Åbo Akademi University

IT00CG19-3002

GPU Programming

Slide set #4: Cosmic Dark Matter / 2-point Angular Correlation

Fall 2023, period 1

Jan Westerholm
Åbo Akademi University

Program Design and Implementation

- To pass this course you are asked to
  - *design*
  - *implement*
  - *run* and
  - *report*

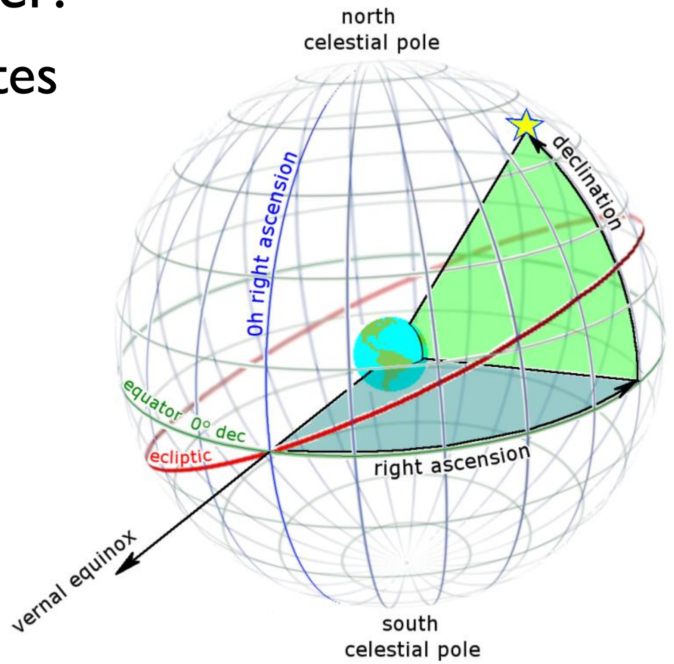
  the calculation of three histograms of the 2-point angular correlation function for two sets of galaxies
  - D: measured set of 100 000 galaxies D
  - R: synthetic random evenly distributed set of 100 000 galaxies

Program Design and Implementation

- Input data
- Figure out what needs to be calculated
- Design your threads and thread blocks
- Implementation on CPU and GPU
- Built in check points?
- Output data

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 in this order:
  - 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



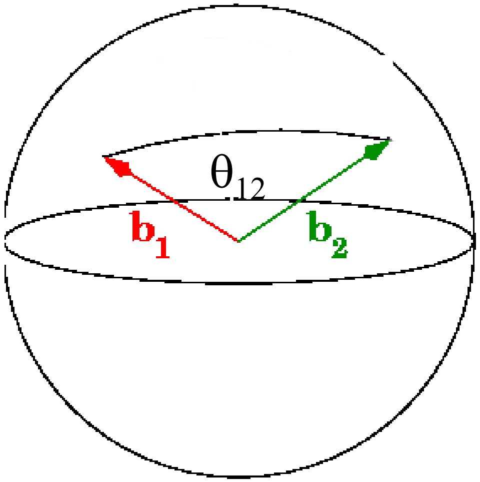
Input Data in moodle

Real data

```
100000
4646.98 3749.51
4644.35 3749.52
4643.24 3748.67
4646.98 3750.89
4641.13 3748.23
4646.75 3750.26
4643.29 3751.99
4640.78 3747
4638.95 3749.74
4647.29 3749.82
4651.65 3749.02
4649.65 3747.16
4646.66 3752.23
4649.38 3749.14
4648.22 3750.32
4639.27 3747.96
4637.3  3749.5
4649.07 3751.94
4640.39 3752.43
```

