**محتويات المقرر COURSE SYLLABUS**

**Master course**

|  |  |
| --- | --- |
| **1. رقم المقرر :** COURSE NO**.** | 203601-3 physics |
| **2. عنوان المقرر :** COURSE TITLE | **فيزياء حسابية Computational Physic** |
| **3. عدد الساعات :** NUMBER OF HOURS | 3 hours (2 hr. theoretical + 1 hr. application with Fortran) |
| **4. الفرقة والفصل الدراسى:**YEAR & SEMISTER  | High graduate students (Master) |
| **5. وصف المقرر :** COURSE DESCRIPTIONThe course introduces the students to principles of numerical techniques and its applications in physics problems. |

**6 . أهداف المقرر :COURSE GOALS**

**\*** Developing the student's problem solving, and skills needed to find numerical solutions and graphs for physics problems.

**7. المتطلب السابق أو المتوازية : PREREQUESITES AND COREQUISITES: الفيزياء الرياضية المتقدمة Advanced Math. Method in Physics**

Course topics: (Syllabus)

1. Computational errors- Floating point- Root finding- Bisection method- Newton's method.
2. Interpolation and Extrapolation.
3. Polynomial approximation. Least square method, and Curve Fitting.
4. Numerical differentiation and integration methods.
5. Numerical solution of ordinary and partial differential equations.
6. Numerical solution for a system of linear equations.

**9. المتطلبات اللازمة للتطبيق العملى للمقرر: معمل حاسب آلى، لغات برمجة، حزم برامج جاهزة بلغة البرمجة المقررة.**

Requirements: Computer Lab., Different Programming Languages, and NAG libraries.

**Numerical analysis**

**Numerical analysis** is the study of [algorithms](http://www.physicsdaily.com/physics/Algorithms) for the problems of *continuous mathematics* (as distinguished from [discrete mathematics](http://www.physicsdaily.com/physics/Discrete_mathematics)). This means it deals mainly with [real variable](http://www.physicsdaily.com/physics/Real_variable) or [complex variable](http://www.physicsdaily.com/physics/Complex_variable) questions, numerical linear algebra over the real or complex fields, the solution of [differential equations](http://www.physicsdaily.com/physics/Differential_equation), and other related problems arising in the [physical sciences](http://www.physicsdaily.com/physics/Physical_sciences) and [engineering](http://www.physicsdaily.com/physics/Engineering).

**General introduction**

Some problems in continuous mathematics can be solved exactly by an algorithm. These algorithms are called *direct methods*. Examples are [Gaussian elimination](http://www.physicsdaily.com/physics/Gaussian_elimination) for solving [systems of linear equations](http://www.physicsdaily.com/physics/System_of_linear_equations) and the [simplex method](http://www.physicsdaily.com/physics/Simplex_method) in [linear programming](http://www.physicsdaily.com/physics/Linear_programming).

However, no direct methods exist for most problems. We might then try to replace the continuous problem by a discrete problem; this process is called *discretization*. Another possibility is to use an [iterative method](http://www.physicsdaily.com/physics/Iterative_method). Such a method starts from a guess and finds successive approximations that hopefully [converge](http://www.physicsdaily.com/physics/Convergence) to the solution. Even when a direct method does exist, an iterative method may be preferable because it is more efficient.

**The generation and propagation of errors**

The study of errors forms an important part of numerical analysis. There are several ways in which error can be introduced in the solution of the problem. [Round-off errors](http://www.physicsdaily.com/physics/Round-off_error) arise because it is impossible to represent all [real numbers](http://www.physicsdaily.com/physics/Real_number) exactly on a [finite-state machine](http://www.physicsdaily.com/physics/Finite-state_machine) (which is what all practical [digital computers](http://www.physicsdaily.com/physics/Digital_computer) are). Truncation errors are committed when an iterative method is [terminated](http://www.physicsdaily.com/physics/Termination) and the approximate solution differs from the exact solution. Similarly, discretization induces a [discretization error](http://www.physicsdaily.com/physics/Discretization_error) because the solution of the discrete problem does not coincide with the solution of the continuous problem.

