確率微分方程式
数値計算プログラム

2017年4月27日

北海道大学理学院数学専攻1年
毛利　光希

本プログラムの目的

不確実なリスクを持つ金融機関に預けたリスク資産の価値X(t)は以下の方程式で表されます。

$$dX_t = \mu X_t \, dt + \sigma X_t \, dW_t$$

これはブラック・ショールズモデルにおけるリスク資産として知られ、このような確率微分方程式の解の挙動を可視化することなどを目的としたプログラムです。

確率微分方程式

確率微分方程式

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t$$

この形で表される確率微分方程式を、擬似乱数を用いた数値計算プログラムで数値解を出す。

（aとbは滑らかな実数値関数である）

２つの近似法

Euler-Maruyama scheme

$$Y_{n+1} = Y_n + a(t_n, Y_n)\Delta t_n + b(t_n, Y_n)\Delta W_n$$

強収束次数が1/2（1/2次の精度）

Milstein scheme

$$Y_{n+1} = Y_n + a(t_n, Y_n)\Delta t_n + b(t_n, Y_n)\Delta W_n + \frac{1}{2}\frac{\partial b}{\partial Y_n} b(t_n, Y_n)(\Delta W_{t_n}^2 - \Delta t_n)$$

強収束次数が1 (1次の精度)