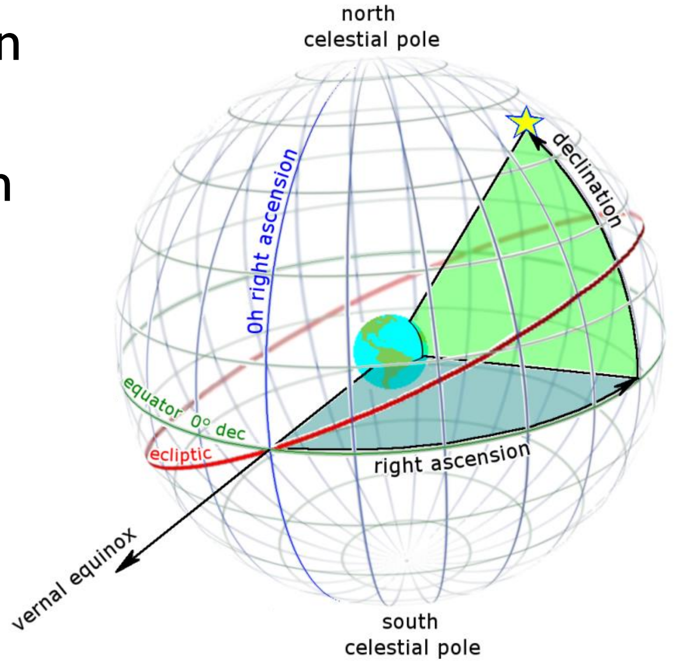
Synthetic data

```
100000
840.961426 387.991697
387.368692 2967.285746
2667.070581 3385.866638
3942.141923 3720.545649
5164.800068 3205.065003
1680.286209 653.214854
3553.889989 152.983872
1011.581615 76.716702
298.341944 3752.723427
4044.605401 2943.458158
1030.720073 2342.109887
54.257097 3769.420145
2478.545632 336.072740
970.044311 3556.987037
840.791266 2610.257310
5272.628047 2441.450374
380.633015 3771.599991
5049.735705 4627.545008
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.
- With DD count the angle between each pair of real-real galaxy (which histogram bin?)
- With DR count the angle between each pair of real-random galaxy
- With RR count the angle between each pair of random-random galaxy
- Check point: how many entries in DR? In DD and RR?



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 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

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

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

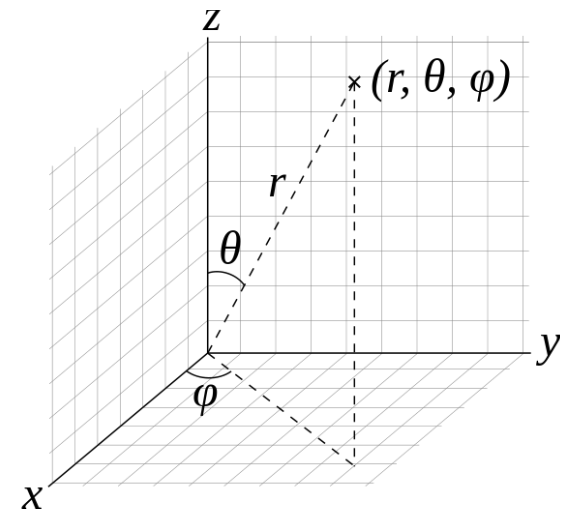
  $$\mathbf{r}_1 \bullet \mathbf{r}_2 = r_{1x} * r_{2x} + r_{1y} * r_{2y} + r_{1z} * r_{2z}$$

- Using the 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 (α_1,δ_1), (α_2,δ_2) is given by

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

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

Threads and Thread Blocks

- Choose your threads and thread blocks
- If you have the time, play around with the size of your thread block and the amount of work one thread does.
- Do you need to synchronize your threads?
- Do you need to use atomic operations?

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
  - calculate and present your results
- Start out with a small set of galaxies, perhaps with known relative angles?

Output Data

- If you have the time, plot the histograms DD and RR to see if there are any visible differences.
- The scientific measure for differences between the distributions of two equally big sets 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 the ω_i values are closer to zero than one, in the range $[-0.5,0.5]$, then D has approximately the same distribution as R, and we have a random distribution of real galaxies
- If the ω_i values are different from zero on the scale of one, then we have a non-random distribution of real galaxies

Short Instructions for dione.abo.fi

- Apply for an account and log on to dione:

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

- Load necessary modules:

```
module load cuda
module load GCC
```

- Compile your program `prog.cu`

```
nvcc -arch=sm_70 prog.cu -o a.out -lm
```

- Run your program `a.out` on the batch queue system

```
srun -p gpu --mem=1G -t=1:00:00 -o prog.out -e prog.err ./a.out real_data sim_data
```