陽解法

計算法

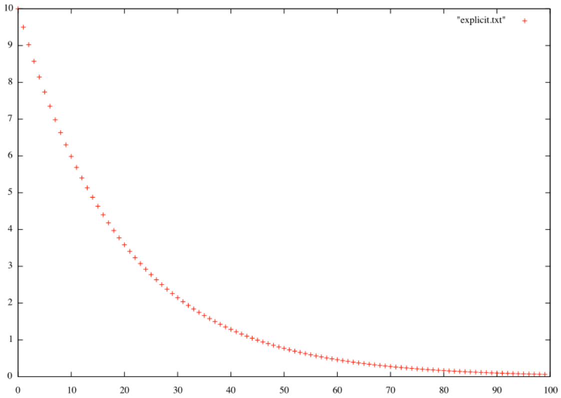
$x_{n+1} - x_n / \delta t = -\lambda x_n$ より

$x_{n+1} = (1 - \lambda \delta t) x_n$

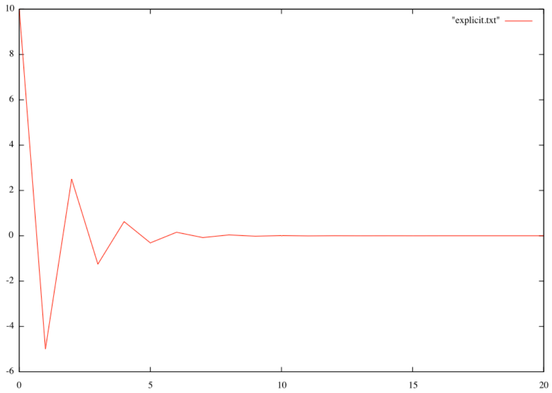
ソース

```
#include <stdio.h>

int main(void)
{
        int lambda = 5; // λ
        double delta = 0.01; // δt
        double x = 10; // x
        int i; //counter

        for (i=0; i<100; i++) {
                printf("%f¥n",x);
                x *= (1 - lambda * delta);
        }
        return 0;
}
```

グラフ



δ t = 0.01



δ t = 0.3

陰解放

計算法

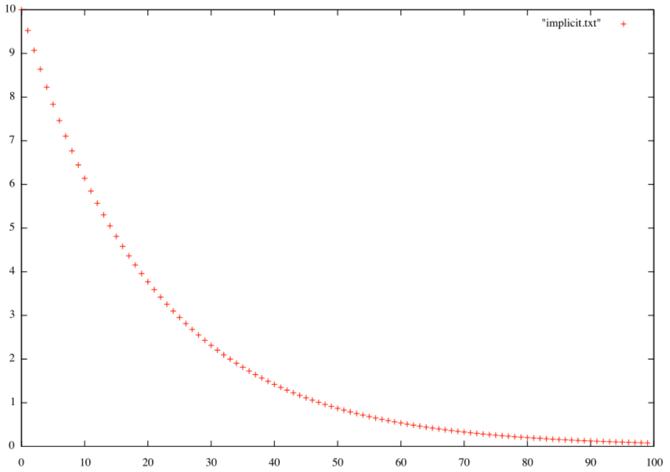
$x_{n+1} - x_n / \delta t = - \lambda x_{n+1}$ より

$x_{n+1} = x_n / (1 + \lambda \delta t)$

ソース

```
#include <stdio.h>

int main(void)
{
    int lambda = 5; // λ
    double delta = 0.01; // δt
    double x = 10; // x
    int i; //counter

    for (i=0; i<100; i++) {
        printf("%f¥n",x);
        x /= (1 + lambda * delta);
    }
    return 0;
}
```

グラフ



t = 0.01

中央差分

計算法

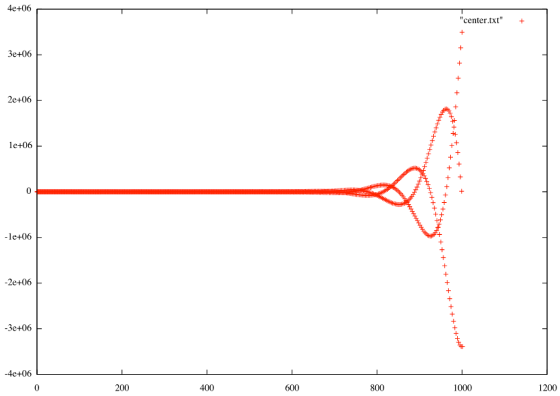
$(x_{n+1} - x_{n-1}) / 2 \delta t = - \lambda x_n$ より

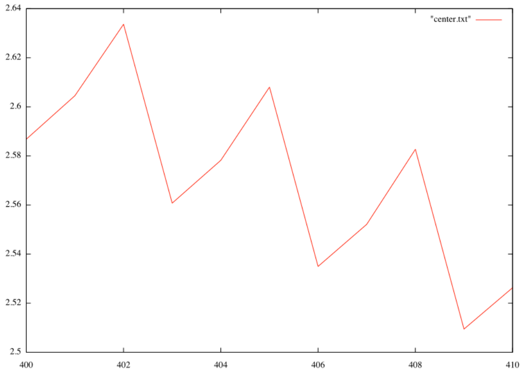
$x_{n+1} = x_{n-1} - 2 \delta t x_n$

ソース

```
#include <stdio.h>

int main(void)
{
    int lambda = 5; // λ
    double delta = 0.001; // δt
    double x1 = 10;
    double x2 = x1 * (1 - lambda * delta);
    double x3 = x1 - 2 * delta * lambda * x2;
    double tmp;
    int i; //counter

        printf("%f¥n",x1);
        printf("%f¥n",x2);

    for (i=0; i<1000; i++) {
        tmp = x3;
        x3 = x1 - 2 * delta * lambda * x2;
        x1 = x2;
        x2 = tmp;
        printf("%f¥n",x3);
    }

    return 0;
}
```

グラフ


"center.txt"
4e+06
3e+06
2e+06
1e+06
0
-1e+06
-2e+06
-3e+06
-4e+06
0
200
400
600
800
1000
1200


"center.txt"
2.64
2.62
2.6
2.58
2.56
2.54
2.52
2.5
400
402
404
406
408
410