数値計算

偏微分方程式の数値解法

偏微分方程式の数値解法

- 変数分離 → 複数の常微分方程式
- 積分方程式化 → 数値積分+代数方程式
- 関数系展開 → 行列の固有値問題、連立一次方程式
- 差分方程式化 → 行列の固有値問題、連立一次方程式、数値解(陽解法)

微分と差分

$$f(x+h) = f(x) + \frac{d}{dx}f(x)h + \frac{1}{2}\frac{d^2}{dx^2}f(x)h^2 + \cdots$$

$$f(x-h) = f(x) - \frac{d}{dx}f(x)h + \frac{1}{2}\frac{d^2}{dx^2}f(x)h^2 - \cdots$$

$$\frac{d}{dx}f(x) \approx \frac{f(x+h)-f(x)}{h}$$ 前進差分

$$\frac{d}{dx}f(x) \approx \frac{f(x)-f(x-h)}{h}$$ 後退差分

$$\frac{d}{dx}f(x) \approx \frac{f(x+h)-f(x-h)}{2h}$$ 中心差分

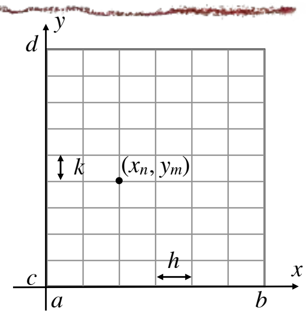
$$\frac{d^2}{dx^2}f(x) \approx \frac{f(x+h)-2f(x)+f(x-h)}{h^2}$$

変数の離散化

$u(x,y)$　　$a \le x \le b$, $c \le y \le d$

$$h = \frac{b-a}{N}$$

$$k = \frac{d-c}{M}$$



$$x_n = x_0 + nh,\ (x_0 = a,\ n = 0,1,2,3,\cdots,N-1)$$

$$y_n = y_0 + mk,\ (x_0 = a,\ m = 0,1,2,3,\cdots,M-1)$$

偏導関数

$u(x,y)$

$$x_n = x_0 + nh,\ (x_0 = a,\ n = 0,1,2,3,\cdots,N-1)$$

$$y_n = y_0 + mk,\ (x_0 = a,\ m = 0,1,2,3,\cdots,M-1)$$

$$\frac{\partial u}{\partial x}(x_n,y_n) = \frac{u(x_{n+1},y_n)-u(x_n,y_n)}{h}$$

$$\frac{\partial u}{\partial y}(x_n,y_n) = \frac{u(x_n,y_{n+1})-u(x_n,y_n)}{k}$$

$$\frac{\partial^2 u}{\partial x^2}(x_n,y_n) = \frac{u(x_{n+1},y_n)-2u(x_n,y_n)+u(x_{n-1},y_n)}{h^2}$$

$$\frac{\partial^2 u}{\partial y^2}(x_n,y_n) = \frac{u(x_n,y_{n+1})-2u(x_n,y_n)+u(x_n,y_{n-1})}{k^2}$$

偏微分方程式 ⇒ 差分方程式

二階線型偏微分方程式

- 放物型微分方程式
  - 熱伝導、拡散方程式
  - 時間に依存するSchrödinger方程式

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial y^2}$$

- 楕円型微分方程式
  - ラプラス・ポアソン方程式
  - 時間に依らないSchrödinger方程式

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f$$

- 双曲型微分方程式
  - 物体の振動、波動方程式

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial y^2}$$

放物型微分方程式

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial y^2}$$

差分方程式化すると

$$\frac{u(x_{n+1}, y_m) - u(x_n, y_m)}{h} = \frac{u(x_n, y_{m+1}) - 2u(x_n, y_m) + u(x_n, y_{m-1})}{k^2}$$

$$u(x_{n+1}, y_m) = ru(x_n, y_{m+1}) + (1-2r)u(x_n, y_m) + ru(x_n, y_{m-1}))$$

$$r = \frac{h}{k^2} < \frac{1}{2}$$

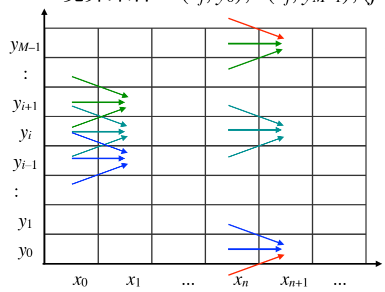
$r : (1-2r) : r$ で平均化

放物型微分方程式

$u(x_{n+1}, y_m) = ru(x_n, y_{m+1}) + (1-2r)u(x_n, y_{m+1}) + ru(x_n, y_{m-1})$

初期条件: $u(x_0, y_i)$, (i=0, 1, 2, ..., M–1)

境界条件: $u(x_j, y_0)$, $u(x_j, y_{M-1})$, (j=0, 1, 2, ..., N–1)

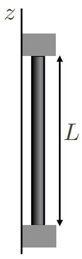


初期条件と境界条件がわかれば次々に $u(x_{n+1}, y_m)$を計算することができる

r<0.5となるようにh, kを選ぶ

放物型微分方程式の例

温度T_0[°C]長さL[m]の金属棒の両端を, 時刻t=0[s]で0 [°C]に保つとき, 棒の温度分布の時間変化$T(t, z)$を求めよ



$$\frac{\partial}{\partial t}T(t,z) = D\frac{\partial^2}{\partial z^2}T(t,z)$$

初期条件 $T(0,z) = T_0$

境界条件 $T(t,0) = T(t,L) = 0$

$u = \frac{T(t,z)}{T_0}$ $x = \frac{D}{L^2}t$ $y = \frac{z}{L}$ とおいて無次元化

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial y^2}$$

初期条件 $u(0,y) = 1$

境界条件 $u(x,0) = u(x,1) = 0$

放物型微分方程式の例

```
#include <stdio.h>
#define N 40
#define M 1000
void outdat(double x, double v[N]);
int main(void) {
  double r, s, dy, x, dx = 0.01;
  double prev[N], next[N];
  int i, j;
  dy = 1.0/(double) N;          /* 刻幅等の設定 */
  r = dx/(dy*dy);
  s = 1.0 - 2.0*r;
  prev[0] = 0.0;                                  /* 境界条件 */
  for (i=1; i<N-1; i++) {prev[i] = 1.0;} /* 初期条件 */
  prev[N-1] = 0.0;                                /* 境界条件 */
```

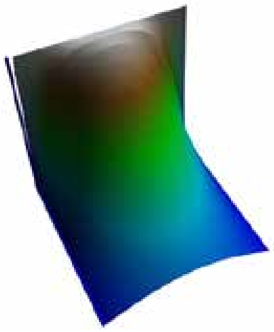
放物型微分方程式の例

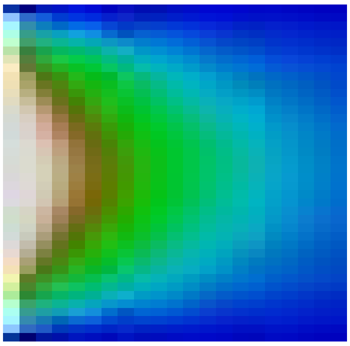
```
  x = 0.0;
  for (j=0; j<M; j++) {
    if (j%100 == 0) {outdat(x, prev);}
    x = x + dx;
    for (i=1; i<N-1; i++) {
      next[i] = r*(prev[i+1] + prev[i-1])
                + s*prev[i];
    }
    for (i=1; i<N-1; i++) {
      prev[i] = next[i];
    }
  }
  return 0;
}
```

放物型微分方程式の例

```
void outdat(double x, double v[N]) {
  int i;
  double y;
  for (i=0; i<N; i++) {
    y = (double) i/(double) N;
    printf("%f %f %f\n", x, y, v[i]);
  }
}
```

放物型微分方程式の例





双曲型微分方程式

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial y^2}$$

差分方程式化すると

$$\frac{u(x_{n+1}, y_m) - 2u(x_n, y_m) + u(x_{n-1}, y_m)}{h^2} = \frac{u(x_n, y_{m+1}) - 2u(x_n, y_m) + u(x_n, y_{m-1})}{k^2}$$

$$u(x_{n+1}, y_m) = r\left(u(x_n, y_{m+1}) + u(x_n, y_{m-1})\right) + 2(1-r)u(x_n, y_m) - u(x_{n-1}, y_m)$$

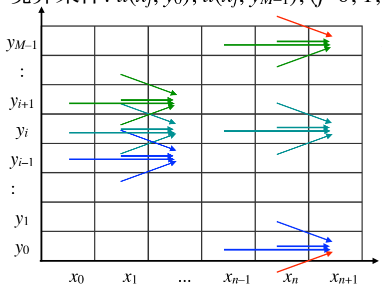
$$r = \frac{h^2}{k^2} < 1$$

双曲型微分方程式

$u(x_{n+1}, y_m) = r(u(x_n, y_{m+1}) + u(x_n, y_{m-1})) + 2(1-r)u(x_n, y_m) - u(x_{n-1}, y_m)$

初期条件: $u(x_0, y_i), u(x_1, y_i), (i=0, 1, 2, ..., M-1)$

境界条件: $u(x_j, y_0), u(x_j, y_{M-1}), (j=0, 1, 2, ..., N-1)$



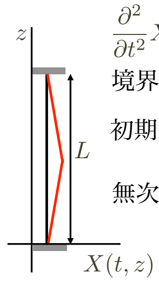
初期条件と境界条件がわかれば次々に $u(x_{n+1}, y_m)$ を計算することができる

$r<1$ となるように

h, k を選ぶ

双曲型微分方程式の例

長さL[m]の弦の両端を固定し時刻t=0[s]に中央を0.5[cm]持ちあげて離した後弦の振動の様子$X(t, z)$ [cm]を求めよ



$$\frac{\partial^2}{\partial t^2} X(t, z) = C \frac{\partial^2}{\partial z^2} X(t, z)$$

境界条件 $X(t, 0) = X(t, L) = 0$

初期条件 $X(0, z) = \begin{cases} \frac{z}{L} \times 1[\text{cm}] & (0 \leq z \leq \frac{L}{2}) \\ (1 - \frac{z}{L}) \times 1[\text{cm}] & (\frac{L}{2} < z \leq L) \end{cases}$

無次元化 $\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial y^2}$

境界条件 $u(x, 0) = u(x, 1) = 0$

初期条件 $u(0, y) = \begin{cases} y & (0 \leq y \leq \frac{1}{2}) \\ 1 - y & (\frac{1}{2} < y \leq 1) \end{cases}$

双曲型微分方程式の例

```
#include <stdio.h>
#define N 40
int main(void) {
  double x = 0.0, next[N], curr[N], prev[N];
  double q, s, h, k = 0.001;
  int i, j, m = 1000;
  h = 1.0/(double) N;
                                /* 刻幅等の設定 */
  q = (k*k)/(h*h);
  s = 2.0*(1.0 - q);
  for (i=0; i<N/2; i++) {
    curr[i] = (double) i/(double) N;     /* 初期条件・境界条件 */
    curr[N-1-i] = curr[i];
  }
  for (i=0; i<N; i++) prev[i] = curr[i];
```

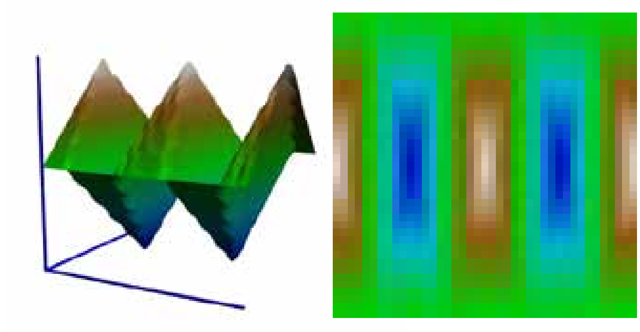
双曲型微分方程式の例

```
  for (j=0; j<m; j++) {
    if (j%100 == 0) outdata(x, curr);
    x = x + k;
    for (i=1; i<N-1; i++) {
      next[i] =q*(curr[i+1] + curr[i-1])
              + s*curr[i] - prev[i];
    }
    for (i=0; i<N; i++) {
      prev[i] = curr[i];
      curr[i] = next[i];
    }
  }
  return 0;
}
```

双曲型微分方程式の例

```
void outdata(double x, double u[N]) {
  int i;
  double y;
  for (i=0; i<N; i++) {
    y = (double) i/(double) N;
    printf ("%f %f %f\n", x, y, u[i]);
  }
}
```

双曲型微分方程式の例



楕円型微分方程式

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f$$

差分方程式化すると

$$\frac{u(x_{n+1}, y_m) - 2u(x_n, y_m) + u(x_{n-1}, y_m)}{h^2} + \frac{u(x_n, y_{m+1}) - 2u(x_n, y_m) + u(x_n, y_{m-1})}{k^2} = f(x_n, y_m)$$

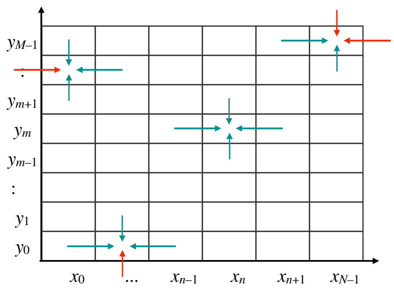
$$u(x_n, y_m) = \frac{1}{4}\{u(x_{n+1}, y_m) + u(x_{n-1}, y_m) + u(x_n, y_{m+1}) + u(x_n, y_{m-1}) - h^2 f(x_n, y_m)\}$$

$$k^2 = h^2$$

楕円型微分方程式

$$u(x_{n+1},y_m)+u(x_{n-1},y_m)-4u(x_n,y_m)+u(x_n,y_{m+1})+u(x_n,y_{m-1})=h^2f(x_n,y_m)$$

