Technische
Universität
Braunschweig

Parallel Computing

Exercise 8

Andres Rodriguez, 9th July 2015

Homework 7 - Remember

- ✓ **Deadline**

  22.07.2015 - 11:59:pm

- ✓ **E-mail**

  Andres Rodriguez
  a.rodriguez-escobar@tu-braunschweig.de

- ✓ **Content**

  ZIP file including - Source code
  - Written report as *.pdf file

Homework 8

Problem definition and Solution Technics

Homework 8

Body interaction:

$$\mathbf{F}_{ij} = G m_i m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3} \quad \Big\}$$

2 Body interaction

$$\mathbf{F}_i = \sum_{j=1, i \neq j}^{n} \mathbf{F}_{ij} = G m_i \sum_{j=1, i \neq j}^{n} m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3} \quad \Big\}$$

Multy-Body interaction

Problem modelling and discretization:

$$\mathbf{F}_i = m_i \frac{d^2 \mathbf{x}_i}{d t^2}$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \frac{\Delta t}{m_i} \mathbf{F}_i^n$$

$$\longrightarrow$$

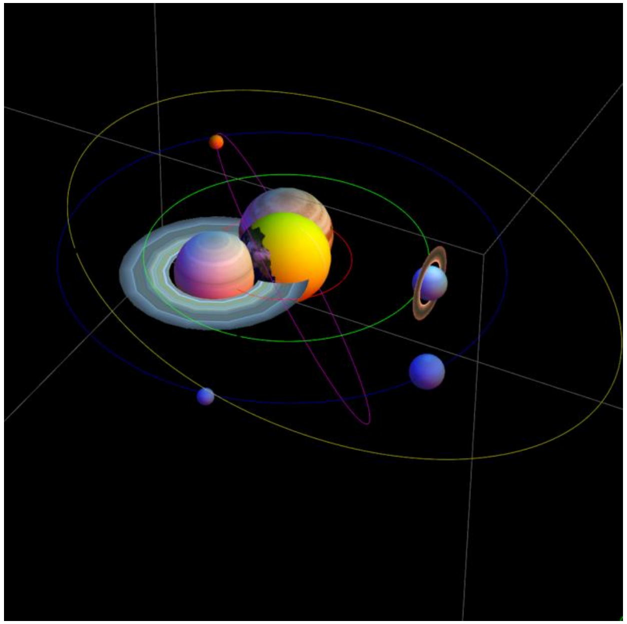
$$\vec{x}_i^{n+1} := \vec{x}_i^n + \Delta t \cdot \vec{v}_i^n$$
$$\vec{v}_i^{n+1} := \vec{v}_i^n + \frac{\Delta t}{m_i} \cdot \vec{F}_i^n$$

Discretization in time
What is the current state of the body?

Homework 8

Body state definition:

$$\vec{x}_i^{n+1} := \vec{x}_i^n + \Delta t \cdot \vec{v}_i^n$$
$$\vec{v}_i^{n+1} := \vec{v}_i^n + \frac{\Delta t}{m_i} \cdot \vec{F}_i^n$$



State descriptor elements:

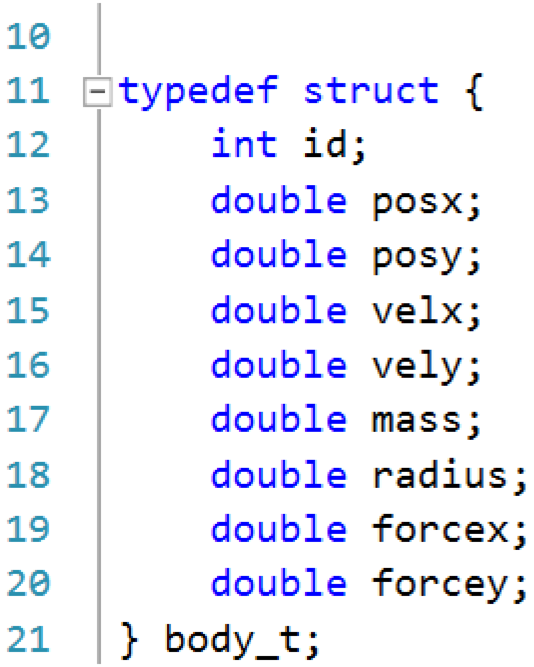
- Position
- Velocity
- Mass
- Force
- Radius

