

# 計算物理基礎学:2025年度 担当:長友英夫

---

## ・講義内容

理工学における数値解析、計算シミュレーションを含む、計算物理学の重要性は高まっています。本講義では、計算物理に必要な数値計算法の基礎、偏微分方程式、力学系、ランダム過程、線形・非線形問題といった基本的な考え方について、具体的な例題をとりあげながら、数理工学の問題を解決するためのベースを培います。特に、解法の原理の理解とともに、自分でプログラミングすることによって数値解析の実習も行いながら学んでもらいます。

プログラミング用のPCは各自で準備し、必要なソフトウェアをインストールして下さい。  
(PCの準備が難しい場合は相談して下さい。)

## ・講義日程: 火曜日 第4限 15:10~16:40

4月 15日、22日

5月 7日(水曜日)、13日、20日、27日

6月 3日、10日、17日、24日

7月 1日、8日、15日、22日、29日、(8月5日)

## ・場所:工学部 GSE情報実習室B

・期末試験は行いません。毎回の出席とレポート提出状況で評価します。

・講義資料の配布、レポート提出、連絡等は大阪大学CLE(授業支援システム)を通して行います。

・講義資料は講義での利用のみ許可します。それ以外のネット等では絶対に共有、公開しないでください。

# 講義内容(予定)

---

1. 数値計算と数値誤差(丸め誤差、打切り誤差)
2. 代数方程式
  - 2-1. 2分法
  - 2-2. ニュートン法
3. 連立方程式
  - 3-1. 直接法: ガウス・ジョルダン法、ガウスの消去法
  - 3-2. 反復法: ヤコビ法、ガウス・ザイデル法
4. 補間法
  - 4-1. ラグランジュの補間多項式
  - 4-2. スプライン関数補間
  - 4-3. 最小2乗法
5. 数値積分
  - 6-1. 台形公式
  - 6-2. シンプソンの公式
6. 常微分方程式
  - 6-1. オイラー法
  - 6-2. ルンゲ・クッタ法
7. Pythonの活用(データ処理など)
8. 偏微分方程式
  - 8-1. 差分法
  - 8-2. 双曲型偏微分方程式(移流方程式解法)
  - 8-3. 楕円型偏微分方程式

# 大規模実験と計算シミュレーション

---

- 費用、時間的理由で実験回数が限られる。
- 技術的な困難さを伴う実験。短い時間(ナノ秒以下)、小さい空間(ミクロン)の現象のため、観察が難しいなど。

## コンピューターシミュレーション



- 自由な条件で模擬実験ができる。
- 任意の仮定を入れることができる。
- 任意の時間、空間の値を知ることができる。



- 結果の妥当性検証
  - シミュレーションのモデルは正しいのか？
  - 計算は正しく行われたか？
- 結果の解釈、ポストプロセス

# コンピューター(計算、数値)シミュレーション

---

- 計算機を用いて現象を模擬する。

模擬＝シミュレーション (Simulation)

- 問題の仮定、スケールを比較的自由に変えることができる。

経費、リスク、時間の節約

- 現実には異なる問題でも数学的には共通のモデルになることがある。

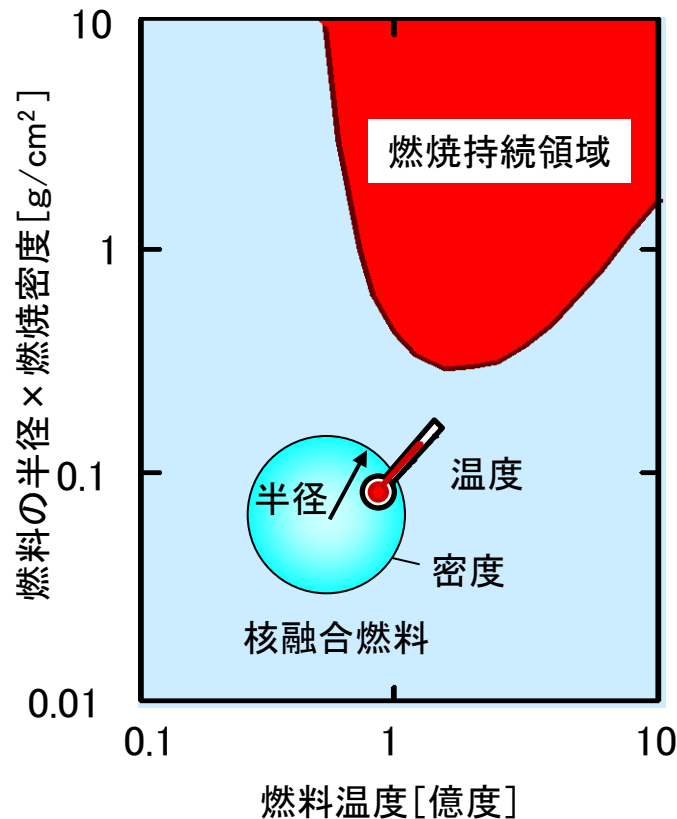
異なる問題も共通の手法で解決できる。

- 仮定、モデルの近似、数学的な誤差、計算誤差がある。

シミュレーションの仮定、限界を十分理解した上でシミュレーションを利用する。(無駄な計算、不正な解を利用しないようにする)

# レーザー核融合の燃焼維持条件(ローソン条件)

物が燃えるのと同様に、核融合反応を持続させるには次の3つの燃焼条件を満たす必要があります。  
(太陽は、重力によって中心部分でこの燃焼条件を満たしています。)



## 条件1: 高い温度



核融合燃料は電気を帯びた粒子(プラズマ)なので、その反発力のため普通の状態では核融合反応は起こりません。そこで、核融合反応を起こすためには燃料の原子核同士を非常に高速(毎秒800km)で衝突させる必要があります。このスピードは燃料を1億度以上に加熱することで得られます。  
(太陽の表面温度は5,500度)

## 条件2: 高い密度



核融合反応を効率良く起こすには、プラズマ同士が衝突する割合を大きくする必要があります。レーザー核融合では1ccの重さが200g以上のプラズマを作り、瞬間的に反応を終了させてエネルギーを取り出します。  
(太陽の中心密度156g/cm³)

