Numerical methods for Physicists

Leif Lönnblad

May 28, 2007

Abstract

Description of the course *Numerical methods for Physicists* (NUM) 7.5 ECTS credits, including five PBL cycles with associated exercises.

1 Introduction

This course is intended to give the basics of some important method used for numerical computations in physics. The course should provide the theoretical and mathematical basis for these methods as well as some practical experience. The course will also give some experience in using programming languages, mainly C and C++.

The course is based on a master-level lecture course (FYS232) given at the Department of theoretical physics in Lund, and also uses some of the exercises developed for that course.

2 Course Overview

2.1 The cycles

The course consists of 5 one week PBL cycles as follows:

- Numeric Integration
- Monte Carlo
- Optimization and minimization
- Ordinary differential equations
- Partial differential equations

The goals for these are detailed below. In addition there are some overall goals, including the understanding of different kinds of errors which are introduced in numerical calculations, and how to minimize these. Another goal is to understand the basic theory of random numbers, and to gain experience in programming from working with programs written in C and C++.

2.2 Numerical Integration

This part introduces the standard numerical integration algorithms such as the *trapezoidal rule*, the *Simpsons rule* and the *Gaussian quadrature* and how these can be derived through interpolation and extrapolation techniques.

2.3 Monte Carlo

Introduction to the Monte Carlo technique for numerical estimates of integrals. This includes some basic properties of random variables, the transformation of distributions of random variables, and the Metropolis algorithm.

2.4 Optimization and minimization

Different techniques for optimization and minimization, such as the Simplex and Conjugate gradient descent algorithms. Also discusses the usage of Monte Carlo techniques, such as *Simulated annealing*. The emphasis is on χ^2 minimization, which is what the students mostly will encounter in HEP.

2.5 Ordinary differential equations

Studies different methods for solving ordinary differential equations and their relative benefits in terms of precision, computational cost and the stability of the solutions. Both explicit methods, such as *Rung-Kutta*, and implicit methods are discussed.

2.6 Partial differential equations

Studies the most common types of partial differential equations, such as the Poisson, diffusion and wave equations. Different methods are discussed and compared in terms of precision, computational cost and the stability of the solutions.

3 Literature and Examination

Most of the course content is covered in

> W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1988).

which is available on-line at http://www.nrbook.com Also other textbooks in the subjects can be used as complementary literature, eg.

> C-E. Fröberg, *Numerical Mathematics*, Addison-Wesley (1985).

Some parts are covered in the lecture notes to the FYS232 course, available on-line at http://www.thep.lu.se/∼leif/fys232/notes.pdf

The examination is a standard written exam for the course (see Appendix).

Appendices: PBL cycles and examination.

Numerical methods for Physicists (NUM)

Cycle 1

Numeric Integration

The nasty integral

— So, did you talk to professor Rechner about evaluating that nasty integral?

— Well, no not really. He was in a hurry and just gave me some crap about how easy numerical integration is "Just replace dx with Δx and $\int$ with $\sum$". And when I asked him about controlling the precision he just said "Think Taylor expansion!" and ran away to catch his bus.

— But didn't you already try to just discretize and sum?

— Yes, but the the result printed out from my program was "NaN" so something must have gone wrong somewhere.

Literature

Most of the course content is covered in

> W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1988).

and for this cycle the most relevant chapters are 3 and 4.

Goals

- Introduction to the local computers
- Lagrange-interpolation
- Richardson extrapolation
- Trapezoidal rule
- Simpsons rule
- Gaussian quadrature

Comments to exercises

Besides familiarizing the students with the computer system, the exercise is to evaluate the integrals of three functions, which are unknown to the student and are only supplied as compiled object files. The functions and the corresponding integrals are

1. $$\int_0^{2\pi} \frac{\sin x}{x} dx$$
2. $$\int_0^1 \frac{1+x^2}{\sqrt{x}} dx$$
3. $$\int_{x^2+y^2<1} e^{-(x^2+y^2)} dx dy$$

Exercise 1

Go through the computer introduction for the course available at http://www.thep.lu.se/∼carl/fys232

Then, in http://www.thep.lu.se/∼leif/fys232/functions.h you will find the function declarations of three functions as follows:

```
#ifdef __cplusplus
extern "C" {
#endif

  /**
   * Integrate this function from 0 to 2*pi. It is a well-behaved,
   * smooth function in the whole interval x>0. For x -> 0, the
   * function goes to a finite constant value.
   */
  double function1(double x);

  /**
   * Integrate this function from 0 to 1. The function has an
   * integrable singularity in x=0 where it diverges as 1/sqrt(x), but
   * is otherwise well-behaved and smooth. In fact it can be written
   * as a 2nd degree polynomial divided by sqrt(x).
   */
  double function2(double x);

  /**
   * Integrate this function inside the unit circle x^2+y^2<1. The
   * function is everywhere smooth and well-behaved.
   */
  double function3(double x, double y);

#ifdef __cplusplus
}
#endif
```

The assignment is to evaluate the integrals of these functions as described in the comment lines. You do not have access to the actual code of these functions, but if you link your program with the object file available in
`http://www.thep.lu.se/~leif/fys232/functions.o`
you can call the functions from your main program. Try different methods of integration for each function.

Write a report where you describe and discuss the different methods you have chosen for each function and discuss the precision of your results.

Numerical methods for Physicists (NUM)

Cycle 2

Monte Carlo

Metropolis

Why is it called the "Metropolis" algorithm? I mean, I can understand the name "Monte Carlo" for using random sampling to numerically evaluate an integral — it really feels like a game of chance like in a casino. But Metropolis? Maybe it has something to do with randomly walking around in the streets of a large city?

In any case, these random methods all seem a bit ... well, random. How can you make sure that you don't accidentally miss a narrow peak in the function you are integrating? It seems to me that even the best possible random number generator you can never really be sure.

Literature

Most of the course content is covered in

> W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1988).

and for this cycle the relevant chapter is 7. The Metropolis algorithm is described in the lecture notes and in the exercise.

Goals

- Random numbers
- Central-limit theorem
- Transformation of random numbers
- Monte-Carlo integration
- The Metropolis algorithm
- Simulated annealing

Comments to exercises

The exercise deals with a simple model for protein folding.

Exercise 2

Go through the computer assignment at
http://www.thep.lu.se/∼carl/fys232_h1/ and write a report on the results.

To compile the `pivot.c` program you need to have the *GNU Scientific Library* package installed on your laptop. As a super user, simply do `rug install gsl` (takes a couple of minutes).

The program is written in `C`. After you have handed in your reports we will discuss how this program would look if it was written in `C++`

Numerical methods for Physicists (NUM)

Cycle 3

Optimization and minimization

Will it fit?

They had several thousand $pp \to \tau^+\tau^- + X$ events, and looking at the invariant mass spectrum $d\sigma/dm_{\tau\tau}$ they could see a small peak over the continuum at around $m_{\tau\tau} = 130$ GeV. So they fitted a Breit-Wigner plus a power-suppressed background and found that the resonance was consistent with $m_{\tau\tau} \approx 130$ GeV and $\Gamma_{\tau\tau} \approx 2$ GeV, but the errors were large, and the significance of the peak was only 1.7 sigma. But it could be the Higgs!

Literature

Most of the course content is covered in

> W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1988).

and for this cycle the relevant chapter 10.

Goals

- Minimization and Optimization
- The downhill simplex method
- Conjugate gradient descent
- C++ programming

Comments to exercises

Students with little or no experience of C++ may need extra tutoring in programming to complete the exercise.

Exercise 3

At http://www.thep.lu.se/∼leif/fys232/PBL/ you will find the file `tautau.dat` which contains simulated LHC data of the distribution in $\tau^+\tau^-$ invariant mass in the range 100 GeV$< m_{\tau\tau} <$ 200 GeV. If you plot the data with `gnuplot` you will find a small peak around $m_{\tau\tau} \approx 135$ GeV.

Write a C++ program to fit this data to a Breit-Wigner peak and a power-suppressed background using eg. Simplex minimization. Extract values for the mass and the width of the peak. Discuss the errors on the extracted values and the significance of the peak.

In http://www.thep.lu.se/∼leif/fys232/PBL/ you will also find the files `FitData.h`, `FitData.cc`, `FitFunction.h`, `FitFunction.cc` and `fitter.cc`, containing a skeleton for the fitting program which you may use. If you implement the calculation of the χ^2 in `FitData.cc` you can compile the program as follows:

```
g++ -o fitting -g -O fitting.cc FitFunction.cc FitData.cc
```

The resulting program is called `fitting` and running it will write out the χ^2 for a chosen set of parameters using just the power-suppressed background.

Numerical methods for Physicists (NUM)

Cycle 4

Ordinary differential equations

Ordinary Differential Equations

— So I think I got the hang of this now. All I need to do to numerically solve this differential equation is to discretize and see where I end up. Then, to understand what kind of errors I get, I just make a Taylor expansion. And looking at the expansion, maybe I can deduce some trick to improve the precision. Now what did that equation look like...?

Literature

Most of the course content is covered in

> W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1988).

and for this cycle the relevant chapter is 16.

