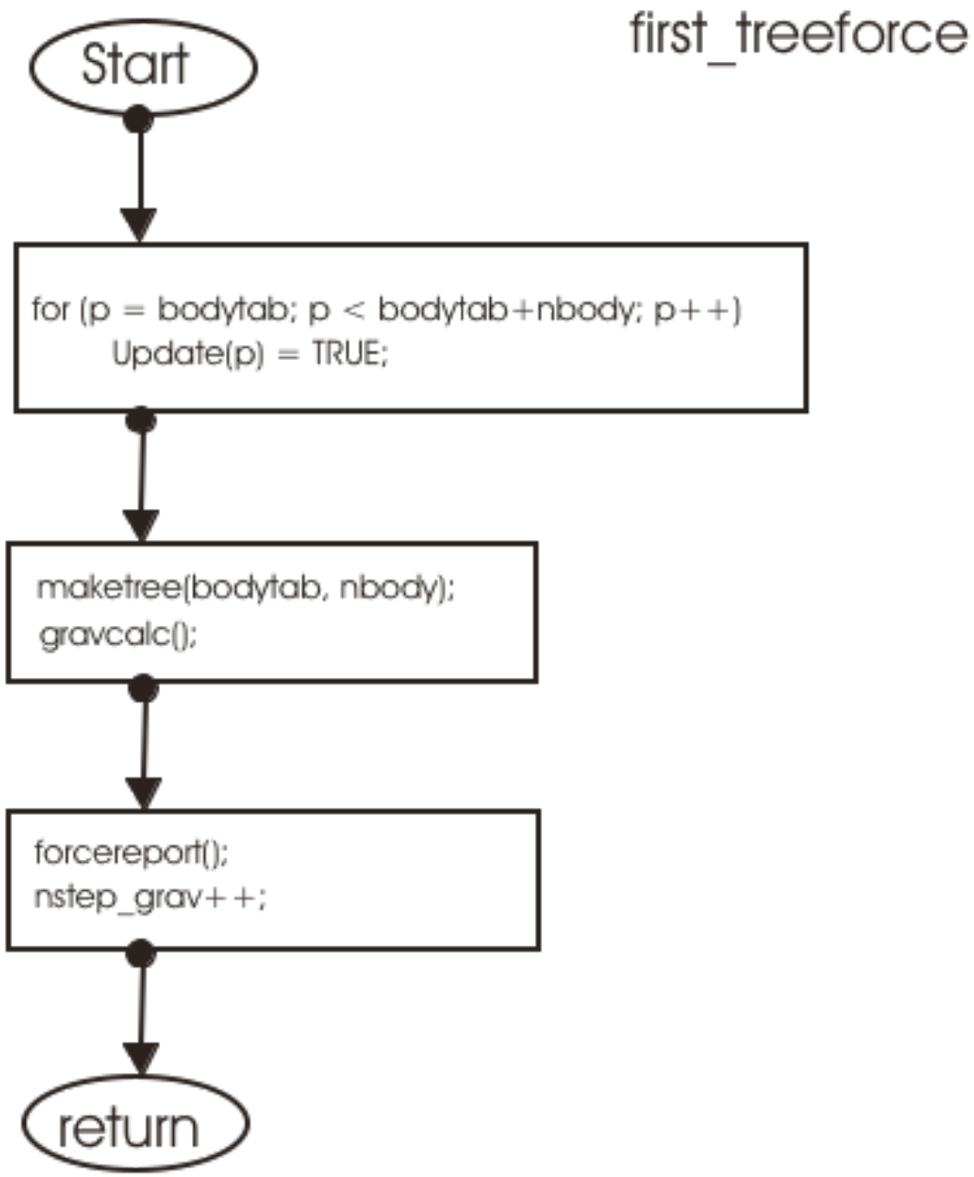
- Data structures
- Routines and functions