## 条件3: 閉じ込める



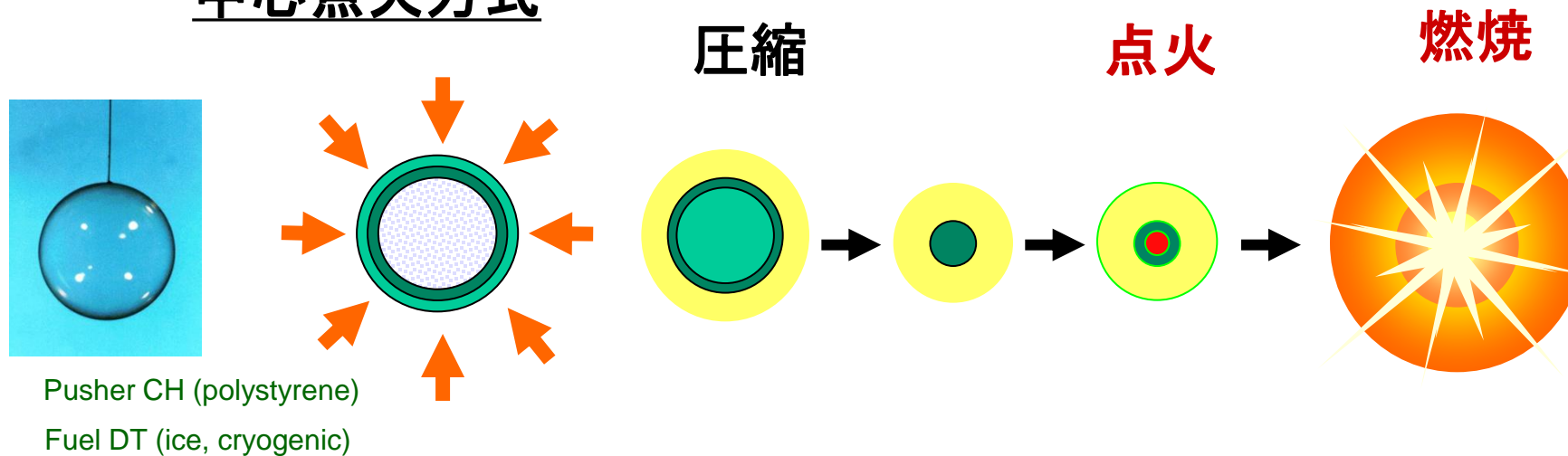
企業経営において投資と収益のバランス点が大切なように、燃焼をある程度維持しないと、核融合反応を起こす前に投入したエネルギーが回収できません。そのためにはある時間以上一定の領域に閉じ込めて燃焼をさせねばなりません。慣性核融合では燃料自体の質量による慣性力で閉じ込めるので、燃料の半径と密度の積がある値以上大きくなければなりません。

# 中心点火方式(直接照射方式)

燃料を充填した球状のシェルを、高強度レーザーを一様に照射し、均一に爆縮する。爆縮によって中心部分が高温、高密度になり、ローソン条件を満たすと点火・燃焼する。(R.E. Kidder (1968), J. Nuckolls (1972))

レーザーの照射一様性の条件が厳しく、大きな爆縮用レーザーが必要になる。[ NIF(米国)、LMJ(フランス)など ]

## 中心点火方式



燃料の密度半径積は、 $\rho R = 0.3 \text{ g/cm}^2$ 程度は必要である。この際、中心部分には点火に必要な10keV以上のホットスポットが形成されている必要があり、これは爆縮の均一性などに依存する。

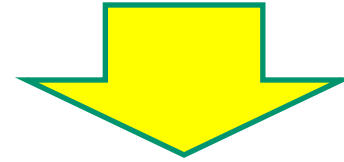
# シミュレーション科学とは

---

- 観測、実験で見出された現象
- 理論的に予想される現象

モデル化

⇒具体的実証手段



大型コンピュータの出現

•**理論、実験**に次ぐ第3の研究法

- 予測
- 新しい知見の創出

**シミュレーション科学**

応用数学、計算機科学、計算機工学

計算物理  
計算化学  
計算生物学  
...

データ処理

**データ科学**

# 計算シミュレーションと情報科学の融合(計算流体力学の例)

『フルードインフォマティクス「流体力学」と「情報科学」の融合』(日本機械学会編, 2010年)

- 流れの予測、診断、制御の必要性
- 実験計測技術
  - 様々な状態量の計測が可能
    - × 任意の広い分布の状態量を正確に計測することはできない
- レーザープラズマシミュレーションの基礎となる数値流体力学(CFD)分野
  - シミュレーション手法の開発の進展
  - 計算機性能の飛躍的向上
    - × 実現象の初期値、境界条件を与えない限り実現象をシミュレーションで再現することはできない。

