DUNE/PDELab Course Material

Release 2.7.0

unknown

Mar 29, 2021

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 About Dune

The Distributed and Unified Numerics Environment (DUNE) is a software framework for the numerical solution of partial differential equations with grid-based methods. Using generic programming techniques it strives for both: high flexibility (efficiency of the programmer) and high performance (efficiency of the program). DUNE provides, among other things, a large variety of local mesh refinement techniques, a scalable parallel programming model, an ample collection of finite element methods and efficient linear solvers.

1.2 About this Course

The course material you are looking at has been used to teach an annual one week long summer school at the Interdisplinary Center for Scientific Computing at Heidelberg University for more than a decade. With the pandemic prohibiting the traditional format, the course has been given virtually for the first time in March 2021. With all our lectures being recorded, we are making the material now available for self study.

1.3 How to study with the Material

The course material is organized as *a number of lectures*. Each lecture consists of the following aspects:

- One or more YouTube videos with the actual lecture that describes theoretical *and* implementation aspects of the lecture topic.
- A PDF document explaining the lecture content in written form.
- An exercise sheet with assignments for you to reflect on the lecture content.

It is important to understand that both the lecture (either in written form or as an actual lecture) and the exercises are necessary to build a profound understanding of the Dune framework.

The exercise material assumes some familiarity with the Linux shell e.g. switching directories with `cd`. If you have never worked with Linux before you might want to consider watching this video after *setting up your environment*:

https://youtu.be/ZmNFWNMlcdM

In order to do the exercises, it is important to understand the directory structure of the course material. This structure results from the modular structure of the Dune ecosystem: The `dune` subdirectory contains sources of all required Dune modules. The `release-build` contains an *out-of-source build* performed by CMake. This build mirrors the directory structure of the source tree, but only contains generated files, build artifacts and simulation results:

```
iwr-course-2021
├── dune
│   ├── dune-common
│   ├── dune-grid
│   ├── ...
│   └── dune-pdelab-tutorials
│       ├── tutorial00
│       │   ├── src
│       │   ├── doc
│       │   └── exercise
│       │       ├── task
│       │       ├── doc
│       │       └── solution
│       ├── tutorial01 ...
│       └── ...
├── release-build
│   ├── dune-common
│   ├── dune-grid
│   ├── ...
│   └── dune-pdelab-tutorials
│       ├── src_dir
│       ├── tutorial00
│       │   ├── src_dir
│       │   ├── src
│       │   │   └── src_dir
│       │   ├── doc
│       │   │   └── src_dir
│       │   └── exercise
│       │       ├── src_dir
│       │       ├── task
│       │       │   └── src_dir
│       │       ├── doc
│       │       │   └── src_dir
│       │       └── solution
│       │           └── src_dir
│       ├── tutorial01 ...
│       └── ...
├── release.opts
└── ...
```

Note: The `src_dir` directories in the build directory are symbolic links to the corresponding source directory. This allows us to always operate directly in the build directory, but quickly jump to the correct source directory using `cd`.

The relevant Dune module for the course material is `dune-pdelab-tutorials`. It contains a subdirectory per lecture. Each of these lecture directories has the structure outlined for `tutorial00` above:

- `src` contains the C++ sources of the executable that is used in the lecture to illustrate the lecture topic. It is a fully functional simulation executable. There is typically additional configuration files that control the simulation in the `src` directory.
- `exercise/task` contains the C++ sources for the exercise. You are not expected to write C++ programs from scratch. Instead, your task is to modify the given program.
- `exercises/solution` contains the full solution to the tasks. Feel free to consult it if you are stuck.
- `doc` and `exercise/doc` contain the Latex sources for the lecture. You do not need to build this yourself.

We have also summarized the structure of the lecture and exercise material in this short video:

https://youtu.be/7mYEHKHhvCg

1.4 Setting up the exercise environment

There is two ways of getting the code and the execution environment for the course material: Directly or within a virtual machine. We only describe the direct setup for recent Debian or Ubuntu systems, where as the virtual machine setup will work on all operating systems.

