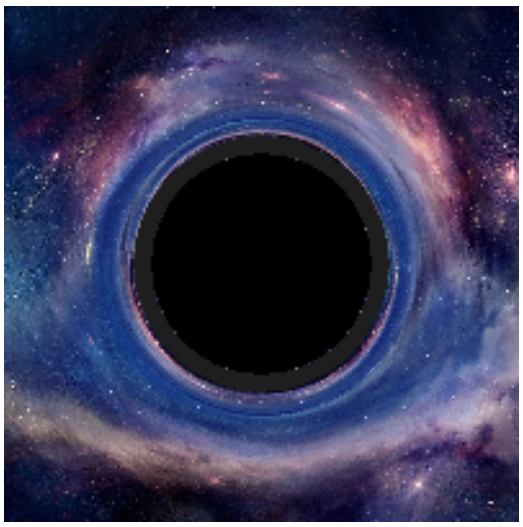
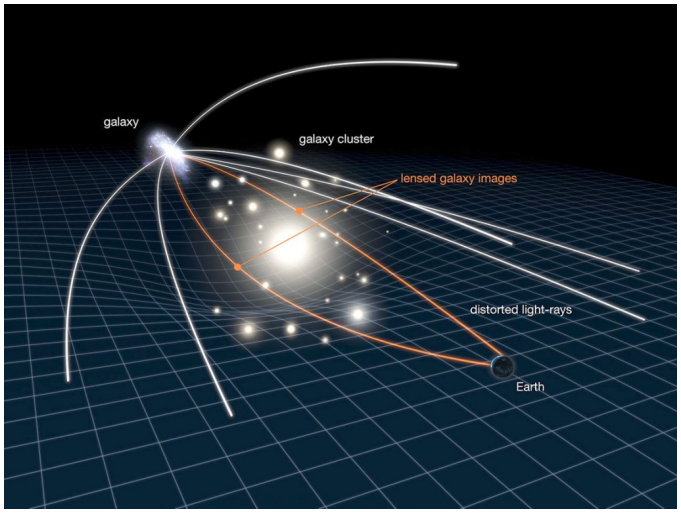
문제 2 : 블랙홀 이미징 (학부: 30 점, 대학원: 20 점)

문제 개요

일반 상대성에 의하면 빛은 중력에 의해 경로가 휘어진다. 특히, 블랙홀과 같은 고중력 천체의 주변을 지나는 빛은 그 휘어짐이 상당하여 관찰자에게 보이는 배경의 상은 심하게 왜곡될 것이다.





실제 블랙홀 근처로 가는 것은 거의 불가능에 가깝지만, 블랙홀 주변의 특정 지점으로 입사되는 다양한 방향의 광선 궤적(빛 다발)을 수치적으로 계산해봄으로서 그 지점에 위치한 가상의 관찰자에게 보일 왜곡된 상을 얻을 수 있다.

구체적으로 광선의 궤적은 일반 상대성에 따라 다음의 1차 상미분 방정식을 만족하며,

$$\frac{dx_i}{dt} = \frac{u_i}{u_0} \qquad (i = 1,2,3)$$

$$\frac{du_\mu}{dt} = -\frac{1}{u_0}\sum_{\alpha=0}^{3}\sum_{\beta=0}^{3}\Gamma_{\mu\alpha\beta}u_\alpha u_\beta \qquad (\mu = 0,1,2,3)$$

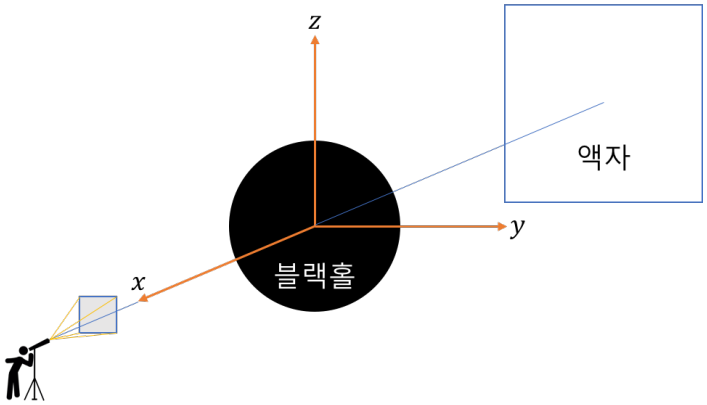
$\Gamma_{\mu\alpha\beta}(x_1, x_2, x_3)$: 블랙홀의 정보를 담고 있는 공간에 대한 함수 (주어짐)