Once an error is generated, it propagates through the calculation. This leads to the notion of [numerical stability](http://www.physicsdaily.com/physics/Numerical_stability): an algorithm is numerically stable if an error, once it is generated, does not grow too much during the calculation. This is only possible if the problem is [well-conditioned](http://www.physicsdaily.com/physics/Condition_number), meaning that the solution changes by only a small amount if the problem data are changed by a small amount. Indeed, if a problem is ill-conditioned, then any error in the data will grow a lot.

However, an algorithm that solves a well-conditioned problem may or may not be numerically stable. An art of numerical analysis is to find a stable algorithm for solving a well-posed mathematical problem.

**Applications**

The algorithms of numerical analysis are routinely applied to solve many problems in science and engineering. Examples are the design of structures like bridges and airplanes (see [computational physics](http://www.physicsdaily.com/physics/Computational_physics) and [computational fluid dynamics](http://www.physicsdaily.com/physics/Computational_fluid_dynamics)), [weather forecasting](http://www.physicsdaily.com/physics/Weather_forecasting), [climate models](http://www.physicsdaily.com/physics/Climate_model), the analysis and design of molecules ([computational chemistry](http://www.physicsdaily.com/physics/Computational_chemistry)), and finding oil reservoirs. In fact, almost all [supercomputers](http://www.physicsdaily.com/physics/Supercomputer) are continually running numerical analysis algorithms.

As a consequence, efficiency plays an important role and a heuristic method may be preferred above a method with a solid theoretic foundation because it is more efficient. Generally, numerical analysis uses [empirical](http://www.physicsdaily.com/physics/Empirical) results of computation runs to probe new methods and analyze problems, though it of course also employs mathematical [axioms](http://www.physicsdaily.com/physics/Axiom), [theorems](http://www.physicsdaily.com/physics/Theorem) and [proofs](http://www.physicsdaily.com/physics/Mathematical_proof).

**Software**

Nowadays, most algorithms are implemented and run on a computer. The [Netlib](http://www.physicsdaily.com/physics/Netlib) repository contains various collections of software routines for numerical problems, mostly in [Fortran](http://www.physicsdaily.com/physics/Fortran) and [C](http://www.physicsdaily.com/physics/C_programming_language). Commercial products implementing many different numerical algorithms include the IMSL and NAG libraries; a free alternative is the [GNU Scientific Library](http://www.physicsdaily.com/physics/GSL). A different approach is taken by the [Numerical Recipes](http://www.physicsdaily.com/physics/Numerical_Recipes) library, where emphasis is placed on clear understanding of algorithms.

There are a number of computer programs used for performing numerical calculations:

* [MATLAB](http://www.physicsdaily.com/physics/MATLAB) is a widely-used program for performing numerical calculations. It comes with its own programming language, in which numerical algorithms can be implemented.
* [GNU Octave](http://www.physicsdaily.com/physics/GNU_Octave) is a near-clone of MATLAB ([free software](http://www.physicsdaily.com/physics/Free_software), [GNU](http://www.physicsdaily.com/physics/GNU) [GPL license](http://www.physicsdaily.com/physics/GPL)).
* [R programming language](http://www.physicsdaily.com/physics/R_programming_language) is a widely used system with a focus on data manipulation and statistics. Several hundred freely downloadable specialized packages are available ([free software](http://www.physicsdaily.com/physics/Free_software), [GNU](http://www.physicsdaily.com/physics/GNU) [GPL license](http://www.physicsdaily.com/physics/GPL)).
* [Scilab](http://www.physicsdaily.com/physics/Scilab), an [open-source](http://www.physicsdaily.com/physics/Open_source) (Scilab license) program for performing numerical calculations.
* [IDL programming language](http://www.physicsdaily.com/physics/IDL_programming_language).
* FreeMat , an [open-source](http://www.physicsdaily.com/physics/Open_source) MATLAB-like environment with a MIT-type license .
* [Python programming language](http://www.physicsdaily.com/physics/Python_programming_language):
	+ [SciPy](http://www.physicsdaily.com/physics/SciPy), a library of scientific tools;
	+ matplotlib , a MATLAB-like plotting library.