If you are on an Debian (e.g. version 10) or Ubuntu (e.g. 20.04 LTS) system, you should open a terminal and execute the following sequence of commands. This will download and build the course material on your computer.

```
sudo apt install git cmake build-essential paraview gmsh libsuitesparse-dev␣
↪libsuperlu-dev openmpi-bin libopenmpi-dev python3-dev python3-pip pkg-config
git clone --recursive https://gitlab.dune-project.org/dune-course/iwr-course-2021.git
cd iwr-course-2021
./install.sh
```

Note: Building the course material can take up to an hour. Be patient and do not close the terminal.

If you are running a different operating system, you can instead work in a virtual machine. In order to do so you need to:

- Install VirtualBox version 6.1 from https://www.virtualbox.org
- Download our VM image from here: https://heibox.uni-heidelberg.de/f/be208766b89f4d2188cb/?dl=1
- Add the VM to VirtualBox and start it.
- Log in to the Debian 10 guest system with the user *dune* and the password *course*

If you have never worked with virtual machines before, you can also watch this video that runs you through the process:

https://youtu.be/IsoC5mDyHzE

CHAPTER

TWO

LECTURES

2.1 C++ for Scientific Computing

The Dune framework is written in modern C++ - mastering Dune will also require expertise in C++. However, providing a full blown introduction to C++ is out of scope of this course. We do however provide this lecture that highlights some aspects of modern C++ that we consider important for Scientific Computing and for using Dune.

The lecture is split into three parts that you can watch below. You can also `download the lecture slides`.

https://youtu.be/IIaD1jMRtHc

https://youtu.be/CuN8GmPnnwI

https://youtu.be/3HQbxVbrNxk

The exercise sheet for this C++ lecture can be `downloaded here`. The material for this exercise can be found in the `c++` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.2 Introduction to Finite Elements

This course focusses on solving PDEs with the Finite Element Method. Similar to the *Introduction to C++*, a rigorous derivation of the Finite Element Method is out of scope of this course. We do however reiterates all of the basic elements of the method that are relevant for its implementation.

We strongly recommend you watching this introduction even if you are already familiar with the Finite Element Method, as the lecture not only introduces the mathematical concepts, it also introduces the mathematical formulations and notation that we will be using during the rest of the course.

The lecture is split into two parts:

https://youtu.be/FAuyWUL9mmc

https://youtu.be/2jrJi3y2wSw

2.3 The Dune Grid Interface

This lecture introduces the core concept of the Dune framework: The Grid Interface. The lecture is split into four parts that can be watched below. You can `download the lecture slides here`.

Part 1: Introduction

https://youtu.be/hTkd2oKoK-k

Part 2: The Grid

https://youtu.be/bnHASaE_qqE

Part 3: Entities & Intersections

https://youtu.be/63d5Y4PieQ4

Part 4: Data Handling

https://youtu.be/RTb-uETPGYI

The exercise sheet for the Grid interface lecture can be `downloaded here`. The material for this exercise can be found in the `gridinterface` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.4 Simulation Workflow

This lecture describes a bunch of topics that are of relevance for understanding Dune and to be able to do the exercises, but that do not really fit with any of the other lectures' topics. You can `download the lecture slides here`.

The first part talks about Dune's modular structure and the CMake-based build system that Dune is using:

https://youtu.be/pSId3V7S2Ec

The next part reiterates on one of the native strengths of Dune compared to competitors: The availability of a variety of grid implementations with very different feature sets. It also explains how grids are constructed using generic factory concepts.

https://youtu.be/hWgAYXhCT0s

The third part briefly explains the configuration file format that is typically used in Dune.

https://youtu.be/lz461yrPL6g

The last part is quite relevant for the exercises as it explains Dune's tackle on visualization. File exports to the VTK format are shown and a hands-on session with ParaView illustrates the intended workflow for the exercises:

https://youtu.be/7iDOni83uTQ

The exercise sheet for this lecture can be `downloaded here`. The material for this exercise can be found in the `workflow` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.5 Elliptic Problems

This lecture will introduce you to the discretization framework dune-pdelab. The lecture is split in two parts. In a first part, we will study the typical model problem: *P1 finite elements for Poisson equation*. In a second part, we will study common generalizations of this model problem, like arbitrary polynomial degree, quadrilateral meshes, nonlinearities etc.