Goals

- Implicit and explicit Euler methods
- Runge-Kutta
- The modified mid-point method
- Richardson extrapolation
- Stability of solutions
- Stiffness in differential equations

Comments to exercises

The exercise goes through the solutions of a simple differential equation using different methods.

Exercise 4

Go through the computer assignment at
http://cbbp.thep.lu.se/∼simon/teaching/RK.html and write a report on the results.

Numerical methods for Physicists (NUM)

Cycle 5

Partial differential equations

Partial Differential Equations

Just as for ordinary differential equations, you need to worry about the stability of your numerical solutions to partial differential equations. A typical example here is the von Neumann stability analysis for the diffusion equation.

But there are also other complications when discretizing partial differential equations. There may be qualitative properties of the continuous solutions which you may want to preserve when doing things numerically. The conservation of probability in the Schrödinger equation is one example.

Literature

Most of the course content is covered in

> W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1988).

and for this cycle the relevant chapter is 19.

Goals

- The diffusion equation
- Wave equations
- The Poisson equation
- von Neumann stability analysis
- The Crank-Nicholson method
- Relaxation methods

Comments to exercises

The exercise studies boundary-value problems for the Poisson equation by using the method of successive over-relaxation.

Exercise 5

Go through the computer assignment at
http://cbbp.thep.lu.se/∼simon/teaching/sor.html and write a report on the results.

Numerical methods for Physicists (NUM)

Written Exam

Examination, Numerical Methods for Physicists, April 26 2007

Allowed calculational aids: "TEFYMA" and pocket calculator. The examination consists of eight problems.

1. Assume you have a function, $f(x)$, which you know behaves as
$$\lim_{x\to 0} f(x) \propto \frac{1}{\sqrt{x}},$$
but is otherwise well behaved and smooth. Give arguments against using Simpsons rule to evaluate
$$\int_0^1 f(x)\,dx.$$
Also argue against using Monte Carlo integration with a uniform random number distribution.

2. For the same integral, describe how you would estimate it using importance sampling. Also describe how you would obtain the necessary random number distribution from a uniform distribution between 0 and 1.

3. For the same integral, derive a suitable two-point Gaussian Quadrature formula.

4. Give the formulae for doing a three-step Richardson extrapolation.

5. Suppose the explicit Euler method is applied to the initial value problem
$$\frac{d\mathbf{y}(x)}{dx} = \mathbf{A}\mathbf{y}(x) \qquad \mathbf{y}(0) = \mathbf{y}_0 \qquad \mathbf{A} = \begin{pmatrix} -2 & -1 \\ -1 & -2 \end{pmatrix}$$
where $\mathbf{y}(x)$ is a two-component vector. How small must the step size h be for the method to be stable?

6. The Levenberg-Marquardt method for minimizing $\chi^2(a_1,\ldots,a_M)$ is based on the update
$$a_k \to a_k' = a_k + \delta a_k, \qquad \text{with} \qquad \sum_{l=1}^{M} \tilde{\alpha}_{kl}\delta a_l = \beta_k$$
$$\beta_k = -\frac{1}{2}\frac{\partial \chi^2}{\partial a_k} \qquad\qquad \tilde{\alpha}_{kl} = \frac{1}{2}\frac{\partial^2 \chi^2}{\partial a_k \partial a_l}(1+\lambda\delta_{kl})$$

where $\lambda > 0$ is a parameter ($\delta_{kl} = 1$ if $k = l$ and $\delta_{kl} = 0$ otherwise). Give a brief motivation for this update based on the behavior for small and large λ, respectively.

7. One wants to sample a discrete probability distribution $P(t)$ by using the Metropolis method. Explain briefly how stationarity of $P(t)$ is achieved in this method.

8. Consider the heat equation $\partial u/\partial t = \partial^2 u/\partial x^2$ with boundary conditions $u(0,t) = u(1,t) = 0$. Discretizing in the x-direction with a step size $a = 1/J$ and using the approximation $\partial^2 u/\partial x^2 \approx (u_{j+1} - 2u_j + u_{j-1})/a^2$ leads to a matrix equation of the form

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}} = \mathbf{A}\mathbf{u}$$

for $\mathbf{u} = (u_1, \ldots, u_{J-1})$. Show that

$$\mathbf{v}^{(k)} = \Big(\sin\frac{k\pi}{J}, \sin\frac{2k\pi}{J}, \ldots, \sin\frac{(J-1)k\pi}{J}\Big) \qquad (k = 1, \ldots, J-1)$$

are eigenvectors of the $\mathbf{A}$ matrix and calculate the eigenvalues.

If we discretize also in t with step size h and use the explicit Euler method, what can be said about the stability of the solution.