

§10 拡散方程式の数値解析手法

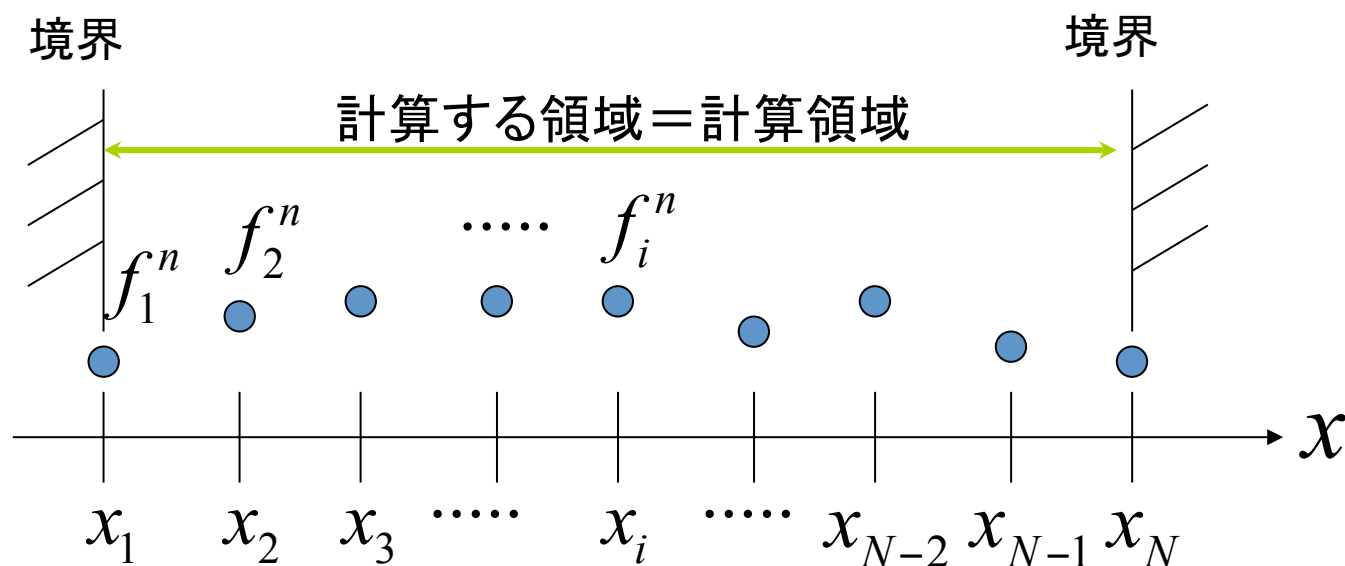
Numerical Method for the Diffusion Eq.



拡散方程式の差分法による離散化

■ 1次元拡散方程式 $\frac{\partial}{\partial t} f(t, x) = \nu \frac{\partial^2}{\partial x^2} f(t, x)$

計算格子点: x_i --- 計算機上で値 f を計算する点



時刻 t_n , 計算格子 x_i における値:

$$f_i^n = f(t_n, x_i)$$



拡散方程式の差分法による離散化

- 1次元拡散方程式 $\frac{\partial}{\partial t} f(t, x) = \nu \frac{\partial^2}{\partial x^2} f(t, x)$

時間微分の差分方法＝Euler陽解法、陰解法、CrankNicolson,...

Euler陽解法を用いることにすれば:

$$\frac{\partial}{\partial t} f(t_n, x_i) \cong \frac{f_i^{n+1} - f_i^n}{\Delta t}$$

$$\frac{\partial}{\partial t} f(t_n, x_i) = \nu \frac{\partial^2}{\partial x^2} f(t_n, x_i) \quad \Rightarrow \quad \frac{f_i^{n+1} - f_i^n}{\Delta t} = \nu \frac{\partial^2}{\partial x^2} f(t_n, x_i)$$



拡散方程式の差分法による離散化

- 陽解法の計算式: $\frac{f_i^{n+1} - f_i^n}{\Delta t} = \nu \frac{\partial}{\partial x^2} f(t_n, x_i)$

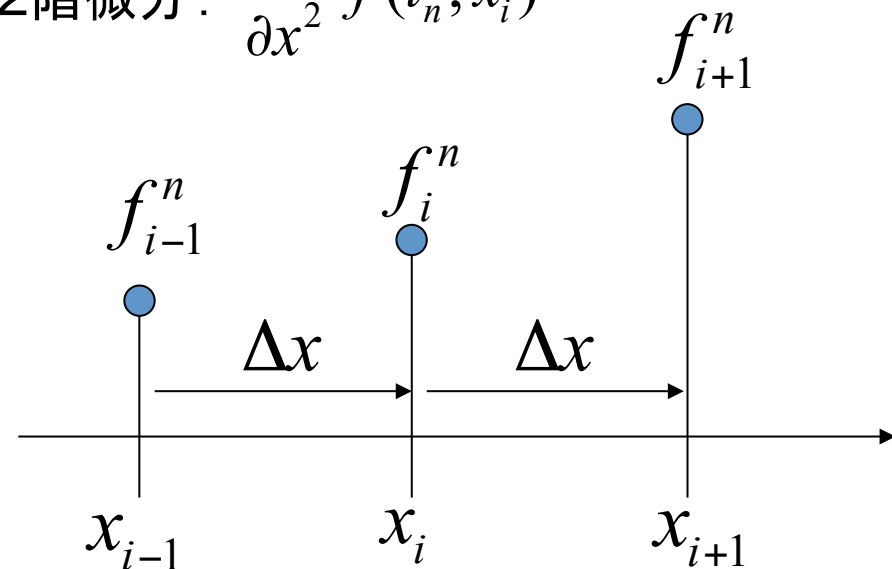
時刻 t_n , 計算格子 x_i における2階微分: $\frac{\partial}{\partial x^2} f(t_n, x_i)$

➡ 中心2階差分

となりあった3点

$$f_{i-1}^n \quad f_i^n \quad f_{i+1}^n$$

を用いて差分近似する。



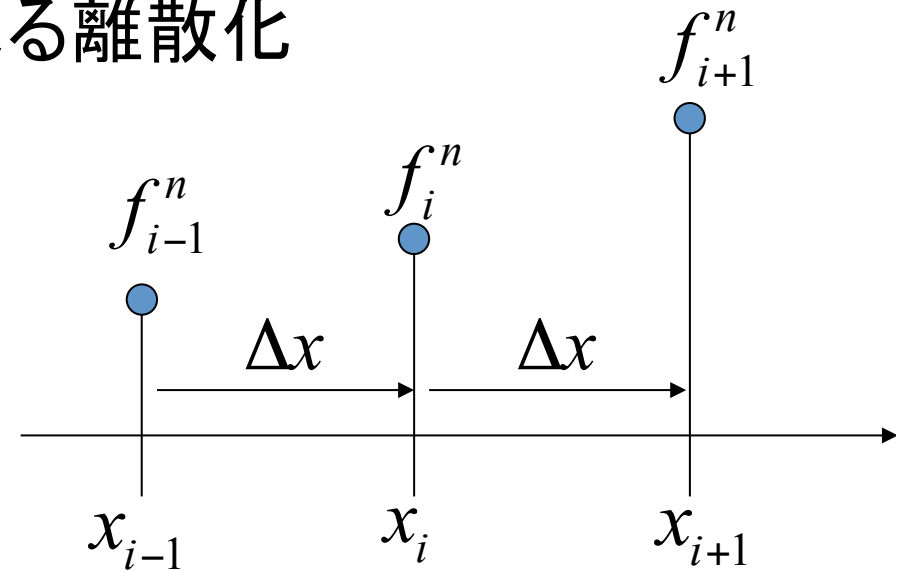
拡散方程式の差分法による離散化

- 中心2階差分

計算格子点の間隔が
等間隔 Δx である。



等間隔計算格子



Taylor展開式:

$$\begin{cases} f^n_{i+1} = f^n_i + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) \Delta x + \frac{\Delta x^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{\Delta x^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) + \dots \\ f^n_{i-1} = f^n_i + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) (-\Delta x) + \frac{(-\Delta x)^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{(-\Delta x)^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) + \dots \end{cases}$$

両辺を足し合わせて、



拡散方程式の差分法による離散化

- 中心2階差分

$$\begin{aligned} f_{i+1}^n + f_{i-1}^n &= f_i^n + f_i^n + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) \Delta x + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) (-\Delta x) \\ &+ \frac{\Delta x^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{(-\Delta x)^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{\Delta x^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) + \frac{(-\Delta x)^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) \\ &+ \dots \end{aligned}$$

整理すると、

$$\begin{aligned} f_{i+1}^n + f_{i-1}^n &= 2f_i^n + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) \Delta x - \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) \Delta x \\ &+ \frac{\Delta x^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{\Delta x^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{\Delta x^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) - \frac{\Delta x^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) \\ &+ O(\Delta x^4) \end{aligned}$$



拡散方程式の差分法による離散化

- 中心2階差分

$$\begin{aligned} f_{i+1}^n + f_{i-1}^n &= f_i^n + f_i^n + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) \Delta x + \left(\frac{\partial f(t_n, x_i)}{\partial x} \right) (-\Delta x) \\ &+ \frac{\Delta x^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{(-\Delta x)^2}{2} \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + \frac{\Delta x^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) + \frac{(-\Delta x)^3}{3!} \left(\frac{\partial^3 f(t_n, x_i)}{\partial x^3} \right) \\ &+ O(\Delta x^4) \end{aligned}$$

