

計算物理 (Computational Physics, PHY.L210)

4Q 火・金 7,8限@南4号館3階 情報ネットワーク演習室 第2演習室

#2: 2023年12月8日(金)

計算物理

関澤 一之

東京工業大学 理学院 物理学系





あれ、ログインできない...??



大丈夫！

※共用の計算機を利用するためには、パスワードを設定する必要があります。
(パスワードを忘れた場合も同様の手続きが必要です。)

以前に利用したことのある人は、自分のログインID (学籍番号) とパスワードで
ログインしてもらえれば大丈夫です。

パスワードの(再)設定方法

1. [東工大ポータル](#) → 「[教育用電子計算機システム\(学内限定\)](#)」
→ 「[教育システム用アカウント パスワード設定画面](#)」へ
2. ログインID (学籍番号) を確認し、設定するパスワードを2回入力
3. [設定/Save password] ボタンをクリックして少し待てば設定完了

※東工大ポータルにアクセスする手段 (携帯等) がない場合：

- 「ログインID: portal パスワード: portal」で仮アカウントにログイン
- 仮アカウントを用いて東工大ポータルにアクセスし、パスワードを設定
- 仮アカウントからログアウトし、自身のアカウントにログインする

後で、時間が許せば東工大のスパコン「TSUBAME3.0」にアクセスしてみます。

まだアカウントの申請をしていない人は、ここで申請しておきましょう。

➤ 東工大ポータル → TSUBAMEポータル → 必要事項を記入して申請するだけ

新規利用申請

利用者区分	学内利用者	
HPCI-ID	<input type="text"/>	← HPCI-IDは関係ないので無視 (お持ちの方は入力してください)
*姓名 (ローマ字)	姓 <input type="text" value="toukou"/>	名 <input type="text" value="hiroshi"/>
姓名 (漢字)	姓 <input type="text" value="東工"/>	名 <input type="text" value="大"/>
姓名 (カナ)	姓 <input type="text" value="トウコウ"/>	名 <input type="text" value="ヒロシ"/>
日本国籍であるか特別永住権を持っている	<input checked="" type="checkbox"/>	
カード種別	*職名	<input type="text" value="アクセスカード(研究員)"/>
	*所属	<input type="text" value="学外"/>
所属詳細		<input type="text"/>
内線番号		<input type="text"/>
内線番号 (その他)		<input type="text"/>
FAX		<input type="text"/>
*連絡用E-Mailアドレス1		<input type="text" value="toukoudai@hogehoge"/>
連絡用E-Mailアドレス2		<input type="text"/>
連絡用E-Mailアドレス3		<input type="text"/>
メールボックス番号		<input type="text"/>
メールボックス番号 (その他)		<input type="text"/>
主な利用目的	<input type="text" value="その他"/>	← たぶん“講義”みたいなのがある？
研究テーマ		<input type="text"/>

授業計画（シラバスより）

授業計画・課題		
	授業計画	課題
第1回	計算環境の準備	共用のPCあるいは自身のPCに計算環境を整える。
第2回	Unix/Fortran入門	Unixの基礎知識を習得する。簡単なFortranプログラムを作成し、基本的な動作を確認する。
第3回	差分法の導入：1次元拡散方程式	差分法によってどのように微分方程式を解くのか、その原理を理解する。
第4回	差分法と数値精度：1次元移流方程式	1次元移流方程式を数値的に解き、様々な差分スキームの精度と計算の安定性を分析する。
第5回	数値流体力学への招待 I：渦度方程式とカルマン渦列	数値流体力学（CFD）の基本知識を習得する。渦度方程式を理解する。
第6回	数値流体力学への招待 II：渦度方程式とカルマン渦列	2次元の渦度方程式を数値的に解き、カルマン渦列を生成する。
第7回	時間に依存しないシュレーディンガー方程式の数値解法 I：ヌメロフ法	水素原子の電子波動関数を記述するシュレーディンガー方程式とヌメロフ法を理解する。
第8回	時間に依存しないシュレーディンガー方程式の数値解法 II：ヌメロフ法	ヌメロフ法を用いて動径方向のシュレーディンガー方程式を数値的に解き、水素原子の電子波動関数を求める。
第9回	時間に依存しないシュレーディンガー方程式の数値解法 III：行列対角化	シュレーディンガー方程式の行列表示を理解する。
第10回	時間に依存しないシュレーディンガー方程式の数値解法 IV：行列対角化	LAPACKを用いてハミルトニアン行列を数値的に対角化し、1次元のシュレーディンガー方程式に対する固有値と固有ベクトルを求める。
第11回	時間依存シュレーディンガー方程式の数値解法：テイラー展開法	時間依存シュレーディンガー方程式に対するテイラー展開法を理解する。
第12回	時間依存シュレーディンガー方程式の数値解法：テイラー展開法	テイラー展開法を用いて1次元時間依存シュレーディンガー方程式を数値的に解き、ポテンシャルに散乱される波束の時間発展を求める。
第13回	量子流体力学への招待 I：時間依存グロス・ピタエフスキー方程式	超流動体を記述するグロス・ピタエフスキー方程式（TDGPE）を理解する。また、1次元TDGPEを解き、ダークソリトンを生成する。
第14回	量子流体力学への招待 II：トポロジカルな励起と関連する物理現象	超流動体に発現するトポロジカルな励起モードの性質を理解する。また、ソリトン励起や量子渦など、超流動体のトポロジカルな励起が関係する様々な物理現象を知る。

初回で得た教訓

➤ “Hello World” も、つまづくとは意外と大変。

目立った問題点：

□ ターミナルの操作に慣れていないこと

- ファイルの場所がわからなくなる
- コマンドがわからない
- 挙動・ルールがわからない

ディレクトリって？

フォルダと一緒にです！



拡張子って？

ファイル名の末尾についているやつです！

file.txt
↑ テキストファイル

本日の目標

➤ 数値計算に必要な環境を整える！（続き）

➤ やることリスト

+ターミナルの使い方

- Slackのインストール・設定
- コンパイラ（**GFortran**）のインストール
- エディタ（**VSCode**）のインストール
- 可視化ソフト（**gnuplot**）のインストール
- SSH接続ソフト（**putty**・**WinSCP**）のインストール
- Tsubame3.0にアクセスしてみる



Windows?

Mac?



