

How to Integrate Parallel Computing in Science Education

A possible outcome of the Computers in Science Education project at UiO

Morten Hjorth-Jensen^{1,2}

¹Department of Physics and Center of Mathematics for Applications, University of Oslo, Norway

²Department of Physics and Astronomy, Michigan State University
East Lansing, Michigan, USA

October 24 2007

State of the Art Educational Approach in the Sciences

Education

- During the last 25 years there has been a considerable focus on different technologies at all levels in the educational ladder.
- Calculators, text editing, e-mail, new learning platforms, digital learning environments etc.
- Much focus on means and technologies, but what about the content, or more important insight about physical systems?
- The basic topics (math, chemistry, physics, ...) are taught more or less in the same fashion as before — unchanged over several decades (does this statement apply to your educational system as well?)

State of the Art Educational Approach in the Sciences

Education

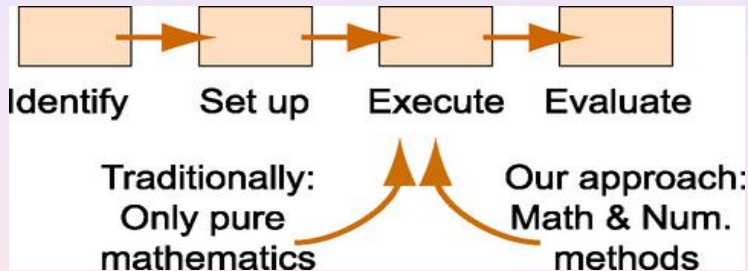
- During the last 25 years there has been a considerable focus on different technologies at all levels in the educational ladder.
- Calculators, text editing, e-mail, new learning platforms, digital learning environments etc.
- Much focus on means and technologies, but what about the content, or more important insight about physical systems?
- The basic topics (math, chemistry, physics, . . .) are taught more or less in the same fashion as before — unchanged over several decades (does this statement apply to your educational system as well?)

Some Observations

Computation in the Sciences

- To compute has always been a central ingredient in the sciences.
- The computer has brought a new dimension to the different fields and we can nowadays do more than 10^{14} floating point operations per second. Soon petascale computers will be available, with 10^{15} floating point operations per second.
- Problems which previously couldn't be solved are solved in a routine-like fashion within seconds or at most minutes.
- All universities have large farms of PCs in student laboratories, can easily be set up as parallel clusters.

What do we want to achieve?



The CSE project

Computations should enter Science Education

- Computation is a fundamental tool to gain new insights and should be included in our teaching.
- Requires development of an algorithmic thinking.
- Our teaching should include an education in basic numerical methods, normally taught in different departments, and often disconnected. In the US several smaller university colleges have implemented something similar to our project, see Computing in Science and Engineering, Vol. **8**, sept/oct issue 2006.
- The students should also learn to develop new methods and learn new tools when needed.
- Need an adequate computational platform.(Example: Python as computing language).

The new Educational Reform (Country dependent) and local Implementations

Univ Oslo's (UiO) information and communication technology (ICT) strategy

Two of five strategic goals with ICT.

- *ICT should be integrated as a pedagogical tool in our education*
- *UiO wishes to develop and increase its employees competence and motivation in the usage of ICT in educational matters.*

More Background for the Educational Reform

From the Faculty of Math and Sciences' strategic plan 2005-2009

Integrate central modern computational tools, instrumentation and techniques, in order to modernize our Math and Science education. Computations and numerical modelling has a central place here.

CSE project

Goals

- Include and integrate a computational perspective in our basic education.
- Give the students realistic examples from our research, this brings our research into the undergraduate teaching at a much earlier stage.
- Postulate : *An algorithmic thinking can enhance physics understanding and insight.*
- We also want to give an education that can build the necessary foundation for a professional work experience that is continuously changing– focus on fundamental (long-lasting) knowledge.

CSE project

Goals

- Include and integrate a computational perspective in our basic education.
- Give the students realistic examples from our research, this brings our research into the undergraduate teaching at a much earlier stage.
- Postulate : *An algorithmic thinking can enhance physics understanding and insight.*
- We also want to give an education that can build the necessary foundation for a professional work experience that is continuously changing– focus on fundamental (long-lasting) knowledge.

Our CSE project

What we do

- Coordinated use of computational exercises and numerical tools in many undergraduate courses.
- Update the scientific staff's competence on computational aspects and give support (scientific, pedagogical and financial) to those who wish to revise their courses in a computational direction.
- Development of courses and exercise modules on computational aspects, both for students and teachers.
- Basic idea: mixture of mathematics, computation, informatics and topics from the physical sciences.
- Centrally rooted in our strategic plans and some of our centers of excellence and several bachelor programs.

Our CSE project

What we do

- Coordinated use of computational exercises and numerical tools in many undergraduate courses.
- Update the scientific staff's competence on computational aspects and give support (scientific, pedagogical and financial) to those who wish to revise their courses in a computational direction.
- Development of courses and exercise modules on computational aspects, both for students and teachers.
- Basic idea: mixture of mathematics, computation, informatics and topics from the physical sciences.
- Centrally rooted in our strategic plans and some of our centers of excellence and several bachelor programs.

Our CSE project

What we do

- Coordinated use of computational exercises and numerical tools in many undergraduate courses.
- Update the scientific staff's competence on computational aspects and give support (scientific, pedagogical and financial) to those who wish to revise their courses in a computational direction.
- Development of courses and exercise modules on computational aspects, both for students and teachers.
- Basic idea: mixture of mathematics, computation, informatics and topics from the physical sciences.
- Centrally rooted in our strategic plans and some of our centers of excellence and several bachelor programs.

Our CSE project

What we do

- Coordinated use of computational exercises and numerical tools in many undergraduate courses.
- Update the scientific staff's competence on computational aspects and give support (scientific, pedagogical and financial) to those who wish to revise their courses in a computational direction.
- Development of courses and exercise modules on computational aspects, both for students and teachers.
- Basic idea: mixture of mathematics, computation, informatics and topics from the physical sciences.
- Centrally rooted in our strategic plans and some of our centers of excellence and several bachelor programs.

Our CSE project

What we do

- Coordinated use of computational exercises and numerical tools in many undergraduate courses.
- Update the scientific staff's competence on computational aspects and give support (scientific, pedagogical and financial) to those who wish to revise their courses in a computational direction.
- Development of courses and exercise modules on computational aspects, both for students and teachers.
- Basic idea: mixture of mathematics, computation, informatics and topics from the physical sciences.
- Centrally rooted in our strategic plans and some of our centers of excellence and several bachelor programs.

Organizational framework

Educational Reform

- The new reform from 2003 (Bachelor+Master+Phd) paved the way for coordinated introduction of computational topics in several bachelor programs.
- We have six large bachelor programs in the sciences which have coordinated bachelor programs with several common mathematics, physics and informatics courses in the first 3-4 semesters.
- Modern software has a low learning threshold and it is easy to visualize and program complicated systems.

Centers of excellence

Four centers of excellence where computations are central. Play an important role in catalyzing cross-disciplinary research and educational projects.

Example of Bachelor Program

Physics, Astronomy and Meteorology (Norwegian text, sorry)

6. semester	Se ønsket studieretning	EXPHIL03 - Examen philosophicum /Valgfritt	Valgfritt
5. semester	Se ønsket studieretning	Se ønsket studieretning	Valgfritt/ EXPHIL03 - Examen philosophicum
4. semester	Se ønsket studieretning	Se ønsket studieretning	Se ønsket studieretning
3. semester	FYS1120 - Elektromagnetisme	AST1100 - Innføring i astrofysikk / GEF1000 - Klimasystemet	MAT1120 - Lineær algebra
2. semester	FYS-MEK1110 - Mekanikk	MEK1100 - Feltteori og vektoranalyse	MAT1110 - Kalkulus og lineær algebra
1. semester	INF1100 - Grunnkurs i programmering for naturvitenskapelige anvendelser	MAT-INF1100 - Modellering og beregninger	MAT1100 - Kalkulus
	10 studiepoeng	10 studiepoeng	10 studiepoeng

- First semester common for most Bachelor programs.
- The mathematics courses MAT1100, MAT1110, MAT1120 og MEK1100 are also common to many Bachelor programs.

How to bake in Computations from day one

Example: The first semester for most Bachelor programs

Three courses the first semester: MAT1100, MAT-INF1100 og INF1100.

- Definition of the derivative in MAT1100 (Calculus and analysis)

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

- Algorithms to compute the derivative in MAT-INF1100 (Mathematical modelling with computing)

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

- The Algorithm is then implemented and used in particular applications in the programming course INF1100, where Python is the default programming language.

How to bake in Computations from day one

Example: The first semester for most Bachelor programs

Three courses the first semester: MAT1100, MAT-INF1100 og INF1100.

- Definition of the derivative in MAT1100 (Calculus and analysis)

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

- Algorithms to compute the derivative in MAT-INF1100 (Mathematical modelling with computing)

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

- The Algorithm is then implemented and used in particular applications in the programming course INF1100, where Python is the default programming language.

How to bake in Computations from day one

Example: The first semester for most Bachelor programs

Three courses the first semester: MAT1100, MAT-INF1100 og INF1100.

- Definition of the derivative in MAT1100 (Calculus and analysis)

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

- Algorithms to compute the derivative in MAT-INF1100 (Mathematical modelling with computing)

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

- The Algorithm is then implemented and used in particular applications in the programming course INF1100, where Python is the default programming language.

How to bake in Computations from day one

Example: The first semester for most Bachelor programs

Three courses the first semester: MAT1100, MAT-INF1100 og INF1100.

- Definition of the derivative in MAT1100 (Calculus and analysis)

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

- Algorithms to compute the derivative in MAT-INF1100 (Mathematical modelling with computing)

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

- The Algorithm is then implemented and used in particular applications in the programming course INF1100, where Python is the default programming language.

Other Examples

Integration by Trapezoidal Rule

- Definition of integration in MAT1100 (Calculus and analysis).
- The algorithm for computing the integral vha the Trapezoidal rule for an interval $x \in [a, b]$

$$\int_a^b (f(x))dx \approx \frac{1}{2} [f(a) + 2f(a+h) + \dots + 2f(b-h) + f(b)]$$

is taught in MAT-INF1100 (Mathematical modelling)

- The algorithm is then implemented in INF1100 (programming course):

Other Examples

Integration by Trapezoidal Rule

- Definition of integration in MAT1100 (Calculus and analysis).
- The algorithm for computing the integral vha the Trapezoidal rule for an interval $x \in [a, b]$

$$\int_a^b (f(x))dx \approx \frac{1}{2} [f(a) + 2f(a+h) + \dots + 2f(b-h) + f(b)]$$

is taught in MAT-INF1100 (Mathematical modelling)

- The algorithm is then implemented in INF1100 (programming course):

Other Examples

Integration by Trapezoidal Rule

- Definition of integration in MAT1100 (Calculus and analysis).
- The algorithm for computing the integral vha the Trapezoidal rule for an interval $x \in [a, b]$

$$\int_a^b (f(x))dx \approx \frac{1}{2} [f(a) + 2f(a+h) + \dots + 2f(b-h) + f(b)]$$

is taught in MAT-INF1100 (Mathematical modelling)

- The algorithm is then implemented in INF1100 (programming course):

Typical implementation in INF1100

Integration by Trapezoidal Rule, easy to parallelize!

```
def Trapez(a,b,f,n)
    h = (b-a)/float(n)
    s = 0
    x = a
    for i in range(1,n,1):
        x = x+h
        s = s+ f(x)
    s = 0.5*(f(a)+f(b)) +s
    return h*s

def f1(x):
    return exp(-x*x)*log(1+x*sin(x))

a = 1; b = 3; n = 1000
result = Trapez(a,b,f1,n)
print result
```

Coordination

More Examples

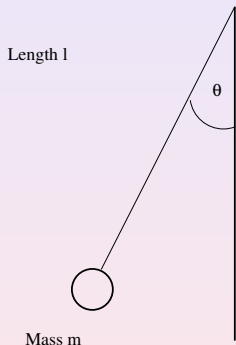
- The courses MAT1100, MAT-INF1100 og INF1100 have many common examples and topics, amongst these ordinary differential equations.
- Differential equations are in turn used widely in our mechanics course, which comes in the second semester, with examples spanning from the classical pendulum to rocket launching.
- Differential equations (partial and ordinary) are in turn used in many many other courses, from electromagnetism to quantum physics.
- The central mathematics courses MEK1100, MAT1110 og MAT1120 develop further numerical exercises and problems, from linear algebra to multi-dimensional integration.
- Teachers in other courses need therefore not use much time on numerical tools— it is naturally included in other courses.

Coordination

More Examples

- The courses MAT1100, MAT-INF1100 og INF1100 have many common examples and topics, amongst these ordinary differential equations.
- Differential equations are in turn used widely in our mechanics course, which comes in the second semester, with examples spanning from the classical pendulum to rocket launching.
- Differential equations (partial and ordinary) are in turn used in many many other courses, from electromagnetism to quantum physics.
- The central mathematics courses MEK1100, MAT1110 og MAT1120 develop further numerical exercises and problems, from linear algebra to multi-dimensional integration.
- Teachers in other courses need therefore not use much time on numerical tools— it is naturally included in other courses.

FYS-MEK1100 (Mechanics), Second Semester



Realistic Pendulum

Classical pendulum with damping and external force

$$ml \frac{d^2\theta}{dt^2} + \nu \frac{d\theta}{dt} + mgsin(\theta) = Asin(\omega t).$$

Easy to solve without approximations and then visualize the solution. Same equation for an RLC circuit

$$L \frac{d^2Q}{dt^2} + \frac{Q}{C} + R \frac{dQ}{dt} = V(t).$$

What can we do with the Pendulum?

