

YMAP ROOT講習会 2023 第6回 (前半)：動くプロット

2023-06-21

Tatsuki Washimi (NAOJ)

自己紹介

鷺見貴生 (ワシミタツキ)

1990年 千葉県生まれ

2009年4月～2013年3月 早稲田大学 応用物理学科

2013年4月～2018年3月 同 修士課程、博士課程

2018年4月～2018年12月 早稲田大学 応用物理学科 助手

2018年12月20日 博士(理学) 取得

ANKOK実験 (暗黒物質探索)

2019年1月～2019年3月 高エネルギー加速器研究機構 研究員

2019年4月～2022年3月 国立天文台 重力波プロジェクト 学振PD

2022年4月～現在 同 特任助教



KAGRA (重力波観測、主に環境雑音対策)

TGraphのアニメーション

例: 2次元ランダムウォーク

animation_tgraph.cc

ここはplotの枠を作るだけ

①まずTGraphを定義し、
値を詰める前にDrawする

②ループの中で1点1点値を
詰めていく

③CanvasをUpdateする

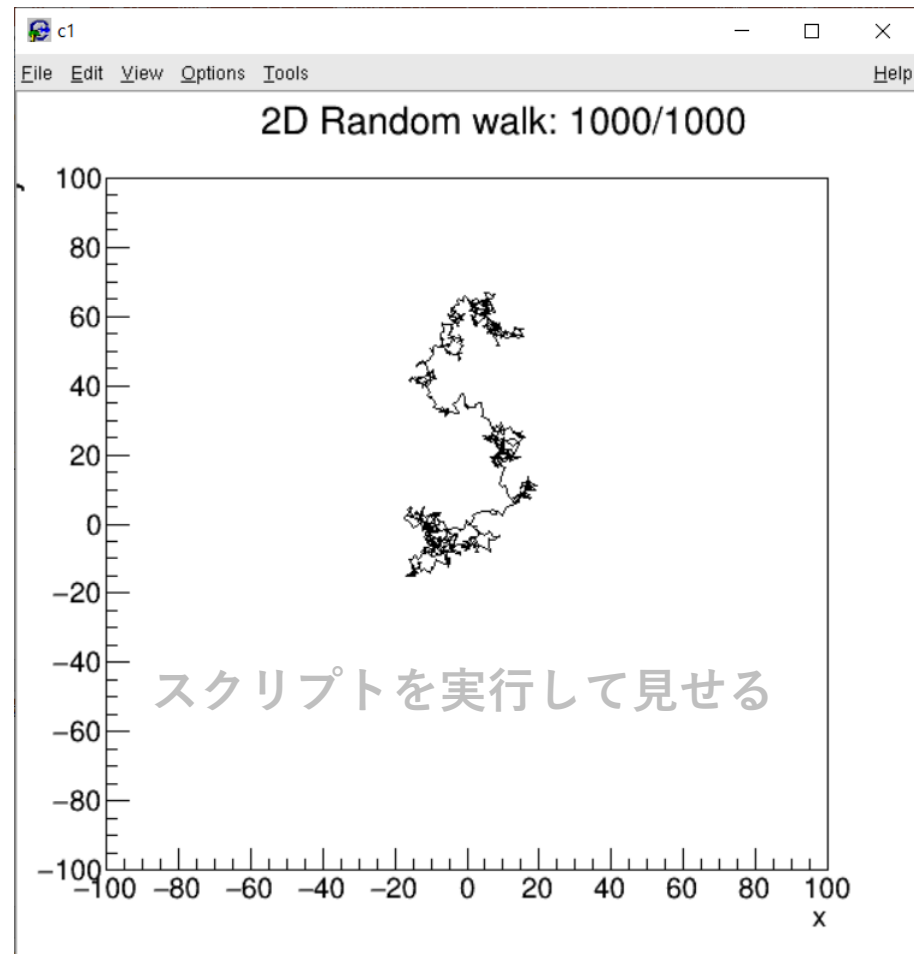
```
#include <string>
#include <TR00T.h>
#include <TStyle.h>
#include <TH2.h>
#include <TGraph.h>
#include <TCanvas.h>

void animation_tgraph(){
    TCanvas *c1 = new TCanvas("c1","c1", 600,600);
    TH2D *h2 = new TH2D("h2","2D Random walk;x;y",101,-100,100,101,-100,100);
    gStyle->SetOptStat(0);
    c1->Draw();
    h2->SetLineWidth(0);
    h2->Draw();

    TGraph *g1 = new TGraph();
    float x = 0;
    float y = 0;
    g1->SetPoint(1,x,y);
    g1->Draw("pl");

    int imax=1000;
    for(int i=0;i<imax+1;i++){
        x += gRandom->Gaus();
        y += gRandom->Gaus();
        g1->SetPoint(i+2,x,y);
        h2->SetTitle(Form("2D Random walk: %i/%i;x;y",i,imax));
        c1->Update();
    }
}
```