0. はじめに：利用環境について

利用環境について

共用PCの仕様について

大岡山キャンパス

南4号館 第二演習室



端末 iMac(21.5 inch, 4K, 2017)
× 93台

✓ 1ユーザあたり10GB利用可

iMac (Retina 4K, 21.5インチ, 2017)	
CPU	3.0GHzクアッドコアIntel Core i5(Turbo Boost使用時最大3.5GHz)
メモリ	8GB 2,400MHz DDR4
ディスク	256GBのフラッシュストレージ
グラフィックス	Radeon Pro 555(2GBビデオメモリ搭載)
ディスプレイ	Retina 4Kディスプレイ
マウス	ロジクール M110sBK
キーボード	BUFFALO BSKBM01 USB接続 Macモデル テンキー付き

➤ 参考：<https://www.edu.gsic.titech.ac.jp/index.php?システム構成>

共用PCのOSについて

➤ 参考：<https://www.edu.gsic.titech.ac.jp/index.php?ソフトウェア>

➤ 2023年12月現在：**macOS・Windowsを選択可能**

※2024年度以降はWindowsのみになる予定
(こちらにいろいろインストールされる)

□ 備考：共用PCのmacOSだと、既にいろいろ入っている：

- ・ コミュニケーション：**Slack**, Zoom, Skype
- ・ コンパイラ：Fortran77, **Fortran90**, C, C++, Python, ...
- ・ 可視化関係：**gnuplot**, ImageMagick, Paraview
- ・ エディタ：**VSCode**, Emacs, Vi/Vim
- ・ その他：**homebrew**

**両方の場合について説明するので
好きな方を選んで構いません
自分のノートPC等を持参してもOK!**

1. Slackのインストール・設定



slack

Slackのインストール

▶ Slack（スラック）とは

- ✓ 2013年8月にアメリカでリリースされた，Salesforce社のコミュニケーションツール。
- ✓ 東工大では2023年1月からすべての学生・教職員にアカウントを提供。
- ✓ チャット感覚で（メールより**気軽に**）グループ全体に見える形で情報をやり取りできる。
- ✓ 履歴が残るので，**誰かの疑問・トラブル・解決策が，他の誰かの助けになる！**

ToDo!

※共用Macの場合Slackは既に入っている

例：Windows 版 Slack

Slack アプリを使えば、ワンクリックでチームと連携できます。

ダウンロード (64 ビット)

1. 公式サイトからダウンロード：<https://slack.com/intl/ja-jp/downloads/windows>

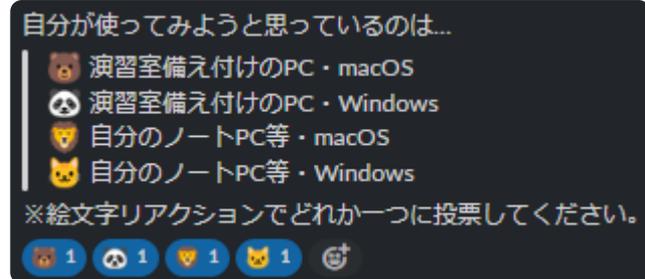
※Windowsの場合，Microsoft StoreアプリでSlackと検索してもOK。

2. ワークスペース「**Team東工大**」にサインイン → 「**tmp-計算物理2023**」というチャンネルを追加

※Webブラウザ版もあり．スマホにアプリを入れてもよい．

3. チャンネルに参加できた人は，これに回答してみてください：

※Pollyというアプリも使ってみたので，
そちらにも同様に回答してみてください。



✓細かいことは考えず，気軽に質問・疑問・トラブルなどを適当に書いてください！

教員とTAに“メンション”してもらえると，気付きやすいかも：
@関澤 @水上 凱斗 @松本侑真（アットマークの後に名前を書く）

もちろん，対面なので
直接聞いてもらってもOK！



Slackで聞いてもらえたら



TAや教員がすぐ対応します！
学生が答えてもOK！

2. VSCodeのインストール



Visual Studio Code

```
test.f90
1 PROGRAM myprog
2 USE test_free, ONLY: test_sig_Sub
3 INTEGER :: alpha
4 REAL(8) :: beta
5 COMPLEX(8) :: gamma
6 CALL test_
7 END PROGRAM myprog
8
```

➤ VSCodeとは（正式にはVisual Studio Code, 単に“ブイエスコード”と呼ぶ）

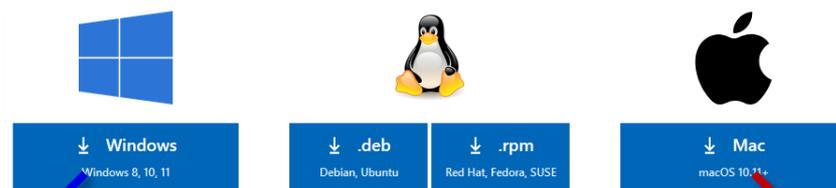
- ✓ 2015年にリリースされた, マイクロソフトが提供するコード開発用のエディタ（無料）
- ✓ 様々な拡張機能を利用できる：シンタックスハイライト, 文字補完, デバッグ機能等
- ✓ Visual Studioとは別物で, 動作が軽く, エディタ内にターミナルを開ける.

便利!



➤ VSCodeのインストール

- ✓ 公式サイト（<https://code.visualstudio.com/download>）にアクセス



Windowsの場合

インストーラ（例：**VSCodeUserSetup-x64-X.X.X.exe**）をダウンロードし, 実行する

※ **PATHへの追加**にチェックを入れておく（後でも設定可）
→ ターミナルで“code”と叩くとVSCodeを開けるようになる

Macの場合

zipファイル（例：**VSCode-darwin-universal.zip**）をダウンロードする（勝手に解凍される？）
→ Visual Studio Codeというアプリケーションが生成されるので, ドラッグ・アンド・ドロップで, サイドメニューの「アプリケーション」に移動する

※ 共用Macの場合既に入っている

➤ 自分好みにVSCodeをカスタマイズしよう！

➤ 拡張機能の追加（推奨）

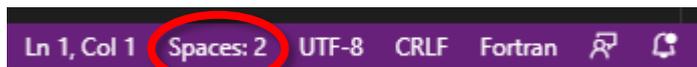
1. ウィンドウ側のサイドバーの**拡張機能（Extensions）**をクリック
2. 表示された検索窓から検索して選択すれば、インストールできる：
 - ✓ [Japanese Language Pack for Visual Studio Code](#)：日本語化したい場合
 - ✓ [Modern Fortran](#)：シンタックスハイライト，文字補完等（**効率化，バグ回避**）
 - ✓ [zenkaku](#)：全角スペースをハイライトしてくれる（**バグ回避**）



他にもいいのがあったら
教えてください！

➤ インデント幅の設定（任意）

→ ウィンドウ右下のSpaces（タブのサイズ）をクリックして変更できる



デフォルトは4（好みの問題）

➤ カラーテーマの変更（任意）

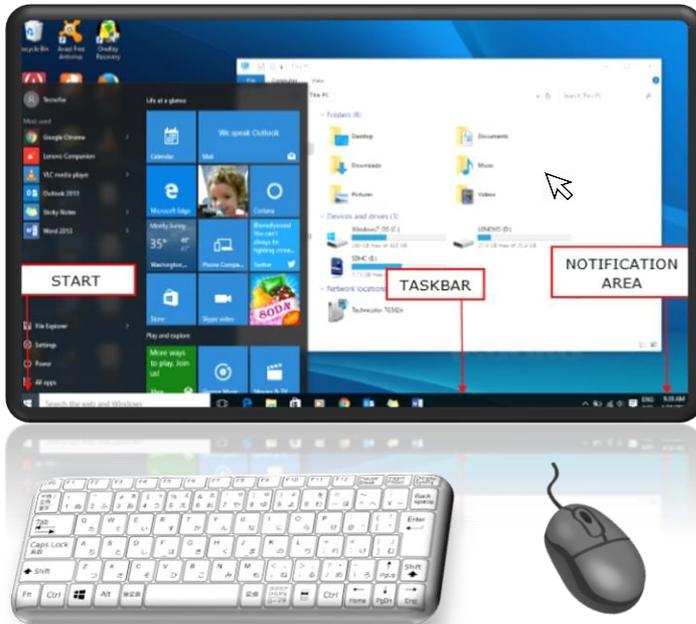
→ **Ctrl+K Ctrl+T**（Controlを押したままKT）→ 様々なテーマを選択・追加できる

