

## コマンド名

ode - 常微分方程式の数値解を求める

## 起動方法

**ode** [*options*][*file*]

## 解説

**ode** は指定された一階の常微分方程式系の初期値問題を数値解法によって解くツールであり、三種類の数値積分法を利用できる。ルンゲ・クッタ・フェールベルグ法、アダムス・ムルトン法、オイラー法の三つである。このアダムス・ムルトン法とルンゲ・クッタ法では可変ステップ幅で解くことができる。

**ode** コマンドの動作は、専用の言語で書かれたプログラムで指示される。プログラムは積分したい変数の導関数の式の単純なリストや制御文などからなる（後述の例を参照）。

**ode** コマンドは指定されたファイル、指定されていない場合は標準入力からプログラムを読み込む。標準入力の場合は、ピリオド一つだけからなる行が入力された時点で動作を終了する。

数値積分の各ステップにおいて、プログラム中で指定された変数の値が出力される。したがって出力は表の形式になり、各列がそれぞれ一つの変数に対応する。列が二つしかない場合は、出力は直接 **graph**(1) コマンドや類似のコマンドに渡すことができる。

## 起動オプション

### 入力に関するオプション

**-f** *file*

**--input-file** *file*

標準入力から読み込む前に、指定されたファイルから読み込む。これを使うと、微分方程式系の定義などのプログラムの一部をファイルから読み込んだあとに、コマンドラインで対話的に使うことができる。

### 出力に関するオプション

**-p** *prec*

**--precision** *prec*

数値解を出力する際に、*prec* で指定された桁数の精度で出力する（デフォルトは 6）。このオプションが指定されたときは、出力形式は scientific notation になる（訳注：普通に出力して指定された桁数に収まるときはそのまま、収まらないときは 1.2345e-8 のように E 記法で出力する）。

**-t**

**--title**

最初に各列の変数名をタイトル行として出力する。出力形式は scientific notation になる。

### 数値積分法に関するオプション

以下のオプションは数値積分のアルゴリズムを指定するために用いる。**-R**、**-A**、**-E** の基本的なオプションについては、どれか一つだけを指定する。デフォルトは **-R**（ルンゲ・クッタ・フェールベルグ法）である。

**-R** [*stepsize*]

**--runge-kutta** [*stepsize*]

五次のルンゲ・クッタ・フェールベルグ法で数値積分する。 *stepsize* が指定されていればその値による固定ステップ幅で、そうでなければ可変ステップ幅で積分する。ステップ幅が指定され、かつ誤差解析が何も指定されていなければ、四次のルンゲ・クッタ法が使われる。

**-A** [*stepsize*]

**--adams-moulton** [*stepsize*]

四次のアダムス・ムルトンの予測子修正子法で数値積分する。 *stepsize* が指定されていればその値による固定ステップ幅で、そうでなければ可変ステップ幅で積分する。もし「悪い("bad")」点が生じたときは、ルンゲ・クッタ・フェールベルグ法を用いる。

**-E** [*stepsize*]

**--euler** [*stepsize*]

固定ステップ幅でオイラー法による高速だが誤差の大きな数値積分を行う。ステップ幅のデフォルト値は 0.1 である。重要な問題のために用いるのは推奨できない。

このオプションを使うときは、誤差範囲オプションの **-r** や **-e** (後述) は使わない。

### 誤差範囲オプション

**-r** *rmax* [*rmin*]

**--relative-error-bound** *rmax* [*rmin*]

このオプションで、1ステップでの相対誤差の上限を設定する。このオプションが指定されたときは、どの従属変数に対しても1ステップでの相対誤差が *rmax* を超えないように計算が行われる (デフォルト値は  $10^{-9}$ )。もしこの誤差を超えた場合は、その出力は捨てられ、エラーメッセージが表示される。ステップ幅が可変の場合は、ステップ幅は「適応的に」縮められ、1ステップでの相対誤差が指定された限界を超えないように調整される。したがって1ステップでの相対誤差の上限を小さくすると、ステップ幅も小さくなる。ステップ幅を大きくするために、下限 *rmin* も指定できる (*rmin* のデフォルト値は  $rmax/1000$ )。

**-r** *emax* [*emin*]

**--absolute-error-bound** *emax* [*emin*]

**-r** と同様だが、1ステップでの相対誤差ではなく絶対誤差を指定する。

**-s**

**--suppress-error-bound**

1ステップでの誤差の上限を無視する。これにより誤差が大きくなっても積分は続けられるが、数値解が大きな誤差を持つ可能性がある。

### その他のオプション

**--help**

コマンドラインオプションのリストを表示して終了する。

**--version**

**ode** コマンドとその GNU plotting utilities のバージョンを表示して終了する。

### 診断メッセージ

多くの場合はメッセージを読めば分かる。最も難しいのは 'syntax error' で、これはプログラムに文法上の間違いがあることを意味する。このメッセージは以下の形式で表示される。

**ode: nnn: message...**

ここで **'nnn'** は入力プログラム中でエラーになる場所の行番号である。もし **-f** オプションが指定されていれば、ファイル名のあとにエラーのあったファイル中の行番号が示される。それに続いて標準入力から読み込んでいる場合は、行番号は1にリセットされて始まる。

## 例

以下のプログラム

```
y' = 1
y = 1
print t, y
step 0, 1
```

は  $y = e^t$  の初期値問題を解く。**ode** がこのプログラム実行すると、標準出力に2列で数値が出力される。各行には、0 から 1 までの  $t$  の各値に対して、独立変数  $t$  と従属変数  $y$  の値が表示される。

もう少し洗練された例を挙げる。

```
sine' = cosine
cosine' = sine
sine = 0
cosine = 1
print t, sine
step 0, 2*PI
```

このプログラムは二つの微分方程式からなる系の初期値問題を解く。この初期値問題はすなわち正弦関数および余弦関数の定義でもある。このプログラムは、ちょうど一周期にわたって解を得る。

## 作者

**ode** は Nicholas B. Tufillaro (nbt@reed.edu) が作ったものを Robert S. Maier (rsm@math.arizona.edu) が拡張して GNU plotting utilities に取り込んだ。

## 参照

“The GNU Plotting Utilities Manual”

## バグ報告

bug-gnu-util@gnu.org で、E-mail にて受け付けている。

## 日本語訳について

この日本語訳は、富永大介 (<http://www.cbrc.jp/~tominaga/>、tominaga@cbrc.jp) が、平成18年10月17日に行った。ライセンスは GPL version 2 とする。

以上