Getting started with ALSVID-UQ [1]

eVITA Winter School 2015

Kjetil Olsen Lye

January 18, 2015

[1] http://www.sam.math.ethz.ch/alsvid-uq/

The material covered in this presentation is also covered with greater detail at the homepage of ALSVID-UQ:

`http://www.sam.math.ethz.ch/alsvid-uq/`

The slides will be available online

Prerequisites

[Ubuntu package names in blue]

- [python2.7]: Python 2.5 or greater (`www.python.org`)
- [g++]: GCC C++ like compiler
- [python-numpy]: numpy [**plots**] (`www.numpy.org`)
- [python-matplotlib] matplotlib [**plots**] (`matplotlib.org`)
- [mayavi2] mayavi [**3D-plots**] (`code.enthought.com/projects/mayavi/`)
- [openmpi-bin] MPI [**multinode**] (`www.open-mpi.org`)

To install these on Ubuntu, issue the command

```
sudo apt-get install python2.7 g++ mayavi2 \
    python-numpy python-matplotlib openmpi-bin
```

Downloading and installing

- Latest version available at `http://www.sam.math.ethz.ch/alsvid-uq/`
- Direct download link: `http://www.sam.math.ethz.ch/alsvid-uq/3311/alsvid-uq-3.0.tar.gz`
- Installation: Untar to suitable destination
- Installation: Copy `alsvid-uq-3.0/configs/local-mayavi_visualization.py` to `alsvid-uq-3.0/local_visualization.py`

Download and installation command (Linux):

```
wget http://www.sam.math.ethz.ch/alsvid-uq/3311/alsvid-uq\
    -3.0.tar.gz
tar xvf alsvid-uq-3.0.tar.gz
cd alsvid-uq-3.0
cp configs/local-mayavi_visualization.py \
    local_visualization.py
```

Running ALSVID-UQ: Interactive mode

- Change directory to `alsvid-uq-3.0/run`

  ```
  cd alsvid-uq-3.0/run
  ```

- Start ALSVID-UQ in interactive mode (gives overview of options)

  ```
  python ../make.py --ask
  ```

- Each presented option has a sane default value

Sample output:

```
python ../make.py --ask
Major Options:
Select equation:
bl          "Buckley-Leverett (scalar nonlinear conservation la
burgers     "Burgers' (scalar nonlinear conservation law)"
euler       "Euler (Euler equations of gas dynamics)"
linadv      "Linear Advection (scalar linear conservation law)"
mhd         "MHD (Magneto-HydroDynamics equations)"
sw          "Shallow Water Equations"
wave        "Wave (Wave equation)"
equation: [mhd]
```

Running ALSVID-UQ: Non-interactive

Can run with all options specified on command line

```
python ../make.py equation:burgers model:sine space:o2eno
```

Short explanation:

equation	equation to solve
model	initial value
space	space solver

Plotting

To plot results, start

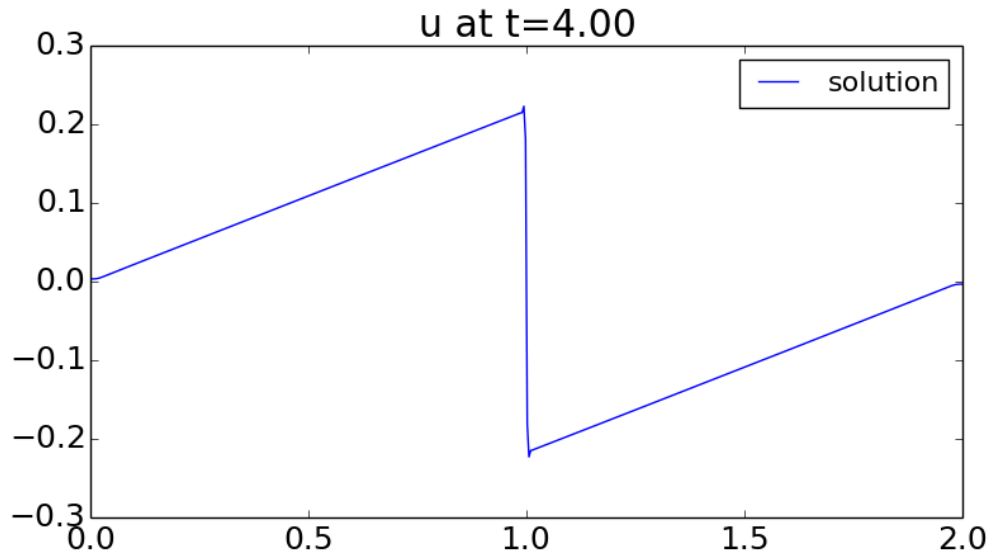
```
python -i ../plot.py
```

Inside the Python shell, type:

```
>>> r.plot(br.U)
```

Explanation:

- `r` the data from the run
- `br` Burgers' specific



OPTS: balancer:static, equation:burgers, flux:det, model:sine, multi:single, rng:well512a, solver:hll, space:o2weno, stats:mean-var
VARS:
INFO: cores: 1, runtime: 0:00:00

Saving more outputs

- Default: Only last timestep saved
- Can add more with NSAVES option:

```
python ../make.py ... NSAVES=10
```

- Can plot specific timestep with

```
>>> r.plot(br.U, ts=3)
```

Bonus slide: Creating custom model

It is relatively straightforward to create custom model

1. Create a new file in `src` with name

   `model-<equation>_<model name>.cpp`

   eg.

   `model-burgers_geilo.cpp`

2. Fill in the newly created file with

```
#include "equation.h"
#include "chaos.h"

///::title:: <your title>
///::vars:: NX=<nx> MAXT=<maxt>
///::consts:: MAXX=<maxx>
///::bc:: {NEUMANN, PERIODIC, MIXED}

void initial_data (PrimitiveVars &v,
     real x, real y, real z) {
  v.u = <value>; // Any C++ is allowed
}
```

Bonus slide: Creating custom model

A concrete example: `model-burgers_geilo.cpp`

```
#include "equation.h"
#include "chaos.h"

///::title:: Sine wave created at Geilo Winter School 2015
///::vars:: NX=512 MAXT=4
///::consts:: MAXX=2
///::bc:: NEUMANN

void initial_data (PrimitiveVars &v,
     real x, real y, real z) {
  v.u = sin(pi*x);
}
```

Run with

```
python ../make.py equation:burgers model:geilo space:o2eno
```