2. ターミナルの基本的な操作方法

▶ コマンド (文字) でPCを操作するアプリケーション

(グラフィカル・ユーザー・インターフェース)

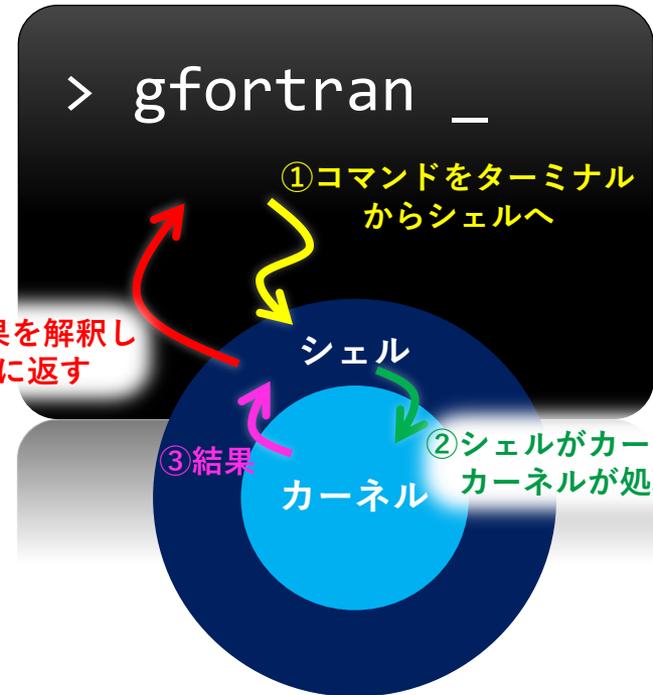
GUI



マウスを使って視覚的に操作

(キャラクター・ユーザー・インターフェース)

CUI※



コマンドのみで操作

- ✓ シェル：人間からの入力をカーネルに伝えるプログラム。
- ✓ カーネル：OSの中核。CPUやメモリとのやり取りをしている。

GUI: https://www.tutorialspoint.com/windows10/windows10_gui_basics.htm

ターミナルの起動方法

➤ 今後よく使うので、覚えておきましょう！

VSCode上では

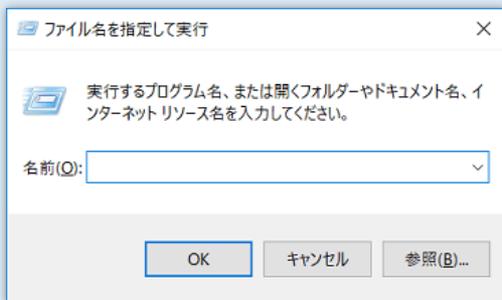
方法1. ファイルを開いた状態で, **Control+@**

(慣れてない人は)
VSCode上で開きましょう

方法2. 上部のメニュー → ターミナル → 新しいターミナル

Windowsの場合

方法1. **Win+r** (Windowsキーを押しながらr)



左下に表示された入力欄に

「**powershell**」もしくは「**cmd**」と入力し, Enter

方法2. **Windowsキー**を押し (検索窓が表示される)

「**powershell**」もしくは「**cmd**」と入力し, Enter

Macの場合

方法1. **Spotlight検索** (**Command+Space**)

→ 「ter」くらい打てば出てくる

方法2. **Dock**にあるアイコンをクリック



➤ 開いているコードとは独立に
開きたいときに便利

もちろん, スタートメニューやDockに登録してもいい

ターミナルで用いる最低限のコマンド

- ✓ 現在位置（カレントディレクトリの絶対パス）の確認

```
> pwd
```

※present working directoryの意味
※Windows/Mac共通

- ✓ カレントディレクトリにあるファイルの確認

```
> ls
```

※listの意味
※Windows (PowerShell) /Mac共通
※Command Promptの場合は「dir」

➤ どこにいるかわからなくなったらとりあえず「pwd」と「ls」を叩く → ディレクトリの階層を表示する「tree」も便利

- ✓ 新しいディレクトリの作成

```
> mkdir ディレクトリ名
```

※make directoryの意味
※Windows/Mac共通

- ✓ カレントディレクトリの移動

```
> cd 行先ディレクトリのパス
```

※change directoryの意味
※Windows/Mac共通

- ✓ ファイルの移動

```
> mv 移動するファイル名 移動先
```

※moveの意味
※Windows (PowerShell) /Mac共通
※Command Promptの場合は「move」

- ✓ ファイルのコピー

```
> cp コピーするファイル名 コピー先
```

※copyの意味
※Windows (PowerShell) /Mac共通
※Command Promptの場合は「copy」

- ✓ ファイルの削除

```
> rm ファイル名
```

※removeの意味
※Windows (PowerShell) /Mac共通
※Command Promptの場合は「del」

ターミナルで用いるその他のコマンド

- ✓ ターミナルの表示をリセットする

```
> clear
```

※Windows (PowerShell) /Mac共通

※Command Promptの場合は「cls」

- ✓ ターミナルを閉じる

```
> exit
```

※Windows/Mac共通

GFortranのエラーメッセージ等が文字化けしてしまうことがある



文字コードの不一致のため

Windowsの場合

- ✓ 文字コードの確認

```
> chcp
> 現在のコード ページ : 932
```

この場合、文字コードはShift_JIS

- ✓ 文字コードをUTF-8に変更

```
> chcp 65001
```

毎回やるのは面倒!



※WindowsPowerShellの場合:

```
> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser -Force
> mkdir C:¥Users¥Username¥Documents¥WindowsPowerShell
> notepad C:¥Users¥Username¥Documents¥WindowsPowerShell¥Microsoft.PowerShell_profile.ps1
```

※起動時に読み込まれるので
設定したいコマンドがあればこれに書く

このファイルに
chcp 65001
と書いて保存する

Macの場合

※Macの場合デフォルトでUTF-8かも

- ✓ 文字コードの確認

```
> locale
```

- ✓ 文字コードの変更: 上部メニューの Terminal → Preferences → Profiles → Advanced → Text Encoding

コード番号	文字コード
932	Japanese (Shift_JIS)
51932	EUC Japanese (euc-jp)
65001	Unicode (utf-8)

ディレクトリのパスについて

- ✓ 絶対パス：PCの最下層から当該ファイルまでのディレクトリ階層
- ✓ 相対パス：カレントディレクトリから当該ファイルまでのディレクトリ階層

例) カレントディレクトリが「C:¥Users¥Username¥」の場合：

先ほどの例でいうと

「C:¥Users¥Username¥Documents¥WindowsPowerShell¥」
が絶対パス。

↑
今はこの階層にいる



Windowsは「¥」（バックスラッシュ; ¥）
Macの場合は「/」（スラッシュ）
がディレクトリの区切り！

カレントディレクトリから見た

「.¥Documents¥WindowsPowerShell¥」
が相対パス。

※ 「.¥」はカレントディレクトリを表す

カレントディレクトリが「C:¥Users¥Username¥Desktop¥」の場合

「..¥Documents¥WindowsPowerShell¥」
が相対パス。

※ 「..¥」は一つ上の階層のディレクトリを表す

→ 2つ上だったら 「..¥..¥」

ためしてみよう！

1. ターミナルを立ち上げてみましょう。
2. カレントディレクトリの絶対パスとそこにあるファイルを確認してみましょう。
3. 「temp」という名前のディレクトリを作成し、その中に移動しましょう。
4. 一つ上の階層のディレクトリに戻りましょう。
5. ディレクトリ「temp」の名前を「test」に変更しましょう。
6. ディレクトリ「test」を削除しましょう。 ※オプションが必要かも（Linuxだと-rf）
7. ターミナルを閉じましょう。

