MONTE\_CARLO\_SYSTEM

Employ Monte Carlo Simulation to compute the area bounded by the x-axis in the interval [mMinX, mMaxX], and the curve of the function which is bounded y-interval [mMinY, mMaxY].

mMinX = -6;

mMaxX = 6;

mMinY = -5.0;

mMaxY = 5.0;

There are four functions:

f0(x) = 5 e-x\*x/4.0 cos(x)

f1(x) = cos(4x)sin(6x) - x/(1+x2)

f2(x) = -3cos(x2)+sin(6x)/( |x-4| + 0.01)

f3(x) = (ex –e-x)/( ex +e-x). This is hyperbolic tangent.

where e is the base of the natural log, which is approximated as 2.718281828.

The minimum number of samples points is 100.

The maximum number of samples points is 1,000,000.

The initial number of sample points is 100,000.

**Key usage**

‘<’: decrease the number of sample points by 10000 each time.

‘>’: increase the number of sample points by 10000 each time.

‘n’, ‘N’: set the number sample points to the minimum number.

‘m’, ‘M’: set the number sample points to the maximum number.

‘r’, ‘R’: reset the number sample points to its initial value.

‘i’, ‘I’: key usage

/\*

Handle the key events based on the key usage.

\*/

bool MONTE\_CARLO\_SYSTEM::handleKeyPressedEvent( int key )

{

……

computeSamples( );

computeArea( );

……

}

/\*

Return the estimated area

\*/

double MONTE\_CARLO\_SYSTEM::getArea() const

/\*

Reset the system state.

Steps:

compute the samples

compute the area

\*/

void MONTE\_CARLO\_SYSTEM::reset( )

/\*

Return the interval [mMinX, mMaxX] of x to [minX, maxX].

\*/

void MONTE\_CARLO\_SYSTEM::getXInterval( double &minX, double &maxX ) const

/\*

Uniformly generate all the random sample points

inside [mMinX, mMaxX] x [mMinY, mMaxY].

The number of sample points is mNumSamples.

\*/

void MONTE\_CARLO\_SYSTEM::computeSamples( )

/\*

Compute the area based on the Monte Carlo Simulation.

Steps:

1. compute the number of samples inside the area

2. compute the ratio which is equal to

the number of samples inside the area

divided

by the total number of samples

3. Based on the ratio, compute the area

4. Display the area value

5. return the area value

\*/

void MONTE\_CARLO\_SYSTEM::computeArea( )

{

……

}

/\*

Return the number of sample points.

\*/

int MONTE\_CARLO\_SYSTEM::getNumSamples( ) const

/\*

Return the coordinates (x,y) of the sample point with index sampleIndex

\*/

void MONTE\_CARLO\_SYSTEM::getSample(int sampleIndex, double &x, double &y ) const

/\*

Based on the function type of the function,

compute the function value for a given value x.

Return the function value.

\*/

double MONTE\_CARLO\_SYSTEM::getValue(double x) const

/\*

Given a point (x,y).

Check if the point lies inside the area.

Return true if it lies inside the area.

Return false otherwise.

The area is defined as the region bounded by

the x-axis and the curve of the function f(x).

Assume that (x,y) lies inside area, then

one of the following conditions must be satisfied.

- If f(x) >= 0 and y >=0, then y <= f(x).

- If f(x) <= 0 and y <=0, then y >= f(x).

Otherwise, (x,y) does not lay inside the area.

\*/

bool MONTE\_CARLO\_SYSTEM::isInsideArea( double x, double y ) const