Many [computer algebra systems](http://www.physicsdaily.com/physics/Computer_algebra_system) such as [Mathematica](http://www.physicsdaily.com/physics/Mathematica) or the [Maple computer algebra system](http://www.physicsdaily.com/physics/Maple_computer_algebra_system) ([free software](http://www.physicsdaily.com/physics/Free_software) systems include [Maxima](http://www.physicsdaily.com/physics/Maxima), [Axiom](http://www.physicsdaily.com/physics/Axiom_computer_algebra_system), [calc](http://www.physicsdaily.com/physics/Calc) and [Yacas](http://www.physicsdaily.com/physics/Yacas)) can also be used for numerical computations.

**Areas of study**

The field of numerical analysis is divided in different disciplines according to the problem that is to be solved.

**Computing values of functions**

One of the simplest problems is the evaluation of a function at a given point. But even evaluating a polynomial is not straightforward: the [Horner scheme](http://www.physicsdaily.com/physics/Horner_scheme) is often more efficient than the obvious method. Generally, it is important to estimate and control [round-off errors](http://www.physicsdaily.com/physics/Round-off_error) arising from the use of [floating point](http://www.physicsdaily.com/physics/Floating_point) arithmetic.

**Interpolation, extrapolation and regression**

[Interpolation](http://www.physicsdaily.com/physics/Interpolation) solves the following problem: given the value of some unknown function at a number of points, what value does that function have at some other point between the given points? A very simple method is to use [linear interpolation](http://www.physicsdaily.com/physics/Linear_interpolation), which assumes that the unknown function is linear between every pair of successive points. This can be generalized to [polynomial interpolation](http://www.physicsdaily.com/physics/Polynomial_interpolation), which is sometimes more accurate but suffers from [Runge's phenomenon](http://www.physicsdaily.com/physics/Runge%27s_phenomenon). Other interpolation methods use localized functions like [splines](http://www.physicsdaily.com/physics/Spline_%28mathematics%29) or [wavelets](http://www.physicsdaily.com/physics/Wavelet).

[Extrapolation](http://www.physicsdaily.com/physics/Extrapolation) is very similar to interpolation, except that now we want to find the value of the unknown function at a point which is outside the given points.

[Regression](http://www.physicsdaily.com/physics/Regression_analysis) is also similar, but it takes into account that the data is imprecise. Given some points, and a measurement of the value of some function at these points (with an error), we want to determine the unknown function. The [least squares](http://www.physicsdaily.com/physics/Least_squares)-method is one popular way to achieve this.

**Solving equations**

Another fundamental problem is computing the solution of some given equation. Two cases are commonly distinguished, depending on whether the equation is linear or not.

Much effort has been put in the development of methods for solving systems of linear equations. Standard methods are [Gauss-Jordan elimination](http://www.physicsdaily.com/physics/Gauss-Jordan_elimination) and [LU-factorization](http://www.physicsdaily.com/physics/LU-factorization). [Iterative methods](http://www.physicsdaily.com/physics/Iterative_method) such as the [conjugate gradient method](http://www.physicsdaily.com/physics/Conjugate_gradient_method) are usually preferred for large systems.

[Root-finding algorithms](http://www.physicsdaily.com/physics/Root-finding_algorithm) are used to solve nonlinear equations (they are so named since a root of a function is an argument for which the function yields zero). If the function is [differentiable](http://www.physicsdaily.com/physics/Derivative) and the derivative is known, then [Newton's method](http://www.physicsdaily.com/physics/Newton%27s_method) is a popular choice. [Linearization](http://www.physicsdaily.com/physics/Linearization) is another technique for solving nonlinear equations.

**Optimization**

*Main article:* [*Optimization (mathematics)*](http://www.physicsdaily.com/physics/Optimization_%28mathematics%29)*.*

Optimization problems ask for the point at which a given function is maximized (or minimized). Often, the point also has to satisfy some [constraints](http://www.physicsdaily.com/physics/Constraint).