全てターミナル上で（マウス・GUIに頼らず）やってください。

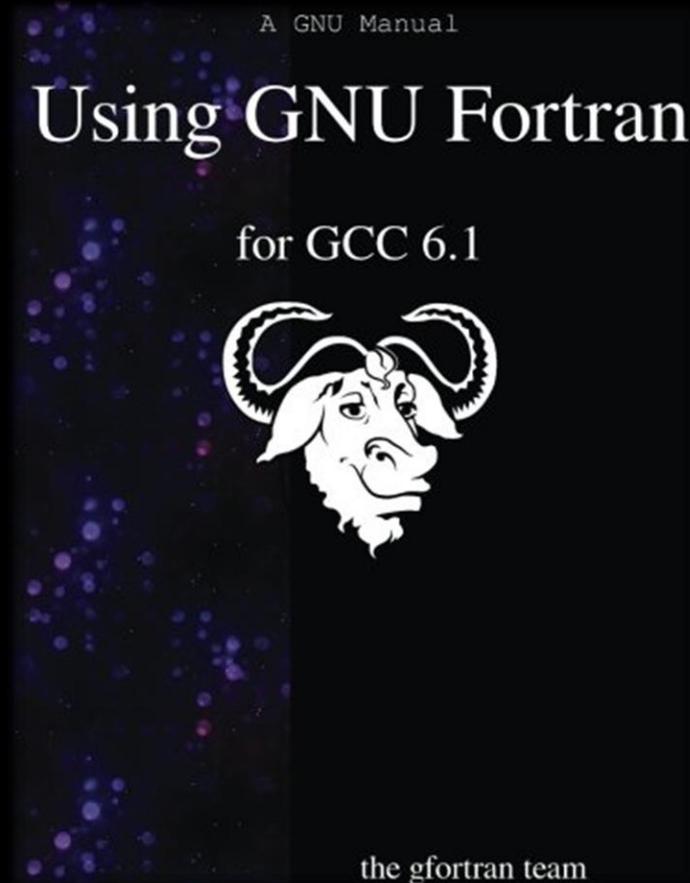
2. GFortranのインストール

```
Terminal File Edit View Search Terminal Help
GFORTRAN(1)
GNU
NAME
  gfortran - GNU Fortran compiler
SYNOPSIS
  gfortran [-c|-S|-E]
           [-g] [-pg] [-Olevel]
           [-Wwarn...] [-pedantic]
           [-Iidir...] [-Ldir...]
           [-Dmacro[=defn]...] [-Umacro]
           [-foption...]
           [-mmachine-option...]
           [-o outfile] infile...

  Only the most useful options are listed here; see below for the
  remainder.

DESCRIPTION
  The gfortran command supports all the options supported by the gcc
  command. Only options specific to GNU Fortran are documented here.

  All GCC and GNU Fortran options are accepted both by gfortran and by
  Manual page gfortran(1) line 1 (press h for help or q to quit)
```



➤ GFortranとは（またはGNU Fortran, 単に“ジーフォートラン”と呼ぶ）

- ✓ GNUコンパイラコレクション（GNU Compiler Collection: GCC）に含まれる, Fortranのコンパイラ（無料）.

=ソースコードをPCが実行可能な形式に変換する（コンパイルする）プログラム

- ✓ GNU（=“GNU is Not Unix!”の再帰的頭字語, 発音は「グヌー」）とは, フリーソフトウェアから構成される, 総合ソフトウェアシステムの名称.



<https://ja.wikipedia.org/wiki/GNU>

➤ なんで今さらFortran?（=“FORmula TRANslation”の略）

- ✓ 1950年代に誕生した, 世界初の高級プログラミング言語.
→ 歴史があるのでマニュアルやライブラリ等が充実（Cでも言えることだけど）
- ✓ コンパイル言語なので実行速度が速く, 並列計算用のプログラミング（OpenMP, MPI, CUDA Fortran）が発達しており, スパコンを用いた最先端の研究でも用いられている（Cでも言えることだけど）
- ❖ C言語と比べて, ポインター等を知らなくても使えるので, 初学者でも簡単に書ける.
- ❖ 教員が熟達している.
- ❖ この講義は数値計算の基礎を学ぶことを目的とし, 次のステップとして, その技術を各自の好きな言語に応用してください ※Fortranで学んだことは無駄にはならない!

※共用Macの場合Homebrewとgfortranは既に入っている

Windowsの場合

1. 「TDM-GCC」の公式ページ (<https://jmeubank.github.io/tdm-gcc/>) にアクセス
2. 実行ファイルをダウンロードする (64bit OSなら「[tdm64-gcc-X.X.X-X.exe](#)」)
3. 実行ファイルを走らせ、指示に従ってインストールする：

↑バージョン

※バックスラッシュ¥=¥
フォントによる

ダウンロードしたexeファイルを実行 → [Create] → 64bitか32bitを選択
→ インストールするフォルダを設定 (例: 「C:¥TDM-GCC-64」)
→ 「type of install」はそのまま「TDM-GCC Recommended, C/C++」でOK※
※ただし、「Components-gcc」のタブを開き、
「fortran」・「openmp」にもチェックを入れること → [Install]

Macの場合

※アンインストール方法：exeファイルを実行 → [Remove]を選択

1. 「Homebrew」をインストールする (もう入っていれば飛ばしてOK)
 - 1-1. 以下のコマンドをターミナルで実行する※Homebrewの公式ページ (<https://brew.sh/>) からコピー

```
> /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- 1-2. パスを通す (「`~/.bash_profile`」に以下の行を追記する)

```
export PATH=$PATH:/opt/homebrew/bin/
```

※ディレクトリの区切り
Windows → 「¥」
Mac → 「/」

2. 「gcc」をインストールする (gfortranも一緒にインストールされる)

```
> brew install gcc
```

※「>」はターミナルを表す
(コマンドには含まれない)

※共用Macの場合gfortranは既に入っている

➤ GFortranが正常にインストールできたことの確認

1. コマンドラインインターフェース (Command Line Interface: **CLI**) [あるいは
キャラクターユーザーインターフェース (Character User Interface: **CUI**)] を開く
(Linuxでは「端末」と呼ばれるので、しばしば「**端末**」とも呼びます)

Windowsの場合

- ✓ **Windows PowerShell**もしくは**コマンドプロンプト**を開く (例: Win+r→cmd)

Macの場合

- ✓ **ターミナル**を開く (例: Command+スペース→Spotlightでターミナルと検索)

2. ターミナルで「gfortran」と入力してエンターを叩く