整理すると、

$$f_{i+1}^n + f_{i-1}^n = 2f_i^n + \Delta x^2 \left(\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \right) + O(\Delta x^4)$$

$$\frac{\partial^2 f(t_n, x_i)}{\partial x^2} = \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2} + O(\Delta x^2)$$

中心2階差分による近似式:

$$\frac{\partial^2 f(t_n, x_i)}{\partial x^2} \cong \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$$



拡散方程式の差分法による離散化

- 中心2階差分のポンチ絵理解

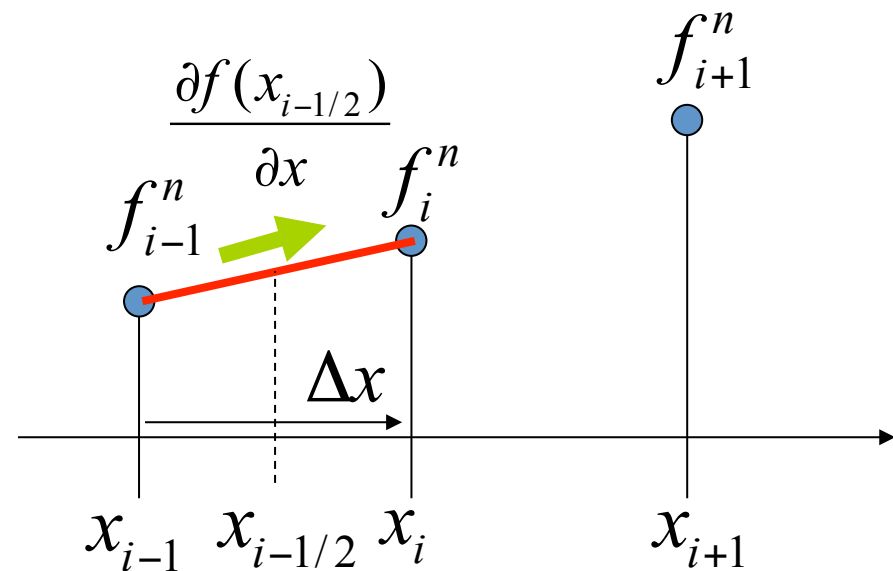
$$\frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2} = \frac{\left(\frac{f_{i+1}^n - f_i^n}{\Delta x} \right) - \left(\frac{f_i^n - f_{i-1}^n}{\Delta x} \right)}{\Delta x}$$

中点 $x_{i-1/2} = \frac{1}{2}(x_i + x_{i-1})$

$$\frac{f_i^n - f_{i-1}^n}{\Delta x} \cong \left(\frac{\partial f(x_{i-1/2})}{\partial x} \right)$$

中点 $x_{i-1/2}$ における

1階微分



拡散方程式の差分法による離散化

- 中心2階差分のポンチ絵理解

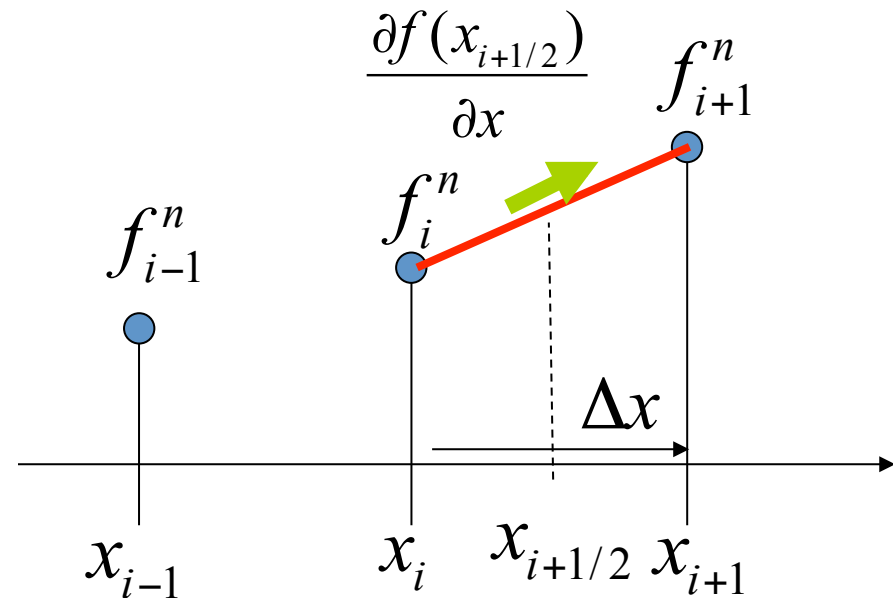
$$\frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2} = \frac{\left(\frac{f_{i+1}^n - f_i^n}{\Delta x} \right) - \left(\frac{f_i^n - f_{i-1}^n}{\Delta x} \right)}{\Delta x}$$

中点 $x_{i+1/2} = \frac{1}{2}(x_i + x_{i+1})$

$$\frac{f_{i+1}^n - f_i^n}{\Delta x} \cong \left(\frac{\partial f(x_{i+1/2})}{\partial x} \right)$$

中点 $x_{i+1/2}$ における

1階微分

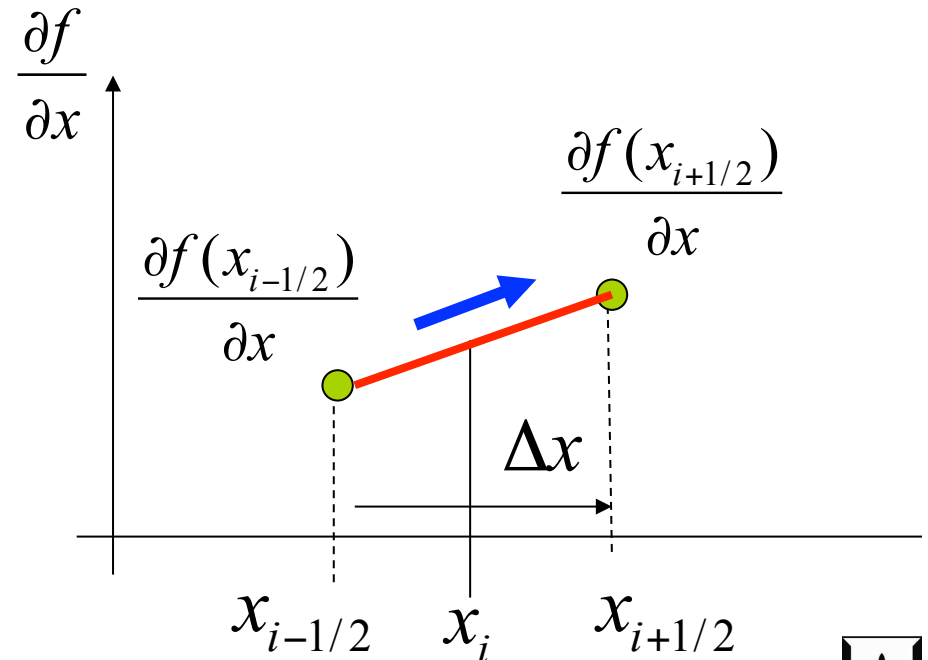
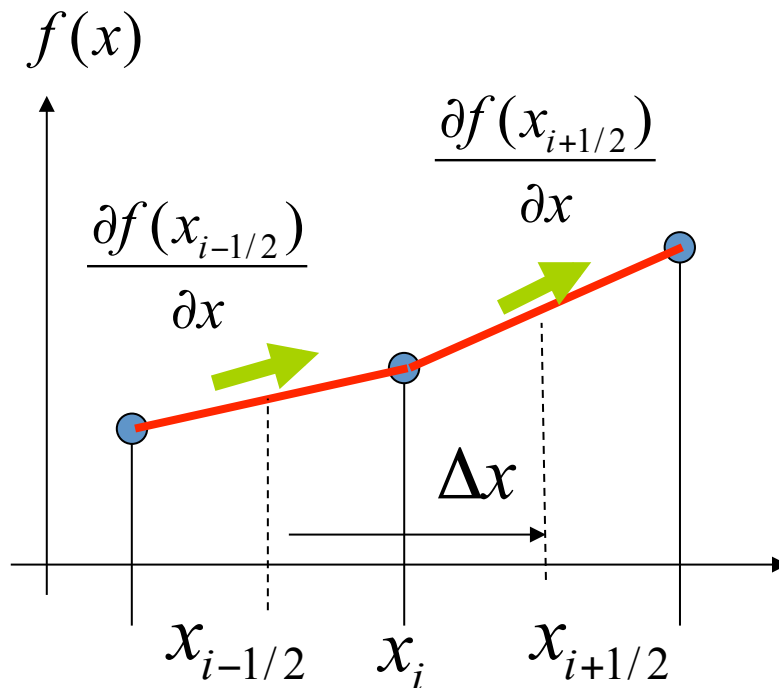


拡散方程式の差分法による離散化

即ち、中心2階微分は中点 $x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}$ 間の1階微分で

$$\frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2} \approx \underbrace{\left(\frac{\partial f(x_{i+1/2})}{\partial x} \right) - \left(\frac{\partial f(x_{i-1/2})}{\partial x} \right)}_{\Delta x} \approx \frac{\partial^2 f(x_i)}{\partial x^2} : 2\text{階微分}$$

x_i における1階微分の傾き



“時間陽解法”＋“空間中心差分”による拡散方程式の離散結果

- 計算式:
$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \nu \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$$

$$\therefore f_i^{n+1} = f_i^n + \frac{\nu \Delta t}{\Delta x^2} (f_{i+1}^n - 2f_i^n + f_{i-1}^n)$$