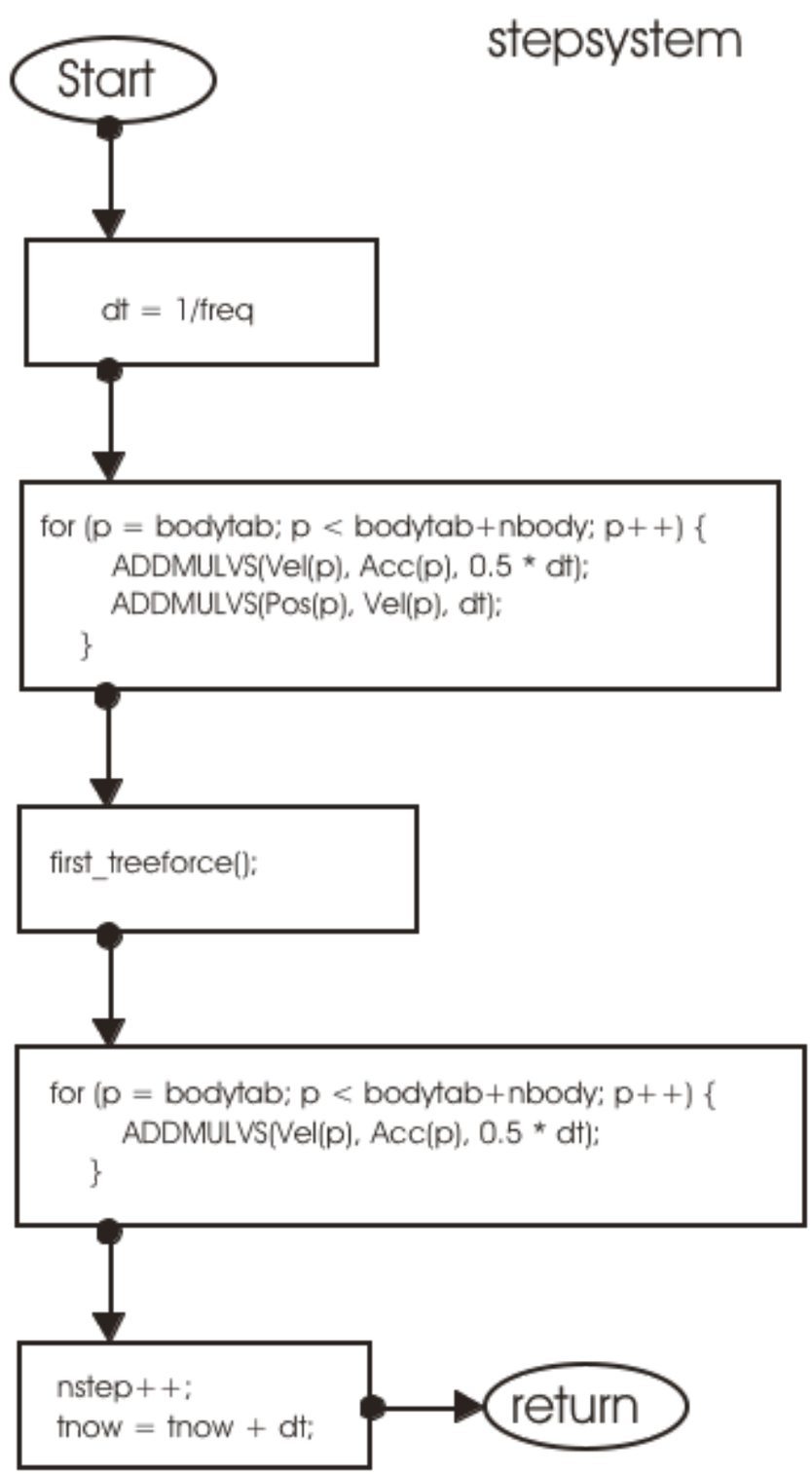
Tree methods:
Flow diagram: Main





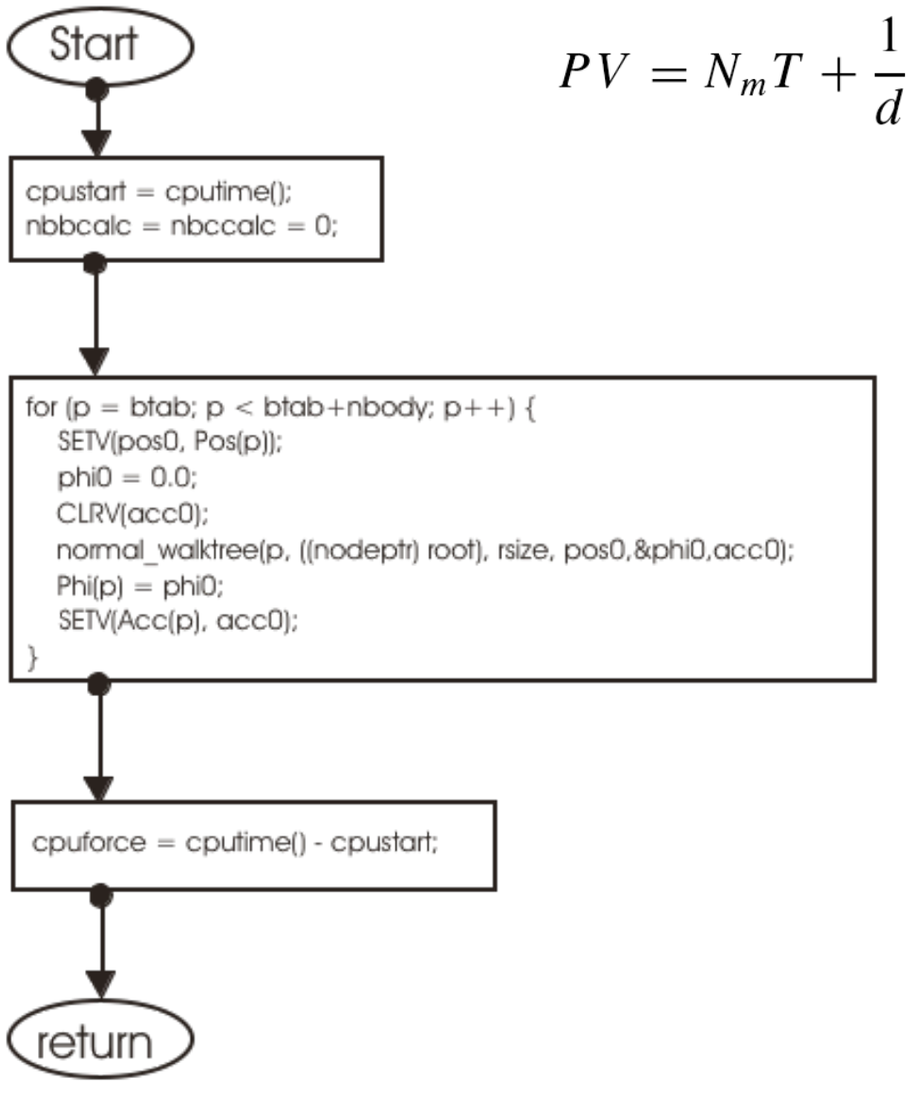