```
> gfortran
gfortran: fatal error: no input files
compilation terminated.
```

上記のようなエラーメッセージが表示されれば、**インストール完了!**



※表示されたのは「コンパイルするファイルが指定されていない」というエラーメッセージ

➤ 実際にプログラムを書いて、実行してみよう！

1. テキストエディタ（メモ帳でもなんでもいい）を使って、以下の文字を書き、適当な名前（例：`Hello_World.f90`）、適当なディレクトリ（例：`C:¥ComptPhys¥Lecture1¥`）に保存する：

HelloWorld.f90

```
program Hello_World
print *, "Hello World!!"
end program Hello_World
```

最初と最後に書くおまじない
「Hello_World」がプログラム名

※全角文字は含めない方がいい
※記号はアンダーバー・ハイフンが無難
(_ - + . ()あたりはOK)

標準出力（端末の画面）に
「Hello World!!」と書き出して
という命令

2. ターミナルを開き、上記ファイルを保存したディレクトリまで移動する：

```
> cd c:¥ComptPhys¥Lecture1¥
```

※「cd」は「change directory」の略

3. 「gfortran」を使ってコンパイル → `Hello_World.exe` という名前の実行ファイルを生成

コンパイルするプログラム名

```
> gfortran -o Hello_World.exe Hello_World.f90
```

「-o 生成する実行ファイル名」 → 省略すると「a.exe」という名前になる

4. 生成された実行ファイルを走らせる

※ディレクトリの区切り
Windows → 「¥」
Mac → 「/」

Windows:

```
> .¥Hello_World.exe
Hello World!!
```

Mac:

```
> ./Hello_World.exe
Hello World!!
```

← プログラムの実行
← 結果



➤ **これで、プログラムを書いて、実行する準備ができた！**

✓ 作業ディレクトリをVSCodeで開く（ターミナルをここで開くため）

→ ウィンドウ上部のメニュー → File → Open Folder → さっきの例だと： `C:¥ComptPhys¥Lecture1¥`

✓ ソースコードをVSCodeで開く

→ Exploreに表示されたファイル（さっきの例だと： `Hello_World.f90`）をダブルクリック

```
≡ Hello_World.f90
1  program Hello_World
2
3     print *, "Hello World!!"
4
5  end program Hello_World
```

← こんな感じに
表示されればOK

✓ VSCode上でターミナルを開く

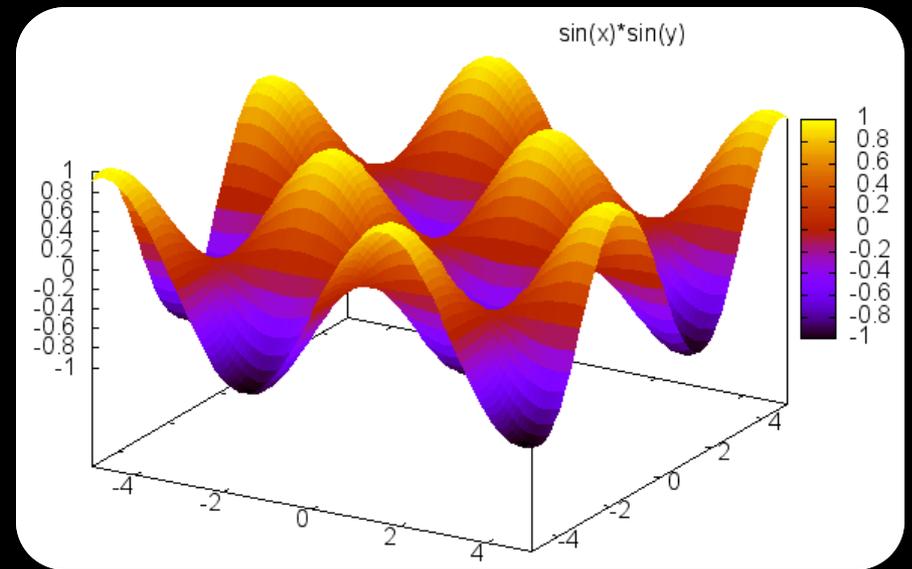
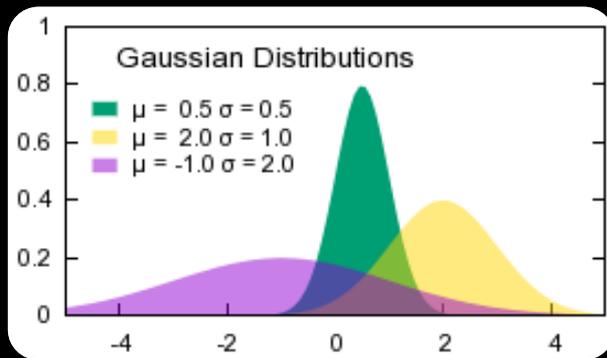
→ ウィンドウ上部のメニュー → Terminal → New Terminal（`Ctrl+Shift+@`）

✓ あとは先ほどと同じ（コンパイル → 実行）

```
> gfortran -o Hello_World.exe Hello_World.f90
> .¥Hello_World.exe ※Macの場合は「./」
Hello World!!
```

← コンパイル
← 実行
← 結果

4. gnuplotのインストール



➤ 計算結果を可視化したい！

➤ gnuplotとは（“ニュープロット”あるいは“ヌープロット”と呼ぶべき）

~~グニユプロット~~
~~ジーエヌユープロット~~



- ✓ 1986年に最初のバージョンがリリースされた、オープンソースの作画ソフト（無料）
- ✓ 「gnu」とあるが「GNU（グヌー, 1983~）」とは無関係
- ✓ 読み方について（[gnuplot Q/A | 1.2 How did it come about and why is it called gnuplot ?](#)より）

（前略）...I (Thomas Williams) wanted to call it “llamaplot” and Colin wanted to call it “nplot.”
We agreed that “newplot” was acceptable but, we then discovered that there was an absolutely ghastly pascal program of that name that the Computer Science Dept. occasionally used.
I decided that “gnuplot” would make a nice pun and after a fashion Colin agreed.

→ 本当は「newplot」としたかったが使われていたので、もじって「gnuplot」とした。



※pun: もじる, ダジャレ

ラマ (Llama)

ヌー (Gnu) ~ New

❖ もちろん, 他のソフトウェア (Pythonのmatplotlib等) でもなんでもいい

ヌー : <https://en.wikipedia.org/wiki/Wildebeest>

ラマ : <https://news.cgtn.com/news/2019-10-25/Spot-the-difference-alpaca-vs-llama-L4JUDBnWso/>

- TAの松本君による解説動画 (<https://youtu.be/YV2YtyhnlCk>)

TOGO ▶ TV

Gnuplotを使って データの可視化・グラフを作成する 基本編

20230520版



TOGO ▶ TV

➤ 計算結果を可視化したい！

➤ gnuplotをインストールする

Windowsの場合

- gnuplotの公式ページ (<http://gnuplot.info/>) にアクセス
→ Download → Primary download site on SourceForgeと進む
→ 一番新しいバージョン (現在: 5.4.8) のフォルダをクリック
→ インストーラ (現在: gp548-win64-mingw.exe) をダウンロード
- ダウンロードしたインストーラを実行する (インストール先の例: C:\Program Files\gnuplot)
- 実行ファイルのあるディレクトリ (例: C:\Program Files\gnuplot\bin) にパスを通す

※明らかにアクセス数が多い



設定を開く → 検索窓に「環境変数」(environment)と入力 → [環境変数の編集]
→ [環境変数...] ([Environment Variables...]) をクリック
→ User variablesの「Path」をクリック → 編集
→ 新規作成 → 「C:\Program Files\gnuplot\bin」と入力 → [OK]

Macの場合

- ✓ ターミナル上で, **Homebrew**を使ってインストールすればOK:

```
> brew install gnuplot
```

※共用Macの場合既に入っている

1. ターミナル上で, gnuplotと叩く :

```
> gnuplot

  G N U P L O T
  Version 5.4 patchlevel 3   last modified 2021-12-24

  Copyright (C) 1986-1993, 1998, 2004, 2007-2021
  Thomas Williams, Colin Kelley and many others

  gnuplot home:      http://www.gnuplot.info
  faq, bugs, etc:   type "help FAQ"
  immediate help:   type "help" (plot window: hit 'h')

  Terminal type is now 'windows'
  Options are '0 color solid butt enhanced standalone'
  Encoding set to 'cp1252'.
  gnuplot>
```

← このようにgnuplotが起動すればOK

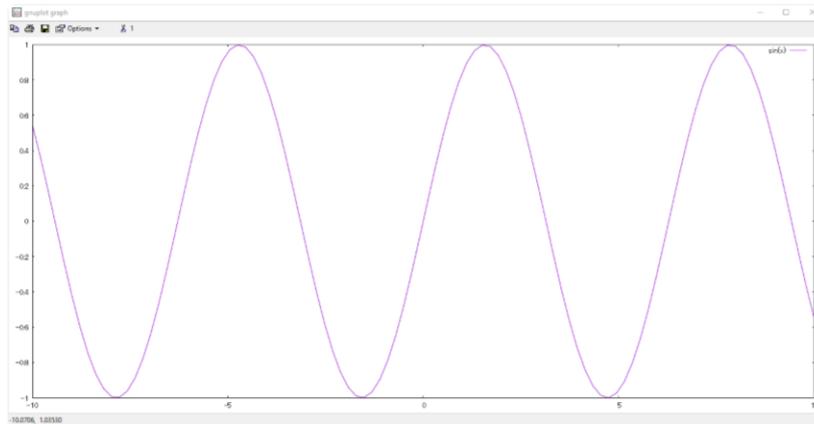
起動しない場合, パスが認識されていない

→ 環境変数を設定した後は,
ターミナルを再起動する

2. 試しに, $\sin(x)$ をプロットしてみる :