Homework 8

Body state definition:

State descriptor elements (2D):

- Position (x , y)
- Velocity (v , u)
- Mass
- Force (F_x , F_y)
- Radius

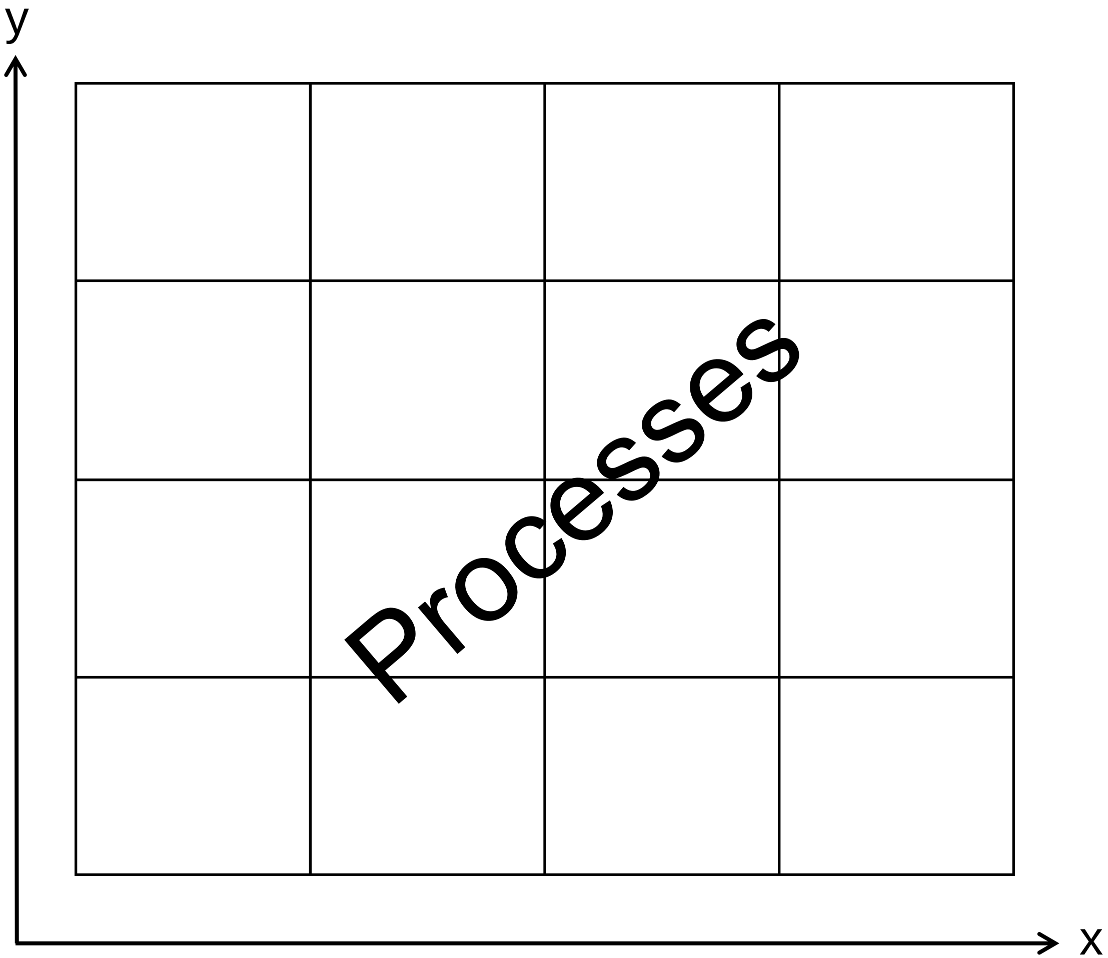


```
typedef struct {
    int id;
    double posx;
    double posy;
    double velx;
    double vely;
    double mass;
    double radius;
    double forcex;
    double forcey;
} body_t;
```