Tree methods: Flow diagram





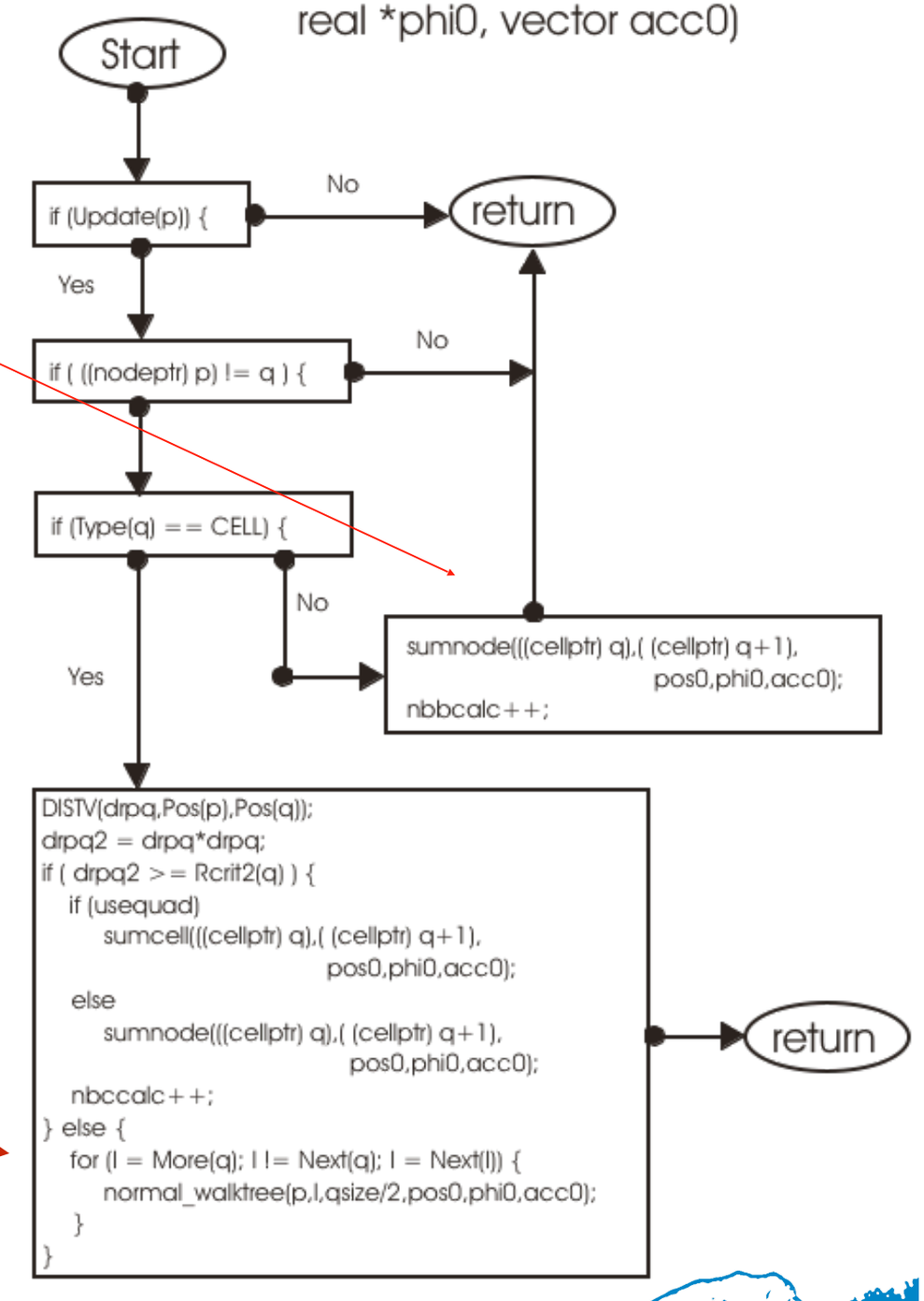
Tree methods:
Walking the tree and force computation