TGraphのアニメーション



スクリプトを実行して見せる

animation_tgraph.cc

```
#include <string>
#include <TR00T.h>
#include <TStyle.h>
#include <TH2.h>
#include <TGraph.h>
#include <TCanvas.h>

void animation_tgraph(){
    TCanvas *c1 = new TCanvas("c1","c1", 600,600);
    TH2D *h2 = new TH2D("h2","2D Random walk;x;y",101,-100,100,101,-100,100);
    gStyle->SetOptStat(0);
    c1->Draw();
    h2->SetLineWidth(0);
    h2->Draw();

    TGraph *g1 = new TGraph();
    float x = 0;
    float y = 0;
    g1->SetPoint(1,x,y);
    g1->Draw("pl");

    int imax=1000;
    for(int i=0;i<imax+1;i++){
        x += gRandom->Gaus();
        y += gRandom->Gaus();
        g1->SetPoint(i+2,x,y);
        h2->SetTitle(Form("2D Random walk: %i/%i;x;y",i,imax));
        c1->Update();
    }
}
```

mean = 0, std dev = 1 の
Gaussianに従って歩幅
を決めている

②ループの中で1点1点値を
詰めていく

③CanvasをUpdateする

GIFアニメとして保存する方法

```
void animation_tgraph(){
    TCanvas *c1 = new TCanvas("c1", "c1", 600,600);
    TH2D *h2 = new TH2D("h2", "2D Random walk;x;y", 101,-100,100,101,-100,100);
    gStyle->SetOptStat(0);
    c1->Draw();
    h2->SetLineWidth(0);
    h2->Draw();
    gSystem->Unlink("animation_tgraph.gif");

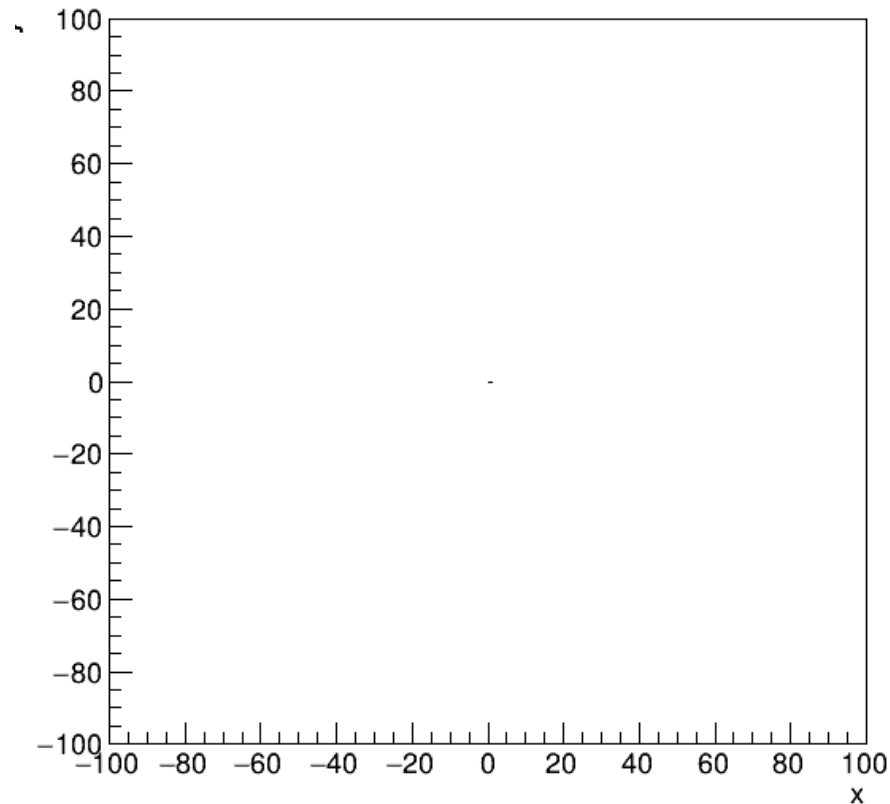
    TGraph *g1 = new TGraph();
    float x = 0;
    float y = 0;
    g1->SetPoint(1,x,y);
    g1->Draw("pl");

    int imax=1000;
    for(int i=0;i<imax+1;i++){
        x += gRandom->Gaus();
        y += gRandom->Gaus();
        g1->SetPoint(i+2,x,y);
        h2->SetTitle(Form("2D Random walk: %i/%i;x;y",i,imax));
        c1->Update();
        c1->Print("animation_tgraph.gif+");
    }
}
```

