Parallel Programming

Cosmic Dark Matter /
2-point Angular Correlation

Fall 2020, period 2

Jan Westerholm
Åbo Akademi University

Program Design and Implementation

- In this exercise you are asked to
  - *design*
  - *implement*
  - *run* and
  - *report*

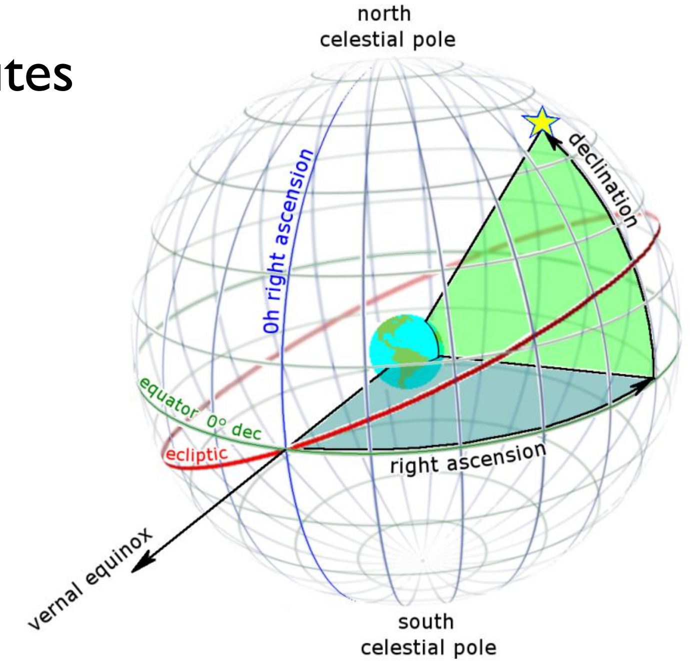
  the calculation of three histograms of the 2-point angular correlation function for two sets of galaxies
  - D: measured set of 100 000 galaxies D
  - R: synthetic random evenly distributed set of 100 000 galaxies

Program Design and Implementation

- Input data
- What needs to be calculated
- Implementation on CPU and GPU
- Output data
- Check points?

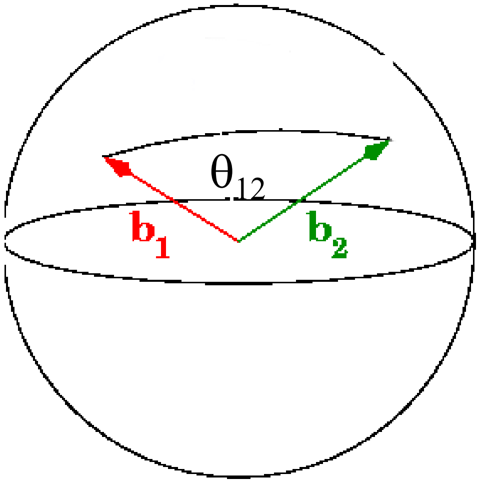
Input Data

- Two lists of N galaxy locations: real measured galaxies and synthetic evenly distributed random galaxies
- For each galaxy, real or synthetic, the list contains the galactic coordinates
  - right ascension α, in arc minutes
  - declination δ, in arc minutes
- Convert from arc minutes to radians by multiplying with $1/60*\pi/180$
- Lists available from moodle