右辺は時刻 t_n における値 f^n のみであるので、全ての計算格子点 x_i について、 $i = 0, 1, 2, 3, \dots, N$ として上の計算式を用いれば、次時刻 t_{n+1} での値 f^{n+1} が求められる。



“時間陽解法”+“空間中心差分”による拡散方程式の離散結果

$$\therefore f_i^{n+1} = f_i^n + \frac{v\Delta t}{\Delta x^2} \left(f_{i+1}^n - 2f_i^n + f_{i-1}^n \right)$$

初期条件:

$$\begin{array}{c} t_0 \quad (f_1^0, f_2^0, \dots, f_{i-2}^0, f_{i-1}^0, f_i^0, f_{i+1}^0, f_{i+2}^0, \dots, f_N^0) \\ \qquad \qquad \qquad \searrow \text{orange} \downarrow \text{yellow} \swarrow \text{purple} \downarrow \text{red} \nwarrow \text{purple} \downarrow \text{red} \nearrow \text{red} \\ t_1 \quad (f_1^1, f_2^1, \dots, f_{i-2}^1, f_{i-1}^1, f_i^1, f_{i+1}^1, f_{i+2}^1, \dots, f_N^1) \\ \qquad \qquad \qquad \vdots \\ t_n \quad (f_1^n, f_2^n, \dots, f_{i-2}^n, f_{i-1}^n, f_i^n, f_{i+1}^n, f_{i+2}^n, \dots, f_N^n) \end{array}$$

境界条件の必要性

境界条件: 境界上の格子点 x_1, x_N の値 f_1^{n+1}, f_N^{n+1} の計算

$$f_1^{n+1} = f_1^n + \frac{v\Delta t}{\Delta x^2} (f_2^n - 2f_1^n + \underline{f_0^n})$$

$$f_N^{n+1} = f_N^n + \frac{v\Delta t}{\Delta x^2} (\underline{f_{N+1}^n} - 2f_N^n + f_{N-1}^n)$$

を計算するためには、領域外の値 f_0^{n+1}, f_{N+1}^{n+1} が必要 \Rightarrow 境界条件で与える

$$f_0^n (f_1^n, f_2^n, \dots, f_{i-2}^n, f_{i-1}^n, f_i^n, f_{i+1}^n, f_{i+2}^n, \dots, f_N^n) f_{N+1}^n$$

$\begin{matrix} ? & \searrow & \downarrow & \swarrow & & & \searrow & \downarrow & \swarrow & ? \\ & & f_1^{n+1} & & f_2^{n+1} & & \dots & & f_{i-2}^{n+1} & & f_{i-1}^{n+1} & & f_i^{n+1} & & f_{i+1}^{n+1} & & f_{i+2}^{n+1} & & \dots & & f_N^{n+1} \end{matrix}$



境界条件の種類

- Dirichlet境界(固定境界)条件: $f_0^n = \alpha$

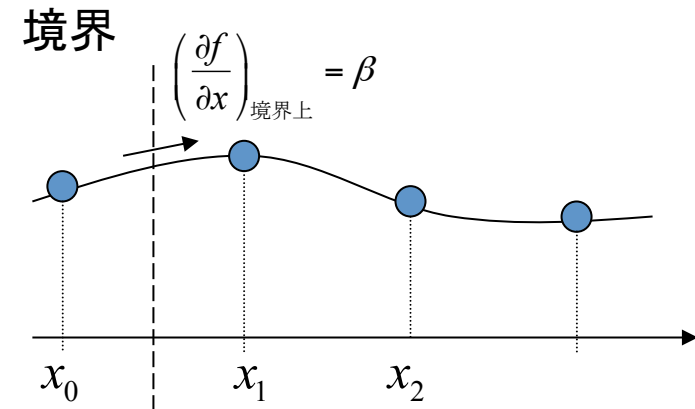
$$\Rightarrow f_1^{n+1} = f_1^n + \frac{v\Delta t}{\Delta x^2} (f_2^n - 2f_1^n + \alpha)$$

- Neumann境界(自由境界)条件:

$$\left(\frac{\partial f}{\partial x} \right)_{\text{境界上}} = \beta$$

境界上での傾き

$$\left(\frac{\partial f}{\partial x} \right)_{\text{境界上}} \cong \frac{f_1^n - f_0^n}{\Delta x}$$



Neumann条件を満たすために以下のように与えればよいことが分かる。

$$\frac{f_1^n - f_0^n}{\Delta x} = \beta \quad \therefore f_0^n = f_1^n - \beta\Delta x$$

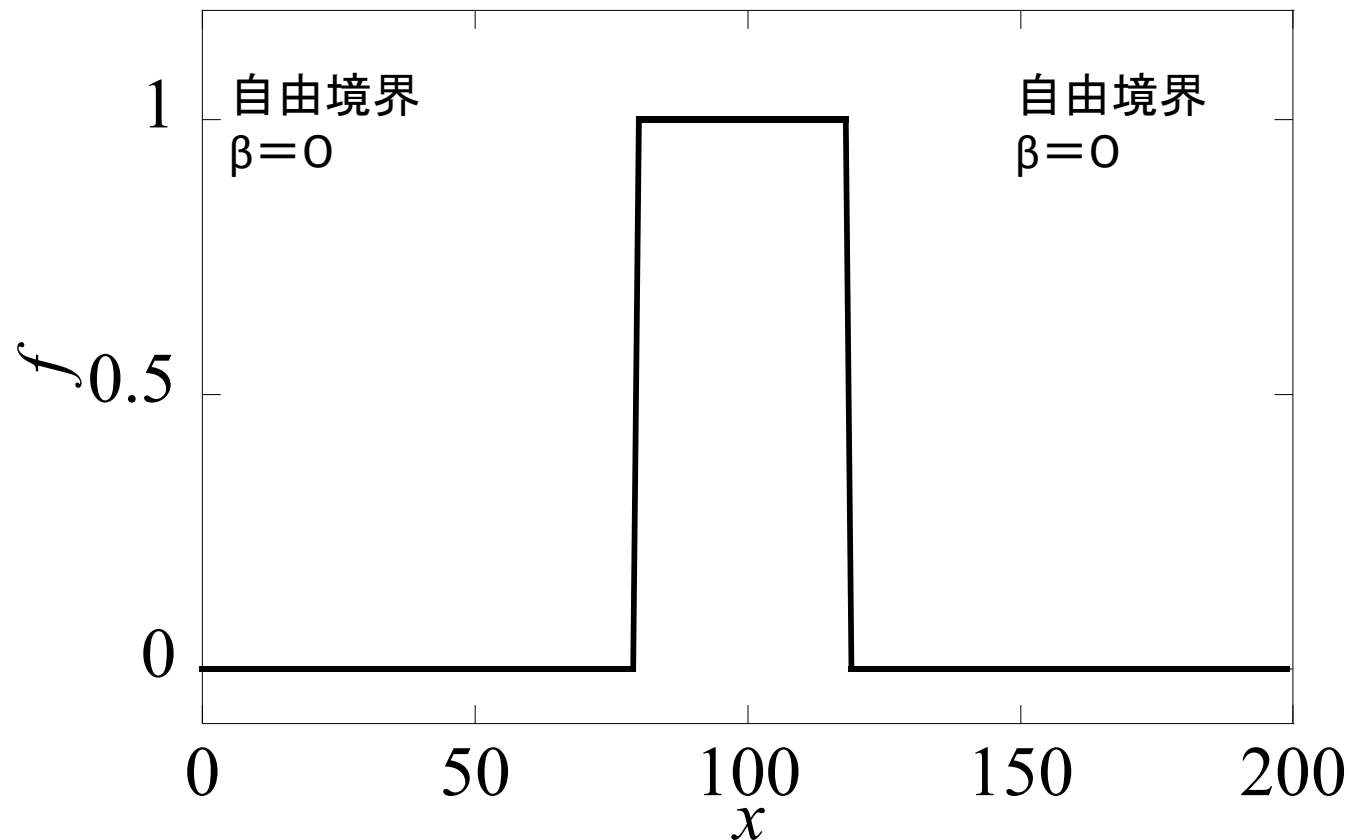
$$\Rightarrow f_1^{n+1} = f_1^n + \frac{v\Delta t}{\Delta x^2} (f_2^n - 2f_1^n + f_1^n - \beta\Delta x)$$