Lots of interesting problems

- Can study chaos, theoretically, numerically and experimentally, can choose 'best' parameters for experimental setup.
- Can test different algorithms for solving ordinary differential equations, from Euler's to fourth-order Runge Kutta methods. Tight connection with algorithm and physics.
- Can make classes of differential equation solvers.
- Can make a general program which can be applied to other scientific cases in later courses, such as electromagnetism (RLC circuits). Students realize that much of the same mathematics enters many physics cases.
- **Can parallelize the ODE solver....., difficult**

More Examples from Physics Courses, 2-5 semester

Second-fourth semester

- 1 Rocket launching with realistic parameters
- 2 How to kick a football and model its trajectory.
- 3 Planet motion and position of planets
- 4 Magnetic fields with various geometries based on Biot-Savart's law
- 5 Harmonic oscillations and various forms of electromagnetic waves.
- 6 Combined effect of different potentials such as the electrostatic potential and the gravitational potential.
- 7 Simple studies of atoms and molecules, and much more

No need to teach computational science methods for solving these systems.

Can we bake in Parallelization at an early stage?

Late: Fifth semester, FYS3150 Computational Physics

The integral we need to solve is the quantum mechanical expectation value of the correlation energy between two electrons, namely

$$\left\langle \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \right\rangle = \int_{-\infty}^{\infty} d\mathbf{r}_1 d\mathbf{r}_2 e^{-2\alpha(r_1+r_2)} \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}.$$

Students use either C++ or Fortran95 as programming languages on a teaching cluster of some 30 standard supermarket PCs. MPI (MPI-1 mainly) as library and MPICH2 as setup (homebuilt).

It took the students on average 30 mins at the lab to become operative with their first MPI program. [http:](http://www.uio.no/studier/emner/matnat/fys/FYS3150/h07/)

[//www.uio.no/studier/emner/matnat/fys/FYS3150/h07/](http://www.uio.no/studier/emner/matnat/fys/FYS3150/h07/), see under projects, project 2 and 4.

Can we bake in Parallelization at an early stage?

Late: Fifth semester, FYS3150 Computational Physics

The tasks they are to solve

- a) Use Gauss-Legendre quadrature and compute the integral by integrating for each variable $x_1, y_1, z_1, x_2, y_2, z_2$ from $-\infty$ to ∞ .
- b) Compute the same integral but now with brute force Monte Carlo and compare your results with those from the previous point. Discuss the differences. With brute force we mean that you should use the uniform distribution.
- c) Improve your brute force Monte Carlo calculation by using importance sampling. Hint: use the exponential distribution. Does the variance decrease? Comment your results.
- d) **Parallelize your code from the previous point and compare the CPU time needed with that from point [c)]. Do you achieve a good speedup?**
- e) The integral has an analytical expression. Can you find it?

Parallel Code is simple, only few MPI function calls

```
.....
MPI_Init(NULL,NULL);
MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
//number of MC cycles
int N = 10000000000;
//number of cycles per process
int local_N = N / numprocs;
.....
//Sum up
MPI_Reduce(&local_sum, &sum, 1, MPI_DOUBLE, MPI_SUM, 0
MPI_Reduce(&local_variance, &variancesum, 1, MPI_DOUBL
if (minrank == 0)
{
    cout << jakobidet*average << " +/- " << jakobidet*var
}
MPI_Finalize();
```

Can we bake in Parallelization at an early stage?

Late: Fifth semester, FYS3150 Computational Physics

- Integration and Monte carlo methods are embarrassingly trivial (ET), nobody (out of 35) had any conceptual problems when parallelizing. I used only two lectures a 45 min each on MPI and parallelization of integrals.
- The students found it easy to implement MPI as long as support was given.
- To use MPI or similar libraries, it is important that the infrastructure has a as low as possible threshold. On average they used 30 mins to have MPI running.
- Now they are solving quantum mechanical problems by Variational Monte Carlo (ET).
- The last project is difficult, parallelize Jacobi's method. I'm not sure if I dare to do that this year!

Challenge: what about the first semester and INF1100?

Integration by Trapezoidal Rule, easy to parallelize!

- Can for example use the *pp* package, **import pp**
- **pp** has many simple functions which are easily called
- It is a matter of explaining how to use these functions. On section in lecture notes, perhaps two lectures.
- Need stable clusters for undergraduate teaching.
- Can we do this in INF1100? That would really enhance the visibility of high-performance computing from day one in our educational ladder!
- ODEs and PDEs more difficult, but can be taught later. Same with parallel linear algebra algos and eigenvalue solvers.