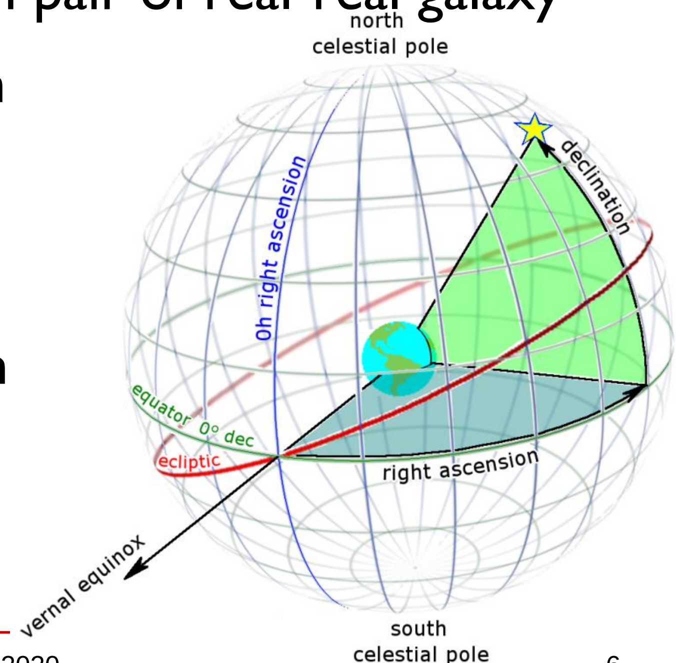
Input Data in moodle

Real data	Synthetic data
100000	100000
4646.98 3749.51	840.961426 387.991697
4644.35 3749.52	387.368692 2967.285746
4643.24 3748.67	2667.070581 3385.866638
4646.98 3750.89	3942.141923 3720.545649
4641.13 3748.23	5164.800068 3205.065003
4646.75 3750.26	1680.286209 653.214854
4643.29 3751.99	3553.889989 152.983872
4640.78 3747	1011.581615 76.716702
4638.95 3749.74	298.341944 3752.723427
4647.29 3749.82	4044.605401 2943.458158
4651.65 3749.02	1030.720073 2342.109887
4649.65 3747.16	54.257097 3769.420145
4646.66 3752.23	2478.545632 336.072740
4649.38 3749.14	970.044311 3556.987037
4648.22 3750.32	840.791266 2610.257310
4639.27 3747.96	5272.628047 2441.450374
4637.3 3749.5	380.633015 3771.599991
4649.07 3751.94	5049.735705 4627.545008
4640.39 3752.43	2849.250851 2945.861883



What Needs to be Calculated

- Initialize the three histograms DD, DR and RR to zero, covering $0 \rightarrow 180$ degrees, bin width $= 0.25$ degrees: DD[0] covers [0, 0.25), DD[1] covers [0.25,0.5), etc.
  - Hint: $0 \rightarrow 90$ degrees is enough in our case
- Fill DD from angles between each pair of real-real galaxy
- Fill DR from angles between each pair of real-random galaxy
- Fill RR from angles between each pair of random-random galaxy
- Check point: how many entries in DR? In DD and RR? Answer: 10 billion (10000000000) !



What Needs to be Calculated

- Given two points on the surface of a sphere, how do we calculate the angle between those two points as seen from the center of the sphere?
- Basic idea: represent each point by a unit vector from the center of the sphere to the surface of the sphere, and calculate the dot product of the two vectors.
- The dot product $\bullet$ between two 3-D vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ is given by $\mathbf{r}_1 \bullet \mathbf{r}_2 = |\mathbf{r}_1|\,|\mathbf{r}_2| \cos(\theta_{12})$ where $|\mathbf{r}|$ is the length of the vector $\mathbf{r}$, and θ_{12} is the angle between the two vectors.
- Here we have unit vectors, hence $|\mathbf{r}_1| = |\mathbf{r}_2| = 1$

What Needs to be Calculated

- The formula we can use is now

$$\theta_{12} = \arccos(\mathbf{r}_1 \bullet \mathbf{r}_2)$$

$$\mathbf{r} = r_x\mathbf{e}_x + r_y\mathbf{e}_y + r_z\mathbf{e}_z$$

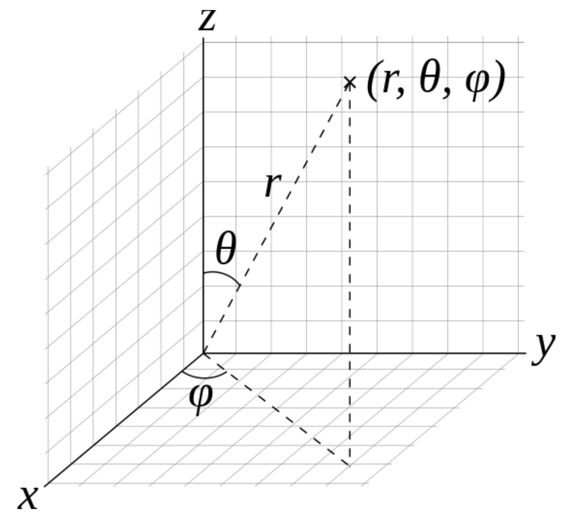
$$\mathbf{r}_1 \bullet \mathbf{r}_2 = r_{1x}*r_{2x} + r_{1y}*r_{2y} + r_{1z}*r_{2z}$$