void normal_gravcalc(bodyptr btab, int nbody)



$$PV = N_m T + \frac{1}{d}\left\langle \sum_{i=1}^{N_m} \boldsymbol{r}_i \cdot \boldsymbol{f}_i \right\rangle$$

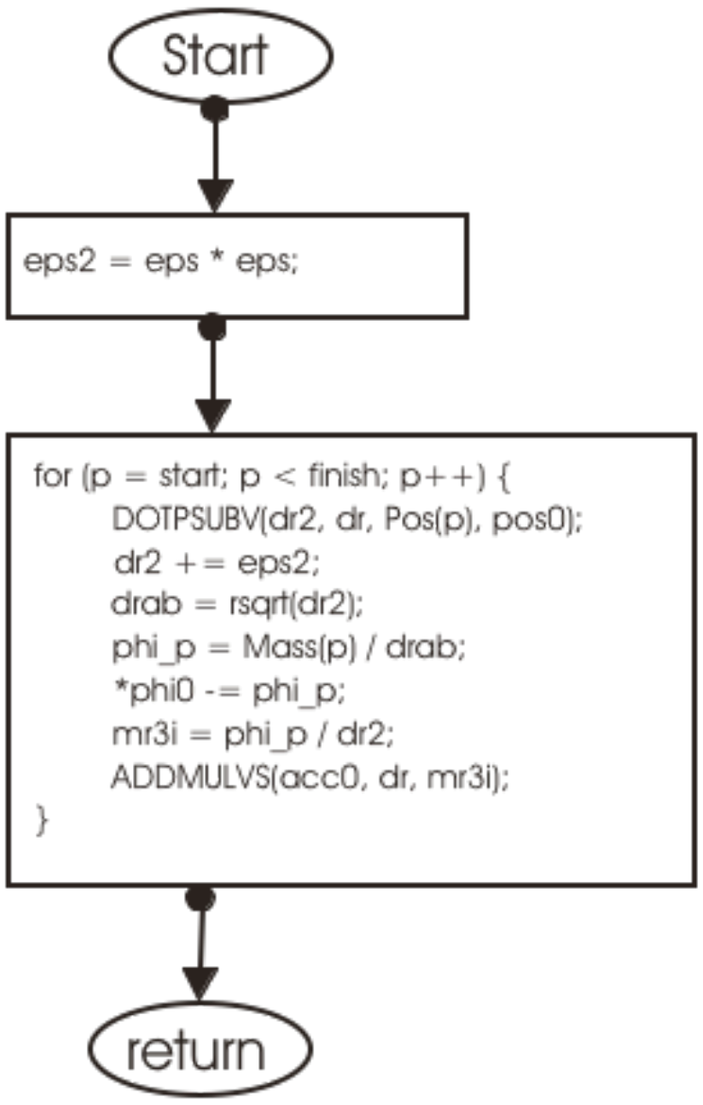
local void normal_walktree(bodyptr p,
nodeptr q, real qsize, vector pos0,
real *phi0, vector acc0)