```
gnuplot> plot sin(x)
```

※ 「plot」は「p」と省略できる（「p sin(x)」と書いても同じ）



← $\sin(x)$ が描画された

5. TSUBAME3.0にアクセスしてみよう

- 総ノード数：540
- 総コア数：15,120
- メモリ：256 GB/node
- 総GPU数：2,160



➤ リモートでスパコンに接続したい！

➤ SSHとは (Secure SHellの略称)

- ✓ リモートコンピュータと通信するためのプロトコル
- ✓ 認証部分を含めネットワーク上の通信がすべて暗号化されるため、安全に通信できる

➤ SSHクライアントのインストール

Windowsの場合

- ✓ 例えば「PuTTY-ranvis」を用いる (ダウンロードサイト: <https://www.ranvis.com/putty>)

→ 64bit OSであれば, 0.79 (2023-08-27) 64bit .7z からダウンロード
→ 解凍 → ファイル一式が入っているフォルダ「PuTTY-ranvis」を
「C:¥Program Files¥」の中に移動する

- PuTTY-ranvisの中にある「putty.exe」と「puttygen.exe」を用いる

Macの場合

- ✓ 最初からターミナル上でSSHコマンドが使える (インストール不要)

SSH公開鍵認証をするために

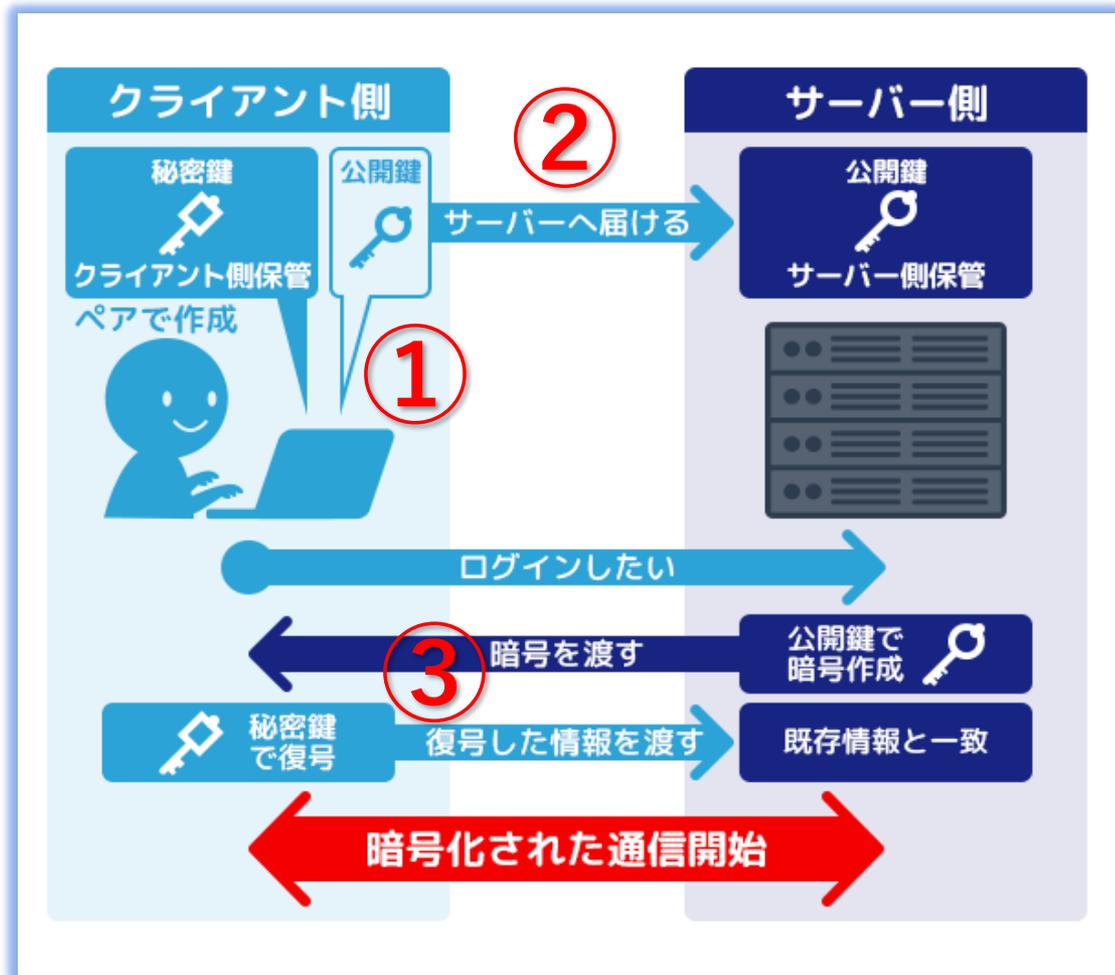
➤ より安全に接続するために、公開鍵認証方式が用いられる

① **秘密鍵**・**公開鍵**ペアの生成

※秘密鍵は自分だけが持つ
人に渡さない・サーバーに置かない

② **公開鍵**をサーバーに登録

③ **秘密鍵**を使ってログイン



概念図 : <https://www.kagoya.jp/howto/it-glossary/server/ssh/>

➤ Windowsの場合、puttygen.exeを用いて鍵のペアを生成する

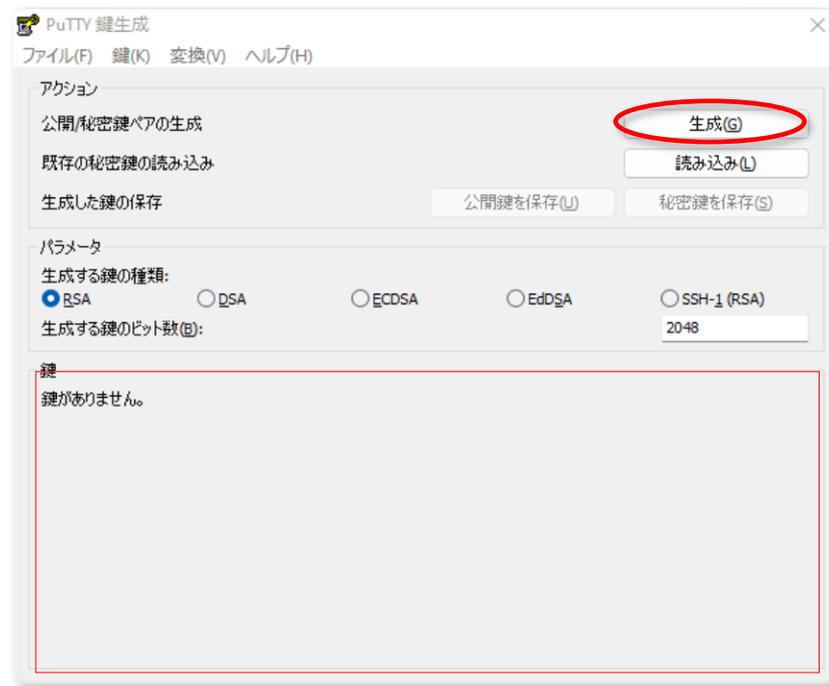
Windowsの場合

1. 「puttygen.exe」を実行する

(例：C:\Program Files\PuTTY-ranvis\puttygen.exe)

2. 「生成」をクリック

→ 下の空白エリアでカーソルを動かす



➤ Windowsの場合、puttygen.exeを用いて鍵のペアを生成する

Windowsの場合

1. 「puttygen.exe」を実行する

(例：C:\Program Files\PuTTY-ranvis\puttygen.exe)

2. 「生成」をクリック

→ 下の空白エリアでカーソルを動かす

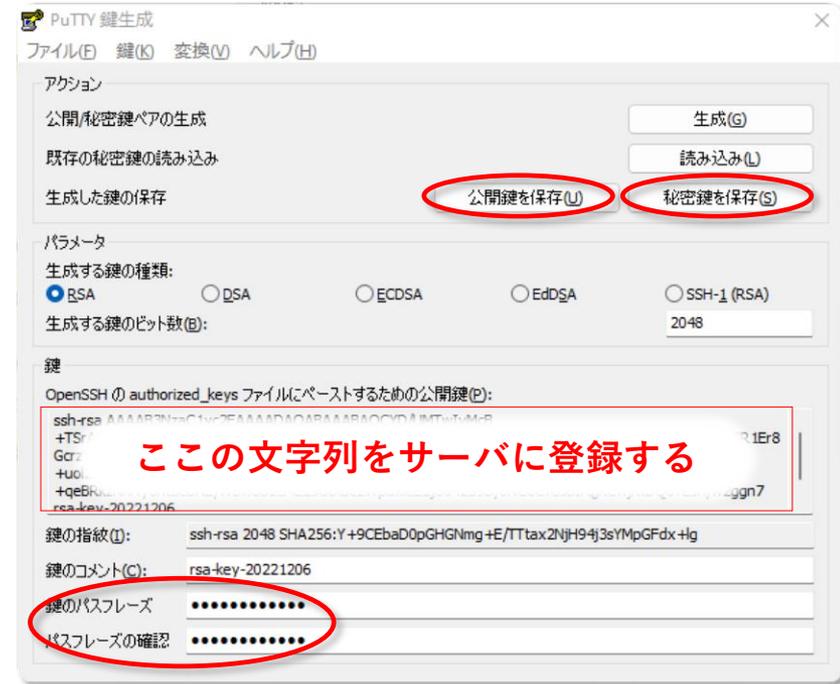
3. 「公開鍵を保存」 → 適当な場所に保存

(例：C:\Program Files\PuTTY-ranvis\keys\id_rsa.pub)

4. 「鍵のパスフレーズ」 (パスワード) を入力×2行

→ 「秘密鍵を保存」 → 適当な場所に保存

(例：C:\Program Files\PuTTY-ranvis\keys\id_rsa.ppk)



※PuttyのPPK形式で作成した公開鍵をアップロードしても接続できません！

※上に示したOpenSSHの公開鍵の部分をコピーしてください

※アップロードする場合、PuTTYgen → Conversions (変換) → Export OpenSSH key として OpenSSH形式の鍵を保存して、それをアップロードしてください。

➤ Macの場合, ssh-keygenを用いて鍵のペアを生成する

Macの場合

ターミナル上で, ssh-keygenを実行すればOK.

```
> ssh-keygen -t rsa
```

```
~/.ssh/id_rsa : 秘密鍵  
~/.ssh/id_rsa.pub : 公開鍵
```

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/Users/(ここはユーザーの名前が入ります)/.ssh/id_rsa): <-----  
Enter passphrase (empty for no passphrase): <----- パスフレーズ  
Enter same passphrase again: <----- パスフレーズ (確認用)  
Your identification has been saved in /Users/(ここはユーザーの名前が入ります)/.ssh/id_rsa.  
Your public key has been saved in /Users/(ここはユーザーの名前が入ります)/.ssh/id_rsa.pub.  
The key fingerprint is:  
99:24:u0:8i:2d:34:e6:82:r5:89:41:p8:a7:58:1p:n2 (ここにメールアドレスが入ります) .local  
The key's randomart image is:  
+---[RSA 2048]----+  
|      . =+.+. |  
|      . + .+ |  
|      . . . . |  
| o + o o + . |  
|o . o o.SB ..+ |  
|..      .+ +00+0 |  
|o .      ..*=.0. |  
|.o      . ..E+.0+|  
|.              .+=0+|  
+-----+  
+-----+
```

そのままエンター

1. 「TSUBAMEポータル」にログイン
2. 左側のメニューにある、「[SSH公開鍵登録](#)」をクリック
3. 先ほど生成した[公開鍵](#)（例：id_rsa.pub）の[中身をコピペ](#) or [ファイルをアップロード](#)する

※Puttyで作った鍵をアップロードする場合、**OpenSSH形式にする必要があります!**

TSUBAMEポータルページ

利用者権限：
アカウント：sekizawa-k-ac
現在の状態：利用中
所属グループ：hp190063

[TSUBAME利用状況]
[シヨブ一覧](#)
[予約ノード一覧](#)

[利用者情報]
[利用者情報表示](#)
[SSH公開鍵登録](#)
[Webサービス利用](#)

[パスワード設定](#)
[利用規約一覧](#)