レーザープラズマでは一般的な流体力学現象より難易度が高い

極限的な計測技術が求められる。  
積分値しか得られないこともある。

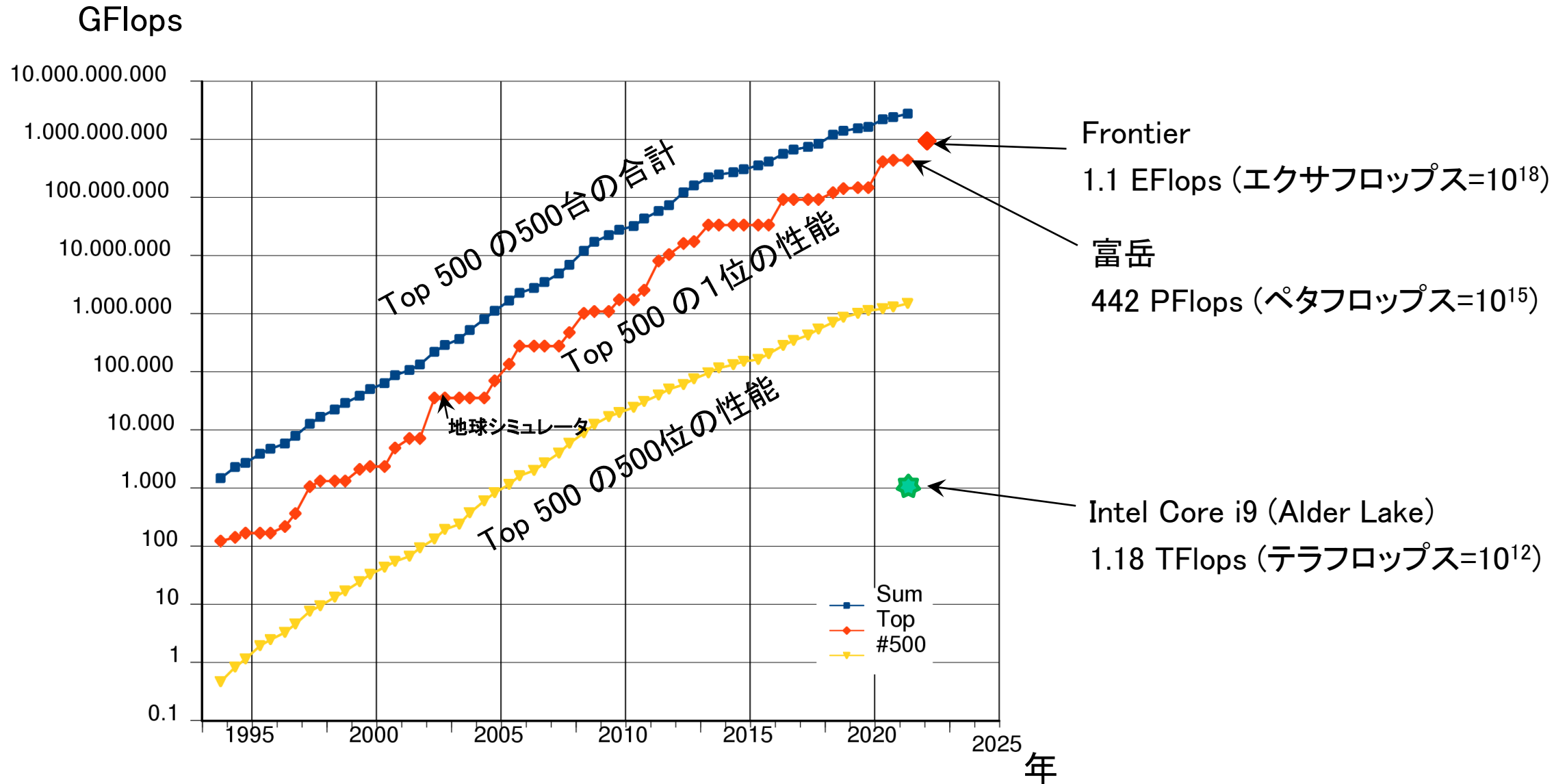
一段と複雑な非線形現象を伴う。  
それらを、数値モデルとして組み込んでいる。

⇒インフォマティクスを活用したデータ補間、関数展開

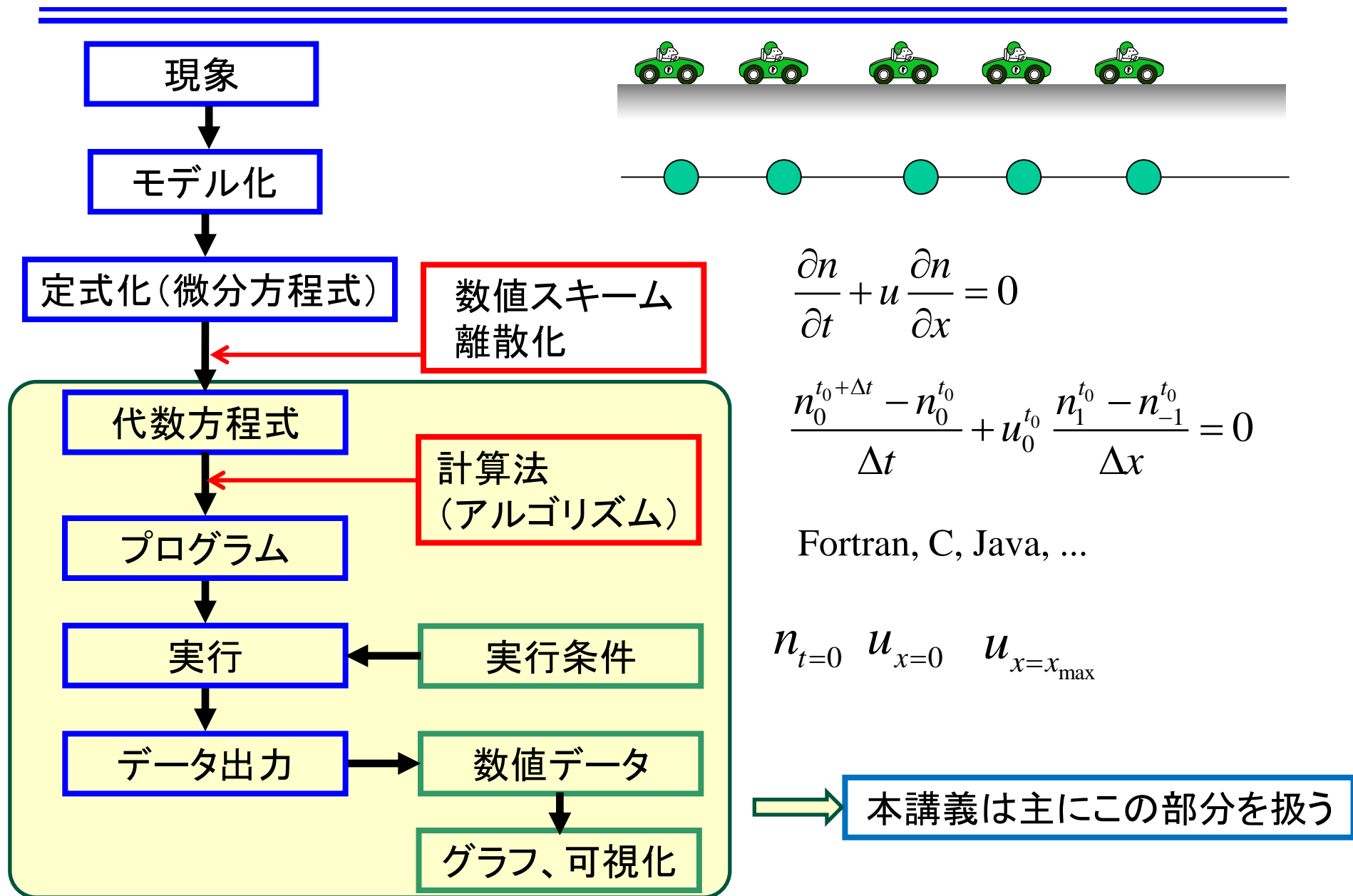
解析手法によるデータ同化に加え、  
機械学習によるデータ同化、その場  
観察への活用が期待できる。