計算例

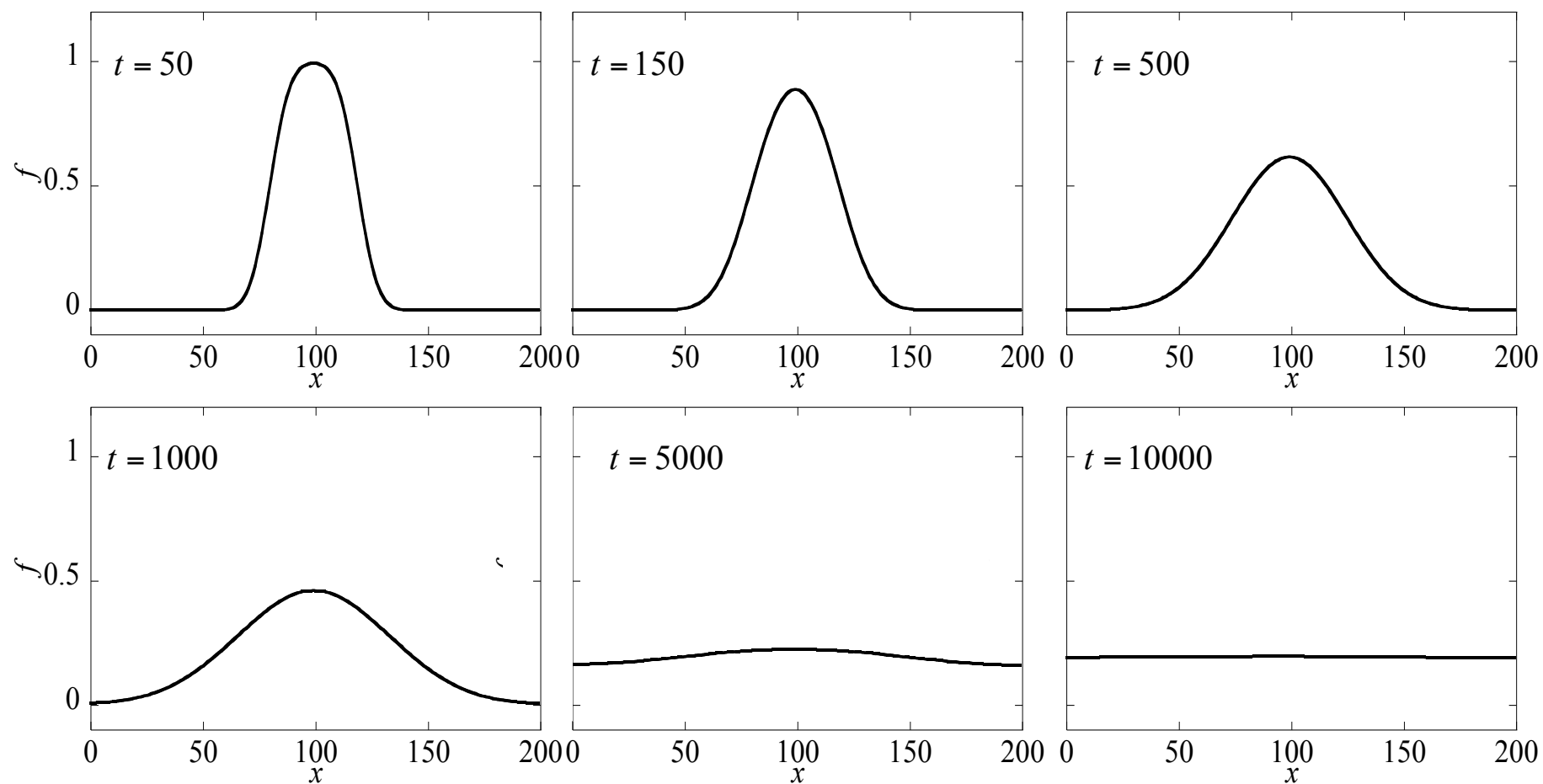
$$\frac{\partial f}{\partial t} = \nu \frac{\partial^2 f}{\partial x^2} \quad \nu = 0.5 \quad N = 200, \quad \Delta x = 1.0, \quad \Delta t = 0.1$$

初期条件: 中心40格子点のみ1、それ以外0



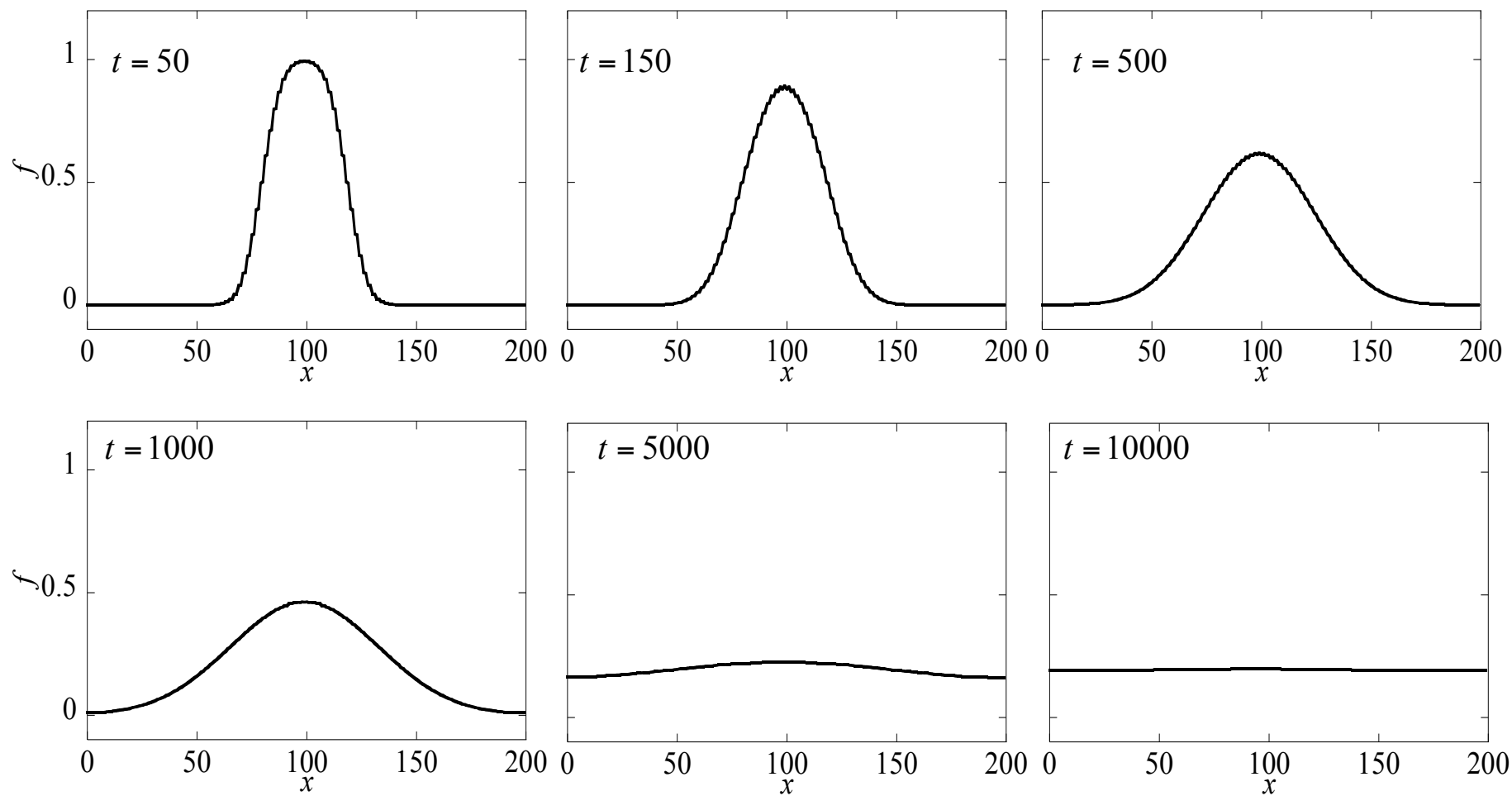
計算例

$$\Delta t = 0.1 \quad (C_{Diff} = 0.1)$$



計算例

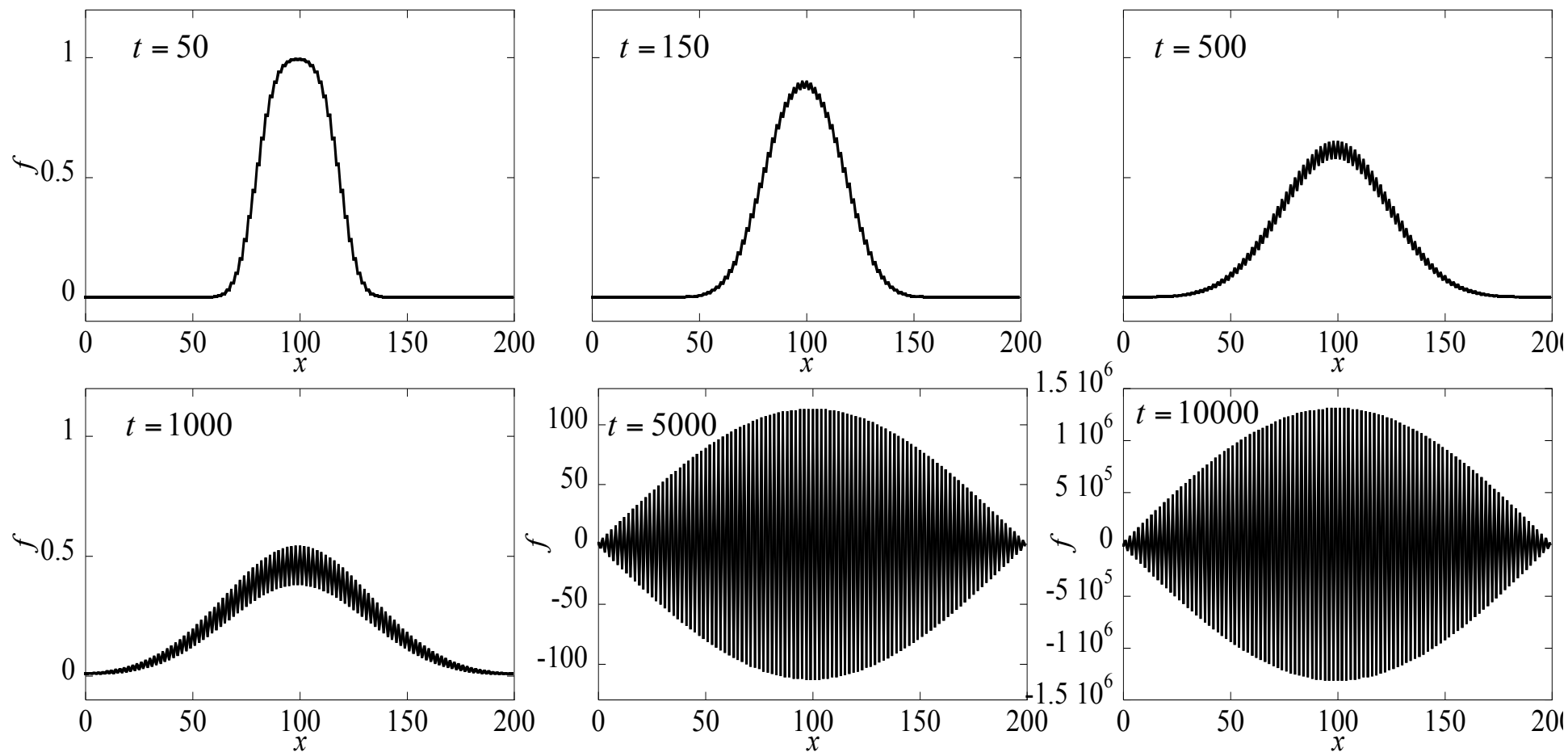
$$\Delta t = 0.999 \quad (C_{Diff} = 0.4995)$$



計算例

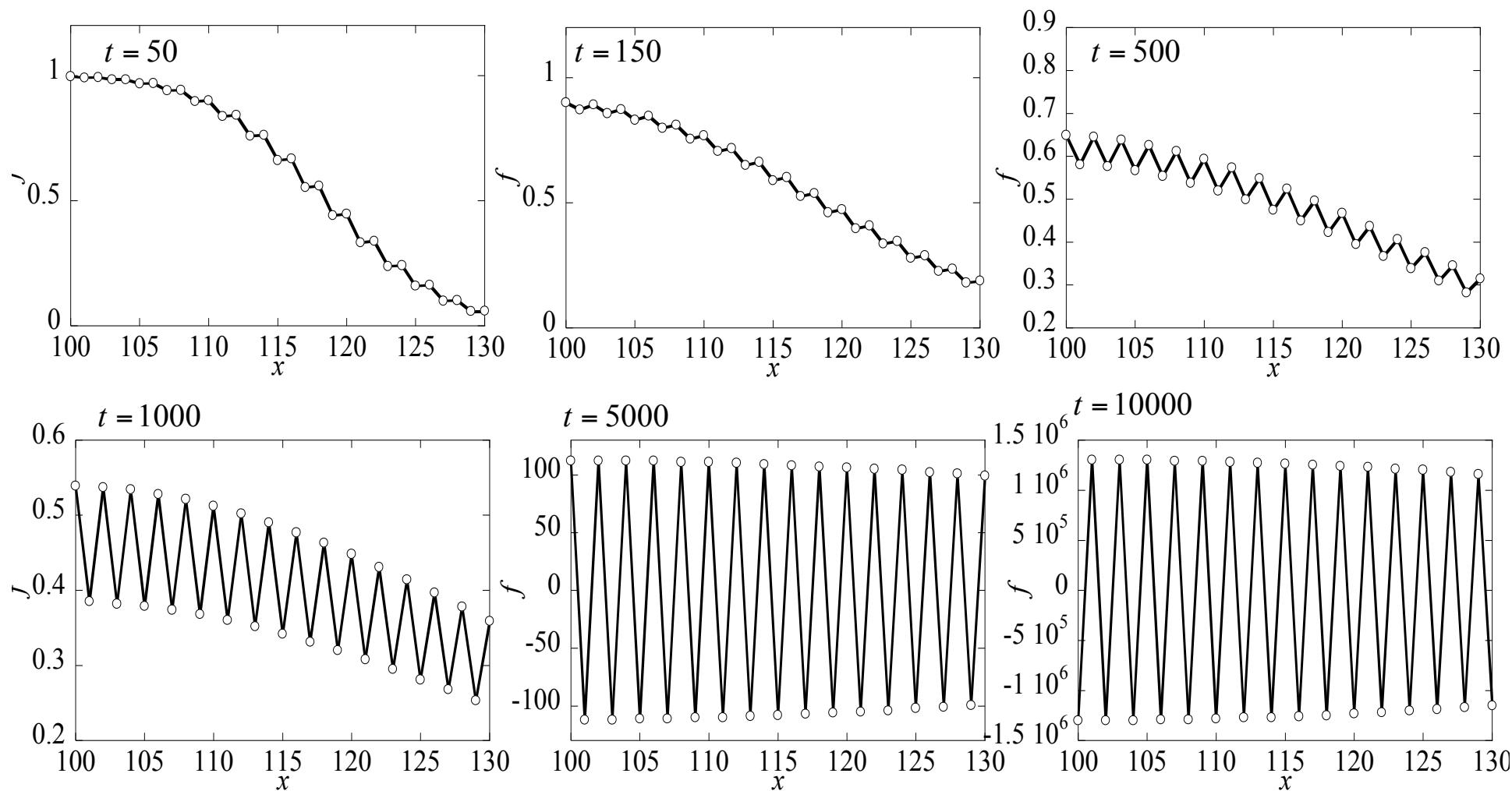
$$\Delta t = 1.001$$

$$(C_{Diff} = 0.505)$$



計算例(拡大図)

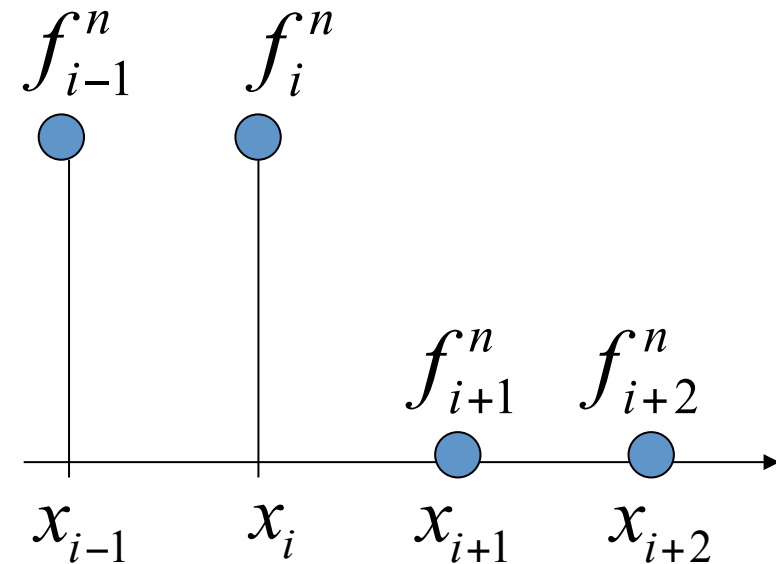
$$\Delta t = 1.001 \quad (C_{Diff} = 0.505)$$



“時間陽解法”+“空間中心差分”での 計算の安定性

$$\begin{cases} f_1 = \dots = f_{i-1}^n = f_i^n = 1 \\ f_{i+1}^n = f_{i+2}^n = \dots = f_N^n = 0 \end{cases}$$

の場合



次時刻での値を計算:

$$f_{i-1}^{n+1} = f_{i-1}^n + \frac{v\Delta t}{\Delta x^2} (f_i^n - 2f_{i-1}^n + f_{i-2}^n) = 1 + \frac{v\Delta t}{\Delta x^2} (1 - 2 + 1) = 1$$

$$f_i^{n+1} = f_i^n + \frac{v\Delta t}{\Delta x^2} (f_{i+1}^n - 2f_i^n + f_{i-1}^n) = 1 + \frac{v\Delta t}{\Delta x^2} (1 - 2 + 0) = 1 - \frac{v\Delta t}{\Delta x^2}$$

$$f_{i+1}^{n+1} = f_{i+1}^n + \frac{v\Delta t}{\Delta x^2} (f_{i+2}^n - 2f_{i+1}^n + f_i^n) = 0 + \frac{v\Delta t}{\Delta x^2} (0 - 0 + 1) = \frac{v\Delta t}{\Delta x^2}$$

$$f_{i+2}^{n+1} = f_{i+2}^n + \frac{v\Delta t}{\Delta x^2} (f_{i+3}^n - 2f_{i+2}^n + f_{i+1}^n) = 0 + \frac{v\Delta t}{\Delta x^2} (0 - 0 + 0) = 0$$



