Uncertaintity quantification with ALSVID-UQ [1]

eVITA Winter School 2015

Kjetil Olsen Lye

January 20, 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

Running with Monte-Carlo

To run with Monte-Carlo, simply add `ML=<number of samples>`:

```
python ../make.py ... ML=<number of samples>
```

eg.

```
python ../make.py equation:burgers model:sine \
    space:o2eno ML=50
```

Plotting mean and variance

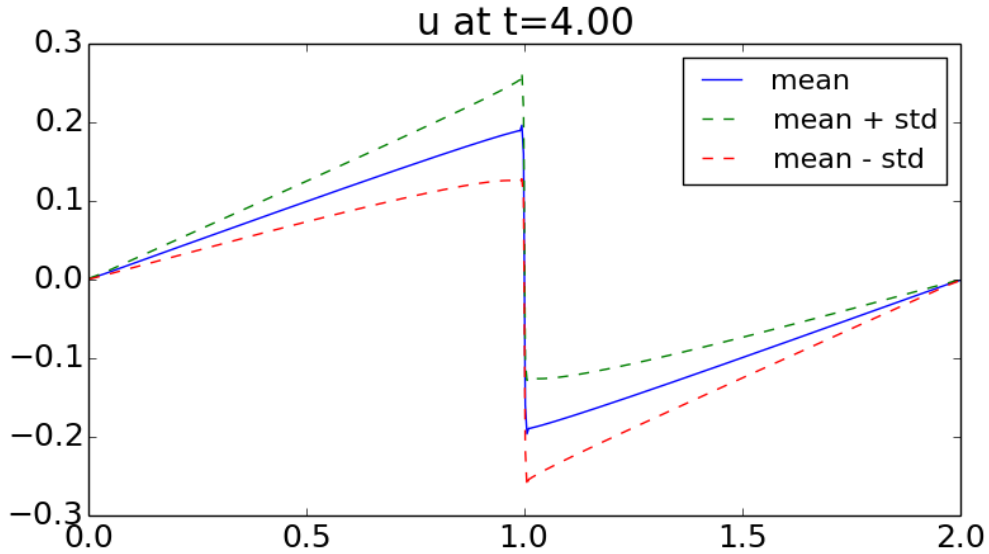
Mean and variance automatically included:

To plot results, start

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

Inside the Python shell, type:

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



OPTS: balancer:static, equation:burgers, flux:det, model:sine, multi:single, rng:well512a, solver:hll, space:o2eno, stats:mean-var
VARS: ML=50
INFO: cores: 1, runtime: 0:00:08
SAMPLES: 50

Running multilevel MC

Add the options `L=<number of levels>` and `ML=<number of samples on finest level>` eg.

```
python ../make.py equation:burgers model:sine \
    space:o2eno ML=4 L=3
```

Bonus slide: Adding stochastic initial data

We will modify our model from yesterday and add some stochastic initial data

1. Add the function `void model_set_chaos_size ()`
2. Fill the function with

```
// Number of uniform i.d.d. variables
CHAOS_SIZE [MODEL] [GLOBAL] [UNIFORM] = 1;
// Number of normal i.d.d. variables
CHAOS_SIZE [MODEL] [GLOBAL] [NORMAL]  = 0;

// These two can be ignored for simple use:
CHAOS_SIZE [MODEL] [LOCAL]  [UNIFORM] = 0;
CHAOS_SIZE [MODEL] [LOCAL]  [NORMAL]  = 0;
```

3. Use

   `chaos [MODEL] [GLOBAL] [UNIFORM] [<random var index>]`

   to gain access to random data in `void initial_data()`

Bonus slide: Adding stochastic initial data

```
#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 model_set_chaos_size () {
  CHAOS_SIZE [MODEL] [GLOBAL] [UNIFORM] = 1;
  CHAOS_SIZE [MODEL] [GLOBAL] [NORMAL]  = 0;
  CHAOS_SIZE [MODEL] [LOCAL]  [UNIFORM] = 0;
  CHAOS_SIZE [MODEL] [LOCAL]  [NORMAL]  = 0;
}

void initial_data (PrimitiveVars &v, real x, real y, real z)
{
  v.u = chaos [MODEL] [GLOBAL] [UNIFORM] [0] * sin(pi*x);
}
```