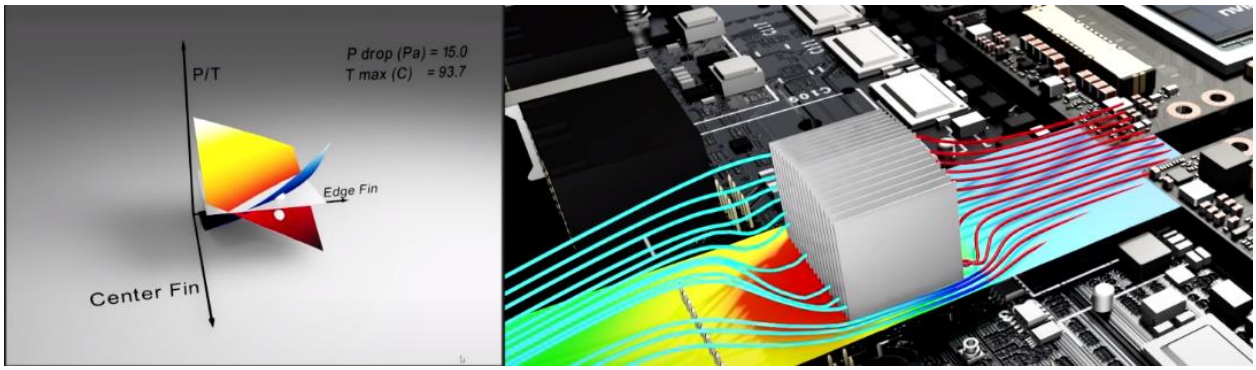
Modulus

A Neural Network Based Partial Differential Equation Solver



Installation Guidelines

Release v21.06 | November 9, 2021

NVIDIA

Notice

The information provided in this specification is believed to be accurate and reliable as of the date provided. However, NVIDIA Corporation ("NVIDIA") does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This publication supersedes and replaces all other specifications for the product that may have been previously supplied.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this specification, at any time and/or to discontinue any product or service without notice. Customer should obtain the latest relevant specification before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer. NVIDIA hereby expressly objects to applying any customer general terms and conditions with regard to the purchase of the NVIDIA product referenced in this specification.

NVIDIA products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on these specifications will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this specification. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this specification, or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this specification. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this specification is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, CUDA, and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

Modulus Installation Guidelines

There are two ways you can install Modulus, by using the Docker image or using Bare Metal Installation. Due to dependencies such as TensorFlow and Horovod, NVIDIA highly recommends using Modulus with the docker image provided, because it contains TensorFlow and Horovod which are required. Using this docker image allows for maximal utilization of the GPUs as well.

System Requirements

Operating System	• Ubuntu 18.04 or Linux 4.18 kernel
Driver & GPU Requirements	• Bare Metal version: NVIDIA driver 465.19 required only if SDF library is used • Docker container: NVIDIA driver 465.19 or higher driver must be used. If using a Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 440.30 or 418.xx however any drivers older than 465 will not support the SDF library. (https://docs.nvidia.com/deeplearning/frameworks/support-matrix/index.html)
Required installations for Bare Metal version	• Python 3.6 • Tensorflow 1.15 • Horovod 0.21.0

Supported Processors	• 64-bit x86
	(this dependency is only when the SDF library is used since the SDF library is compiled on x86. If you need the SDF compiled on Power9 architecture then please e-mail us at: modulus-team@exchange.nvidia.com)
	
	• NVIDIA GPU based on the following architectures:
	  ○ Nvidia Ampere GPU Architecture (A100)
	  ○ Volta (V100, Titan V, Quadro GV100)
	  ○ Turing (T4, Quadro RTX series)
	  ○ Pascal (P100, P40, P4, Titan Xp, Titan X)

	All studies in the User Guide are done using V100 on DGX-1. A100 has also been tested.

NOTE: To get the benefits of all the performance improvements (e.g. AMP, multi-GPU scaling, etc.), use the NVIDIA Tensorflow container for Modulus. This container comes with all the prerequisites and dependencies and allows you to get started efficiently with Modulus.