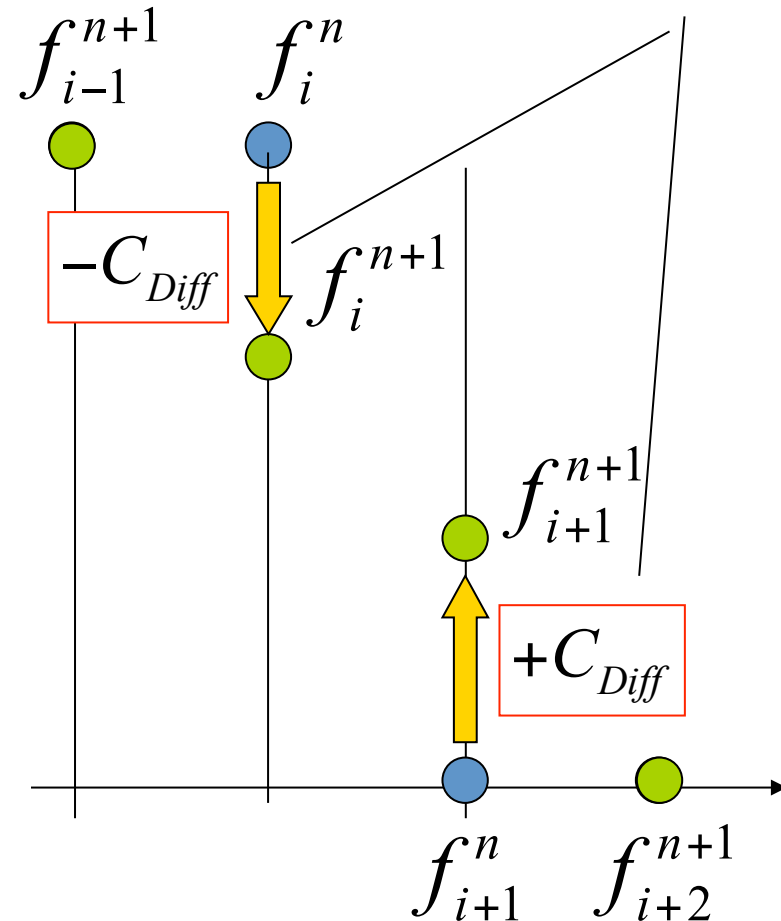
“時間陽解法”+“空間中心差分”での 計算の安定性

同じ分量だけ増減

次時刻での値を計算:

$$\left\{ \begin{array}{l} f_{i-1}^{n+1} = 1 \\ f_i^{n+1} = 1 - \frac{v\Delta t}{\Delta x^2} = 1 - C_{diff} \\ f_{i+1}^{n+1} = \frac{v\Delta t}{\Delta x^2} = C_{diff} \\ f_{i+2}^{n+1} = 0 \end{array} \right.$$

$$C_{Diff} = \frac{v\Delta t}{\Delta x^2}$$



“時間陽解法”+“空間中心差分”での 計算の安定性

$$C_{Diff} = \frac{v\Delta t}{\Delta x^2} \text{ は時間刻み幅に比例する}$$

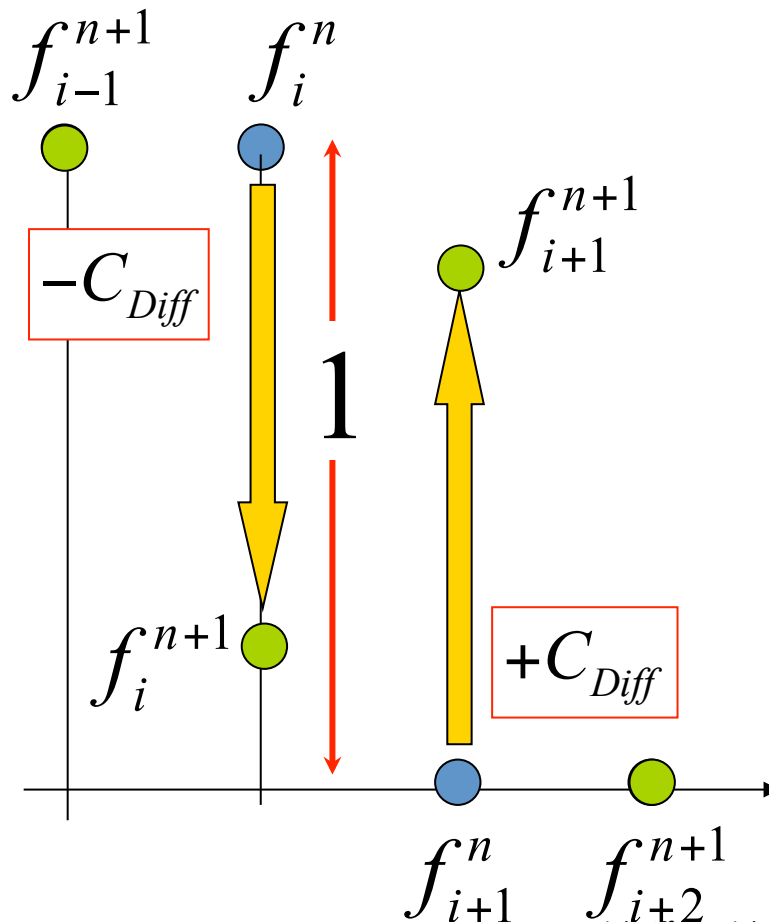
仮に Δt が大きく

$$C_{Diff} = \frac{v\Delta t}{\Delta x^2} > \frac{1}{2}$$

であった場合を考えると、

$$f_i^{n+1} = 1 - C_{diff} < \frac{1}{2}$$

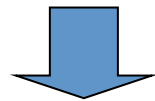
$$f_{i+1}^{n+1} = C_{diff} > \frac{1}{2}$$



“時間陽解法”+“空間中心差分”での 計算の安定性

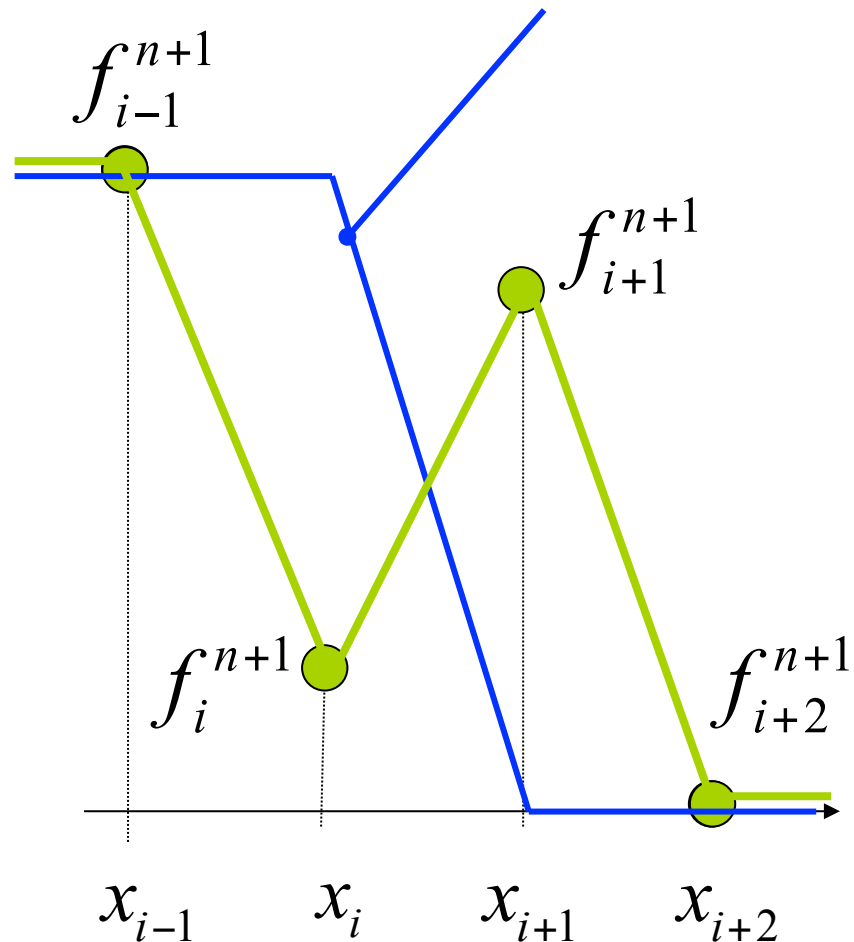
もともとの拡散の現象
とは異なり、高密度の
場所が急に現れる！

密度変化のあるところ
には必ず振動が生じる！



数値不安定
(計算破綻)