방정식에 대한 수치해를 구하는 데에는 다음의 4차 룽게-쿠타 방법이 널리 사용된다.

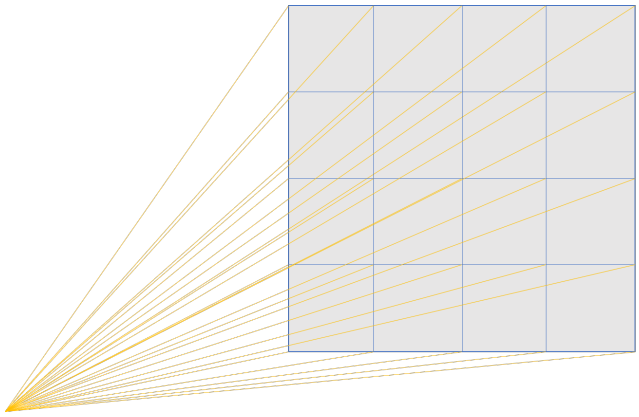
$$f(x) \equiv \frac{dx}{dt}$$
$$k_1 \leftarrow f(x)$$
$$k_2 \leftarrow f\left(x + \frac{1}{2}k_1\Delta t\right)$$
$$k_3 \leftarrow f\left(x + \frac{1}{2}k_2\Delta t\right)$$
$$k_4 \leftarrow f(x + k_3\Delta t)$$
$$x(t + \Delta t) \leftarrow x(t) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\Delta t$$

문제 설명

아래 그림과 같이 관찰자와 액자 사이에 블랙홀을 배치했을 때 관찰자에게 입사되는 다양한 방향의 광선 궤적을 4차 룽게-쿠타 방법으로 시뮬레이션 한다.



입사되는 광선의 방향은 관찰자 앞에 놓인 화면(회색 정사각형)에 들어오는 것만으로 한정한다. 화면의 해상도는 가로x세로 픽셀의 숫자로 주어지며 화면의 모든 픽셀 점 마다 해당 지점으로 입사되는 광선의 과거 궤적을 추적한다.



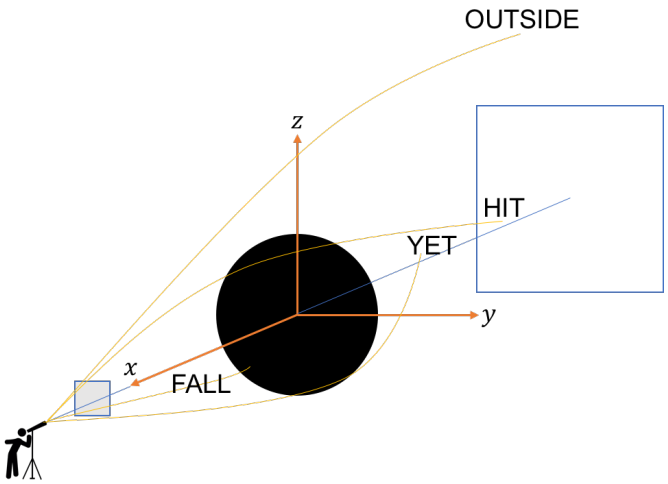
충분한 시간 간격 동안 추적한 뒤, 그 궤적에 따라 광선을 다음과 같이 4가지로 분류한다.