Homework 8

Space abstraction from the processes

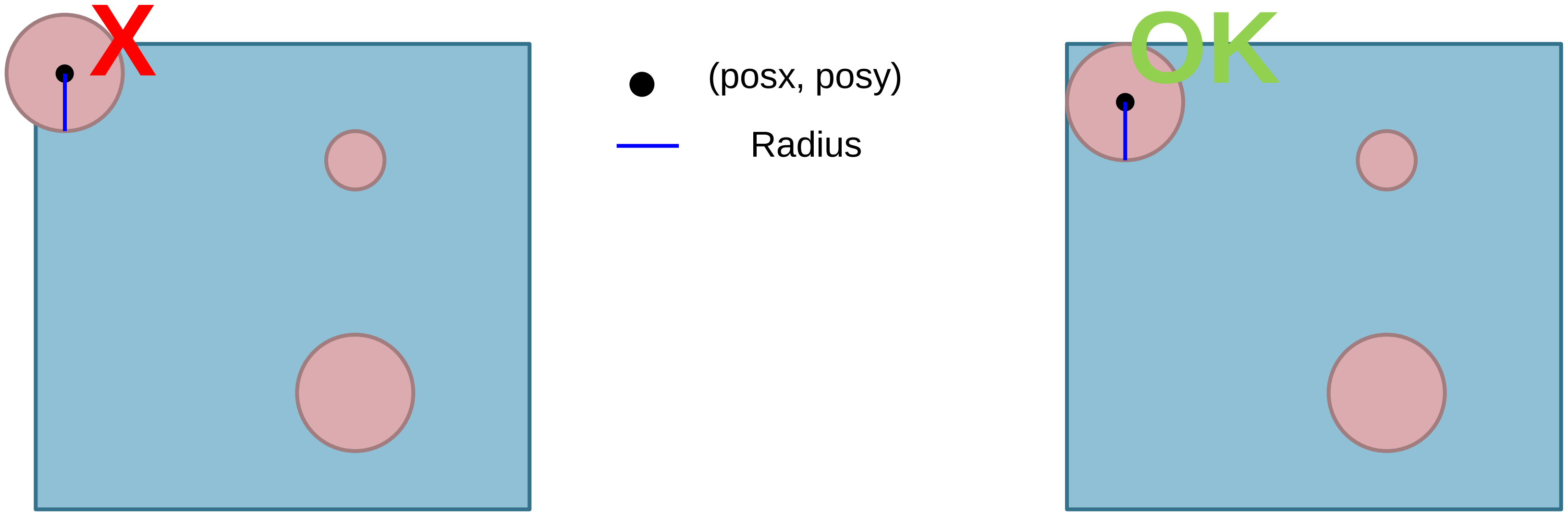
All processes contain the whole bodie descriptors but can identify their belongin (According to current location) bodies



Homework 8

Body state definition:

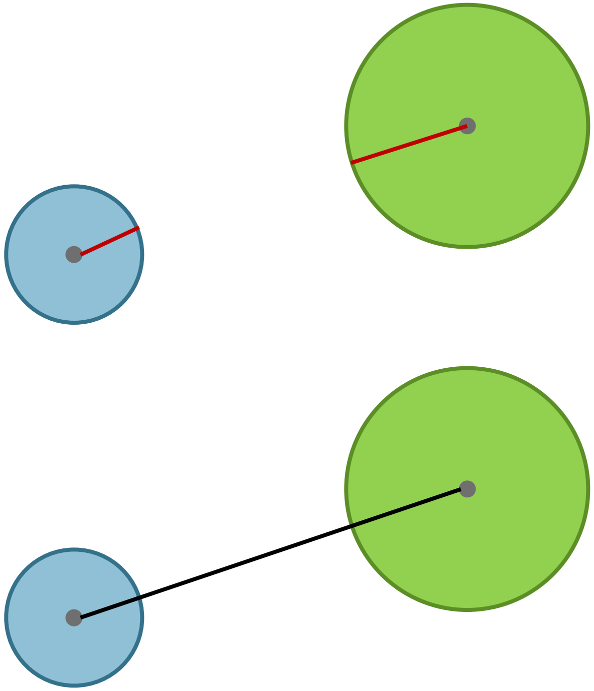
Initialize your bodies to fit or be created within the defined space:



Homework 8

Callculating Collitions:

Compare sum of Radiuses Vs Euclidean distance to check collisions:



What happens with the body motion after collition?

Some online available code for elastic collisions:
http://www.plasmaphysics.org.uk/programs/coll2d_cpp.htm

Homework 8

Inter-process communication of the body descriptors

MPI Allows to consider custom data structures as custom datatypes:

Example:

```
//Normal C/C++ style struct
typedef struct {
    int var;
    char string[STRING_LENGTH];
    double foo;
} bar;

//********************************************************************************************************************************************************************************

int count = 3;                                                      //Counts the number of declarations within the struct “bar”.
int lengths[3] = {1, STRING_LENGTH, 1};                             //Number of elements of the types declared within the struct “bar”.

MPI_Datatype barDatatype;                                           //Declare a new type for the MPI interface.

MPI_Aint offsets[3] = {0, sizeof(int), sizeof(int) + STRING_LENGTH}; //Offsets or starting points of each per type correspondent elements.
MPI_Datatype types[3] = {MPI_INT, MPI_CHAR, MPI_DOUBLE};            //Vector of the MPI Type versions of the orginal in-struct elements.

MPI_Type_struct(count, lengths, offsets, types, &barDatatype);      //Creates the new MPI data structure and assign to barDatatype and
ID.
MPI_Type_commit(&barDatatype);                                      //Make MPI aware of the new MPI struct.
```

Homework 8

Inter-process communication of the body descriptors

Other “Manual” option for customized data transmission is MPI_Pack and MPI_Unpack:

Example:

```
//Normal C/C++ style struct
typedef struct {
    int var;
    char string[STRING_LENGTH];
    double foo;
} bar;

//*****************************************************************************************************************************************************************************************

buffsize =  sizeof(int) + STRING_LENGTH*size(char) + sizeof(double);                  //Needs to be in bytes.
double *positions = malloc(buffsize);                                                 //Memory allocation for the send buffer.

MPI_Pack(&bar.var, 1, MPI_INT, positions, buffsize, &pos, MPI_COMM_WORLD);
MPI_Pack(bar.string, STRING_LENGTH, MPI_CHAR, positions, buffsize, &pos, MPI_COMM_WORLD);
MPI_Pack(&bar.double, 1, MPI_DOUBLE, positions, buffsize, &pos, MPI_COMM_WORLD);

double *otherpos = malloc(buffsize);                                                  //Memory allocation for the receive buffer.
MPI_Bcast(otherpos, buffsize, MPI_PACKED, i, MPI_COMM_WORLD);                         //Ex. of communicate the packed data.

MPI_Unpack(otherpos, buffsize, &pos, &var_dst, 1, MPI_INT, MPI_COMM_WORLD);
MPI_Unpack(otherpos, buffsize, &pos, string_dst, STRING_LENGTH, MPI_CHAR, MPI_COMM_WORLD);
MPI_Unpack(otherpos, buffsize, &pos, &foo_dst, 1, MPI_DOUBLE, MPI_COMM_WORLD);
```

Homework 8

Solution Video

Homework 8

„Advanced“ Examples Using – GPGPU

https://www.youtube.com/watch?v=XLMU6o7n4E4
https://www.youtube.com/watch?v=XUBIlJ9uaZU