前の時刻での分布



“時間陽解法”+“空間中心差分”での 計算の安定性

“時間陽解法”+“空間中心差分”の場合の安定条件

$$C_{Diff} = \frac{v\Delta t}{\Delta x^2} < \frac{1}{2}$$

実際には、時間刻みに関する条件に書き換え

$$\Delta t < \frac{\Delta x^2}{2v}$$

Δx と v の大きさに応じて、 Δt を上 の条件式を満たすように十分に小さな値として計算を行うことになる。

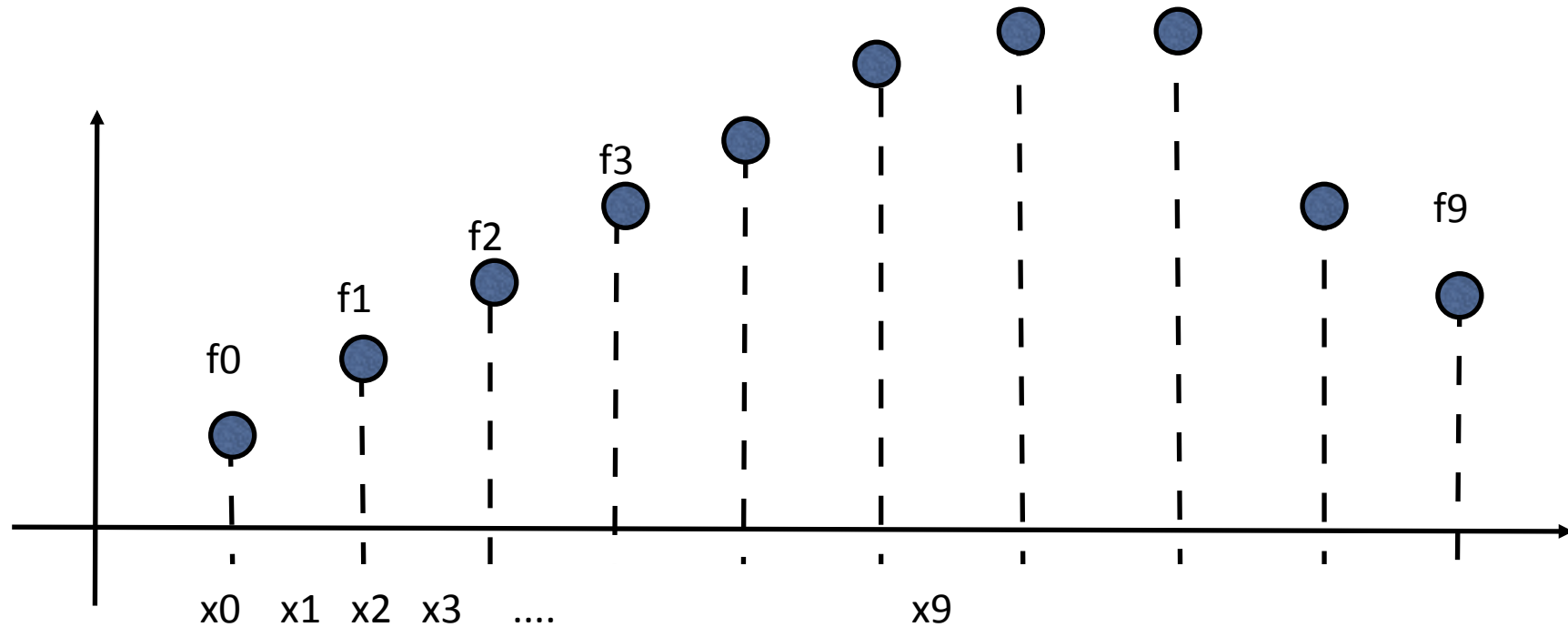


配列を使う

- 複数の値を格納する変数: 配列

一次元の座標: $x_0, x_1, x_2, x_3, \dots, x_9$

関数の値: $f_0, f_1, f_2, f_3, \dots, f_9$



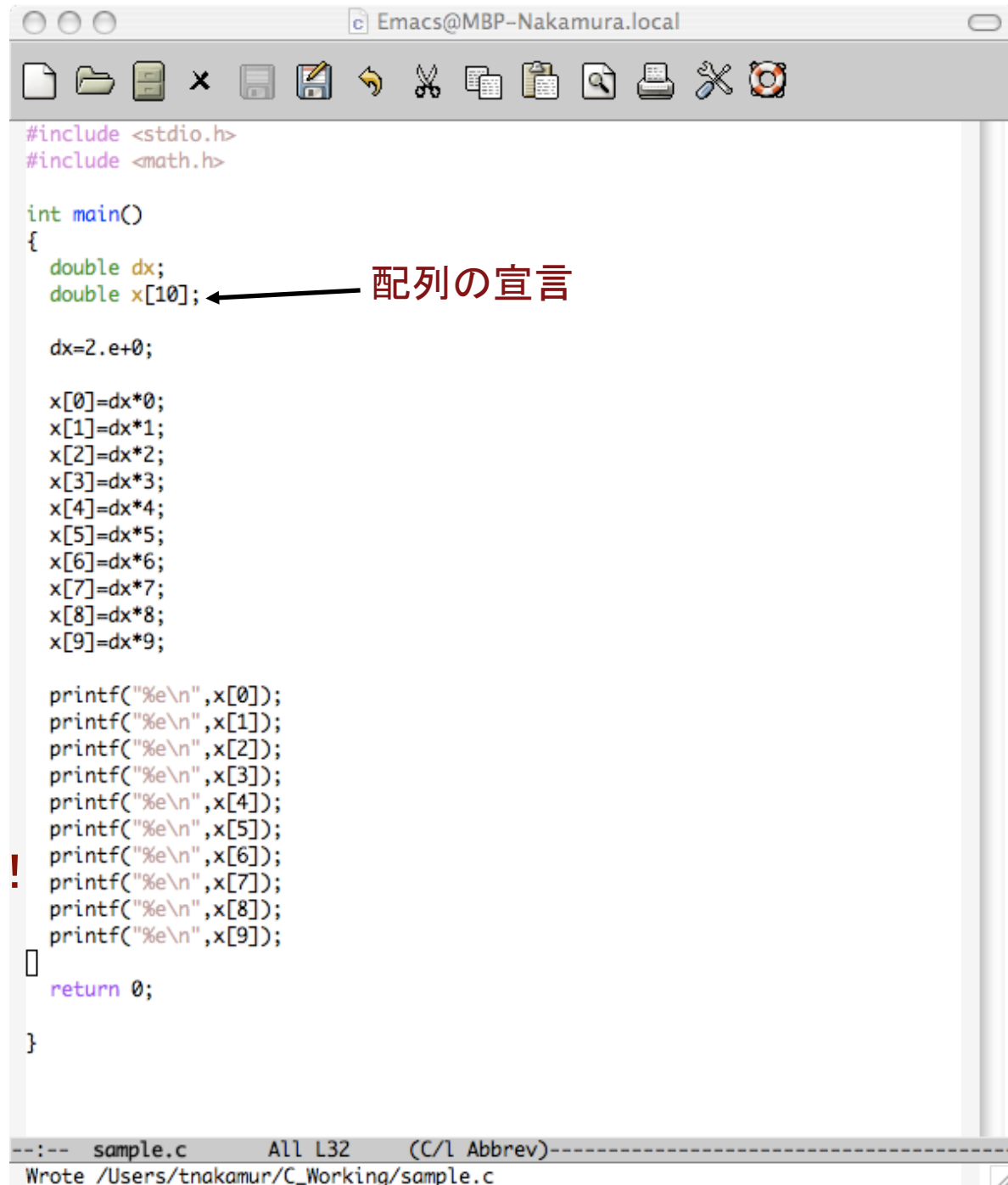
配列の宣言

`double x[10];`

double型の変数として、
`x[0], x[1], x[2], ..., x[8], x[9]`

の10個を使えるように
にする。

※`x[1] ~ x[10]`では無い！！



```
#include <stdio.h>
#include <math.h>

int main()
{
    double dx;
    double x[10]; ← 配列の宣言

    dx=2.0;

    x[0]=dx*0;
    x[1]=dx*1;
    x[2]=dx*2;
    x[3]=dx*3;
    x[4]=dx*4;
    x[5]=dx*5;
    x[6]=dx*6;
    x[7]=dx*7;
    x[8]=dx*8;
    x[9]=dx*9;

    printf("%e\n", x[0]);
    printf("%e\n", x[1]);
    printf("%e\n", x[2]);
    printf("%e\n", x[3]);
    printf("%e\n", x[4]);
    printf("%e\n", x[5]);
    printf("%e\n", x[6]);
    printf("%e\n", x[7]);
    printf("%e\n", x[8]);
    printf("%e\n", x[9]);

    return 0;
}
```

--:-- sample.c All L32 (C/l Abbrev)-----
Wrote /Users/tnakamur/C_Working/sample.c