The recording of the first lecture about *P1 finite elements for Poisson equation* is available below. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

P1 finite elements for Poisson equation - Theory Part:

https://youtu.be/puYblvIR9qY

P1 finite elements for Poisson equation - Implementation Part:

https://youtu.be/nW6kUDMCkbQ

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial00` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

In the second part of this lecture, we will study common generalizations of this model problem, like arbitrary polynomial degree, quadrilateral meshes, nonlinearities etc. The recording are embedded below. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

Theory Part:

https://youtu.be/mbQikv3ob0U

Implementation Part:

https://youtu.be/zjgRQm5z3to

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial01` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.6 Instationary Problems

In this lecture, the (nonlinear) heat equation will be solved with PDELab. The explained concepts apply to all instationary PDE problems. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

https://youtu.be/GBPuTEm3yGY

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial03` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.7 Finite Volumes

In this lecture, the Cell-Centered Finite Volume Method will be introduced and implemented in PDELab. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

https://youtu.be/DL-pjA7A0PY

There is currently no exercise sheet for this lecture.

2.8 Systems of PDEs

In this lecture, the wave equation will be discretized as a system of PDEs and solved with PDELab. The explained concepts apply to all systems of PDEs. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

https://youtu.be/Tx5gmH-7Jzk

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial04` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.9 Adaptivity

In this lesson the nonlinear Poisson equation will be solved with adaptive mesh refinement, introducing the relevant Dune and PDELab concepts for error estimation and mesh adaptation. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

https://youtu.be/KEffRt-Molw

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial05` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.10 Parallelization

This lecture explains how MPI parallelism is incorporated into the Dune framework. A written form of this tutorial is available `for download`. You can also `download the lecture slides`.

https://youtu.be/9X7pWHGf9JU

https://youtu.be/Cr57YL_veU8

https://youtu.be/WYoJdUI5YCg

https://youtu.be/CIfoXYjjya0

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial06` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

2.11 Code Generation with Python

The Python programming language is increasingly popular in scientific applications. Recently, we invested work into adding a code generation toolchain to dune-pdelab. It allows to express the weak formulation of your PDE in a domain specific language (UFL) and automatically generates the necessary integration kernels (aka the `LocalOperator`) from that input. This lesson explains the use of that system. You can also `download the lecture slides`.

https://youtu.be/m26jHtaoeRE

The exercise sheet for this lesson can be `downloaded here`. The material for this exercise can be found in the `tutorial09` subfolder of the `dune-pdelab-tutorials` module. Check *How to study with the Material* for details on how to find the material.

CHAPTER

THREE

QUESTIONS AND ANSWERS

Asking questions to experienced developers is vital for becoming an expert with any piece of scientific software. In order to get your questions about Dune answered there is the following possibilities:

- Subscribe to the Dune mailing list to receive important notifications and be able to ask general questions about Dune.
- Subscribe to the PDELab mailing list for topics of PDELab specifically.
- Sign up at the Dune Gitlab instance to be able to see developer discussions and report issues.

If you have questions about the exercises from this course material specifically, we will also try to host a Q&A session through a video conferencing tool at some point. We will advertise this session through the PDELab mailing list.

CHAPTER

FOUR

LICENSING AND COPYRIGHT

The course material on this site is licensed as follows:

- The YouTube videos are licensed CC BY 4.0.
- The linked PDF documents are licensed CC BY-SA 3.0.
- All the contained code snippets are additionally licensed BSD-3.

The list of copyright holders is:

- Peter Bastian
- Markus Blatt
- Christian Engwer
- René Heß
- Olaf Ippisch
- Dominic Kempf
- Ole Klein
- Piotr Minakowski
- Steffen Müthing
- Marian Piatkowski