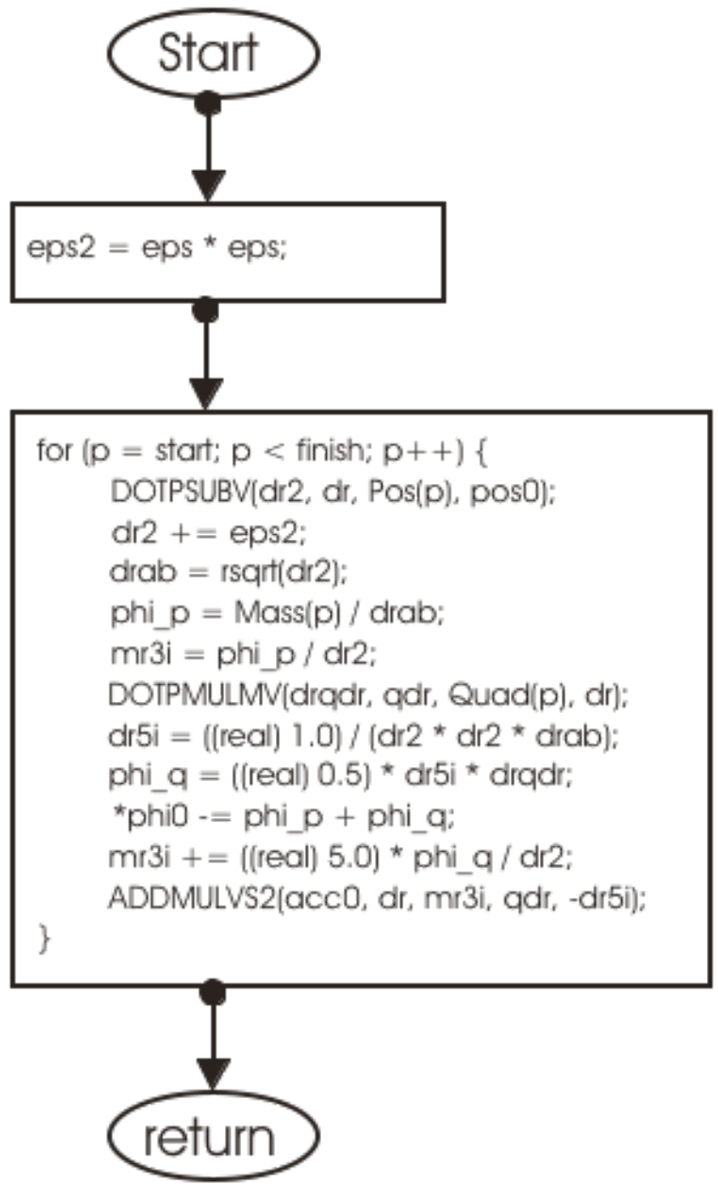
Tree methods: Sum of nodes and cells

$$\psi(\mathbf{r}) = \frac{m}{r} + \frac{1}{2}\sum_{i,j} Q_{ij}\frac{x_i x_j}{r^5} + \ldots$$

local void sumnode(cellptr start, cellptr finish,
vector pos0, real *phi0, vector acc0)



local void sumcell(cellptr start, cellptr finish,
vector pos0, real *phi0, vector acc0)



Running gbsph

```
[lapmar:Codigos/research/gbsph] mar% ./gbsph -help
./gbsph
  Method=gbsph01                    Hierarchical General Body and SPH (gbsph) code
  gravcalc_method=barnes            Method to use
  force_models=                     Gravitation calculation method to use
  Parameter_File=                   Other forces to include
  datanaly=false                    File with the info of parameters to use
  param_comp=false                  Data analysis
  in=                               Parameters Computation
  out=                              Input file with initial conditions
  outpv=                            Output file of N-body frames
  freq=32.0                         Output file of N-body pos-vel frames
  eps=0.025                         Fundamental integration frequency
  theta=1.0                         Density smoothing length
  usequad=false                     Force accuracy parameter
  dm_lambda=1.0                     If true, use quad moments
  dm_alpha=0.0                      Dark matter lambda, range of interaction
  eps_pot=1.0                       Dark matter alpha, intensity of the force
  sigma_pot=1.0                     External potential energy
  x_pot=7.0                         External potential length
  y_pot=0.0                         External potential x-position
  z_pot=0.0                         External potential y-position
  model=                            External potential z-position
  param_model=                      Model to test
  datanaly_type=                    Parameters Model to compute
  options=                          Data analysis type to be done
  tstop=2.0                         Various control options
  freqout=4.0                       Time to stop integration
  nbody=4096                        Data output frequency
  seed=123                          Number of bodies for test run
  save=                             Random number seed for test run
  restore=                          Write state file as code runs
  sphparam=false                    Continue run from state file
  VERSION=1.02                      Save SPH parameters
                                    M.A. Rodriguez-Meza 1999-2004
```


Tree methods:
Numerical details: *N*-body

For larger *N* numerical algorithms have a efficiency that depends on:

- Improvement of precision in the reduction of time step.
- Time step adaptive in response to boundary conditions.
- Increase of computational complexity as *N* increases.

Tree methods: Numerical details: *N*-body

Para *N* grande los algorítmos numéricos tienen una eficiencia que depende de:

- Crecimiento de la complejidad computacional con *N* : **Notación "O"**

Cuando observamos alguna función $f(n)$ en términos de su razón de crecimiento, algunos puntos saltan a la vista.