# 世界のスーパーコンピューター TOP500



# 計算物理学における数値解析の流れ



# 数値計算法に求められる条件

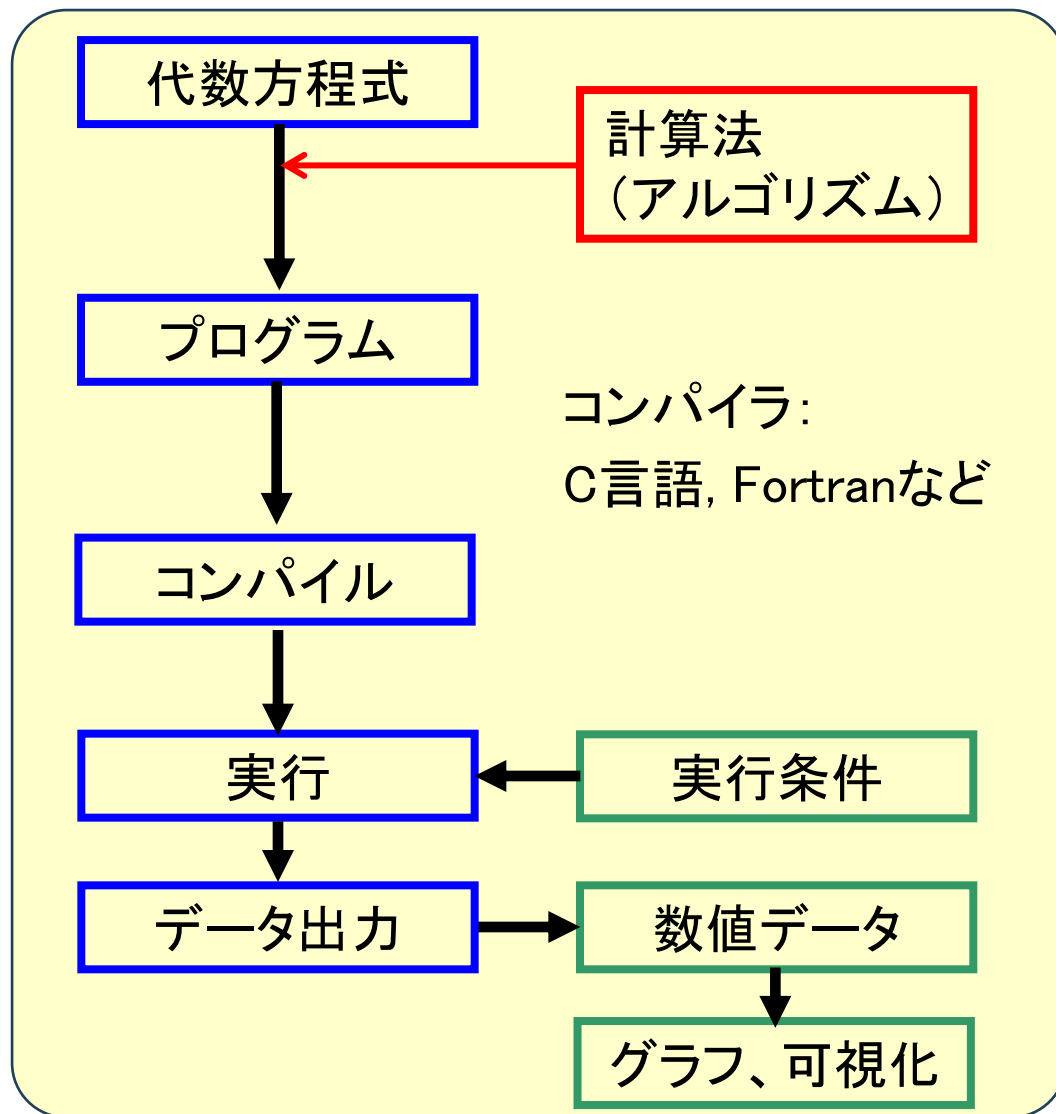
---

## 数値計算法に求められる条件

1. 一般性がある。
2. 計算が早い
3. 精度が高い。結果が信頼できる。
4. プログラムが単純である。
5. 現有の計算機資源で計算可能である。

# 一般的な計算・シミュレーションの手順

プログラムの作成から結果が得られるまでの流れ



⇒人間が理解しやすく、マシンの依存性を少なくする言語  
コンパイルすることによって、それぞれのコンピュータ用のマシン語に翻訳され実行ファイルを生成する。

⇔インタプリタ言語  
(コンパイラが不要な言語)  
Python、Java Script、Ruby

# 計算物理基礎学: プログラミング環境の準備について

---

この講義では講義内容に沿ったプログラミングの演習によって、講義内容の理解を深めてもらいます。  
そのための環境を構築してください。コンパイラは、C言語を準備して下さい。  
大きく分けて3つの方法があります。

方法1 コンパイラ(Visual Studio) とグラフ作成ソフト(gnuplot)をWindowsマシンに直接インストールする。  
[比較的容易]

方法2 WindowsのWSL2 (Windows Subsystem for Linux) を使って、Linuxの仮想環境を構築し、  
そのLinuxにコンパイラとグラフ作成ソフトをインストールする。 [ちょっと複雑]

方法3 WindowsにCygwinをインストールする。(WindowsにLinux風の環境を構築しますが、  
あまりお勧めしませんが、既に他の講義でインストールしている場合はそのまま使えます。)

方法1と方法2について次ページ以降に概要を説明します。

※MacOSの場合: Homebrewをインストールした上で、コンパイラをインストールするようです。  
詳細はネットで調べてください。

# 方法1 コンパイラとグラフ作成ソフトをWindowsマシンに直接インストール

---

コンパイラは、C言語をインストールして下さい。

- C言語(これが一番普及していると思われる)  
科学計算に広く使われています。無料版のMicrosoftのVisual Studio Communityでも基本的なプログラミングを行います。
- マイクロソフトの Visual Studio Community のサイト  
<https://visualstudio.microsoft.com/ja/vs/community/>
- グラフ作成  
グラフ作成ソフトのインストールについては、必要になるときに説明します。

# 方法1 ② Visual Studio Community 2022をインストールする

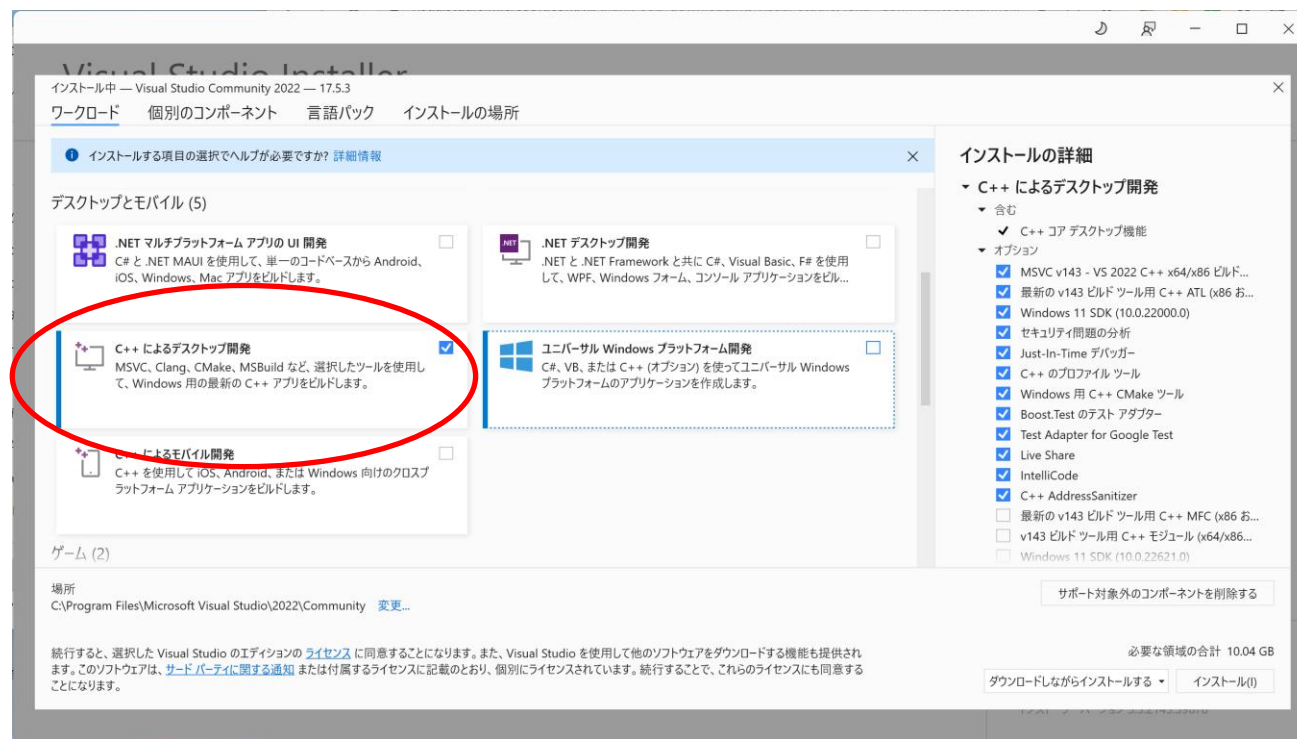
- ・ マイクロソフトの Visual Studio Community のサイトからインストーラをダウンロード

