

シミュレーション物理2 プログラミングの基本

大槻東巳

出席の取り方

- Mail address:
- 件名(subject)に**学生番号**, **氏名**を書くこと。
本文は空でもいいです。

プログラムの基本

1. 問題の解析, 無次元化など
2. データの型, アルゴリズムなどを考える
3. プログラム書き(coding)-->ソースコード
4. コンパイル(ソースコードをコンピュータが解釈できるようにすること)-->実行ファイル
5. 実行とテスト, エラーの除去(compile-time error, run-time error, logic error)
6. メンテナンス

今日の課題

- プログラムを作ってみる。
- サーバに転送する。
- サーバ上でコンパイル
- サーバ上で実行

Step 1の具体例

$$m \frac{d^2 \vec{r}}{dt^2} = -\frac{GMm}{r^3} \vec{r}$$

$$\rightarrow \frac{d^2 \vec{r}}{dt^2} = -\frac{GM}{r^3} \vec{r} \rightarrow 2D(\text{using angular momentum cons.})$$

$$\rightarrow r/l = \tilde{r}, t/T = \tilde{t}, \frac{d^2 \tilde{r}}{d\tilde{t}^2} = -\left(\frac{GMT^2}{l^3}\right) \frac{1}{\tilde{r}^3} \tilde{r}$$

$$\rightarrow \text{set } \frac{GMT^2}{l^3} = 1$$

$$\rightarrow \frac{d^2 \tilde{r}}{d\tilde{t}^2} = -\frac{1}{\tilde{r}^3} \tilde{r}, \tilde{r} = (\tilde{x}, \tilde{y})$$

Note: This is nothing but Kepler's law!

Step 2: data

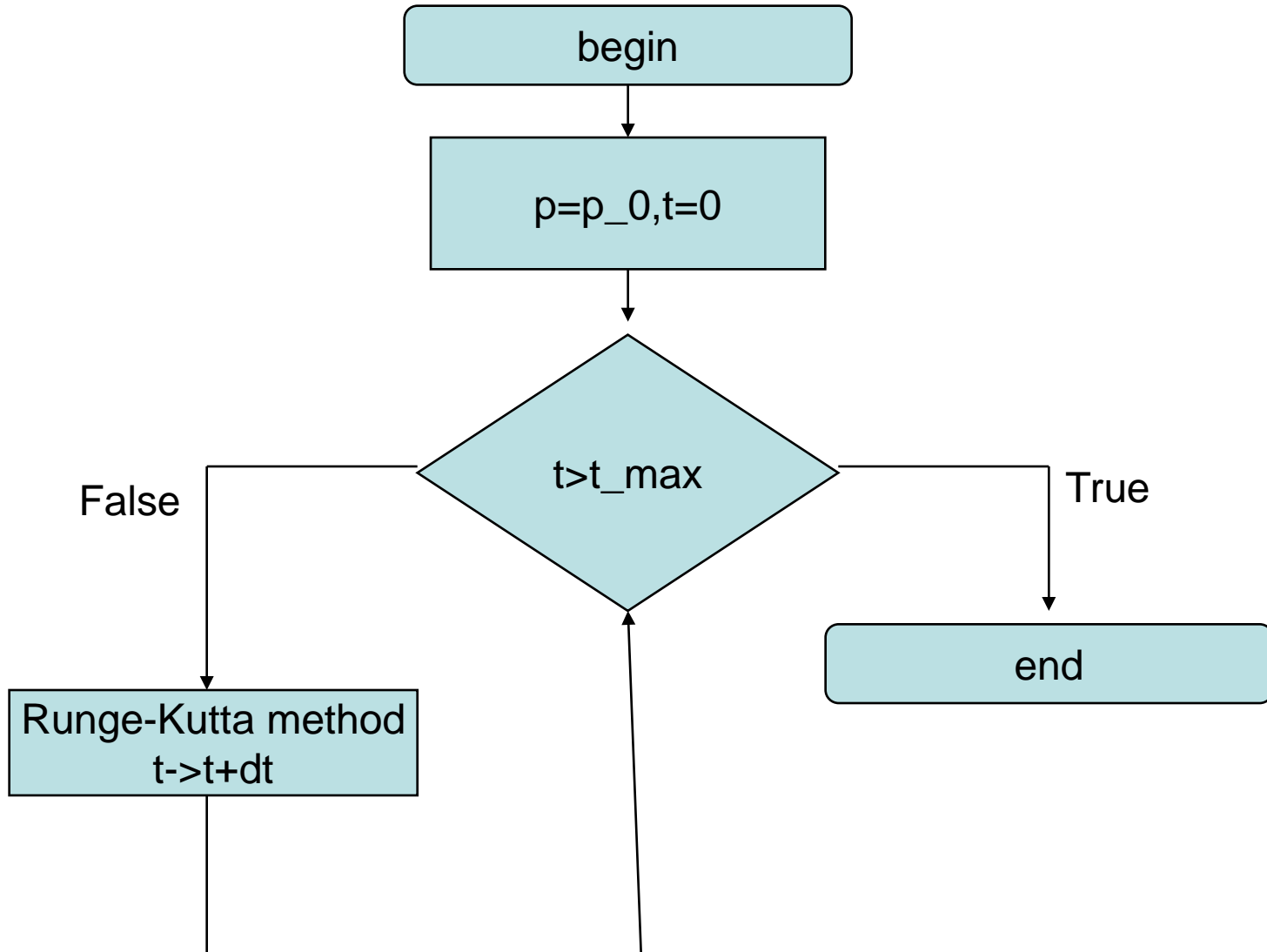
Data: $x, y, v_x, v_y, t, t_{\max}, dt$

Algorithm:

$$\dot{x} = v_x, \dot{y} = v_y, \dot{v}_x = \frac{-x}{\sqrt{x^2 + y^2}}, \dot{v}_y = \frac{-y}{\sqrt{x^2 + y^2}}$$

$$\vec{p}(t) = (x, y, v_x, v_y)$$

Step 2: flow chart



Example

- Riemann Zeta function:
 - 物理によく出てくる
 - 素数の分布などにも有効
 - 1億円の懸賞問題にもなってる ($\zeta(x) = 0, \operatorname{Re} x = \frac{1}{2}$)

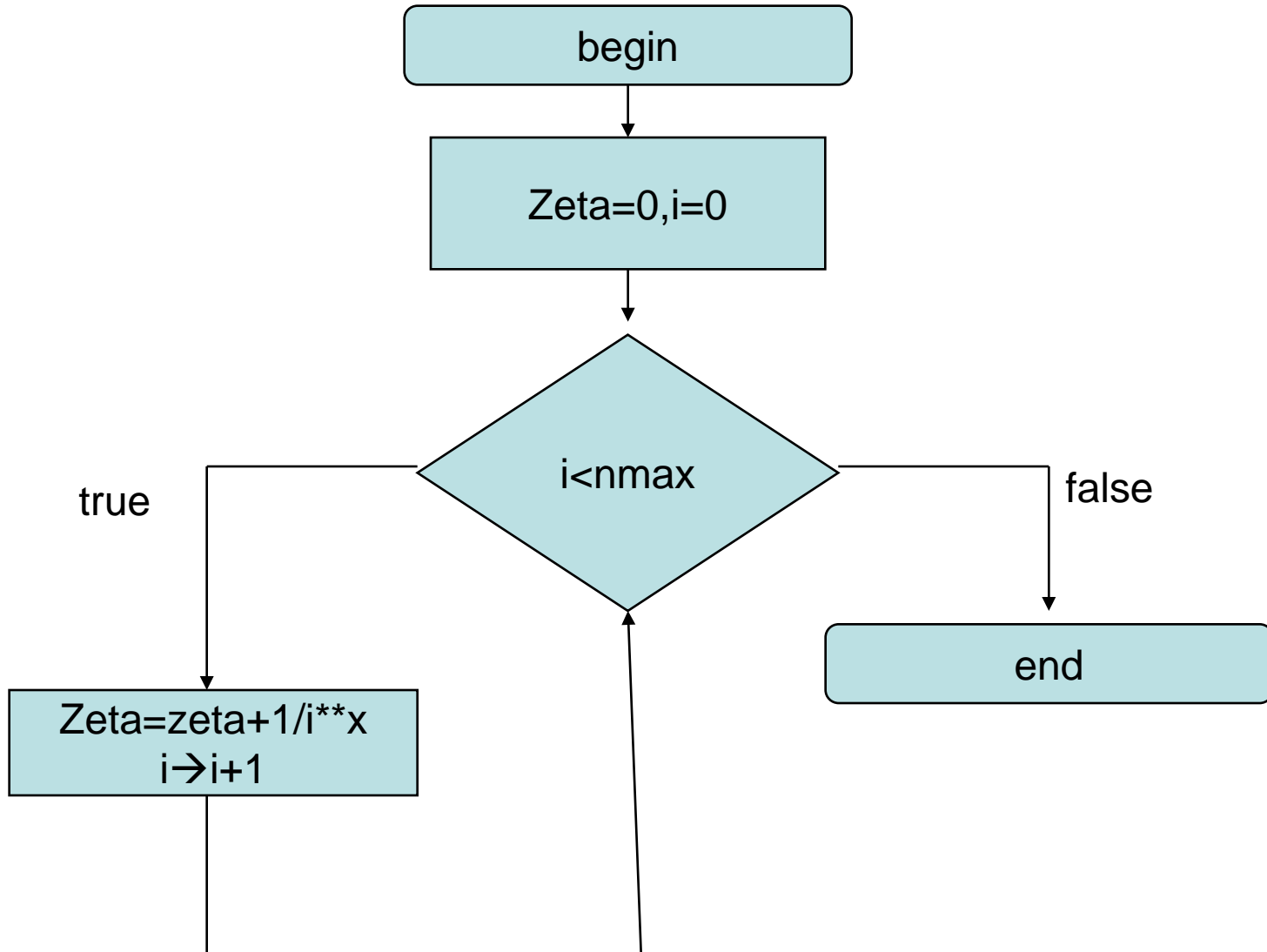
$$\zeta(x) = \sum_{n=1}^{\infty} \frac{1}{n^x},$$

$$\zeta(2) = \frac{\pi^2}{6}, \zeta(4) = \frac{\pi^4}{90}$$

Step 1

- この問題はもう、無次元化されている
- Algorithm
 - とりあえず大きい数(n_{\max})までたしてみればい
だらう
- データ
 - 整数: i (loopに使う), n_{\max}
 - 実数: x, ζ

Flow chart



- コンパイル

- f90 filename(必ず.f90で終わるファイル)
- a.outというファイルができるのでそれを実行(a.outと打ち込む)
- もしa.outでなく、たとえばzetafunctionという名前の実行ファイル(キーボードで打ち込むと結果が出るものを実行ファイルという)がほしければ
 - f90 -o zetafunction zeta.f90
- zetafunctionが実行ファイル, zeta.f90がソースファイル

コンピュータールームCでの手順

- メモ帳でプログラムを書く
- ftpでファイルをdahlmanに転送
- dahlmanにlogin (teratermを使う)
- プログラムをコンパイル
 - f90 zeta.f90
 - f90 -o zetafunction zeta.f90
- zetafunction と打ち込んで実行

```
program Zeta_Function
!-----
! This is a program to calculate Riemann Zeta function
!2005/4/20 Written by T. Ohtsuki
!-----
implicit none ! Always begin with this statement
real, parameter::zero=0.0
real:: zeta,x
integer,parameter::nmax=1000000
integer::i

Print *, "Enter x"
Read *, x

zeta=zero
SumOverl: do i=1,nmax
zeta=zeta+1./real(i)**x
end do SumOverl

print *, zeta

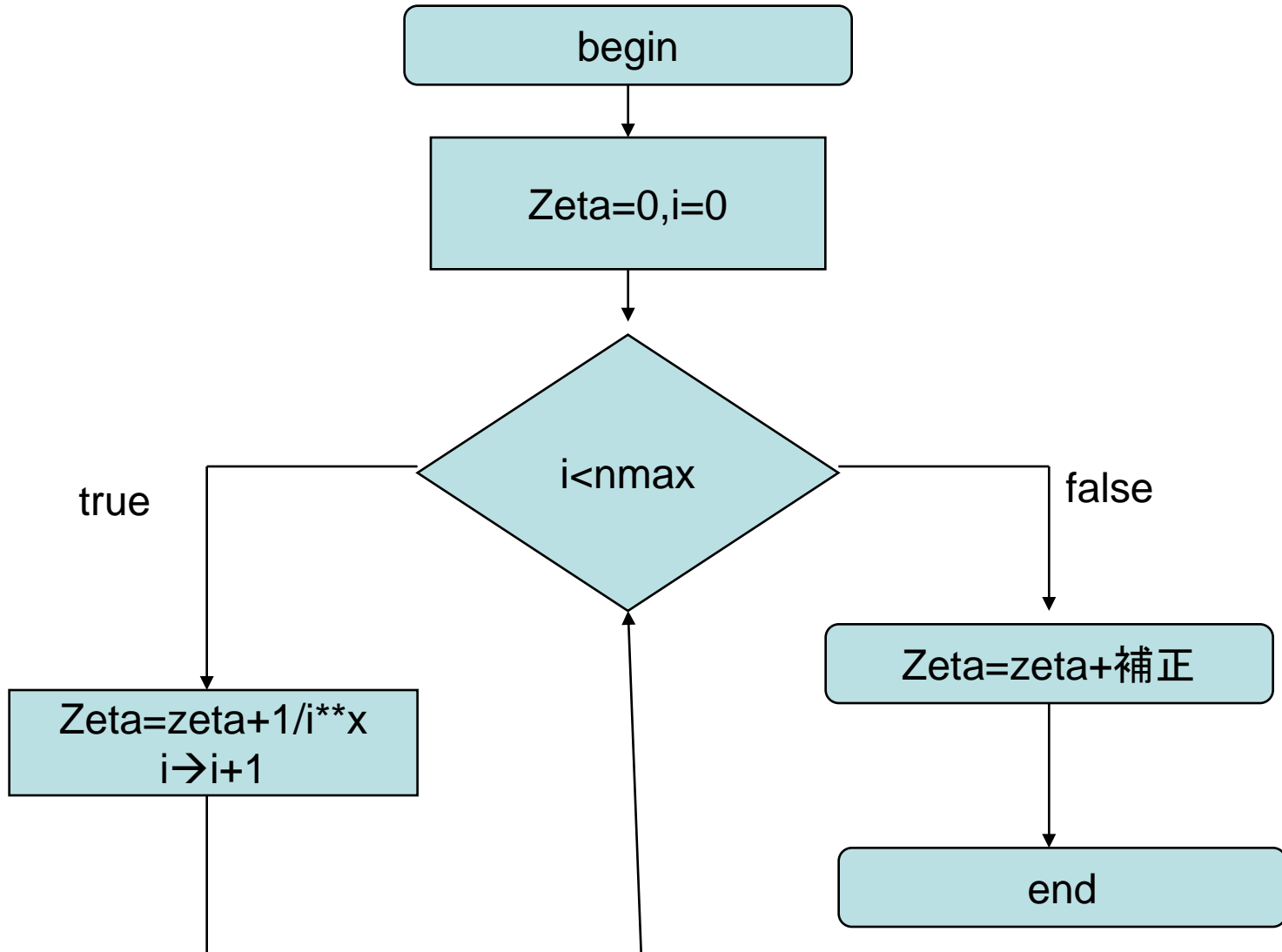
stop
end
```

無視した $n \geq n_{\max}$ の項の取り扱い

- 和を積分で置き換える

$$\sum_{k=n}^{\infty} \frac{1}{k^x} \approx \int_n^{\infty} dk \frac{1}{k^x} = \frac{-1}{x-1} \left[\frac{1}{k^{x-1}} \right]_n^{\infty} = \frac{1}{n^{x-1}}$$

Flow chart



```
program Zeta_Function
```

```
!-----
```

```
! This is a program to calculate Riemann Zeta function
```

```
! 2005/4/20 Written by T. Ohtsuki
```

```
! revised by improving the sum using integral correction
```

```
!-----
```

```
implicit none ! Always begin with this statement
```

```
real, parameter::zero=0.0
```

```
real:: zeta,x
```

```
integer,parameter::nmax=1000000
```

```
integer::i
```

```
Print *, "Enter x"
```

```
Read *, x
```

```
zeta=zero
```

```
SumOverl: do i=1,nmax
```

```
zeta=zeta+1./real(i)**x
```

```
end do SumOverl
```

```
zeta=zeta+1./real(nmax+1)**(x-1)/(x-1.)
```

```
print *, zeta
```

```
stop
```

```
end
```