

Windows Or Linux Installation Instructions

JAN CAS

December 2022

Contents

1	Requirements	1
2	WSL & Ubuntu	2
2.1	Installing WSL	2
2.2	Installing Ubuntu	2
3	setting up DSGF c	2
3.1	installing the compiler	3
3.1.1	installing build-essential for linux	3
3.1.2	installing the MKL toolkit	3
3.1.3	installing the oneAPI compiler	4
3.1.4	installing Lapacke	4
3.1.5	finding your compiler version	4
3.2	downloading the DSGF code	5
3.2.1	Setting up GitHub SSH keys	5
3.2.2	cloning the DSGF c code	5
3.2.3	initializing the compiler	5
3.2.4	(Optional) permanently initializing the compiler	5
3.2.5	running it for the first time	6
3.3	Making sure the result is right	6

1 Requirements

This document serves as both the Linux and Windows installation instruction. Here are the requirements for running the DSGF code:

- Windows

Windows 10 or

For x64 systems: Version 1903 or later, with Build 18362 or later.

For ARM64 systems: Version 2004 or later, with Build 19041 or later.

Windows 11

- Linux & Windows

GitHub account

For Windows go to section 2. For Linux skip to section 3.

2 WSL & Ubuntu

On Windows we recommend using WSL 2 coupled with a Debian distribution such as Ubuntu

2.1 Installing WSL

To install WSL open the windows powershell as a administrator and type

```
$ WSL —install
```

This will install WSL 2 on your windows machine and allow you to install Linux distributions which will run native on the windows machine.

Ubuntu should also be installed automatically installed. However in the case that it isn't refer to the next section 2.2.

Once the installation is done you need to restart your machine for the changes to take effect

For more info about installing refer to <https://learn.microsoft.com/en-us/windows/wsl/install>

2.2 Installing Ubuntu

To install Ubuntu straight on your windows machine just go to the windows store and search for Ubuntu. Once installed it will prompt you to create a username and password.

Once you have your user on Ubuntu make sure to run:

```
$ sudo apt update
$ sudo apt upgrade
```

3 setting up DSGF c

This section will walk you through how to set up the DSGF C99 code on your machine assuming there is a Ubuntu distribution installed on it.

3.1 installing the compiler

Before installing the compiler make sure to update the distribution

```
$ sudo apt update  
$ sudo apt upgrade
```

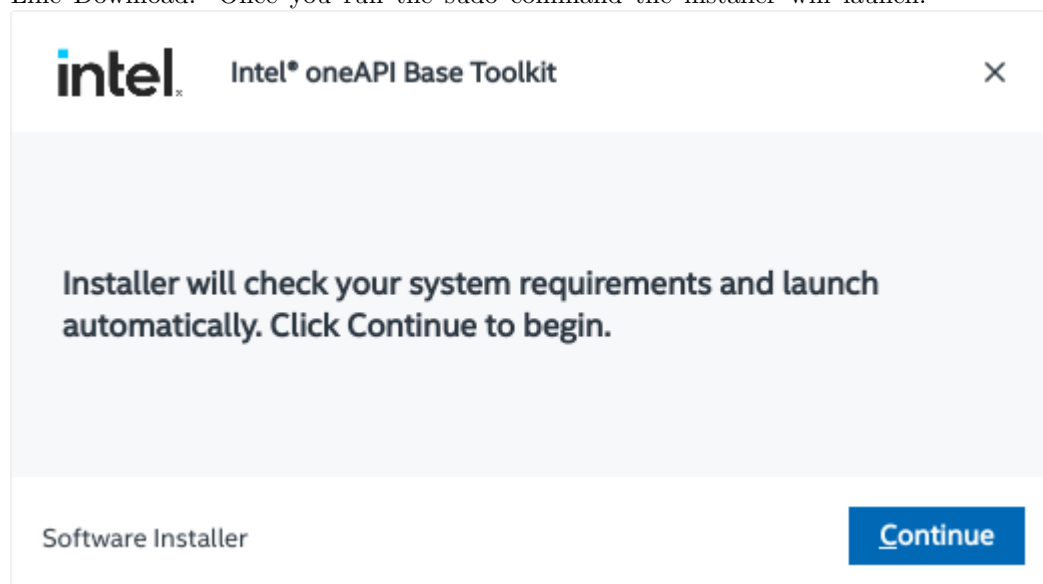
3.1.1 installing build-essential for linux

The compiler requires build-essential and pkg-config to be installed. Run:

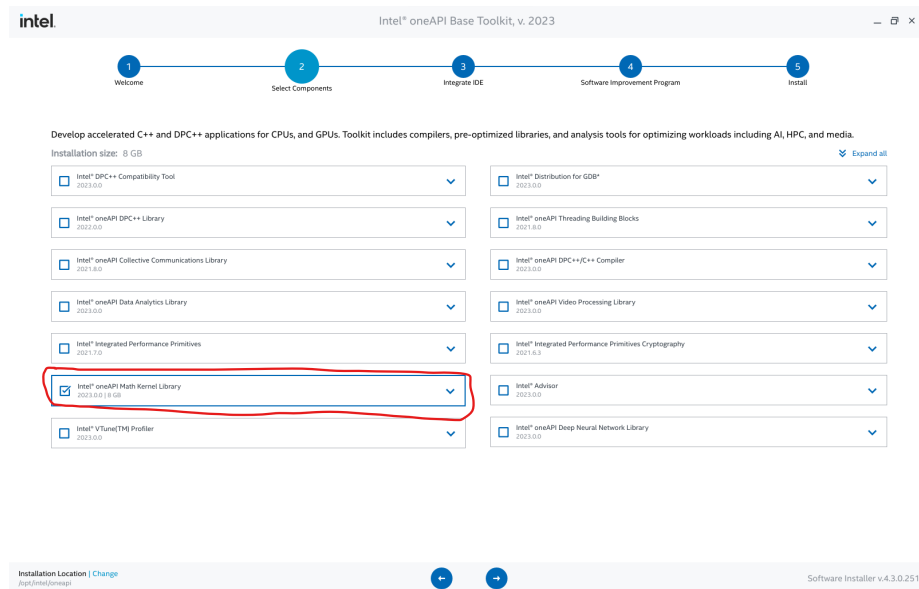
```
$ sudo apt install pkg-config build-essential
```

3.1.2 installing the MKL toolkit

The code uses the intel oneAPI compiler coupled with the MKL toolkit. Installation instructions are detailed on the [Intel Website MKL](#) under the Command Line Download. Once you run the sudo command the installer will launch.



once pressing continue this screen will pop up. Press Custom installation and only select the MKL toolbox as is shown in the next picture.



3.1.3 installing the oneAPI compiler

To install the Intel C compiler navigate to [Intel Website Compiler](#) and follow the instructions for the Command Line Download instructions.

3.1.4 installing Lapacke

The program relies on the Lapacke library. Here are the installation instructions

```
$ sudo apt update
$ sudo apt install liblapack3
$ sudo apt install liblapack-dev
$ sudo apt install libopenblas-base
$ sudo apt install libopenblas-dev
$ sudo apt install liblapacke-dev
```

3.1.5 finding your compiler version

Later when you try to run the code you will need to source the compiler so you will need to compiler version. To find the compiler version navigate to where it is installed. Usually this is at `opt/intel/oneapi/compiler/`. There you will be able to read the version number. For example:

```
ianlc@DESKTOP-6SK06PK:/opt/intel/oneapi/compiler$ ls
2023.0.0 latest
```

in this picture you can see that the version number is 2023.0.0 which is what

will be used throughout this setup. If you have a different version number make sure to substitute it when told.

3.2 downloading the DSGF code

As stated in the requirements you need a GitHub account in order to access the code.

3.2.1 Setting up GitHub SSH keys

Follow these [instructions](#) to set up ssh keys on Ubuntu.

3.2.2 cloning the DSGF c code

To get the DSGF c99 code open ubuntu and type:

```
$ git clone git@github.com:Discrete-System-Greens-Function/DSGF_c.git
```

3.2.3 initializing the compiler

These commands need to be run every time you reopen the shell and want to run the DSGF code. The version number was found in [3.1.5](#). 2023.0.0 is used here.

```
$ source /opt/intel/oneapi/mkl/2023.0.0/env/vars.sh
$ source /opt/intel/oneapi/compiler/2023.0.0/env/vars.sh
```

(Optional) To permanently initialize the compiler follow the instructions in the next section [3.2.4](#).

3.2.4 (Optional) permanently initializing the compiler

To initialize the compiler and MKL automatically every time you open shell do the following:

1. navigate to `/etc/profile.d/`
2. create a new file called `intel_compiler_env_vars.sh`
3. past the following into this file `. /opt/intel/oneapi/setvars.sh`

Once you have done that every time you open the shell the following should be displayed at the top:

```
:: initializing oneAPI environment ...  
-bash: BASH_VERSION = 5.1.16(1)-release  
args: Using "$@" for setvars.sh arguments:  
:: clck -- latest  
:: compiler -- latest  
:: debugger -- latest  
:: dev-utilities -- latest  
:: inspector -- latest  
:: itac -- latest  
:: mkl -- latest  
:: mpi -- latest  
:: tbb -- latest  
:: oneAPI environment initialized ::  
  
janlc@DESKTOP-6SK06PK:~$
```

Now you don't have to do steps in 3.2.3 every time you open the shell and can go straight to running the code.

3.2.5 running it for the first time

To compile and run the code type

```
$ ./run_DSGF.sh
```

This will run the code with the sample parameters.

3.3 Making sure the result is right

Once you have run the code from above make sure that the output looks like this:

```
End of frequency loop  
Total conductance at 3.000000e+02 K= 9.386028e-12
```