<https://visualstudio.microsoft.com/ja/vs/community/>

- ・ インストールする途中でインストールする項目を選択する画面になる。

「C++によるデスクトップ開発」を選択する。

他の項目は特に必要ない。



# Visual Studio Community 中の「開発者コマンドプロンプト」を使う

1. インストールした後、Windowsのスタートメニューから「すべて」を押してすべてのアプリを表示
2. 「Visual Studio 2022」の「Developer Command Prompt for VS 2022」（日本語で[開発者コマンド プロンプト for VS 2022]となっている場合があります。）

この操作が面倒であれば、ショートカットをデスクトップ、またはタスクバーに入れておくと簡単に起動させられます。

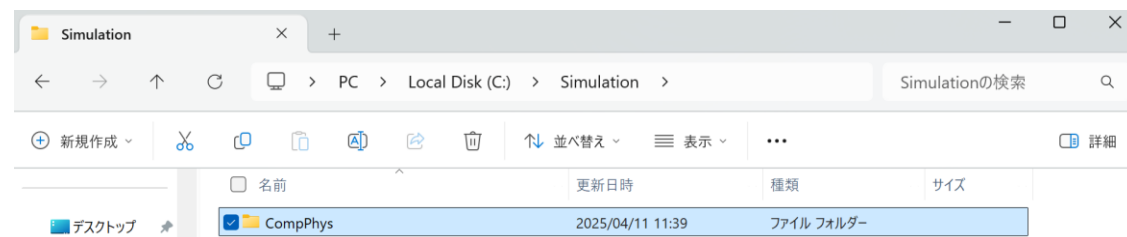
- ① スタートメニューのDeveloper Command Prompt for VS 2022を右クリックして「タスクバーにピン留め」を選択する。  
または;
- ② 「ファイルの場所を開く」を選びショートカットのあるフォルダを開き、ショートカットをデスクトップへコピーする。



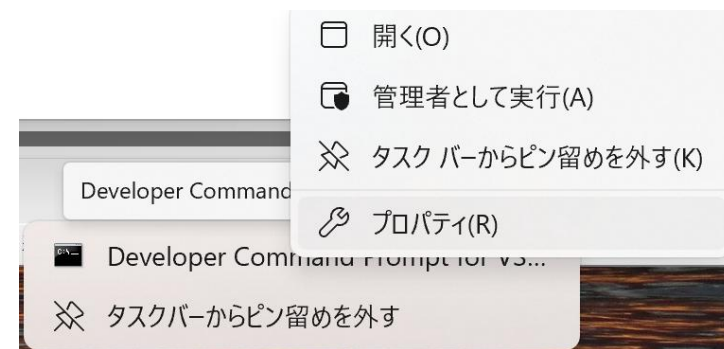


# Visual Studio Community 中の「開発者コマンドプロンプト」を使うための環境設定

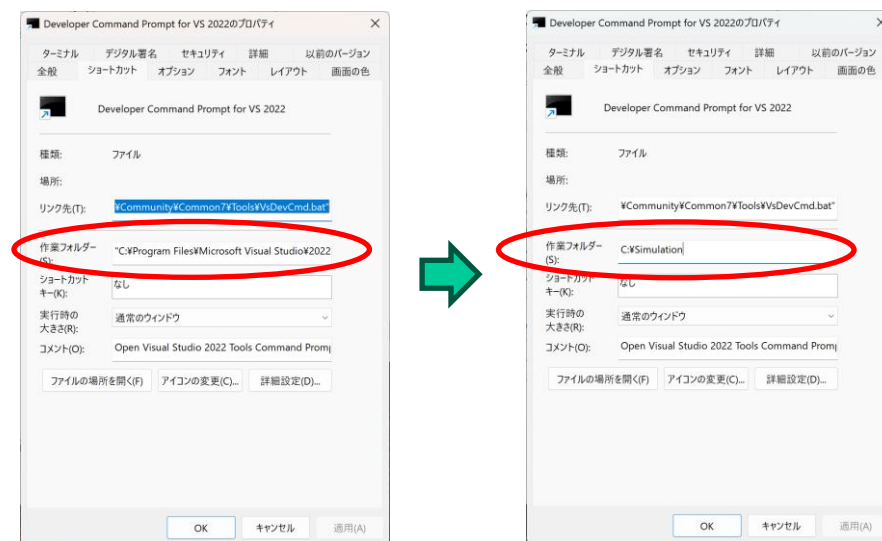
3. プログラムのソースコードを置く場所を決めます。  
必要なら新しいフォルダを作成します。  
以下の例では C:\Simulation とします。



- 2で作成したデスクトップのショートカット、あるいはタスクバーにピン留めされた「Developer Command Prompt for VS 2022」を右クリックしてプロパティを開きます。



4. プロパティの中の「全般」 「作業フォルダ」を3. で決めた作業用フォルダに修正します。



## 「Visual Studio 2022」のコンパイラの使い方（① 直接使う場合）

---

5. 前のページで作成したショートカットから「Developer Command Prompt for VS 2022」を起動させます。  
ショートカットを作成していない場合は起動した後、コマンドプロンプトの中で “cd” コマンドを使ってソースファイルの置いてあるフォルダへ移動します。

例えば、ディスクCのDATAというフォルダに移動したいのであれば、

```
> cd c:\DATA
```

（¥記号はバックスラッシュ記号で表示される場合もあります）

6. コンパイルするためのコマンドは”cl.exe” です。単に、”cl”だけでも同じです。

例えば “sample.c”というソースファイルをコンパイルしたいときは

```
> cl sample.c
```

とすればコンパイルできます。sample.exeと言う実行ファイルが作成されますので、  
ここで

```
> sample.exe
```

と打ち込めば実行できます。

詳細はマイクロソフトのサイト:

<https://docs.microsoft.com/ja-jp/cpp/build/walkthrough-compile-a-c-program-on-the-command-line?view=vs-2019>

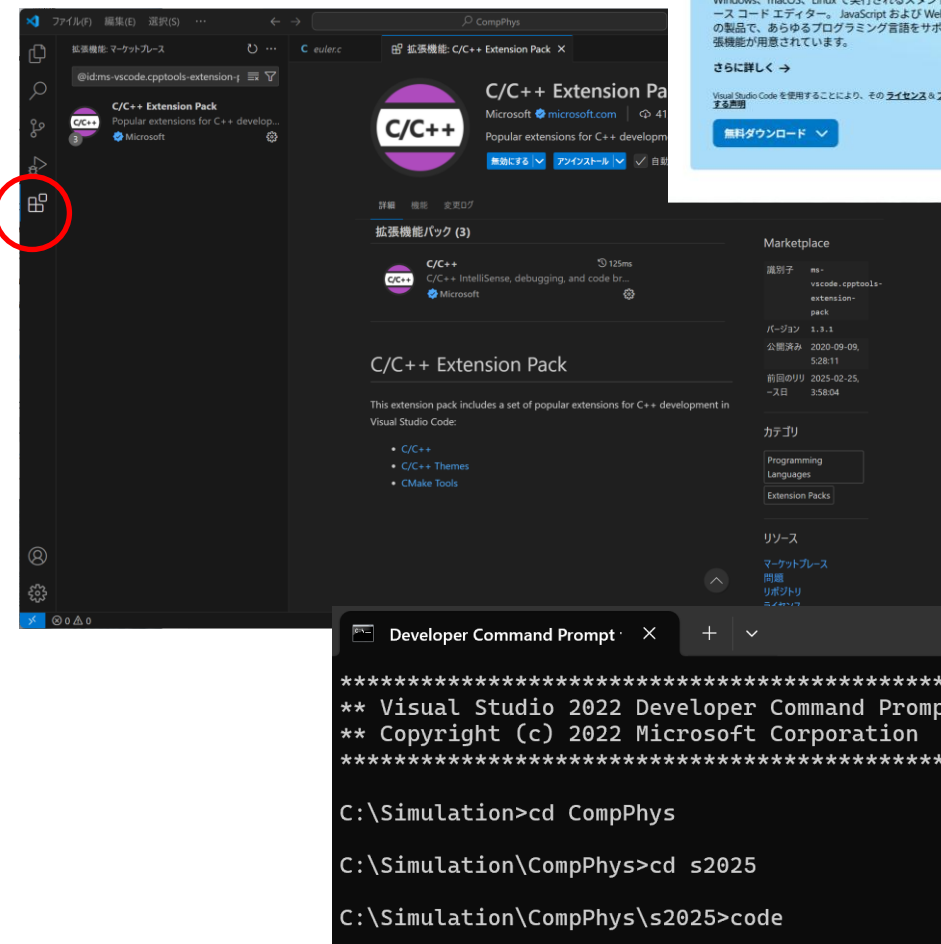
あるいは初心者向けの説明が書いてあるWebサイトなど;

<https://www.javadrive.jp/cstart/install/index5.html>

# 「Visual Studio 2022」のコンパイラの使い方（② Visual Studio Codeから使う）

Visual Studioのダウンロードサイトから Visual Studio Codeをダウンロード、インストールする。

Visual Studio Codeを起動  
拡張機能から「C/C++」、  
「Japanese Language Pack for VS Code」  
をインストール



『「Visual Studio 2022」のコンパイラの使い方（① 直接使う場合）』で作成したコマンドプロンプトから “code”と打ち込むとエディタと実行用コンソールなどがまとめて表示される。

## 方法2 ①WindowsにLinuxの仮想環境を構築する(WSL2: Windows Subsystem for Linux)

---

- Linuxについて

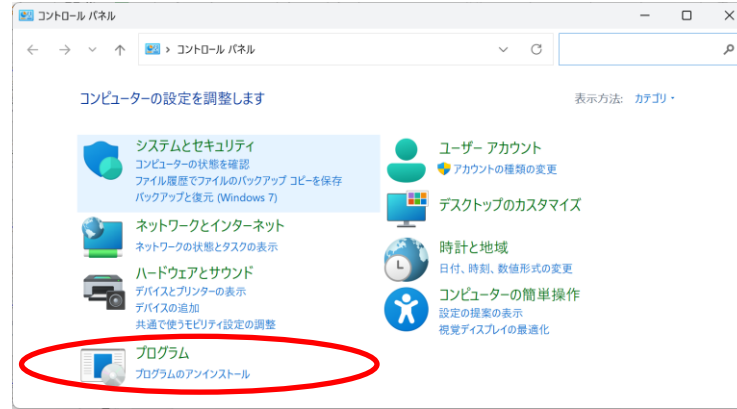
Linux専用マシンではなくても、Windowsの仮想環境にLinuxを動かせる環境を準備すると、高度な利用もできるようになります。

- 最新バージョンのWindows11で、メモリー が8GB以上、ディスクはできたら256GB以上有るとそれなりに使えると思います。
- LinuxにC, Fortran, gnuplotなど必要なソフトをすべてインストールできます。  
例えば、管理者権限(root)で、”apt install gnuplot”でgnuplotがインストールできます。
- インストール方法は、色々なWebサイトに紹介されています。WSL2, ubuntu, xwindowなどをキーワードとして検索してみてください。例えば;  
<https://note.com/hiro20180901/n/nc798a07485e2>  
<https://www.kkaneko.jp/tools/wsl/wsl2.html>

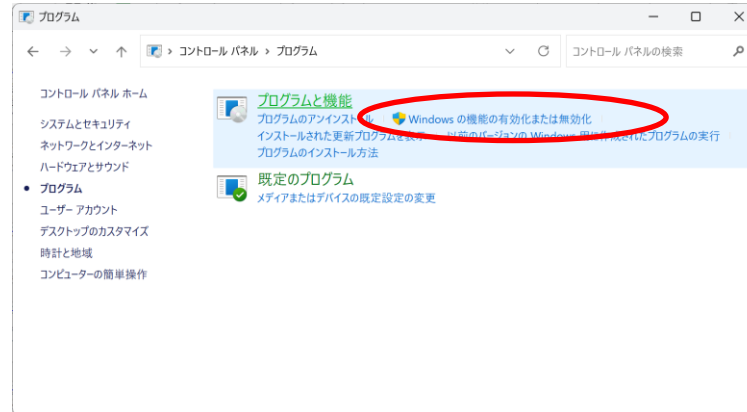
- グラフ作成  
Linux環境にgnuplotをインストールすることもできます。

## 方法2 ②WindowsにLinuxの仮想環境を構築する(WSL2: Windows Subsystem for Linux)

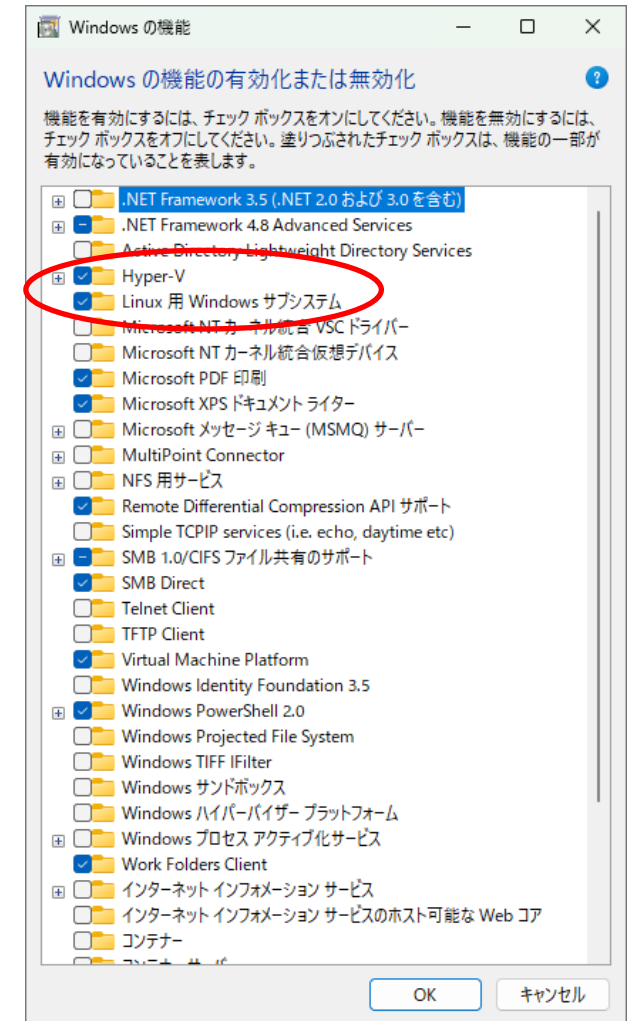
- メニューから「コントロールパネル」を開く



- 「プログラムと機能」の中の「Windowsの機能の有効化または無効化」を開く

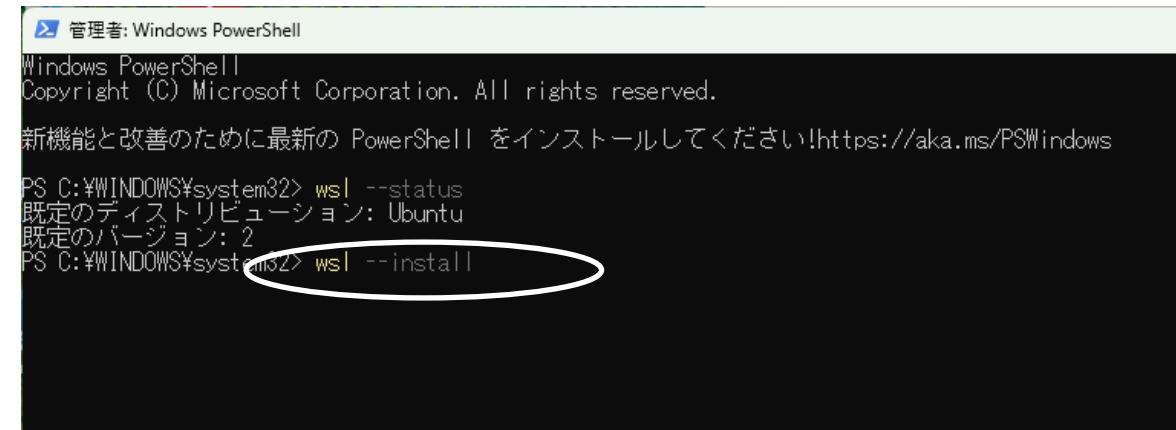


- 「Windowsの機能」の中の「Hyper-V」と「Linux用Windowsサブシステム」にチェックを入れる。



## 方法2 ③WindowsにLinuxの仮想環境を構築する(WSL2: Windows Subsystem for Linux)

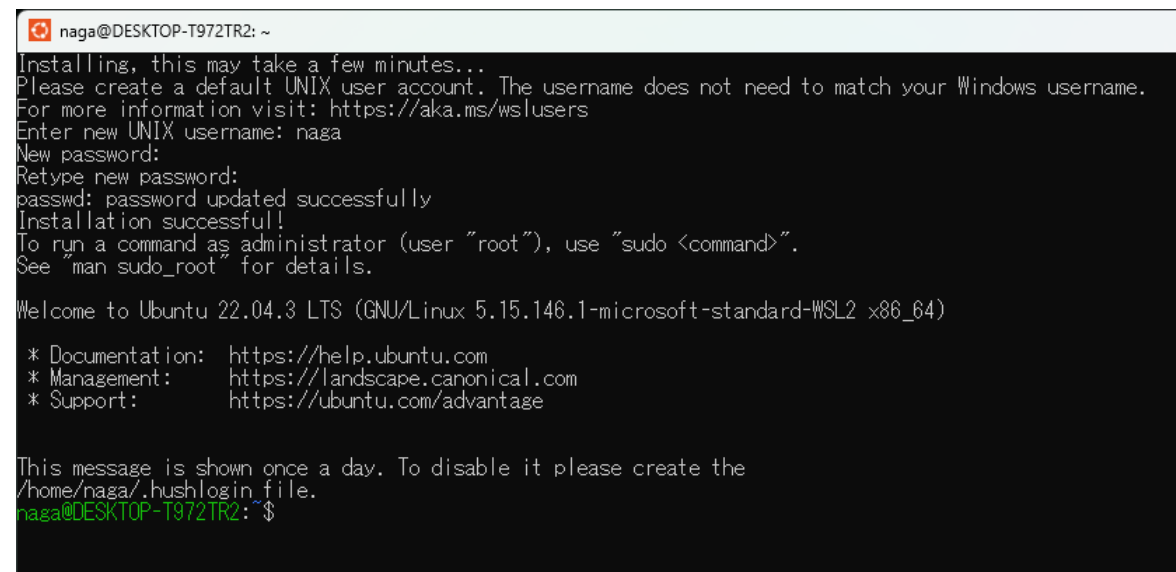
- Windowsメニューから「ターミナル」を起動  
(最新のPowerShellをインストールしてくださいは無視)
- “wsl –status”を入力する。右のような表示になればLinuxをインストールする準備ができている。
- “wsl –install”を入力すると、Ubuntuがインストールされる。  
途中、Linuxユーザー名やパスワードを入力するように言われるので従う。



```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新機能と改善のために最新の PowerShell をインストールしてください!https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --status
既定のディストリビューション: Ubuntu
既定のバージョン: 2
PS C:\WINDOWS\system32> wsl --install
```



```
naga@DESKTOP-T972TR2: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: naga
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/naga/.hushlogin file.
naga@DESKTOP-T972TR2:~$
```

## 方法2 ④WindowsにLinuxの仮想環境を構築する(WSL2: Windows Subsystem for Linux)

- Windowsメニューから「ubuntu」を起動  
タスクバーにピン留めしておくと簡単に起動できます。
- Linuxが起動できるようになったらCコンパイラ (gcc)をインストールします。

“sudo apt install build-essential”