この2行を追加する

animation_tgraph.cc

2D Random walk: 0/1000



ただし、保存しない場合と比べて実行速度が非常に遅くなるので、普段使いにはおススメしない

- バグチェック等が済んで完成したスクリプトに対して、プレゼン用の1例として使うなど
- -b オプション(バッチモード)を付けて描画せずに走らせると、幾分速くなる

```
$ root -l -b -q animation_tgraph.cc
```

TH1Dのアニメーション

例: 2次元ランダムウォーク

①まずTH1Dを定義し、値を詰める前にDrawする

②ループの中で1点1点値を詰めていく

③CanvasをUpdateする

```
#include <string>
#include <TR00T.h>
#include <TStyle.h>
#include <TH2.h>
#include <TH1.h>
#include <TCanvas.h>

void animation_th1d(){
    TCanvas *c1 = new TCanvas("c1","c1", 800,600);
    gStyle->SetOptStat(0);
    c1->Draw();

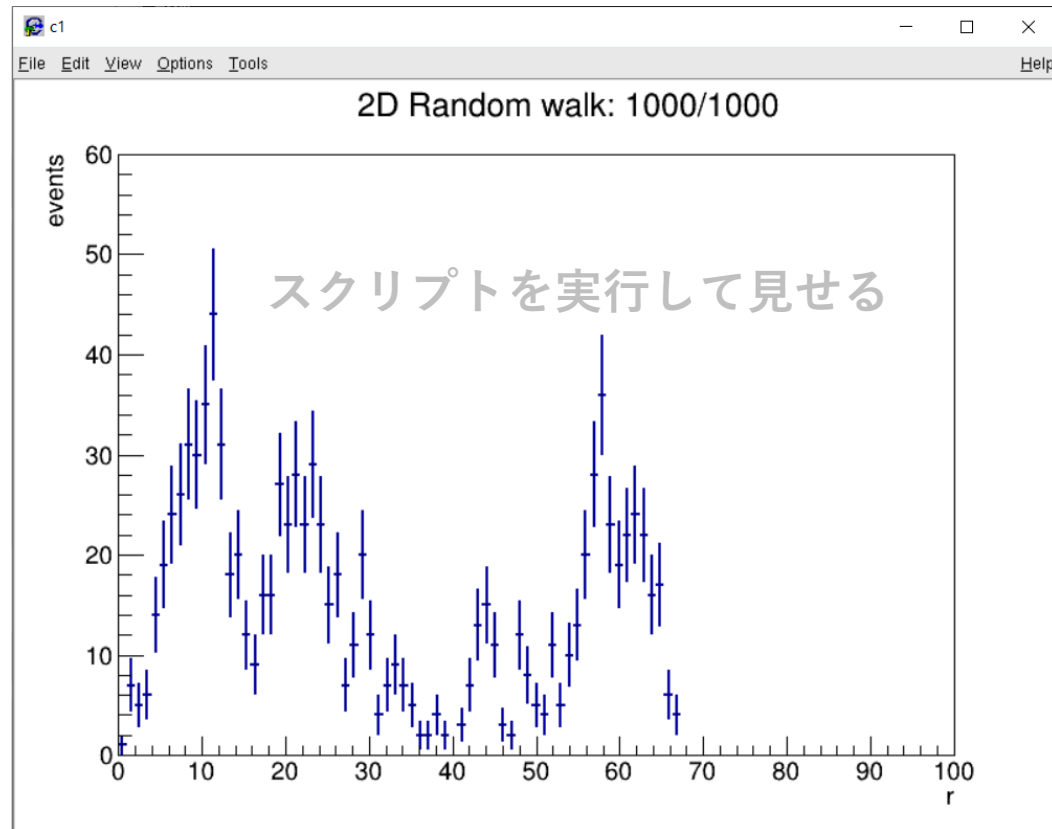
    TH1D *h1 = new TH1D("h1","2D Random walk;r;events",101,0,100);
    float x = 0;
    float y = 0;
    float r = 0;
    h1->SetLineWidth(2);
    h1->Draw("e");
    h1->GetYaxis()->SetRangeUser(0,60);

    int imax=1000;
    for(int i=0;i<imax+1;i++){
        x += gRandom->Gaus();
        y += gRandom->Gaus();
        r = pow(x*x+y*y,0.5);
        h1->Fill(r);
        h1->SetTitle(Form("2D Random walk: %i/%i;r;events",i,imax));
        c1->Update();
    }
}
```