- Using spherical coordinates (r,θ,φ), the Cartesian components of **r** are given by

$$r_x = r*\sin(\theta)*\cos(\varphi)$$

$$r_y = r*\sin(\theta)*\sin(\varphi)$$

$$r_z = r*\cos(\theta)$$



Cartesian: x, y, z
Spherical: r, θ, φ

What Needs to be Calculated

- Finally, what is the connection between the spherical coordinates θ, φ and the galactic coordinates right ascension α and declination δ ?
- Answer: $\varphi = \alpha$ and $\theta = 90 - \delta$
- $r_1 \bullet r_2 = \sin(\theta_1)\cos(\varphi_1)\sin(\theta_2)\cos(\varphi_2) + \sin(\theta_1)\sin(\varphi_1)\sin(\theta_2)\sin(\varphi_2) + \cos(\theta_1)\cos(\theta_2)$

$$= \cos(\delta_1)\cos(\alpha_1)\cos(\delta_2)\cos(\alpha_2) + \cos(\delta_1)\sin(\alpha_1)\cos(\delta_2)\sin(\alpha_2) + \sin(\delta_1)\sin(\delta_2)$$

$$= \cos(\delta_1)\cos(\delta_2)\,[\cos(\alpha_1)\cos(\alpha_2) + \sin(\alpha_1)\sin(\alpha_2)] + \sin(\delta_1)\sin(\delta_2)$$

$$= \cos(\delta_1)\cos(\delta_2)\cos(\alpha_1 - \alpha_2) + \sin(\delta_1)\sin(\delta_2)$$

What Needs to be Calculated

- Final result: the angle θ_{12} between two galaxies $(\alpha_1, \delta_1), (\alpha_2, \delta_2)$ is given by

$$\theta_{12} = \mathrm{arccos}\big(\mathrm{sin}(\delta_1) * \mathrm{sin}(\delta_2) + \mathrm{cos}(\delta_1) * \mathrm{cos}(\delta_2) * \mathrm{cos}(\alpha_1 - \alpha_2)\big)$$

- Data type for α_i, δ_i : single or double precision floating point numbers?
- Data layout in memory?

Output Data

- Plot the histograms DD and RR to see if there are any visible differences.
- The scientific measure for differences between the two distributions of galaxies is

  $\omega_i(\theta) = (DD_i - 2*DR_i + RR_i)/RR_i$

  DD_i, DR_i, RR_i = value in histogram bin i
- If all ω_i are close to zero, in the range $[-0.5,0.5]$, then we have a random distribution of real galaxies (the two distributions are statistically the same)
- If the ω_i values are < -1.0 or $> +1.0$, then we have a non-random distribution of real galaxies

GPU: Threads and Thread Blocks

- Choose your threads and thread blocks
- If you have the time, play around with the size of your thread blocks and the amount of work one thread does.
- Do you need to synchronize your threads?
- Do you need to use atomic operations?
- Is it a good idea to read the galaxy data within a thread block into *shared* memory?

Implementation on CPU and GPU

- Use the very basic program template from the lecture slides
  - read the data from the files, manipulate as needed
  - transfer data to the GPU (or use unified memory)
  - start the kernel(s)
  - transfer data back to the CPU
  - present your results
- Start out with a small set of galaxies, perhaps with known relative angles?

Short Instructions for dione.abo.fi

- Log on to dione:

```
ssh user_name@dione.abo.fi
```

- Load necessary modules:

```
module load openmpi          // MPI
module load CUDA             // CUDA
```

- Compile your program

```
mpicc -O3 mpiprog.c -o mpiprog -lm
nvcc gpuprog.cu -o gpuprog -lm
```

- Run your program on the batch system

```
srun -o prog.out -e prog.err mpiprog real_data sim_data
srun -p gpu -n 100 gpuprog real_data sim_data
```

Hints and help

- Input data is available from moodle.abo.fi
- Reference results and further explanations available from

  D. Bard, M. Bellis, M.T.Allen, H. Yepremyan, J.M. Kratochvil, "*Cosmological calculations on the GPU*", **Astronomy and Computing**, Vol. 1 (2013) 17–22
- Report the total time for your program to calculate the estimators $\omega_i(\theta)$.