注意

※Milstein schemeの方が精度が高いですが、偏微分を求める必要があります。

使用方法（テスト）

1. 「EM_scheme.c」または「Mil_scheme.c」のどちらかと、「mt19937-64.c」を同時にコンパイルします。

   →$ cc　○○_scheme.c　mt19937-64.c　-lm

2. 生成されたa.outを実行します。

   →$ ./a.out > a.dat

3. Gnuplotなどでa.datを可視化します。

   →gnuplot > plot “a.dat” with lines

使用方法（メイン）

4. “EM_scheme.c”または”Mil_scheme.c”をテキストエディタなどで開きます。

5. Function_A_dt と Function_B_dW の部分を任意の形に書き換えます。

（”Mil_scheme.c”を利用する場合はFunction_dB_dW（偏微分）も書き換えます。）

6. 1～3の過程に戻ります。

使用方法（応用）

7. Xtの初期条件はメイン関数の中(X0)で定義されています。適宜変更してください。

8. NはとあるT時間Tに対する時間の分割数です。Nを大きくすると、全体の時間はそのままに刻み幅が小さくなります。

（※この時確率変数の数が増えるためグラフの形が変わったように見えます。）

9. 解析解との比較をする場合は後述。

追加事項

1. 方程式の離散化と乱数の発生方法

2. 実際に計算してみた例

3. 解析解との比較

ブラウン運動の生成

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t$$

離散化

時間分割 $0 = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = T$

初期条件 $Y_0 = X_0$

$$Y_{n+1} = Y_n + a(t_n, Y_n)\Delta t_n + b(t_n, Y_n)\Delta W_n$$

（Euler-Maruyama scheme）

ただし　$\Delta t_n = t_{n+1} - t_n, \Delta W_n = W_{t_{n+1}} - W_{t_n}$

ブラウン運動の生成

$$\Delta t_n = \delta t\ (\text{一定})$$

ΔW_n これを乱数を用いて近似する

ブラウン運動の性質から ΔW_n はそれぞれ独立で、
$\Delta W_n \in N(0, \Delta t_n)$ を満たしている

メルセンヌ・ツイスタの一様乱数から正規分布に従う乱数を生成する

ブラウン運動の生成

Polar法を用いたアルゴリズム

２つの一様乱数

→２つの正規分布に従う確率変数