- また、ときどきLinuxのアップデートをしてください。

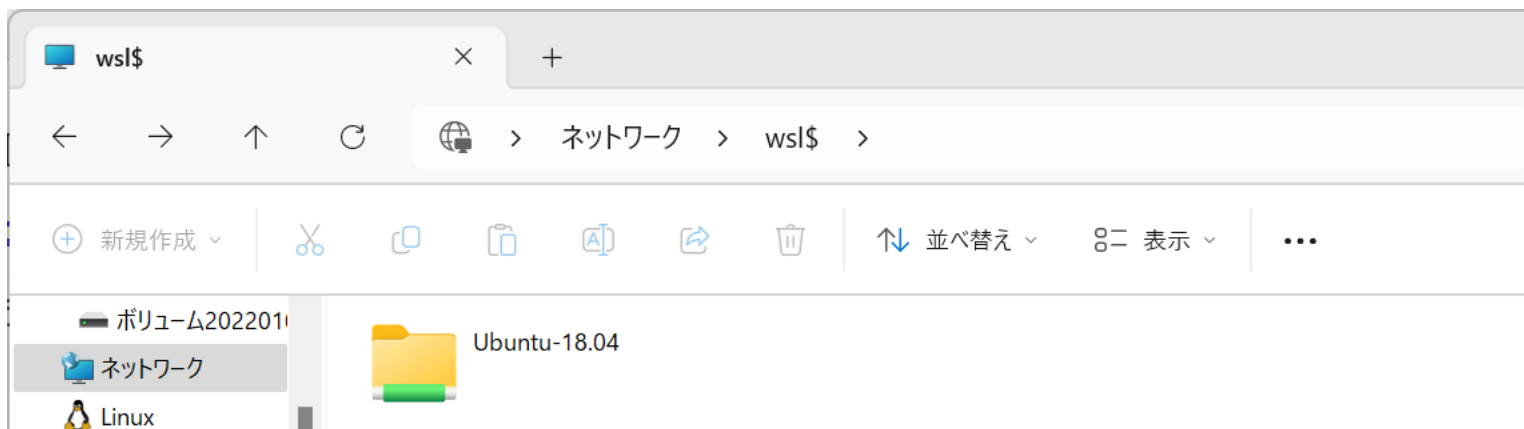
“sudo apt update”

“sudo apt upgrade”

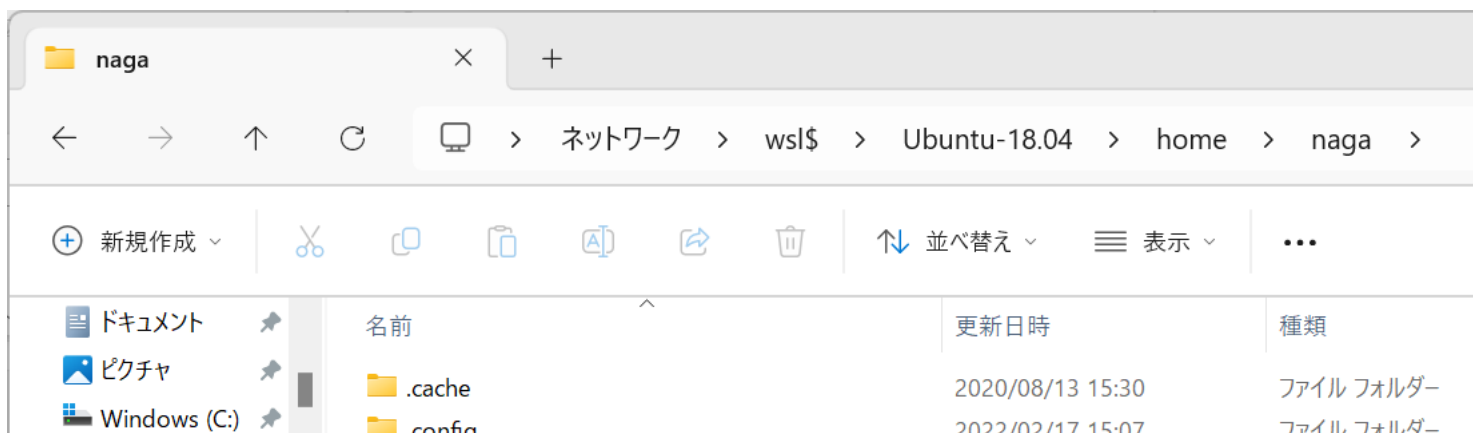


## 方法2 ⑤ WSLでLinux (ubuntu) をインストールした場合 WindowsからLinuxのファイルを見る方法

エクスプローラのアドレスバーに”\\wsl\$”と打ち込むとLinuxの一番上のディレクトリが見えます。



Linuxの一番上のディレクトリから home -> ユーザー名で自分のホームのファイルが見えます。



ただし、この場合はLinuxの管理者権限でファイル进行操作できるので、誤って大切なファイルを消去する可能性がありますので慎重に操作して下さい。



## 方法2 ⑥ WSLでLinux (ubuntu) をインストールした場合 LinuxからWindowsのファイルを見る方法

---

Linuxのコマンド `cd` を使います。“`cd /mnt/c`” で WindowsのCドライブが見えます。

シンボリックリンクさせておくと、Windowsのショートカットのようなリンクを作成できます。

例えば、Cドライブの直ぐしたにDATAというフォルダを作成したとします。

LinuxからDATAのディレクトリへ移動したいとき、通常であれば、Linuxにログインした後

```
> cd /mnt/c/DATA
```

で移動しますが、ログインした場所(home)にリンクを作成しておくと

```
> ln -s /mnt/c/DATA DATA
```

とするとリンクが作成されます。確認のため、

```
>ls -l
```

で見ると、

```
xxxxxx user user xxxxx xxx 2024 DATA -> /mnt/c/DATA
```

となり、リンク(ショートカット)として利用できます。

# インストールしたプログラミング環境の確認方法

CLEにサンプルプログラム hello.c がおいてあります。  
あるいは、テキストエディターで右のプログラムを入力して下さい。

1. コンパイルできるか確認する
2. 実行ファイルができているか確認する
3. 実行する

## Visual Studioの場合

```
c:\¥Simulation¥CompPhys¥s2024>cl hello.c  
Microsoft(R) C/C++ Optimizing Compiler Version 19.35.32216.1 for x86  
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
hello.c  
Microsoft (R) Incremental Linker Version 14.35.32216.1  
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
/out:hello.exe  
hello.obj
```

```
c:\¥Simulation¥CompPhys¥s2024>dir  
....(省略:hello.exeが作成されていることを確認)  
c:\¥Simulation¥CompPhys¥s2024>hello  
HelloWorld
```

hello.c

```
#include <stdio.h>  
  
void main()  
{  
    printf("HelloWorld¥n");  
}
```

## Linux のgcc の場合

```
> gcc hello.c  
gcc hello.c  
  
> ls -l  
-rwxrwxrwx 1 naga naga 8304 Apr 16 11:06 hello.c  
-rwxrwxrwx 1 naga naga 8304 Apr 16 11:08 a.out  
  
> ./a.out  
HelloWorld  
>
```