animation_th1d.cc

TH1Dのアニメーション

例: 2次元ランダムウォーク



③CanvasをUpdateする

```
#include <string>
#include <TR00T.h>
#include <TStyle.h>
#include <TH2.h>
#include <TH1.h>
#include <TCanvas.h>

void animation_th1d(){
    TCanvas *c1 = new TCanvas("c1","c1", 800,600);
    gStyle->SetOptStat(0);
    c1->Draw();

    TH1D *h1 = new TH1D("h1","2D Random walk;r;events",101,0,100);
    float x = 0;
    float y = 0;
    float r = 0;
    h1->SetLineWidth(2);
    h1->Draw("e");
    h1->GetYaxis()->SetRangeUser(0,60);

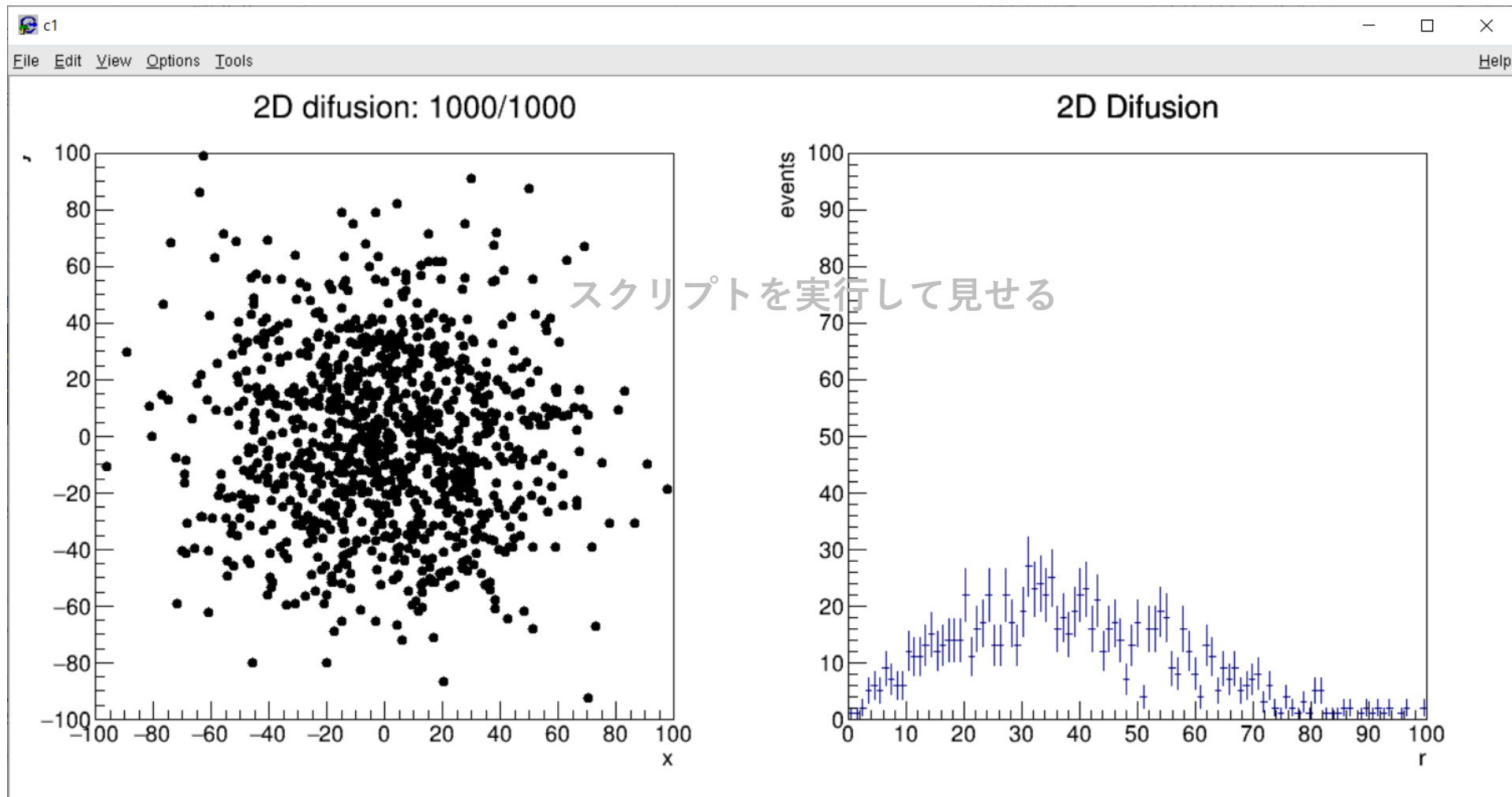
    int imax=1000;
    for(int i=0;i<imax+1;i++){
        x += gRandom->Gaus();
        y += gRandom->Gaus();
        r = pow(x*x+y*y,0.5);
        h1->Fill(r);
        h1->SetTitle(Form("2D Random walk: %i/%i;r;events",i,imax));
        c1->Update();
    }
}
```