Modulus with Docker Image (Recommended)

Install the Docker Engine

To start working with Modulus, ensure that you have Docker Engine installed. The steps to install docker can be found here: https://docs.docker.com/engine/install/ubuntu/

You will also need to install the nvidia docker toolkit found here: https://github.com/NVIDIA/nvidia-docker. This should work on most debian based systems: `sudo apt-get install nvidia-docker2`. Running Modulus in the docker image while using SDF library may require nvidia-container-toolkit version greater or equal to 1.0.4.

To run the docker commands without `sudo`, add yourself to the `docker` group by following the steps 1-4 found in Manage Docker as a non-root user: https://docs.docker.com/engine/install/linux-postinstall/

Install Modulus

Download the Modulus docker container.
Once downloaded, load the Modulus container into docker using the following command (This may take several minutes):

```
docker load -i modulus_image_v21.06.tar.gz
```

Once complete, `Loaded image: modulus:21.06` will get printed in the console.

Using the Modulus examples

All examples can be found in the Modulus examples tarball.
Once the tarball is downloaded, you can run the docker image and mount the Modulus examples using:

```
tar -xvzf ./Modulus_examples.tar.gz
docker run --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 \
           --runtime nvidia -v ${PWD}/examples:/examples \
           -it modulus:21.06 bash
```

To verify the installation has been done correctly, you can run the following commands:

```
cd helmholtz/
python helmholtz.py
```

If you see `network_checkpoint_hemholtz/` directory created after the execution of the command (~5 min), the installation is successful.

Note: If you intend to use the quadrature functionality of Modulus (e.g. User Guide Section 9.6), please install the `quadpy` package inside the container using the following commands:

```
pip install quadpy
```

Modulus Bare Metal Installation

While NVIDIA recommends using the docker image provided to run Modulus, installation instructions for Ubuntu 18.04 are also provided. Modulus requires Cuda to be installed. For compatibility with TensorFlow 1.15, use Cuda 10.2 or later. Modulus requires Python 3.6 or later.

Other dependencies can be installed using:

```
pip3 install matplotlib transforms3d future typing numpy quadpy\
             numpy-stl==2.11.2 h5py sympy==1.5.1 termcolor psutil\
             symengine==0.6.1 numba Cython chaospy
pip3 install -U https://github.com/paulo-herrera/PyEVTK/archive/v1.1.2.tar.gz
```

Note: Currently, Modulus has only been tested for `numpy-stl 2.11.2` , `sympy 1.5.1` , `symengine 0.6.1` and `pyevtk 1.1.2` versions. Using other versions for these packages might give errors.

Once all dependencies are installed, the Modulus source code can be downloaded.
Modulus can be installed from the Modulus source tar ball using:

```
tar -xvzf ./Modulus_source.tar.gz
cd ./Modulus/
python setup.py install
```

To run examples using the STL point cloud generation you will need to put `libsdf.so` in your library path and install the accompanying PySDF library. This can be done using,

```
export LD_LIBRARY_PATH=$(pwd)/Modulus/external/pysdf/build/:${LD_LIBRARY_PATH}
cd ./Modulus/external/pysdf/
python setup.py install
```

Using the Modulus examples

All examples can be found in the Modulus examples tarball.
To verify the installation has been done correctly, you can run the following commands:

```
tar -xvzf ./Modulus_examples.tar.gz
cd examples/helmholtz/
python helmholtz.py
```

If you see `network_checkpoint_hemholtz/` directory created after the execution of the command (~5 min), the installation is successful.

Note: To verify the installation of SDF library and the STL geometry support, you can run the following:

```
cd examples/aneurysm/
python aneurysm.py
```

Running Jobs using Multiple GPUs

Use these steps to run jobs using multiple GPUs:

1. Find out the available GPU devices. This can be done using:

```
nvidia-smi
```

2. Run the multi GPU job using `horovodrun -np #GPUs` . The below command shows how to run a job using 2 GPUs.

```
cd examples/ldc/
horovodrun -np 2 python ldc_2d.py
```