HIT : 액자를 관통한 경우

FALL : 블랙홀 근처에서 빠져나올 수 없는 경우

OUTSIDE : 실험실로부터 멀어져 더 이상 돌아오지 않는 경우

YET : 시뮬레이션 시간이 충분하지 않아 위의 3가지로 분류되지 않을 경우



한 광선에 대해 HIT, FALL, OUTSIDE가 결정이 되면 더 이상 추적할 필요가 없기 때문에 해당 광선은 시뮬레이션을 멈춘다. 모든 광선의 종류가 HIT, FALL, OUTSIDE 중 하나로 결정되었거나 지정한 시간 간격만큼 추적이 진행되었다면 시뮬레이션을 모두 멈추고 광선의 종류를 파일로 출력한다. HIT에 해당하는 광선의 경우 액자와 만난 지점의 y좌표와 z좌표를 파일로 출력한다.

지금까지 설명한 시뮬레이션 방법에 대한 순차 코드가 주어진다. 이 순차 코드와 같은 출력 결과를 내는 병렬 코드를 제출하라. 단, 다음에 제시된 유의사항을 위반하는 경우 제출한 병렬 코드의 성능과 관계없이 0점으로 평가됨에 유의한다.

참고 및 유의 사항

1. Makefile이 주어진다. 채점시 주어진 Makefile이 수정없이 사용된다.
2. 앞서 소개한 시뮬레이션 방법을 준수하는 범위 내에서 코드를 자유롭게 수정 할 수 있다.
3. 병렬화에 반드시 MPI가 사용되어야 한다.
4. 디버깅의 편의를 위해 다음의 기능을 가진 툴이 제공된다.
    A. 출력 파일 검증
        i. 사용방법 : $ ./tool data
        ii. 프로그램이 출력한 파일이 올바른 파일인지 검증한다.
    B. 출력 파일 비교
        i. 사용방법 : $ ./tool data1 data2
        ii. 두 출력 파일의 내용을 비교한다.

iii. 비교되는 내용은 다음과 같다.

1. 서로 다른 종류로 판별된 광선의 비율
2. 두 출력 파일 모두 HIT으로 판별된 광선들에 대한 y 좌표 차이, z 좌표 차이의 root mean square

C. 이미지 파일 생성

i. 사용방법 : $ ./tool data input_png_image output_png_image

ii. 출력 파일의 내용을 이용하여 input_png_image가 액자에 놓여있다고 가정했을 때 관찰자가 보는 상의 output_png_image를 생성한다.

iii. 이미지 파일은 input과 ouput 모두 png 형식만 가능하다. 툴이 input_png_image 파일을 읽지 못하는 경우 png 이미지의 프로파일을 일반 RGB로 변경한다.

5. 병렬 코드가 실행되면서 프로세스 마다 부하의 불균형이 발생할 수 있다. 이 경우 작업을 일찍 끝낸 프로세스가 다른 프로세스가 끝날 때 까지 기다려야 하므로 컴퓨팅 자원이 낭비된다. 이 문제를 해결할 수 있다면 더 좋은 병렬화 성능을 얻을 수 있을 것이다.

평가 방법

1. 제출한 병렬 코드를 주어진 Makefile 로 컴파일 한 뒤 대학원은 32 core(4 node x 8 core) , 학부는 20 core (1 node x 20 core) 에서 실행되는 시간을 time command 로 측정한다.
2. 주어진 순차 코드가 출력한 파일과 제출한 병렬 코드가 출력한 파일의 내용이 동일해야 한다. 다만, 코딩 방식의 차이에 의해 미세한 수치적 오차가 발생할 수 있기 때문에 다음 수준의 오차를 허용한다.
    a. 서로 다른 종류로 판별된 광선의 수 / 전체 광선의 수 $\leq 10^{-3}$
    b. 두 출력 파일 모두 HIT 으로 판별된 광선에 대해
        i. y 좌표 차이의 root mean square $\leq 10^{-10}$
        ii. z 좌표 차이의 root mean square $\leq 10^{-10}$
3. 위의 조건이 만족되지 않는 경우 정확성 미달로 결격 처리한다.