情報処理論2a

6章「積分」

(数値積分, Monte Carlo法)

中島康彦

§6. 1 今日の作業ディレクトリを作る

1. % **cd** ⇒ ホームディレクトリへ移動
2. % **mkdir chap18** ⇒ ディレクトリchap18を作成
3. % **cd chap18** ⇒ ディレクトリchap18へ移動
4. % **netscapeを使ってdata18をchap18へダウンロード**
5. % **tar xvf data18** ⇒ サンプルデータの複写

```
pi1.c
pi2.c
ndist1.c
ndist2.c
```

§6. 2 二次曲線による近似 ⇒ シンプソン公式

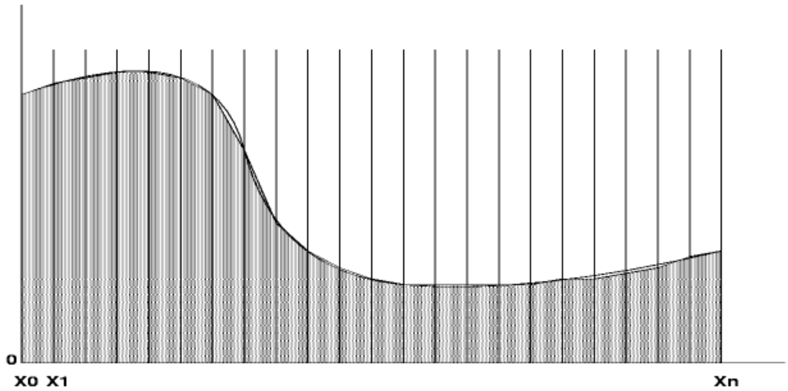
- y=f(x)の積分区間を[X0,Xn]，XiとXi+1の差をd
- (Xi-d,Yi-1),(Xi,Yi),(Xi+d,Yi+1)の3点を通る2次曲線の区間[Xi-1,Xi+1]の積分は

  $y=Ax^2+Bx+C$の場合

  $\{(X_i+d)^3-(X_i-d)^3\}A/3+\{(X_i+d)^2-(X_i-d)^2\}B/2+\{(X_i+d)-(X_i-d)\}C$ つまり

  $(Y_{i-1} + 4Y_i + Y_{i+1})d/3$
- 区間[X0,Xn]の積分の近似値は

  $\{(Y_0+4Y_1+Y_2)+(Y_2+4Y_3+.)..(.+Y_n)\}d/3$



§6. 2 シンプソン公式(続き)

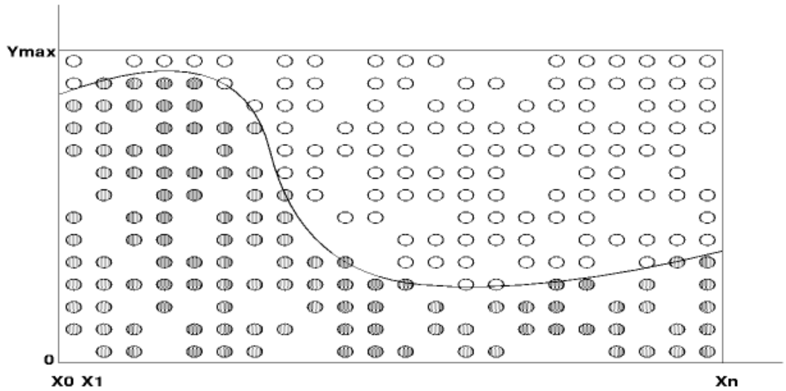
```
#define X0 0.0
#define Xn 2.0
double f(double x);

main(int argc, char **argv)
{
  int n, i, j;
  double d, x, S;

  if (argc != 2 || sscanf(argv[1], "%d", &n) != 1) {
      fprintf(stderr, "usage: %s N\n", argv[0]);
      exit(1);
  }
  n = n/2*2; d = (Xn-X0)/(double)n;
  x = X0;     S = f(x);
  for (i=1; i<=n; i+=2) {
    x = X0+d*i;      S += 4.0*f(x);
    x = X0+d*(i+1); S += 2.0*f(x);
  }
  S -= f(x);
  S = S*d/3.0;
  printf("S=%27.20e\n", S); /* 解の表示 */
  exit(0);
}
```

§6. 3 乱数を用いる方法 ⇒ モンテカルロ法

- y=f(x)の積分区間を[X0,Xn]，区間[X0,Xn]におけるyの最小値が0を下回らない，yの最大値がYmaxを上回らないとき，
- グラフ上に，X0≦Ri≦Xn，0≦Si≦Ymaxを満たすランダムな座標の点(Ri,Si)を散りばめる.
- 長方形領域の面積(Xn-X0)Ymaxに「Si<f(Ri)を満たす点が総数に占める割合」を乗じることにより，積分を求める.



§6. 3 モンテカルロ法(続き)

```
#define X0 0.0
#define Xn 2.0
#define Ymax 2.0
double f(double x, double y);

main(int argc, char **argv)
{
  int n, seed, i, j;
  double x, y;

  if (argc != 3 || sscanf(argv[1], "%d", &n) != 1
                || sscanf(argv[2], "%d", &seed) != 1) {
      fprintf(stderr, "usage: %s loop seed\n", argv[0]);
      exit(1);
  }
  srand(seed);
  for (i=0, j=0; i<n; i++) {
    x = (double)rand()*(Xn-X0)/(double)RAND_MAX;
    y = (double)rand()* Ymax  /(double)RAND_MAX;
    if (f(x, y) < 0.0)
      j++;
  }
  printf("S=%27.20e\n", (Xn-X0)*Ymax*(double)j/(double)n); /* 解の表示 */
  exit(0);
}
```

§6. 4 半径2の円の4分の1(答えはπ)

$x^2 + y^2 = 4 \quad (0 \leqq x \leqq 2,\ 0 \leqq y \leqq 2)$

積分区間を[0,2]とする.

▶ シンプソン公式の場合

$f(x) = \sqrt{(4 - x^2)} \quad (0 \leqq x \leqq 2)$

▶ モンテカルロ法の場合

区間[0,2]におけるyの最小値が0を下回らない.
yの最大値が2を上回らない.

0≦Ri≦2, 0≦Si≦2を満たすランダムな座標の点(Ri,Si)を散りばめる.

長方形領域の面積4に「R*R + S*S < 4を満たす点が総数に占める割合」を乗じる.

§6. 5 プログラムおよび入力例

pi1.c … シンプソン公式によるもの

▶ プログラムの引数はN(積分区間の分割数)

```
double f(double x)
{
  return (sqrt(4.0 - x*x));
}
```

pi2.c … モンテカルロ法によるもの

▶ プログラムの引数はN(点の総数)および乱数の種

```
double f(double x, double y)
{
  return (x*x + y*y - 4.0);
}
```

§6. 6 コンパイルと実行

1．コンパイルする.

```
% gcc pi1.c -o pi1 -lm
```

2．実行

```
% ./pi1
usage: ./pi1 N

% ./pi1 100
S= 3.14113320533922646405e+00

% ./pi1 10000
S= 3.14159219433824032919e+00

% ./pi1 1000000
S= 3.14159265313050140023e+00
```

§6. 6 コンパイルと実行(続き)

1．コンパイルする.

```
% gcc pi2.c -o pi2 -lm
```

2．実行

```
% ./pi2
usage: ./pi2 loop seed

% ./pi2 100 1
S= 3.19999999999999984013e+00

% ./pi2 10000 1
S= 3.13560000000000016485e+00

% ./pi2 1000000 1
S= 3.14306399999999985795e+00
```

§6. 7 正規密度関数(原始関数が存在しない)

$\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$ 標準偏差σ，平均μ，[-∞,+∞]の積分は1

$\pi^{-0.5}e^{-x^2}$ 標準偏差σ=1/√2，平均μ=0，[-∞,+∞]の積分は1

積分区間を[0,5]とする.

- シンプソン公式の場合

  f(x) = $\pi^{-0.5}e^{-x^2}$ (0≦x≦5)

- モンテカルロ法の場合

  区間[0,5]におけるyの最小値が0を下回らない.
  yの最大値が1/√πつまり1を上回らない.
  0≦Ri≦5, 0≦Si≦1を満たすランダムな座標の点(Ri,Si)を散りばめる.
  長方形領域の面積5に「Si < f(Ri)を満たす点が総数に占める割合」を乗じる.

§6. 8 プログラムおよび入力例

ndist1.c … シンプソン公式によるもの

- プログラムの引数はN(積分区間の分割数)

```
double f(double x)
{
  return (1.0/sqrt(M_PI)*exp(-x*x));
}
```

ndist2.c … モンテカルロ法によるもの

- プログラムの引数はN(点の総数)および乱数の種

```
double f(double x)
{
  return (1.0/sqrt(M_PI)*exp(-x*x));
}
```

§6. 9 コンパイルと実行

1. コンパイルする.

```
% gcc ndist1.c -o ndist1 -lm
```

2. 実行

```
% ./ndist1
usage: ./ndist1 N

% ./ndist1 100
S= 4.99999999999230837489e-01

% ./ndist1 10000
S= 4.99999999999233279979e-01

% ./ndist1 1000000
S= 4.99999999999229560732e-01
```

§6. 9 コンパイルと実行(続き)

1. コンパイルする.

```
% gcc ndist2.c -o ndist2 -lm
```

2. 実行

```
% ./ndist2
usage: ./ndist2 loop seed

% ./ndist2 100 1
S= 3.49999999999999977796e-01

% ./ndist2 10000 1
S= 4.91499999999999992450e-01

% ./ndist2 1000000 1
S= 4.99495000000000022311e-01
```

§6. 10 例題

正規密度関数を計算する際にπ(M_PI)を用いていた．このことを利用し，πを用いない密度関数の積分により，πの近似値を求める方法を考えよ．

ヒント:積分が√πとなるように密度関数を変形する．

§6. 11 今日の課題

半径2の球の8分の1の体積(答えは4π/3)をモンテカルロ法により求めるプログラムを作成せよ．

また，loop数を100，10000，1000000，100000000と変化させて，計算結果がどのように変化するか調査せよ．

ヒント:pi2.cにz座標を追加する．変更箇所はわずか8行．

宛先: nakashim@econ.kyoto-u.ac.jp
件名: unix2-学生番号

今日はここまで