$N(\mu, \sigma^2)$ に従う確率変数ができる

$\mu = 0, \sigma = \sqrt{\Delta t_n}$

実例

簡単な確率微分方程式を解いてみる。

$$dX_t = (X_t)\ dt + (\sqrt{2}X_t)\ dW_t$$

$$X_0 = 1.0$$

この確率微分方程式の解析解は

$$X_t = \exp(\sqrt{2}W_t)$$

だとわかっている。これと数値解を比較する。

（ここで解析解は確率過程であり、一意に定まるものではないため、次ページのグラフは一例であることにご注意ください。）

Euler-Maruyama scheme

T = 1.0

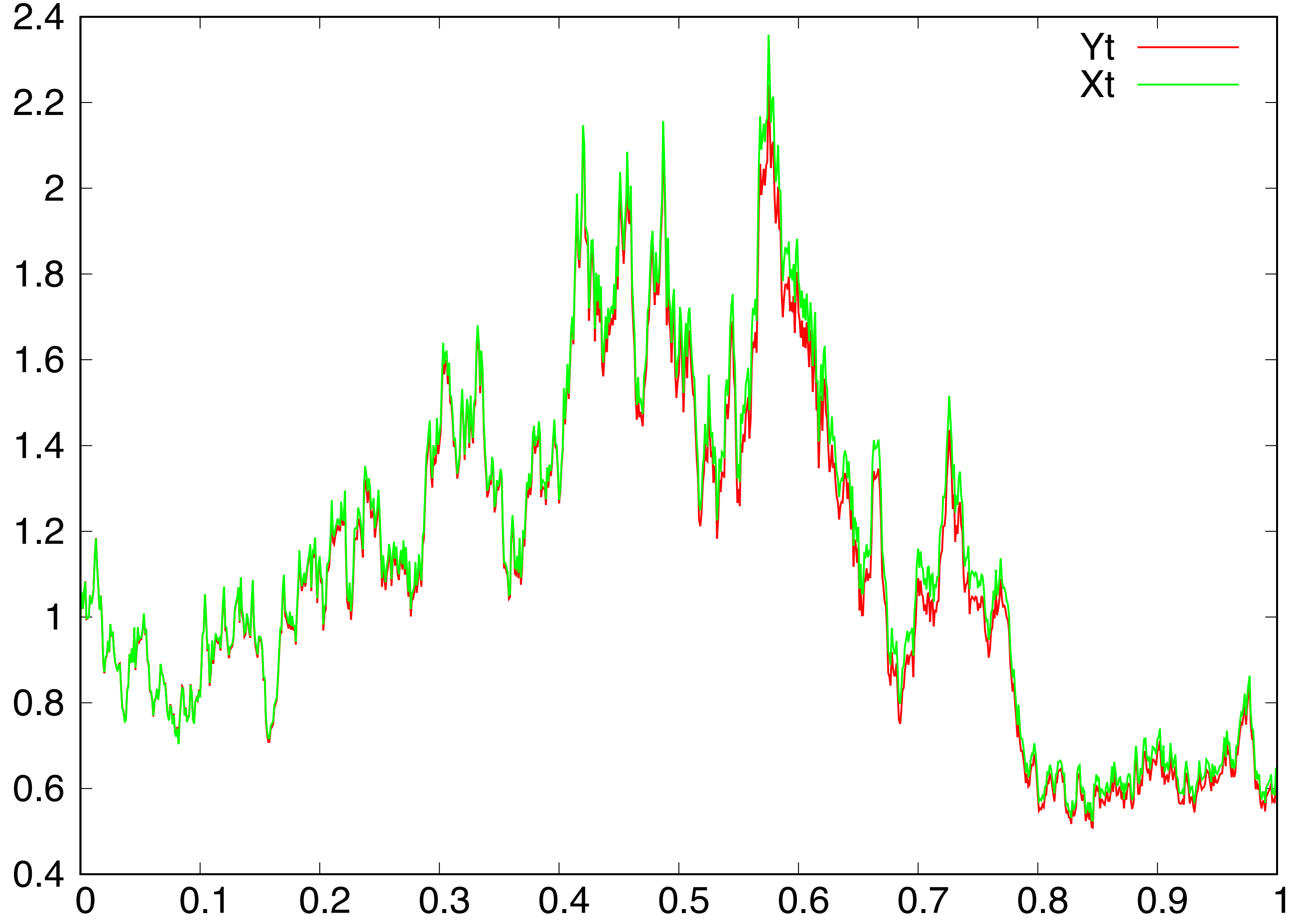
（時間幅）

N = 1000

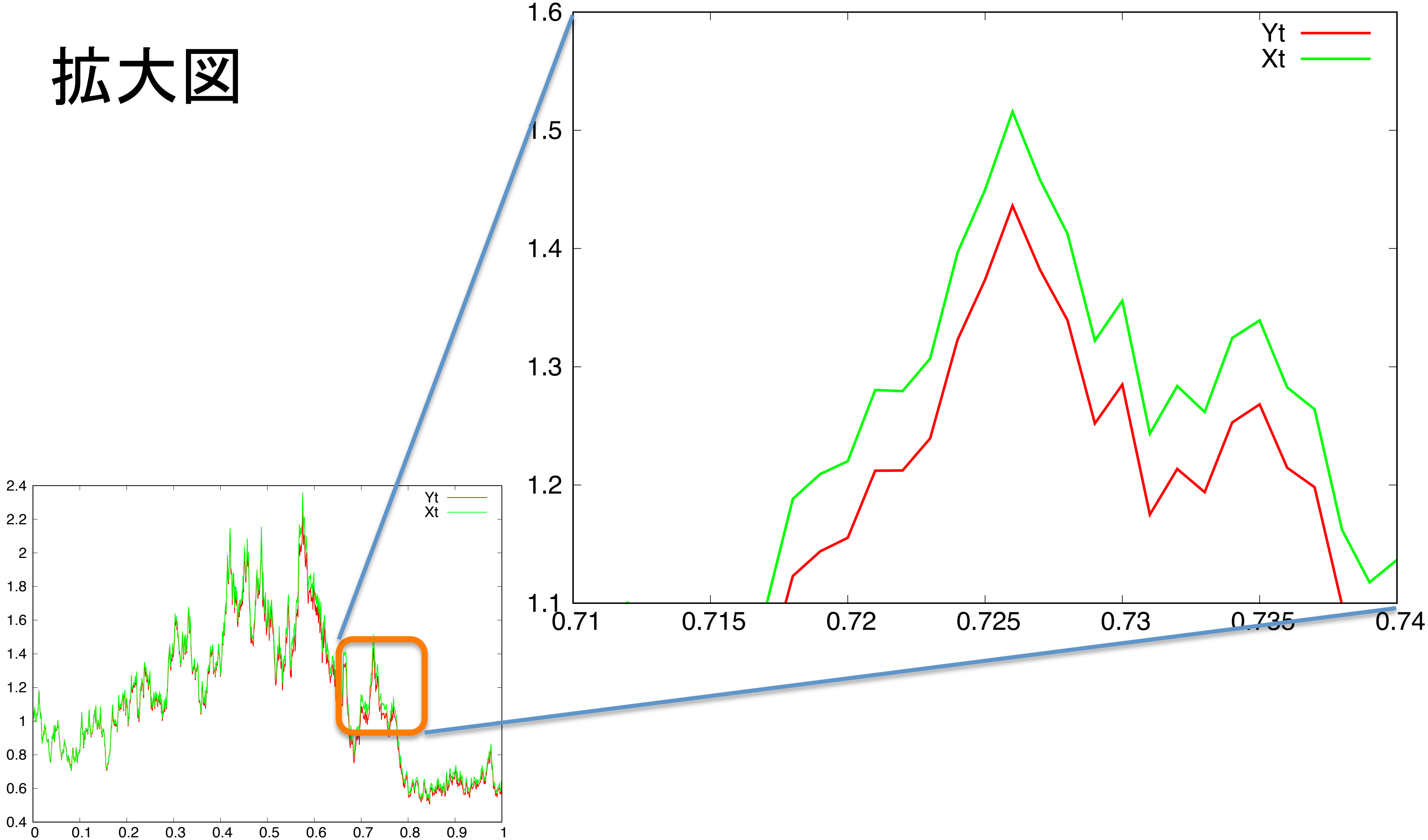
（時間刻み数）

赤 : 数値解

緑 : 解析解



拡大図

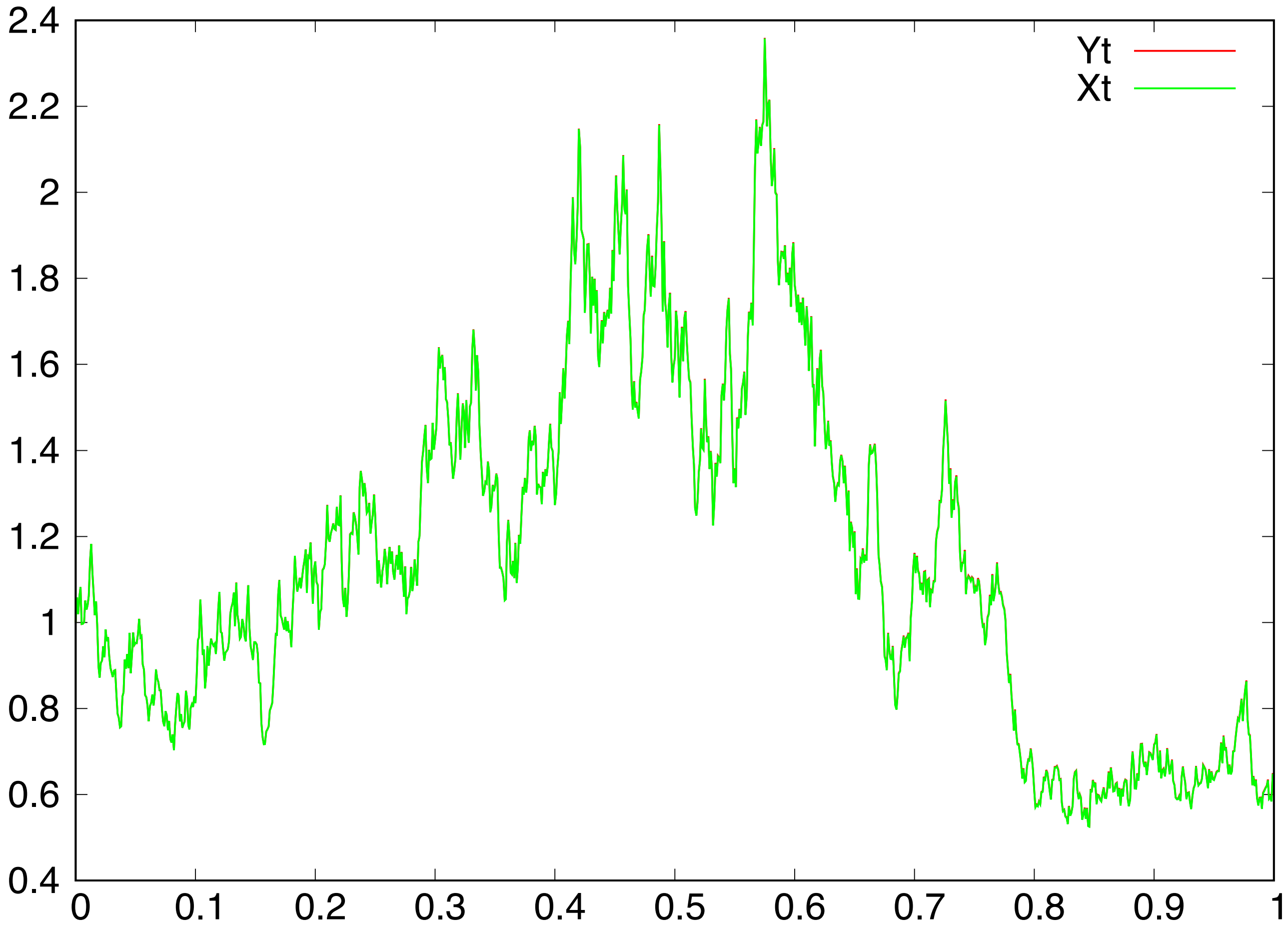

Yt
Xt
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.6
1.5
1.4
1.3
1.2
1.1
0.71
0.715
0.72
0.725
0.73
0.735
0.74

