IPVS – Universität Stuttgart WS 17/18
Abteilung Simulation großer Systeme
Prof. Miriam Mehl
Malte Brunn, Amin Totounferoush, Dr.-Ing. Nehzat Emamy

Numerical Simulation
Exercise sheet 4

Erweiterung: Uncertainty Quantification
Hand in via email to amin.totounferoush@ipvs.uni-stuttgart.de until 16.01.2018, 18:00
Contact for quesions about the framework malte.brunn@ipvs.uni-stuttgart.de

4.1 Uncertainty Quantification

In the framework of this exercise, we want to quantify the uncertainties referring to the actual Reynolds number. For this purpose, we first use the Monte-Carlo method. This has the advantage that it is relatively easy to implement and has a dimension-independent convergence. Second, we implement the quadrature with trapezoidal rule, which has a better convergence. However, its convergence depends on the dimension of the parameter space.

4.1.1 Studies on the driven cavity flow

Consider a normal distribution of the Reynolds number around the expectation $\mu = 1500$ with the standard deviation of $\sigma = \frac{1000}{6}$. Estimate the expectation of the u-velocity and its standard deviation at the grid points $(120, 5)$, $(64, 64)$ and $(5, 120)$ over the time. Employ a 128×128 grid for this purpose and set a proper constant time step such that the same times could be obtained for all runs.

a) Execute 2000 simulations for Monte-Carlo runs and estimate the expectation and variance of the requested velocity. In order to estimate the rate of convergence, compute both values (expectation and variance) at the last time step after 500 and 1000 runs.

b) Approximate the expectation and standard deviation using the trapezoidal rule considering 50, 100 and 200 nodes. Limit the integral to the interval $[\mu - 3\sigma; \mu + 3\sigma]$.

Changes to the implementation To quantify the uncertainties, you must add the following functionalities to your code:

- Extend your program for:

  a) Generation of the random values of the Reynolds number with respect to the above given normal distribution (for the Monte-Carlo method) and equidistant nodes in the interval $[\mu - 3\sigma; \mu + 3\sigma]$ for the trapezoidal rule,
  b) creation of the input data with the generated parameters (Reynolds numbers),
  c) the possibility to start the flow solver code, and
  d) the computation of the expectation and variance from the gathered output values.

- Generate the normally distributed random numbers around the expectation $\mu = 1500$ with the standard deviation of $\sigma = \frac{1000}{6}$ by the many possibilities that C++ provides since its 2011 standard [13] *(importance sampling)*. Or use the uniformly distributed random

[13] http://www.cplusplus.com/reference/random/ (C++11)

numbers [14] which you have to weight with respect to the probabilities when computing the expectation and variance *(weight uniformly distributed sampling).*

- To start programs from other programs, you could use the C++ function `system`[15].

- Compute the expectation and variance of the velocity after each time step according to the formulas in the lecture notes. Set an array of the size `[(tEnd / deltaT) + 1]`. Moreover, put away the vtk output (even if it is difficult for you ☺) and write instead a new function that writes the velocity components at the related points in a text file that you parse later. Here, there is the csv format: use one line for each time step and use one column for the velocity components of each point.

Note I: Take care that the many simulations require a non negligible amount of time! With a simulation time of 3 minutes for your flow solver, you need 75 hours of computing time for 1500 simulations.
Implement your control program or script such that it is relatively error tolerant with respect to breakups. For this purpose, separate the program sequence in a computation step and a following evaluation step. Initially, in the computation step, the program runs the required number of simulations and writes the determined values including the used sample (from the parameter space) in the persistent memory (the sample could be written to the name of the file, for example). Besides, use the command `nohup`[16] in order that your programs runs on the background even when you close your ssh-session.

Note II: Notice that the you can perform independent simulations in parallel.

Besides submitting your code (control program and flow solver), hand in plots for each measurement point ($(120, 5)$, $(64, 64)$ and $(5, 120)$) and each method (for the trapezoidal rule: also for each resolution), in which the expectation and standard deviation are plotted over the time.

4.2 Questions

- Perform convergence studies based on the results for different samples/number of nodes for both methods. Think about the results if they agree with your expectation (and the theoretical results from the lecture).

- Make an approximate plot of the distribution of the Reynolds number as well as the velocity based on the taken samples at the three given grid points. Use only the values at time t_{end} of your simulations for this matter.

- Assume that you have a program to use a quadrature rule on a regular m-dimensional grid in the parameter space, which can evaluate integrals with accuracy $\mathcal{O}(h^p)$. h is here the grid size of the regular grid in the parameter space. From which dimension m does the Monte-Carlo method converge faster than this quadrature rule in order to approximate the expectation?

[14] http://www.cplusplus.com/reference/cstdlib/rand/
[15] http://www.cplusplus.com/reference/cstdlib/system/
[16] http://linux.die.net/man/1/nohup