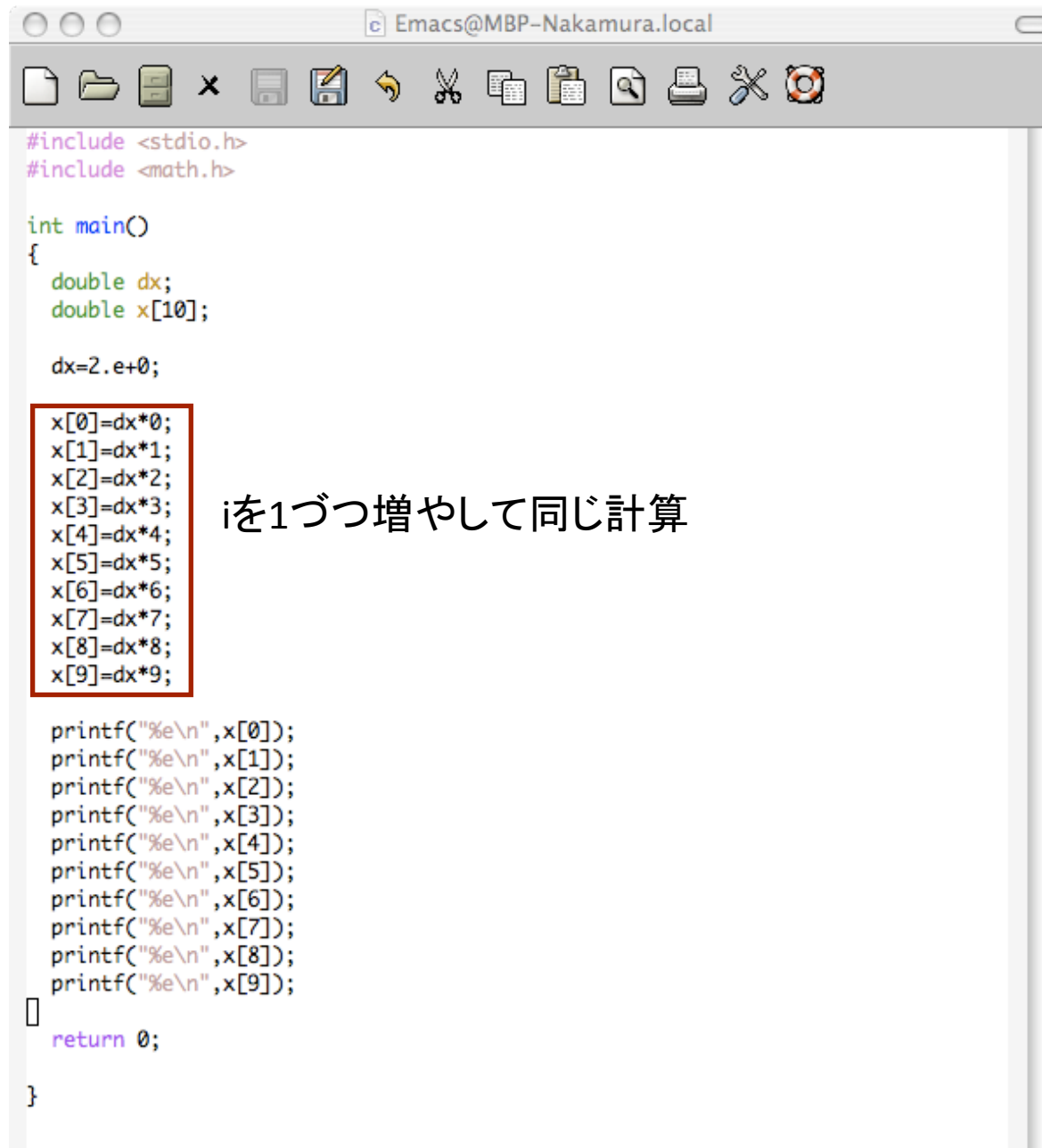


実行結果

```
MBP/Users/tnakamur/C_Working 4> cc -o run sample.c -lm
MBP/Users/tnakamur/C_Working 5> ./run
0.000000e+00 x[0]
2.000000e+00 x[1]
4.000000e+00 x[2]
6.000000e+00 x[3]
8.000000e+00 x[4]
1.000000e+01 x[5]
1.200000e+01 x[6]
1.400000e+01 x[7]
1.600000e+01 x[8]
1.800000e+01 x[9]
MBP/Users/tnakamur/C_Working 6> □
```



forループを用いた繰り返し



```
#include <stdio.h>
#include <math.h>

int main()
{
    double dx;
    double x[10];

    dx=2.0;

    x[0]=dx*0;
    x[1]=dx*1;
    x[2]=dx*2;
    x[3]=dx*3;
    x[4]=dx*4;
    x[5]=dx*5;
    x[6]=dx*6;
    x[7]=dx*7;
    x[8]=dx*8;
    x[9]=dx*9;

    printf("%e\n",x[0]);
    printf("%e\n",x[1]);
    printf("%e\n",x[2]);
    printf("%e\n",x[3]);
    printf("%e\n",x[4]);
    printf("%e\n",x[5]);
    printf("%e\n",x[6]);
    printf("%e\n",x[7]);
    printf("%e\n",x[8]);
    printf("%e\n",x[9]);

    return 0;
}
```

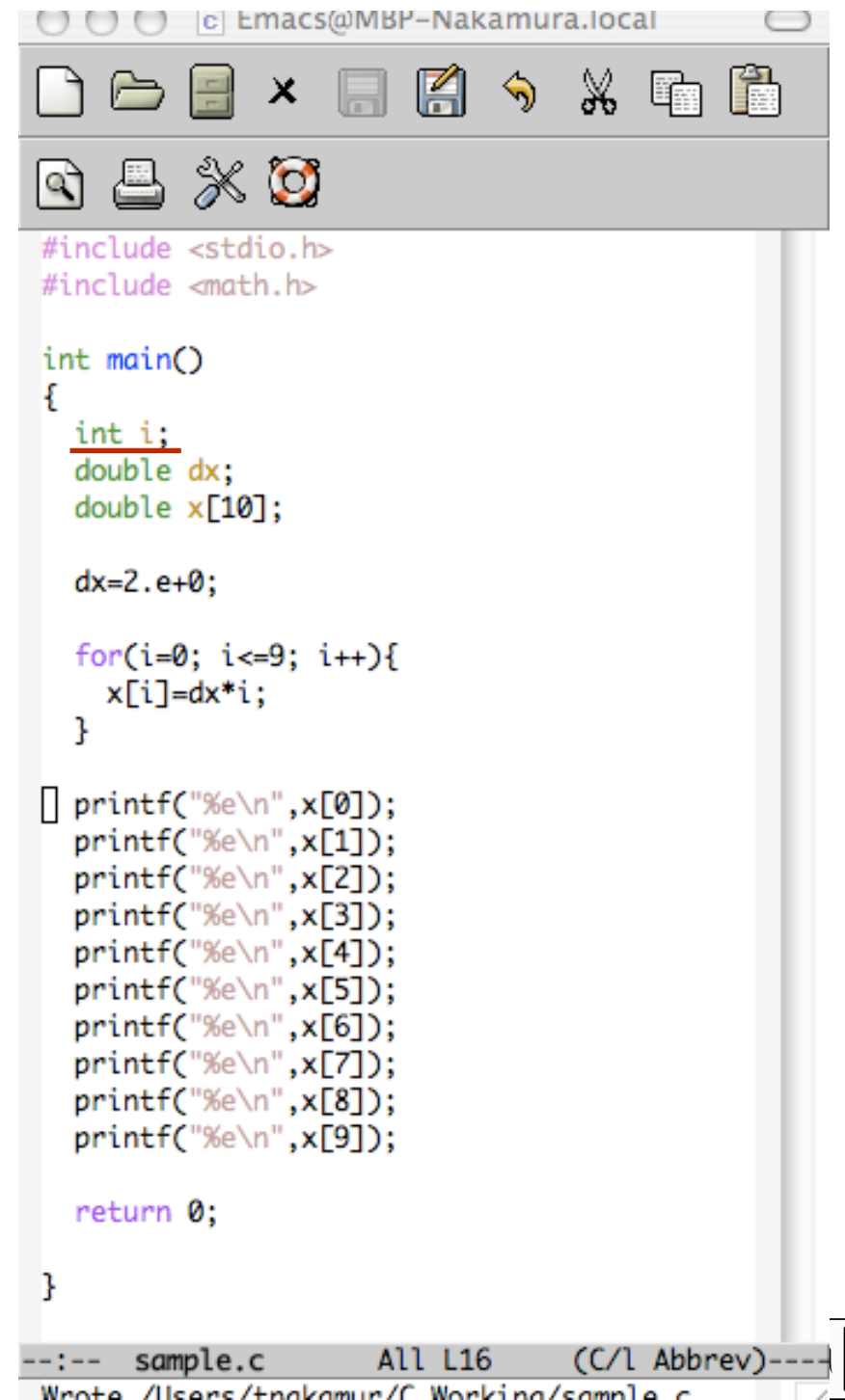
iを1ずつ増やして同じ計算



forループを用いた繰り返し

```
for(i=0; i<=9; i++){  
    x[i]=dx*i;  
}
```

iが9以下である間で、
i=0からiを一つずつ増やしながら括弧の中
を計算する。



```
#include <stdio.h>  
#include <math.h>  
  
int main()  
{  
    int i;  
    double dx;  
    double x[10];  
  
    dx=2.e+0;  
  
    for(i=0; i<=9; i++){  
        x[i]=dx*i;  
    }  
  
    printf("%e\n",x[0]);  
    printf("%e\n",x[1]);  
    printf("%e\n",x[2]);  
    printf("%e\n",x[3]);  
    printf("%e\n",x[4]);  
    printf("%e\n",x[5]);  
    printf("%e\n",x[6]);  
    printf("%e\n",x[7]);  
    printf("%e\n",x[8]);  
    printf("%e\n",x[9]);  
  
    return 0;  
}
```

--:-- sample.c All L16 (C/l Abbrev)----

```
Emacs@MBP-Nakamura.local

#include <stdio.h>
#include <math.h>

int main()
{
    int i;
    double dx;
    double x[10];

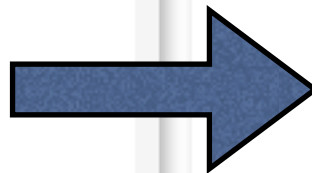
    dx=2.e+0;

    for(i=0; i<=9; i++){
        x[i]=dx*i;
    }

    printf("%e\n",x[0]);
    printf("%e\n",x[1]);
    printf("%e\n",x[2]);
    printf("%e\n",x[3]);
    printf("%e\n",x[4]);
    printf("%e\n",x[5]);
    printf("%e\n",x[6]);
    printf("%e\n",x[7]);
    printf("%e\n",x[8]);
    printf("%e\n",x[9]);

    return 0;
}
```

--:-- sample.c All L16 (C/l Abbrev)-----
Wrote /Users/tnakamur/C_Working/sample.c



```
Emacs@MBP-Nakamura.local

#include <stdio.h>
#include <math.h>

int main()
{
    int i;
    double dx;
    double x[10];

    dx=2.e+0;

    for(i=0; i<=9; i++){
        x[i]=dx*i;
    }

    for(i=0; i<=9; i++){
        printf("%e\n",x[i]);
    }

    return 0;
}
```

--:-- sample.c All L16 (C/l Abbrev)-----
Beginning of buffer