$h=k$となるようにN, Mを選ぶ



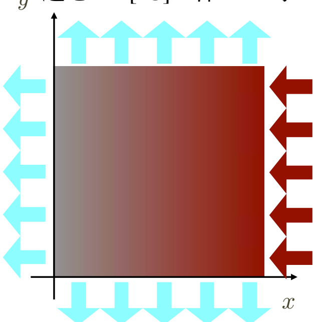
境界条件:

$u(x_0, y_i)$,
$u(x_{N-1}, y_i)$,
$(i=0, 1, 2, ..., M-1)$
$u(x_j, y_0)$,
$u(x_j, y_{M-1})$,
$(j=0, 1, 2, ..., N-1)$

がわかれば $u(x_n, y_m)$ についての連立一次方程式になる

楕円型微分方程式の例

一辺の長さL[m]の正方形の板の一つの辺をT=100[°C]に、他の三辺をT=0[°C]に保った時の板の温度分布$T(x,y)$を求めよ。



$$\frac{\partial^2}{\partial x^2}T(x,y) + \frac{\partial^2}{\partial y^2}T(x,y) = 0$$

境界条件

$$T(x,0) = T(x,L) = T(0,y) = 0$$
$$T(L,y) = 100$$

無次元化

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$
$$u(x,0) = u(x,1) = u(0,y) = 0$$
$$u(1,y) = 1$$

楕円型微分方程式の例

```
#include <stdio.h>
#include <math.h>
#define N 41
#define M 10000
#define DELTA 0.0001
double gz(double next[N][N], double prev[N][N]);
int main(void) {
  int i, j, k;
  double err, next[N][N], prev[N][N];
  for (i=0; i<N; i++) {
    next[0][i] = 0.0;
    next[i][0] = 0.0;         /* 境界条件 */
    next[i][N-1] = 1.0;
  }
  for (i=1; i<N-1; i++) {
    next[N-1][i] = 0.0;
  }
```

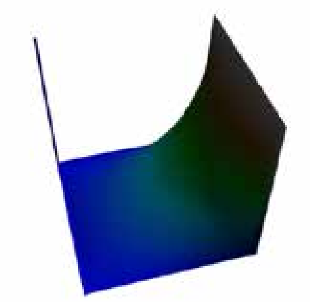
楕円型微分方程式の例

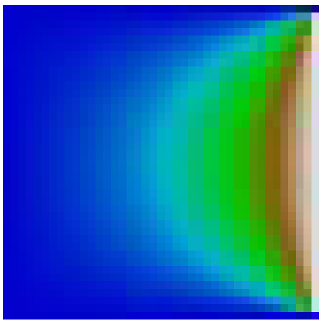
```
  for (i=0; i<N; i++) {
    for (j=0; j<N; j++) { prev[i][j] = next[i][j]; }
  }
  for (k=0; k<M; k++) {
    err = gz(next, prev);
    if (err < DELTA) {
      for (i=0; i<N; i++) {
        for (j=0; j<N; j++) { printf(" %f", next[i][j]);}
        printf("\n");
      }
      return 0;
    }
  }
  printf("NOT CONVERGENT\n");
  return 1;
}
```

楕円型微分方程式の例

```
double gz(double next[N][N], double prev[N][N]) {
  int i, j;
  double err = 0.0;
  for (i=1; i<N-1; i++) {
    for (j=1; j<N-1; j++) {
      next[i][j] = (prev[i+1][j] + prev[i-1][j]
                  + prev[i][j+1] + prev[i][j-1])/4.0;
      err = err + fabs(prev[i][j] - next[i][j]);
      prev[i][j] = next[i][j];
    }
  }
  return err;
}
```

楕円型微分方程式の例





固有値方程式型微分方程式

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \lambda u$$

差分方程式化では

境界条件を満足するように固有値を決めるのが困難

$$Lu = \lambda u$$

$$u = c_0\varphi_0 + c_1\varphi_1 + c_2\varphi_2 + \cdots = \sum_n c_n\varphi_n$$

$$\sum_n c_n L\varphi_n = \lambda \sum_n c_n\varphi_n \qquad (\langle\varphi_m|\varphi_n\rangle = \delta_{m,n})$$

$$\sum_n \langle\varphi_m|L|\varphi_n\rangle c_n = \lambda c_m$$

固有値方程式型微分方程式

$$\sum_n \langle\varphi_m|L|\varphi_n\rangle c_n = \lambda c_m$$

$$\begin{pmatrix} \langle\varphi_0|L|\varphi_0\rangle & \langle\varphi_0|L|\varphi_1\rangle & \cdots \\ \langle\varphi_1|L|\varphi_0\rangle & \langle\varphi_1|L|\varphi_1\rangle & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \end{pmatrix} = \lambda \begin{pmatrix} c_0 \\ c_1 \\ \vdots \end{pmatrix}$$

行列の固有値方程式