- Los términos constantes se expresan como $O(1)$. Cuando analizamos el tiempo de corrida de un algoritmo aplicamos esta regla cuando tenemos una tarea que sabemos se ejecutará en cierto tiempo sin importar el tamaño de los datos que son procesados. Formalmente decimos que para alguna constante c, $O(c) = O(1)$.

- Las constantes multiplicativas se omiten. Cuando analizamos el tiempo de corrida de un algoritmo aplicamos esta regla cuando tenemos un número de tareas que se ejecutan en el mismo tiempo. Por ejemplo, tres tareas cada una corre en el tiempo $T(n) = n$, el resultado es $O(3n)$ lo que se simplifica a $O(n)$. Formalmente se establece que $O(cT) = cO(T) = O(T)$ para alguna constante c.

- La suma se hace tomando el máximo. Cuando analizamos el tiempo de corrida de un algorítmo aplicamos esta regla cuando una tarea se ejecuta después de otra. Por ejemplo, si $T_1(n) = n$ y $T_2(n) = n^2$, describen dos tareas ejecutadas secuencialmente, el resultado es $O(n) + O(n^2)$, lo que se simplifica a $O(n^2)$. Formalmente decimos $O(T_1)+O(T_2) = O(T_1+T_2) = \max(O(T_1)+O(T_2))$.

Tree methods: Numerical details: *N*-body

Para *N* grande los algorítmos numéricos tienen una eficiencia que depende de:

- Crecimiento de la complejidad computacional con *N* : **Notación "O"**

- La multiplicación no cambia pero es escrita de una forma más compacta. Cuando analizamos el tiempo de corrida de un algorítmo aplicamos esta regla cuando una tarea hace que otra sea ejecutada un cierto número de veces para cada iteración de sí misma. Por ejemplo, en un ciclo anidado cuyas iteraciones exteriores son descritas por T_1 y cuyas iteraciones interiores por T_2, si $T_1(n) = n$ y $T_2(n) = n$, el resultado es $O(n)O(n)$, o $O(n^2)$. Formalmente, $O(T_1)O(T_2) = O(T_1T_2)$.

Tree methods: Numerical details: *N*-body

Para *N* grande los algorítmos numéricos tienen una eficiencia que depende de:

- Crecimiento de la complejidad computacional con *N* : **Notación "O"**

Veamos un ejemplo. Supongamos que tenemos un algoritmo cuyo tiempo de corrida está descrito por la función $T(n) = 3n^2 + 10n + 10$. Usando las reglas de la notación-O, podemos simplificar esta función como:

$$O(T(n)) = O(3n^2 + 10n + 10) = O(3n^2) = O(n^2) \quad ,$$

lo que indica que el término con n^2 será el que contribuya con el mayor tiempo de corrida cuando n se hace arbitrariamente grande. Específicamente, si $n = 10$ tenemos:

Tiempo de corrida para $3n^2$:	$3(10)^2/(3(10)^2 + 10(10) + 10)$	=	73,2 %
Tiempo de corrida para $10n$:	$10(10)/(3(10)^2 + 10(10) + 10)$	=	24,4 %
Tiempo de corrida para 10:	$10/(3(10)^2 + 10(10) + 10)$	=	2,4 %

Vemos que n^2 se lleva la mayoría del tiempo de corrida. Para $n = 100$ tenemos:

Tiempo de corrida para $3n^2$:	$3(100)^2/(3(100)^2 + 10(100) + 10)$	=	96,7 %
Tiempo de corrida para $10n$:	$10(100)/(3(100)^2 + 10(100) + 10)$	=	3,2 %
Tiempo de corrida para 10:	$10/(3(100)^2 + 10(100) + 10)$	<	0,1 %

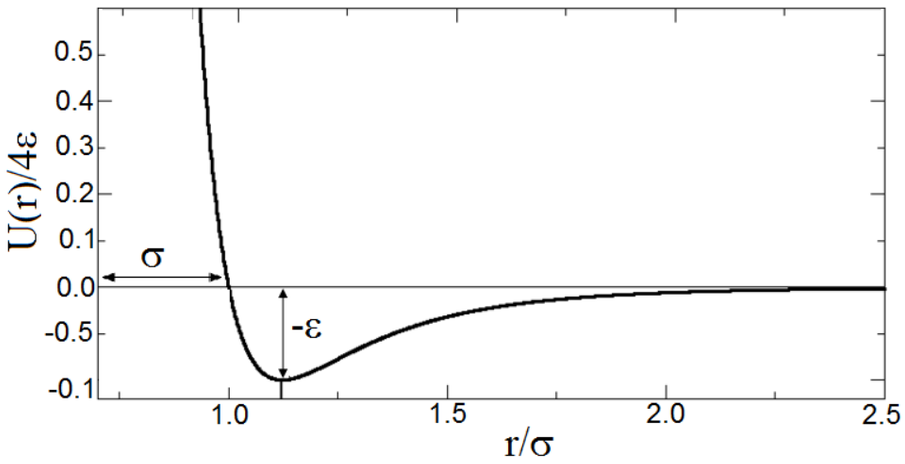
Aquí vemos que el término cuadrático se lleva prácticamente todo el tiempo de la corrida y los otros términos se hacen comparativamente despreciables. El lector puede imaginar qué pasaría para $n = 10^6$.

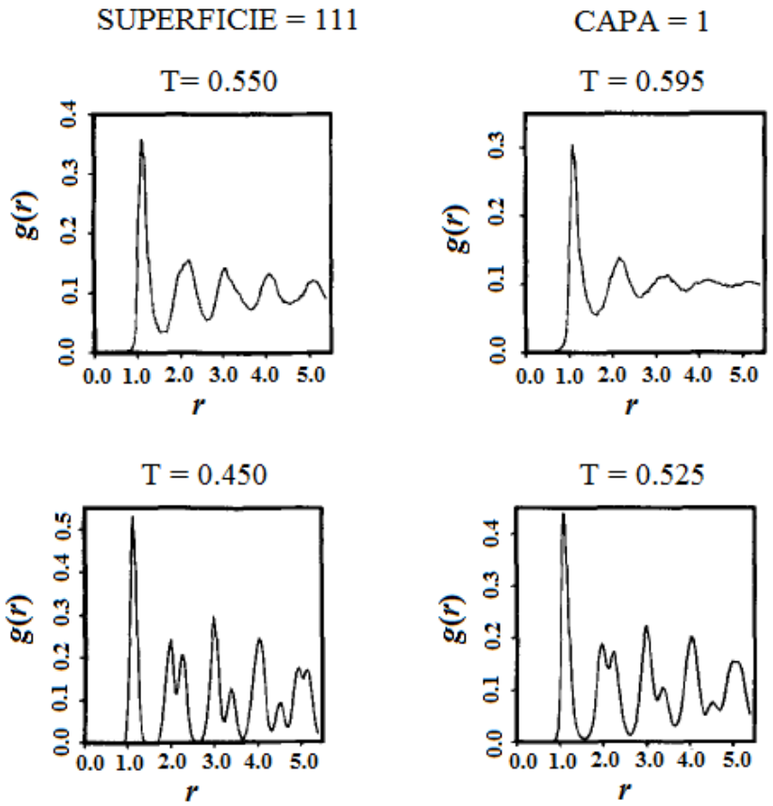
Tree methods: Numerical details

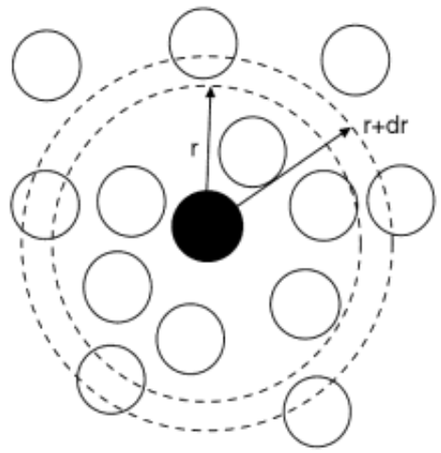
Homework: why tree method is O(*N* Log *N*)?

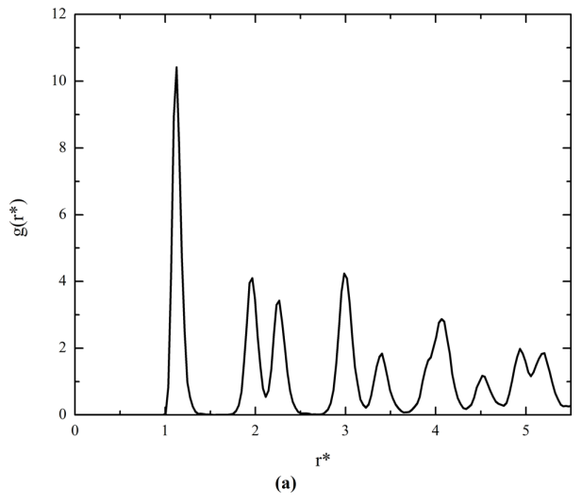
The liquid structure (2D) can be simulated using molecular dynamics: simplest case Lennard-Jones potential

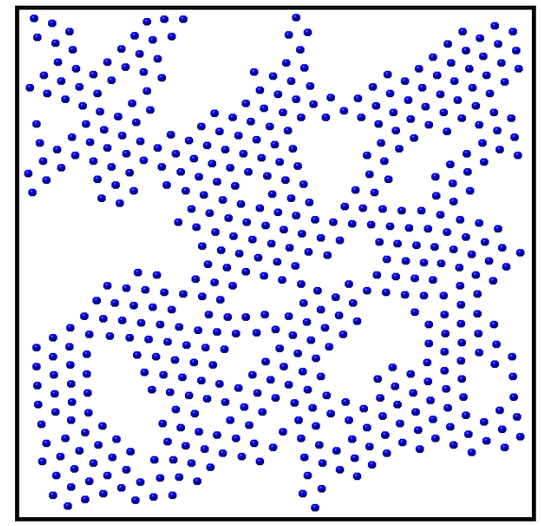
$$\mathbf{F} = m\,\mathbf{a}$$

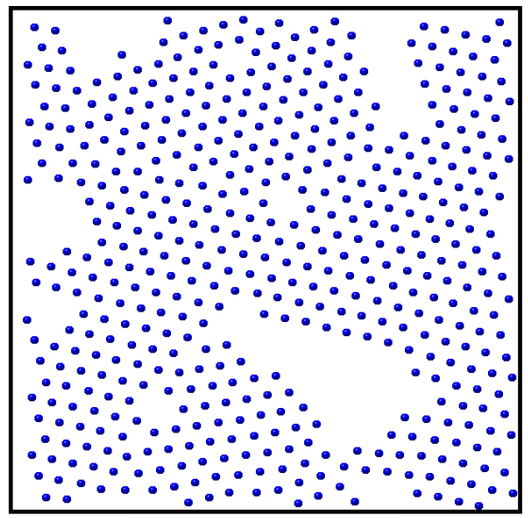


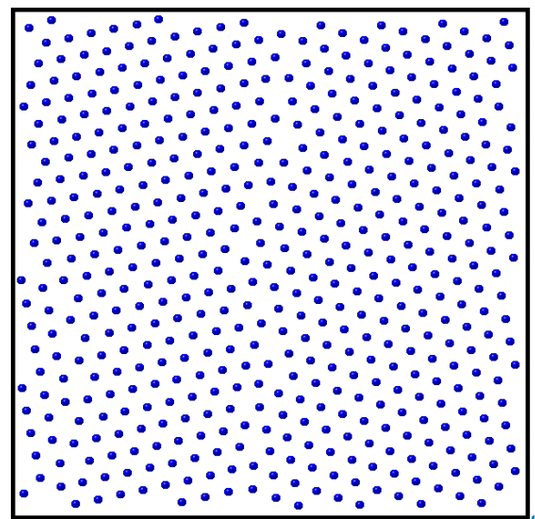







(a)


(b)


(c)

Conclusions:
Tree methods. (a)

We have seen:

- Concept of binary tree.
- Examples in *N*-body simulations.

Conclusions:
Tree methods. (b)

- Start with the lagrangian
- Background study (cosmological principle)
- Change CAMB or CLASS and generate power spectrum a initial z. Then produce Initial condition
- Newtonian limit, using a perturbed metric
- Implement Poisson and Klein-Gordon equations in an N-body version
- Obtain geodesics (Newton like equations).
- Modified SPH in order to take into account the modifications coming from scalar fields…
- Analyse snaps: two-point correlation function, power spectrum, halos catalogs, voids DF, etcetera…

References and material

- Cosmología numérica y estadística: NagBody kit (http://bitbucket.org/rodriguezmeza). Mario A. Rodríguez-Meza. And: https://github.com/rodriguezmeza/NagBody_pkg.git

- Métodos numéricos en astrofísica, capítulo 1, Método de N-cuerpos en astrofísica. (https://www.researchgate.net/publication/316582859_Metodo_de_N-Cuerpos_en_Astrofisica)

- La estructura a gran escala del universo. Capítulo 22 en Travesuras cosmológicas de Einstein et al. https://www.researchgate.net/publication/316582400_La_estructura_a_gran_escala_del_universo_simulaciones_numericas

- https://www.researchgate.net/profile/Mario_Rodriguez-Meza

- https://www.researchgate.net/publication/314281416_Los_agujeros_negros_y_las_ondas_del_Dr_Einstein

- M.A. Rodriguez-Meza, Adv. Astron. 2012, 509682 (2012). arXiv: 1112.5201. (https://www.researchgate.net/publication/51967093_A_Scalar_Field_Dark_Matter_Model_and_Its_Role_in_the_Large-Scale_Structure_Formation_in_the_Universe)

See you!