

Sprawozdanie z projektu: Całkowanie Monte Carlo przy użyciu klasy TFoam z Pakietu ROOT

Justyna Peciak Teresa Rodak

10 lutego 2016

1 Wstęp

Niniejszy raport został sporządzony w celu przedstawienia całkowania Monte Carlo przy użyciu klasy TFoam z Pakietu ROOT. Dokument zawiera treści opisujące bibliotekę ROOT w sposób ogólny (jak również sposób jej instalacji), klasę TFoam, oraz opis całkowania Metodą Monte Carlo. Na końcu został umieszczony opis programu napisanego w języku c++, który to wykorzystuje klasę TFoam do całkowania Monte Carlo.

2 Informacje ogólne

2.1 Całkowanie Metodą Monte Carlo

Załóżmy, że chcemy obliczyć całkę z funkcji $f(x)$ w przedziale $\langle xp; xk \rangle$. Definicja całki znaczonej mówi, że wartość całki równa jest polu obszaru pod wykresem krzywej w zadanym przedziale całkowania. Ustalamy że wartość funkcji w obszarze całkowania mieszczą się w przedziale $\langle yp; yk \rangle$. Pole prostokąta wyznaczonego przez przedział całkowania: $\langle xp; xk \rangle$ oraz zakres wartości funkcji w tym przedziale: $\langle yp; yk \rangle$ jest prosty do wyznaczenia i wynosi:

$$P = |xk - xp| * |yp - yk|$$

Metoda Monte Carlo polega na wylosowaniu n punktów znajdujących się w obrębie wspomnianego prostokąta. Przybliżoną wartość całki wyznaczamy z proporcji: stosunek pola pod krzywą do pola prostokąta jest równy stosunkowi liczby punktów pod krzywą do wszystkich wylosowanych punktów. Błędy całkowania maleją odwrotnie proporcjonalnie do pierwiastka z liczby próbek:

$$1/\sqrt{N}$$

Dokładność wyniku uzyskanego tą metodą jest zależna od liczby sprawdzeń i jakości użytego generatora liczb pseudolosowych. Zwiększanie liczby prób nie zawsze zwiększa dokładność wyniku, ponieważ generator liczb pseudolosowych ma skończenie wiele liczb losowych w cyklu.

2.2 Biblioteka ROOT

Obecne eksperymenty fizyczne zbierają ogromne ilości danych. Danych tych nie da się analizować ręcznie choćby ze względu na ich ilość dlatego do ich obróbki wykorzystuje się komputery.

W CERN-ie (Europejska Organizacja Badań Jądrowych) opracowano więc narzędzie, które miało ułatwić analizowanie danych. Narzędziem tym jest właśnie ROOT czyli platforma programistyczna na której opierają się najważniejsze eksperymenty fizyki wysokich energii (min. eksperymenty przy LHC - wielki zderzacz hadronów).

Zalety ROOT'a:

- Mniejsza ilość kodu do zapisania - wykorzystuje się dotychczasowe elementy kodu
- Kod jest bardziej niezawodny - elementy pochodzące ze szkieletu zostały już wcześniej przetestowane.
- Biblioteka pozwala na koncentrowanie się na konkretnych zagadnieniach - interfejs, grafika oraz połączenia pomiędzy elementami są dostarczane w ramach szkieletu

ROOT składa się z kilku części:

- Biblioteki c++ - dostarczają one gotowe klasy, które umożliwiają pisanie gotowych programów w c++ dzięki czemu można w znaczny sposób skrócić czas pracy. W środowisku ROOT dopasowanie funkcji, tworzenie histogramu, ustawienie różnych opcji rysowania itp. można zrobić zaledwie w kilku liniach kodu.
- Interpreter CINT - można uruchomić wpisując w terminalu polecenie root. Umożliwia uruchamianie skrypty napisanych z biblioteki ROOT'a. Można też napisać własny kawałek kodu i go szybko uruchomić.
- Kompilator ACLiC - jest to kompilator, który z makr potrafi stworzyć pliki współdzielone.

Biblioteka zawiera bardzo dużo klas:

- TObject - definiuje domyślne ustawienia dla wszystkich obiektów w ROOT
- TGraph, TGraphErrors, TGraph2D, TMultiGraph - klasy związane z wykresami
- TRandom1, TRandom2, TRandom3 - klasy generujące liczby losowe (według rozkładu 1,2,3-d)
- TMath - klasa z funkcjami matematycznymi
- TFoam - symulator generatora liczb losowych.

2.3 Klasa TFoam

TFoam to ogólnego przeznaczenia "komórkowy" generator zdarzeń - liczb losowych. Wysoką wydajność generatora MC uzyskuje się poprzez podzielenie dziedziny całkowania na małe komórki. Komórki mogą być n-wymiarowe. Siatka komórek nazywana jest "pianą"(ang. foam) wtrwarzana jest w procesie binarnego podziału komórek.

Przykładowe metody klasy

- `TFoam::MakeEvent` - generuje losowy punkt / wektor według rozkładu zdefiniowanego przez użytkownika.
- `TFoam::TRandom` - to podstawowa klasa ROOT'a do generowania liczb losowych.
- `TFoam::GetMCvect` - przy pomocy tej metody użytkownik może uzyskać punkt/wektor wygenerowany metoda Monte Carlo.
- `TFoam::GetIntegMC` - zwraca wartość całki obliczonej metodą Monte Carlo.
- `TFoamIntegrand` - funkcja całkująca.

3 Sposób instalacja biblioteki ROOT

3.1 Instalacja - Krok 1

Pierwszym krokiem przy instalacji ROOT'a jest pobranie ze strony <https://root.cern.ch/downloading-root> paczki z plikami wykonywalnymi oraz należy je rozpakować w katalogu docelowym.

- `gunziproot_v6.04.14.Linux - ubuntu14 - x86_64 - gcc4.8.tar.gz`
- `tarxvfroot_v6.04.14.Linux - ubuntu14 - x86_64 - gcc4.8.tar`

Następnie pobieramy kod źródłowy. Najlepiej pobrać najnowszą wersję. Paczkę należy rozpakować w tym samym katalogu.

- `tarzxvfroot_v2.25.xx.source.tar.gz`

Przechodzimy do tego katalogu i upewniamy się że znajduje się w nim plik Makefile.

3.2 Instalacja - Krok 2

Aby poprawnie zainstalować ROOT'a musimy jeszcze pobrać dodatkowe paczki. Można zrobić to poleceniem:

- `sudoapt-getinstallbuild-essentialgfortransubversionxorg-devlibxml2-devlibmysqlclient-devlibftw3-devlibssl-devlibglu1-mesa-devautomakeautoconflibtoolcurllibncurses5-devbinutilslibX11-devlibXpm-devlibXft-devlibXext-dev`

Czasem trzeba doinstalować dodatkowe biblioteki zgodnie z informacjami wyświetlanymi w terminalu.

3.3 Instalacja - Krok 3

Kolejną niezbędną czynnością jest ustawienie zmiennych środowiskowych. Są one niezbędne do poprawnej kompilacji i prawidłowego działania programu. Aby to zrobić otwieramy plik `.bashrc`, który znajduje się w katalogu domowym `home/`:

- `gedit /.bashrc`

i na końcu tego pliku dopisujemy:

```
export ROOTSYS=/opt/root
export PATH=$PATH:$ROOTSYS/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib
```

3.4 Instalacja - Krok 4

Teraz możemy już skonfigurować instalację komendą

- `./configure`

Kiedy już skonfigurujemy ROOT'a możemy przejść do kompilacji.

W tym celu wystarczy wpisać w konsoli:

- `make`

Kompilacja jest dość długim procesem - może trwać ponad godzinę.

