

2階微分方程式共通

特別付録

```
#include <stdio.h>
#include <math.h>
double dydx(double x, double y, double z);
double dzdx(double x, double y, double z);
int main(void) {
    int i, n = 100;
    double x = 0.0, y = 0.0, z = 0.0;
    double dx = 0.01;
    return 0;
}
```

各アルゴリズムのプログラム

問題にあわせて変更する

32

33

2階微分方程式共通

```
double dydx(double x, double y, double z) {
    return z;
}
double dzdx(double x, double y, double z) {
    double ret;
    ret = 1.0 - 2.0*z - y;
    return ret;
}
```

問題にあわせて変更する

34

オイラー法

```
double dy, dz;
printf("%f %f %f\n", x, y, z);
for (i=0; i<n; i++) {
    dy = dydx(x, y, z)*dx;
    dz = dzdx(x, y, z)*dx;
    x = x + dx;
    y = y + dy;
    z = z + dz;
    printf("%f %f %f\n", x, y, z);
}
```

35

修正オイラー法

```
double dy1, dy2, dz1, dz2;
printf("%f %f %f\n", x, y, z);
for (i=0; i<n; i++) {
    dy1 = dydx(x, y, z)*dx;
    dz1 = dzdx(x, y, z)*dx;
    dy2 = dydx(x + dx, y + dy1, z + dz1)*dx;
    dz2 = dzdx(x + dx, y + dy1, z + dz1)*dx;
    x = x + dx;
    y = y + (dy1 + dy2)/2.0;
    z = z + (dz1 + dz2)/2.0;
    printf("%f %f %f\n", x, y, z);
}
```

36

ルンゲ・クッタ法

```
double dy1, dy2, dy3, dy4, dz1, dz2, dz3, dz4;
printf("%f %f %f\n", x, y, z);
for (i=0; i<n; i++) {
    dy1 = dydx(x, y, z)*dx;
    dz1 = dzdx(x, y, z)*dx;
    dy2 = dydx(x + dx/2., y + dy1/2., z + dz1/2.)*dx;
    dz2 = dzdx(x + dx/2., y + dy1/2., z + dz1/2.)*dx;
    dy3 = dydx(x + dx/2., y + dy2/2., z + dz2/2.)*dx;
    dz3 = dzdx(x + dx/2., y + dy2/2., z + dz2/2.)*dx;
    dy4 = dydx(x + dx, y + dy3, z + dz3)*dx;
    dz4 = dzdx(x + dx, y + dy3, z + dz3)*dx;
    x = x + dx;
    y = y + (dy1 + 2.0*(dy2 + dy3) + dy4)/6.0;
    z = z + (dz1 + 2.0*(dz2 + dz3) + dz4)/6.0;
    printf("%f %f %f\n", x, y, z);
}
```

37