[課金管理]
[支払コード管理](#)
[予算コード承認依頼\(0\)](#)

[グループ]
[所属グループ管理](#)
[グループ作成](#)
[履歴表示](#)

SSH公開鍵追加

SSH公開鍵をコード入力または、ファイルアップロードにて追加することができます。

SSH公開鍵コードを入力して追加ボタンを押してください。

公開鍵コード入力

ここにコピペ
→ 追加

○ OpenSSH形式
× PPK形式

登録するSSH公開鍵をアップロードしてください。

SSH公開鍵ファイル: ファイルが選択されていません。

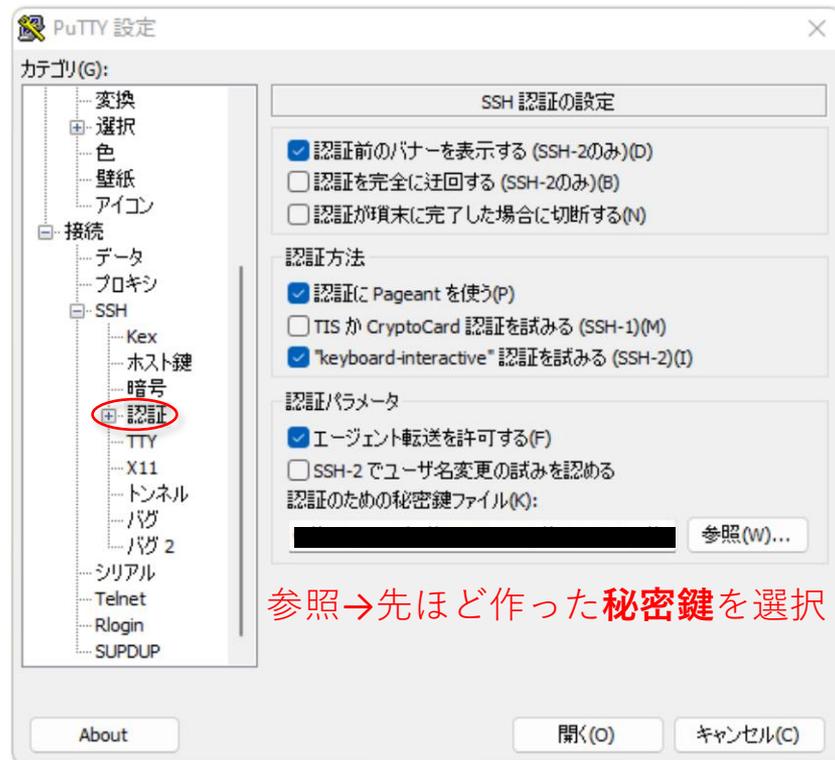
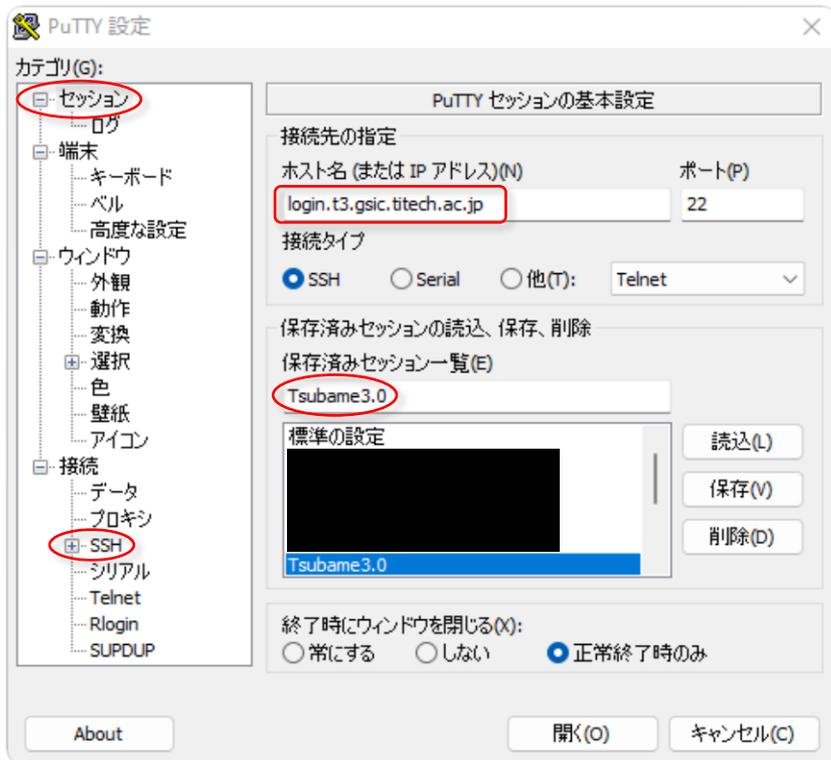
↑ここからアップロード

Windowsの場合

putty.exeを起動 → 下記の設定を入力

→ 「セッション→保存済みセッション一覧の下の箱に名前を入力 → 保存」

→ 「開く」を押すと、ターミナルが開く → ユーザー名・パスワードを入力してログイン



参照 → 先ほど作った秘密鍵を選択

Macの場合

秘密鍵を指定して，ログインノードにSSH接続する：

```
> ssh -i ~/.ssh/id_rsa 学籍番号@login.t3.gsic.titech.ac.jp
```

パーミッションの設定：

```
> chmod 700 ~/.ssh  
> chmod 600 ~/.ssh/id_rsa
```

これで他人は見ることができない

こんな感じのメッセージが出れば，ログイン成功！



```
Using username "sekizawa-k-ac".  
Authenticating with public key "imported-openssh-key"  
Passphrase for key "imported-openssh-key":  
Last login: Thu Feb  3 10:39:54 2022 from 131.112.52.220  
-----  
Last modified: 2022-12-05 16:00:00 JST  
  
** Do not run heavy programs like ISVs on login nodes login[01]. **  
  
(The current TSUBAME 3.0 operational status)  
https://www.t3.gsic.titech.ac.jp/      Twitter:@Titech_TSUBAME  
-----  
Currently Loaded Modulefiles:  
  1) cuda/11.0.3           3) intel-mpi/21.5.1  
  2) intel/21.5.0         4) fftw/3.3.9  
Currently Loaded Modulefiles:  
  1) cuda/11.0.3           3) intel-mpi/21.5.1  
  2) intel/21.5.0         4) fftw/3.3.9  
sekizawa-k-ac@login0:~> █
```



本日の目標

➤ 数値計算に必要な環境を整える！

➤ やることリスト

全部できましたか？
是非自分のPCにも設定してみてください！

- ✓ Slackのインストール・設定
- ✓ コンパイラ (**GFortran**) のインストール
- ✓ エディタ (**VSCode**) のインストール
- ✓ 可視化ソフト (**gnuplot**) のインストール
- ✓ SSH接続ソフト (**putty**・**WinSCP**) のインストール
- ✓ Tsubame3.0にアクセスしてみる

Kazuyuki Sekizawa

Associate Professor

Department of Physics, School of Science

Tokyo Institute of Technology

2-12-1 O-Okayama, Meguro, Tokyo 152-8551, Japan

sekizawa @ phys.titech.ac.jp

About me: <http://sekizawa.fizyka.pw.edu.pl/english/>

About us: <https://nuclphystitech.wordpress.com/>

See also:

