

数值計算

垣谷公徳
17号館3階
電子メール: kimi@ee.ous.ac.jp

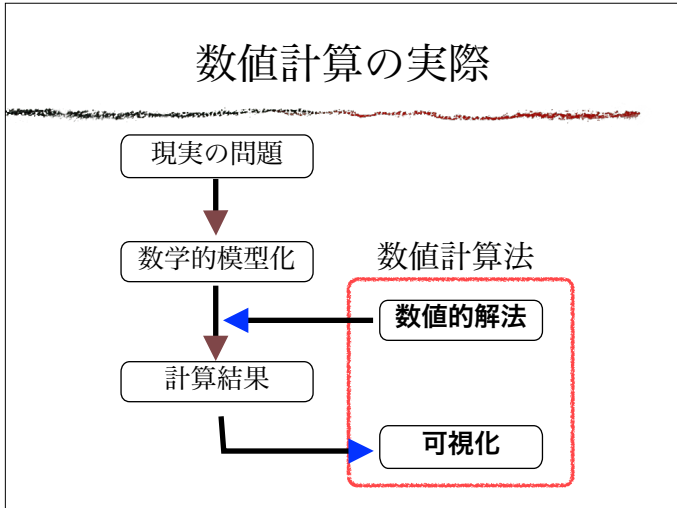
1

講義資料

◦ <http://sstxp.ee.ous.ac.jp/sstlab/>



2



3

- ## 数值計算の応用範囲
- 軌道計算
 - 物質設計 (電子状態・原子配列)
 - 気象予測
 - 機械設計 (エンジン・空力特性)
 - 電磁波解析 (アンテナ・導波管)
 - 回路設計 (回路シミュレータ)
 - ロボット制御
 - 画像処理・音声処理・暗号化

4

- ## 問題の数学的模型化
- 軌道計算
 - ニュートン方程式
 - 二つの対象物しか解析的に解けない
 - 人工衛星+地球+月
 - 物質設計
 - シュレディンガー方程式
 - 特別な場合しか解けない
 - 電子1個についてしか解けない

5

- ## 問題の数学的模型化
- 気象予測
 - 気温・気圧・地形
 - 熱拡散
 - 流体 (大気) の運動
 - 機械設計 (エンジン・空力特性)
 - 燃焼室内のガスの運動
 - ボディ形状による空気抵抗
 - 流体方程式

6

問題の数学的模型化

- 電磁波解析 (アンテナ・導波管)
 - マクスウェル方程式
 - ポアソン方程式
 - ラプラス方程式
- 回路設計
 - 半導体デバイスの静・動特性
 - 電気デバイス (LCR) の諸性質
 - 連立微分方程式

7

問題の数学的模型化

- ロボットの制御
 - 束縛条件のある力学問題
 - ラグランジュ関数・ハミルトン関数
 - 正準方程式 (解析力学)
- 画像処理・音声認識
 - CTスキャン・MRIの画像化
 - 微分方程式の逆問題
 - フーリエ変換・ウェーブレット変換

8

問題の数学的模型化

- 暗号化
 - 擬似乱数
 - 素因数分解
- 数値解析模型構築の困難な問題
 - モンテカルロシミュレーション
 - 擬似乱数
 - 遺伝的アルゴリズム

9

プログラミング言語の分類

- 汎用プログラミング言語
 - 低水準言語
 - 機械語・アセンブリ言語
 - 高水準言語
 - FORTRAN90 (77 or 95)
 - c (c++)
 - python
 - ...

10

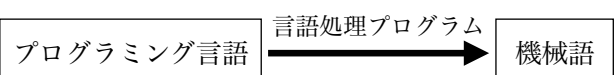
プログラミング言語の分類

- 汎用プログラミング言語
 - 高水準言語
 - 手続き型言語
 - 非手続き型言語
 - 構造化言語
 - オブジェクト指向言語

11

言語処理プログラム

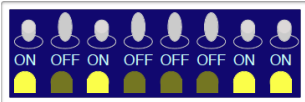
- 機械語
 - CPUが実行できる唯一の命令列
 - 通常0/1の2進数で表現される
 - 人間に理解し易いように16進数が使われることもある



12

機械語

- 実験用マイクロコンピュータボードで入力SWに対応するLEDを点灯させる



```
1101101111111001
1101001111111000
```

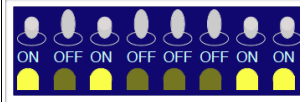
```
DB F9
D3 F8
```

ただし、入力SWのポート番号は11111001(F9)、出力LEDのポート番号は11111000(F8)とする

13

アセンブリ言語

- 機械語と一対一に対応



```
IN A, (F9) ;DB F9
OUT (F8), A ;D3 F8
```

入力SWのポート番号は(F9), 出力LEDのポート番号は(F8)
プロセッサはZ80

```
IN A, (??) → DB ??
OUT (??), A → D3 ??
```

14

低水準言語

- 機械語
- アセンブリ言語
- メリット
 - プログラムを小さくすることができる
 - プログラムを高速に実行することができる
- デメリット
 - 人間に解りにくい
 - 特定のハードウェアに依存する

15

高水準言語

- 人間の言語表現に近い構造をもつ
- 記述に、汎用性・可搬性がある
- 特定のハードウェアに依存しない
- 例
 - Fortran, COBOL, pascal, c, c++, java,
 - python, lisp, prolog, SQL

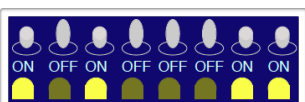
16

高水準言語

- c言語

```
unsigned int a;
a = get_SW_status();
set_LED_on(a);
```

入出力関数は専用ライブラリで用意されなければならない



- Fortran

```
INTEGER A
READ (9, *) A
WRITE (8, *) A
```

入出力装置番号8/9はシステムにより定義されていないなければならない

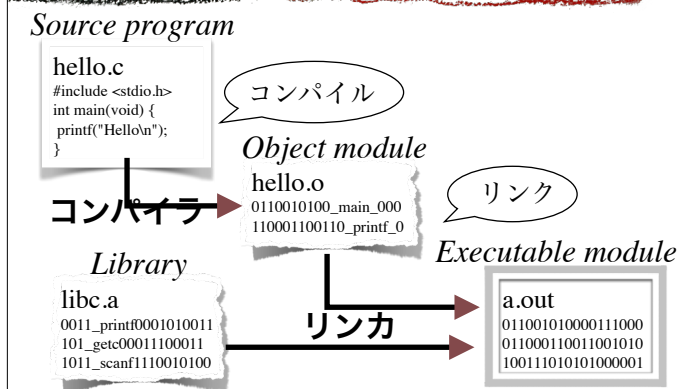
17

言語処理プログラムの種類

- コンパイラ(翻訳型)
 - 高水準言語で書かれたプログラムを機械語に一括翻訳した後、実行する
 - 高速、エラーを取り除くのが難しい
- インタープリタ(解釈型)
 - 高水準言語で書かれたプログラムを一行ずつ機械語に解釈し逐次実行する
 - 低速、扱い易い

18

プログラム作成から実行まで



19

プログラム作成・実行手順

- 原始プログラム(Source program)の作成
- 原始プログラムをコンパイルし目的プログラム(Object module)に変換
- 目的プログラムを関係編集(link)し、実行可能プログラム(Executable module/Load(able) module)に変換
 - リンク(関係編集)
 - リンケージエディタ(リンカ)によって、目的プログラムのサブルーチン、関数のアドレス定義を行いライブラリなどと結合させて実行可能プログラムを作成する
- 実行可能プログラムをメモリ上にロードし実行する

20

プログラム作成用語集

- 原始プログラム(source program/source code/source file)
 - プログラム言語によって記述されたもの
- 目的プログラム(object module/object code/object file)
 - コンパイラによって変換されたもの
 - アドレスなどが未定義のため実行不可
- 実行可能プログラム(executable file/load module/command)
 - リンカ (リンケージエディタ) により、プログラムの実行に必要なサブルーチンや関数などを含めた計算機で実行可能なもの

21

プログラミング言語

- 数値計算に用いられる主なプログラミング言語
 - FORTRAN 66/77/90/95
 - c (c++, java)
 - python etc.

22

FORTRAN言語の特徴

- 最古の高水準言語
- 汎用構造化言語 (FORTRAN77以降)
 - FORTRAN90以降はオブジェクト指向
- 数値計算に適した言語仕様
- 過去の豊富な資産 (ライブラリ)
- 数値計算の分野では圧倒的なシェア

23

c言語の特徴

- 低水準な処理も可能な高水準言語
- 汎用構造化言語
 - オブジェクト指向言語(c++, java)の基礎
- ソフトウェア技術者の教養的知識
- 最新のライブラリ(過去の資産は使えない)
- 数値計算には不向き(FORTRANに比べて)

24