The field of optimization is further split in several subfields, depending on the form of the objective function and the constraint. For instance, [linear programming](http://www.physicsdaily.com/physics/Linear_programming) deals with the case that both the objective function and the constraints are linear. A famous method in linear programming is the [simplex method](http://www.physicsdaily.com/physics/Simplex_method).

The method of [Lagrange multipliers](http://www.physicsdaily.com/physics/Lagrange_multipliers) can be used to reduce optimization problems with constraints to an unconstrained optimization problems.

**Evaluating integrals**

*Main article:* [*Numerical integration*](http://www.physicsdaily.com/physics/Numerical_integration)*.*

Numerical integration, also known as numerical [quadrature](http://www.physicsdaily.com/physics/Quadrature_%28mathematics%29), asks for the value of a definite [integral](http://www.physicsdaily.com/physics/Integral). Popular methods use some [Newton-Cotes formula](http://www.physicsdaily.com/physics/Newton-Cotes_formula), for instance the midpoint rule or the trapezoid rule, or [Gaussian quadrature](http://www.physicsdaily.com/physics/Gaussian_quadrature). However, if the dimension of the integration domain becomes large, these methods become prohibitively expensive. In this situation, one may use a [Monte Carlo method](http://www.physicsdaily.com/physics/Monte_Carlo_method), a [quasi-Monte Carlo method](http://www.physicsdaily.com/physics/Quasi-Monte_Carlo_method), or, in modestly large dimensions, the method of [sparse grids](http://www.physicsdaily.com/physics/Sparse_grid).

**Differential equations**

*Main articles:* [*Numerical ordinary differential equations*](http://www.physicsdaily.com/physics/Numerical_ordinary_differential_equations)*,* [*Numerical partial differential equations*](http://www.physicsdaily.com/physics/Numerical_partial_differential_equations)*.*

Numerical analysis is also concerned with computing (in an approximate way) the solution of [differential equations](http://www.physicsdaily.com/physics/Differential_equation), both ordinary differential equations and [partial differential equations](http://www.physicsdaily.com/physics/Partial_differential_equation).

Partial differential equations are solved by first discretizing the equation, bringing it into a finite-dimensional subspace. This can be done by a [finite element method](http://www.physicsdaily.com/physics/Finite_element_method), a [finite difference](http://www.physicsdaily.com/physics/Finite_difference) method, or (particularly in engineering) a [finite volume method](http://www.physicsdaily.com/physics/Finite_volume_method). The theoretical justification of these methods often involves theorems from [functional analysis](http://www.physicsdaily.com/physics/Functional_analysis). This reduces the problem to the solution of an algebraic equation.

**History**

The field of numerical analysis predates the invention of modern computers by many centuries. In fact, many great mathematicians of the past were preoccupied by numerical analysis, as is obvious from the names of important algorithms like [Newton's method](http://www.physicsdaily.com/physics/Newton%27s_method), [Lagrange interpolation polynomial](http://www.physicsdaily.com/physics/Lagrange_polynomial), [Gaussian elimination](http://www.physicsdaily.com/physics/Gaussian_elimination), or [Euler's method](http://www.physicsdaily.com/physics/Euler%27s_method).

To facilitate computations by hand, large books were produced with formulas and tables of data such as interpolation points and function coefficients. Using these tables, often calculated out to 16 decimal places or more for some functions, one could look up values to plug into the formulas given and achieve very good numerical estimates of some functions. The canonical work in the field is the [NIST](http://www.physicsdaily.com/physics/NIST) publication edited by [Abramowitz and Stegun](http://www.physicsdaily.com/physics/Abramowitz_and_Stegun), a 1000-plus page book of a very large number of commonly used formulas and functions and their values at many points. The function values are no longer very useful when a computer is available, but the large listing of formulas can still be very handy.

The [mechanical calculator](http://www.physicsdaily.com/physics/Mechanical_calculator) was also developed as a tool for hand computation. These calculators evolved into electronic computers in the [1940s](http://www.physicsdaily.com/physics/1940s), and it was then found that these computers were also useful for administrative purposes. But the invention of the computer also influenced the field of numerical analysis, since now longer and more complicated calculations could be done.