Na koniec wyświetla się informacja aby przed uruchomieniem ROOT'a wykonać polecenie:

- `.../root/bin/thisroot.sh`

3.5 Instalacja - Krok 5

Przy wpisaniu w konsolę komendy `root` włącza się program. Jeśli nie chcemy żeby wyświetlał się



ekran startowy ROOT'a przy wpisywaniu komendy `root` w konsoli należy dodać `-l`.

4 ROOT w praktyce

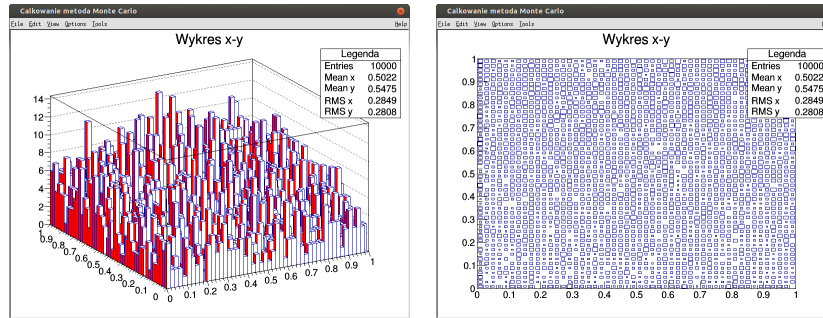
4.1 Kod programu głównego

```
using namespace std;

int main()
{
    cout<<"--- Tworzenie wykresu ---"<<endl;
    TH2D *hst_xy = new TH2D("Legenda", "Wykres x-y", 100,0,1.0, 100,0,1.0); //Definicja parametrów wykresu -
                                                                    (Legenda, Tytuł wykresu,
                                                                    wymiary osi x i y)

    Double_t *MCvect = new Double_t[2]; //Dwuwymiarowy wektor generowany przez MC run.
    TRandom *PseRan= new TRandom3(); //Tworzy generator liczb losowych.
    PseRan->SetSeed(4357);
    TFoam *FoamX = new TFoam("FoamX"); //Tworzy symulator generatora liczn losowych.
    FoamX->SetkDim(2); //Liczba wymiarów
    FoamX->SetnCells(500); //Liczba celi (domyślnie jest to 2000)
    FoamX->SetRhoInt(Funkcja); //Ustawia dwuwymiarowy rozkład.
    FoamX->SetPseRan(PseRan); //Ustawia generator liczb losowych.
    FoamX->Initialize(); //Inicjalizuje symulator (może to chwilę potrwać).

    // Tworzenie okna do wyświetlenia wykresu.
    TCanvas *cKanna = new TCanvas("cKanna", "Calkowanie metoda Monte Carlo", 800,600); //Tworzy okno
                                                                    wyświetlające wykres
                                                                    ("klasa", "Tytuł", wymiary okna)
    cKanna->cd();
}
```

Rysunek 2: Typy wykresów

Funkcja, która była poddana całkowaniu:

```
#include "TMath.h"

double Funkcja(int nDim, double *Xarg) ///Dwuwymiarowy rozkład cel.
{
    double x=Xarg[0];
    double y=Xarg[1];
    double Fun=0;
    Fun +=TMath::Sin(x+1.0)*TMath::Cos(y-1.0);
    return Fun;
}
```

5 Podsumowanie

Program ROOT służy do przechowywania, analizy i prezentacji danych. Oparty jest o programowanie obiektowe, więc pełne zrozumienie tego programu możliwe jest dopiero po opanowaniu podstaw języka C++. Osobom, które nie posiadają zaawansowanej wiedzy na temat programowania oraz znajomości systemu operacyjnego Linux, już sama instalacja ROOT'a może przysporzyć wiele problemów. Ponadto instalacja jest czasochłonna.

Literatura

- [1] <https://root.cern.ch/>
- [2] <https://root.cern.ch/doc/master/annotated.html>
- [3] <https://root.cern.ch/doc/master/classTFoam.html>
- [4] <http://arxiv.org/abs/physics/0210061>