Milstein scheme

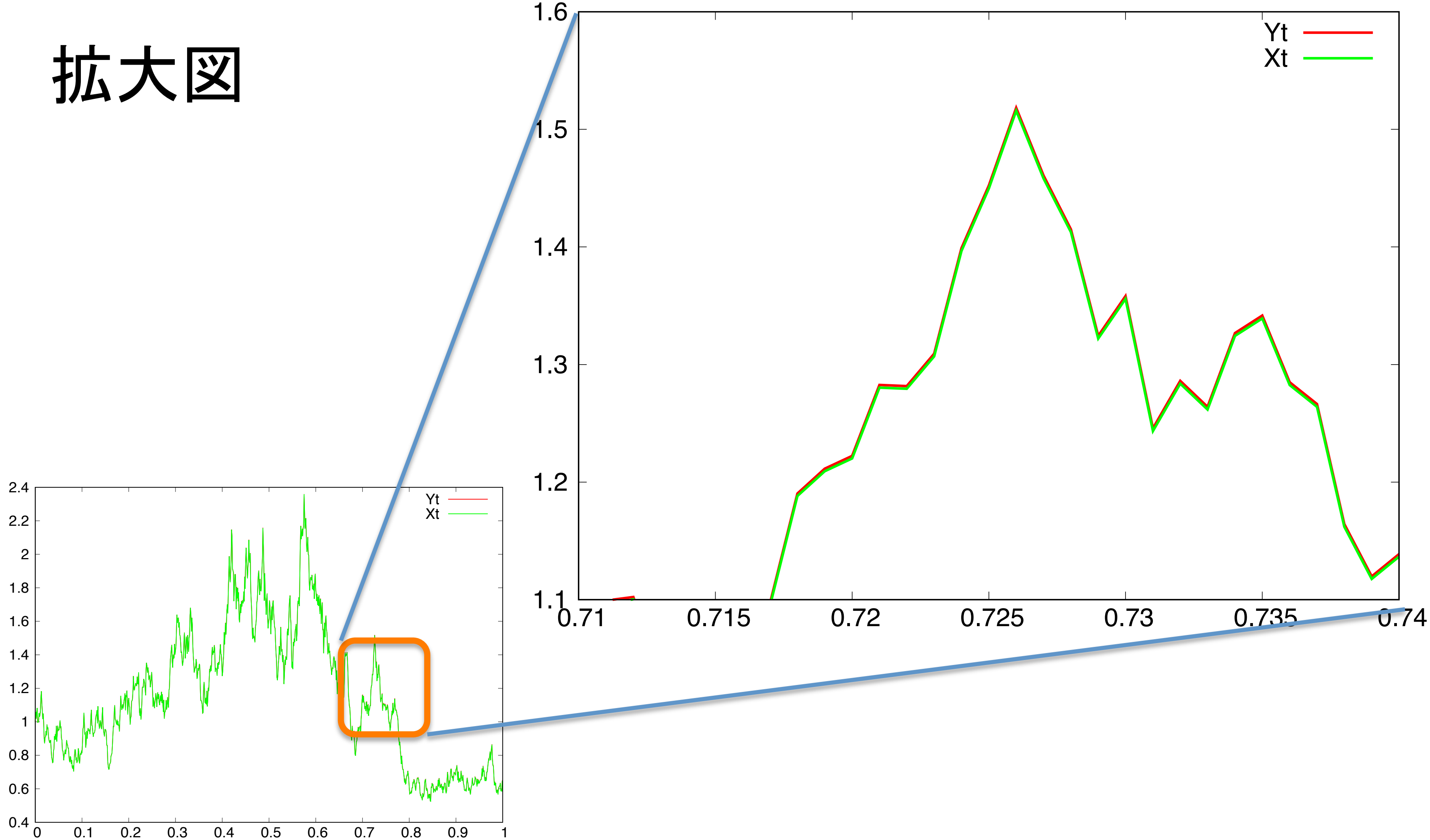
T = 1.0
（時間幅）

N = 1000
（時間刻み数）

赤 : 数値解
緑 : 解析解



拡大図



Yt
Xt
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.6
1.5
1.4
1.3
1.2
1.1
0.71
0.715
0.72
0.725
0.73
0.735
0.74

数値比較

EM_scheme

N	err
1000	0.054549999
4000	0.020677397
16000	0.011561287
64000	0.005103039
256000	0.003442725
1024000	0.001209479
4096000	0.000661487

Mil_scheme

N	err
1000	0.0018673600
4000	0.0004496050
16000	0.0000856447
64000	0.0000181859
256000	0.0000062320
1024000	0.0000015257
4096000	0.0000002810

Nは時間分割数、E(・)は平均値を表します。

$err_i = E(|Y_{in} - X_{in}|)$ $\leftarrow 1 \leq n \leq N$ における平均です。

$err = E(err_i)$ ←100回の試行で計算しています。

数値比較

EM_schemeとMil_schemeを比べると、
Nを4倍するにつれEM_schemeは1/2、
Mil_schemeは1/4になっているため、それぞれ
1/2次、1次の精度が出ていることがわかります。

（dt(刻み幅) = T / N）

Nを4倍　→　dt を 1/4倍

1/2次精度 = 1/2倍ずつ

1次精度 = 1/4倍ずつ

解析解との比較

前述の実例のように数値解と解析解を比較してみたい場合は、下記の例のように書きます。

（例） $dX_t = \mu X_t \; dt + \sigma X_t \; dW_t$

解析解は $X_t = X_0 \exp(\sigma W_t + (\mu - \frac{\sigma^2}{2})t)$

メイン関数内のXt=～とある部分の～を消して

$X_0 = X0$

$W_t = W\_new$

と置き換えてください。

解析解との比較

前ページの例にあった

$$dX_t = \mu X_t\ dt + \sigma X_t\ dW_t$$

の比較はすでにできるように作ってあります。

プログラム上部で定義している SDE_alpha_MouriとSDE_beta_Mouriが上記のμとσに対応しています。以下の部分（１６８行あたり）の//を、下部から上部に変えると比較バージョンができます。

```
printf("%f %f\n", t, Y);
//printf("%f %f %f\n", t, Y, Xt);
```

解析解との比較

Gnuplotで比較する場合

gnuplot> plot “a.dat” using 1:2 with lines, “a.dat” using 1:3 with lines

と入力すると、1番目が数値解、2番目が解析解となって同時プロットされます。