animation_th1d.cc

始点からの距離を計算して
ヒストグラムに詰める

応用例：粒子の拡散

- TGraphとTH1Dを同時にアニメーションにする
- 多数の粒子(今回は1000個)を同時に動かす



応用例：粒子の拡散

- TGraphとTH1Dを同時にアニメーションにする
- 多数の粒子(今回は1000個)を同時に動かす

```
void difusion(){
    TCanvas *c1 = new TCanvas("c1","c1", 1200,600);

    c1->Divide(2,1);
    c1->cd(2);
    TH1D *h1 = new TH1D("h1","2D Difusion;r;events",101,0,100);
    h1->Draw("e");
    h1->GetYaxis()->SetRangeUser(0,100);

    c1->cd(1);
    TH2D *h2 = new TH2D("h2","2D Difusion;x;y",101,-100,100,101,-100,100);
    gStyle->SetOptStat(0);
    c1->Draw();
    h2->SetLineWidth(0);
    h2->Draw();

    TGraph *g1 = new TGraph();
    const int N = 1000;
    float x[N] = {0};
    float y[N] = {0};
    float r = 0;
```

difusion.cc

1000個の粒子

CanvasをDivideして描画する場合、ヒストグラムは毎回Drawし直さないと(最後しか)更新されない。

```
int imax=1000;
for(int i=0;i<imax+1;i++){
    h1->Reset();
    for(int j=0;j<N;j++){
        x[j] += gRandom->Gaus();
        y[j] += gRandom->Gaus();
        r = pow(x[j]*x[j] + y[j]*y[j],0.5);
        h1->Fill(r);
    }

    delete g1;
    TGraph *g1 = new TGraph(N,x,y);
    c1->cd(1);
    g1->SetMarkerStyle(8);
    g1->Draw("p");
    h2->SetTitle(Form("2D difusion: %i/%i;x;y",i,imax));

    c1->cd(2);
    h1->Draw("e");

    c1->Update();
}
}
```

グラフの点が追加されるのではなく丸々置き換わるので、いちいち消して作り直す必要がある

動くプロットの使いどころ

- ◆自分で書いたシミュレーションの経過を見る(今回紹介した例)
 - ・ 望み通りの挙動をしているのか確認、バグチェック
 - ・ プレゼンテーション用アニメーション
- ◆実験中のモニタープロット
 - ・ イベントディスプレイ (レートが高い場合は適当に間引いて)
 - ・ 統計データや環境モニタのディスプレイ
- ◆GUIアプリを作る
 - ・ 例えば重力波業界では、ROOTで作られた”diaggui”というツールが使われています

注意：今回紹介した方法はあくまで小技です。おそらく公式が意図した方法ではないので、挙動が不安定(途中で止めるとターミナルが固まるなど)だったり、実行環境によって使えなかったりします。

真面目にやる場合は、この辺りを参照してください。

- ・ <https://root.cern.ch/root/html/doc/guides/users-guide/WritingGUI.html>
- ・ https://root.cern/doc/master/anim_8C.html