



Universidad de Sonora  
División de Ciencia Exactas y Naturales  
Departamento de Física  
Licenciatura en Física

## Física computacional I

Eje formativo:	Profesional		
Requisitos:	Ecuaciones diferenciales I		
	Programación y lenguaje Fortran		
Carácter:	Obligatorio		
Horas:	Teoría	Taller	Laboratorio
	2	4	0
Créditos:	08		
Servicio del:	Departamento de		
	Física		

### 1. Introducción

En el estudio de los diferentes campos de la Física, al establecer modelos matemáticos para el estudio de los fenómenos, es común que no se pueda dar una solución exacta al problema matemático inherente al modelo, por lo que se hace necesario recurrir a soluciones obtenidas mediante el empleo de métodos numéricos a través del uso de computadoras. La Física Computacional es un enfoque multidisciplinario que combina a la Física, a las Ciencias de la Computación y a las Matemáticas Aplicadas, para resolver problemas aplicados de interés en el mundo real. El área de aplicación es muy amplia, va desde la modelación ambiental, manejo de residuos tóxicos, diseño de materiales, transporte de aguas subterráneas, estructura de la materia y las partículas elementales, análisis de imágenes, etc. El campo de la física computacional, requiere conocer no sólo el funcionamiento de las computadoras y sistemas operativos, los temas específicos de física y matemáticas, sino también

conocer los algoritmos computacionales involucrados y cómo estos se pueden combinar para realizar un trabajo de forma integral.

Las herramientas modernas de cómputo simbólico, de cómputo numérico y de visualización científica, se han convertido en aliados insustituibles, en el estudio de problemas lineales y no lineales interesantes en las ciencias aplicadas. Estas herramientas proporcionan lenguajes de alto nivel para facilitar la modelación y extender el alcance de la mente humana para el estudio y comprensión de los fenómenos. En la actualidad, existen una gran diversidad de herramientas computacionales propietarias y de software libre, que nos permiten trabajar en la modelación matemática de fenómenos físicos y de otras ciencias. Así mismo, el desarrollo de nuevas arquitecturas computacionales para el cómputo de alto rendimiento (programación vectorial y en paralelo), facilitan la simulación de problemas mucho más complejos.

La evolución de sistemas de álgebra computacional, como *Axiom*, *Derive*, *Macysma*, *Maple*, *Mathematica*, *Maxima*, *MuPad*, *Yacas* y otros, permiten el estudio de problemas prácticos donde las tareas de manipulación algebraica de sistemas complejos de ecuaciones son prácticamente imposibles de trabajarlos sin la ayuda de estas herramientas. Por otro lado existen sistemas que incluyen herramientas para cómputo numérico, como *Matlab*, *Mathematica*, *Octave*, *RLaBplus*, y otros, que de igual manera se han convertido en herramientas de uso general para auxiliar en las simulaciones numéricas de uso cotidiano. En cuanto a lenguajes de programación utilizados en análisis numérico, además de los lenguajes tradicionales como *Fortran90*, *Ansii C*, *C++* o *Pascal*, se cuenta con herramientas complementarias desarrolladas en lenguajes orientados a objetos como *Java*, *Python* y otros. Desde la generalización de la Web en Internet y la consolidación de los movimientos de software libre, la colaboración de la comunidad de programadores en Internet han llevado a cristalizar repositorios ricos en recursos de herramientas de software como *SourceForge.Net*, al igual que *Netlib.Org* se convirtió en el repositorio de herramientas de software matemático para el análisis numérico.

El conjunto de estas herramientas enriquecen y extienden la aplicabilidad de los métodos de la física-matemática tradicionales, y es en este contexto que surge la necesidad de desarrollar en el estudiante de la Licenciatura en Física las habilidades computacionales que constituyen el núcleo central de la asignatura *Física computacional I*, que ha sido dividido en dos partes. En la primera parte se estudian los sistemas de ecuaciones lineales y las herramientas empleadas para manipular matrices, así como la solución de problemas de valores y vectores propios, mientras que en la segunda parte se introduce al estudiante en el estudio y conocimiento de los métodos empleados en la resolución de ecuaciones diferenciales ordinarias, así como en las herramientas para realizar simulación computacional.

## 2. Objetivo general

Este curso tiene como objetivo desarrollar las capacidades del estudiante para la incorporación de herramientas de sistemas de cómputo simbólico, numérico y de visualización científica en la solución de problemas lineales y no lineales que se manejan en el campo de la Física-Matemática; al finalizar el curso el estudiante será capaz de analizar y aplicar métodos computacionales para resolver problemas físicos que impliquen la solución de problemas algebraicos o matriciales, así como aquellos en que se involucren ecuaciones diferenciales ordinarias.

## 3. Objetivos específicos

Durante el desarrollo del curso el estudiante desarrollará experiencias de aprendizaje mediante la realización de algunas de las siguientes actividades: exposición de alguno de los temas, realización de un reporte final o prácticas en el laboratorio de cómputo, que lo vinculen con las actividades típicas del desarrollo de la profesión del Físico, ya sea en el campo de la docencia, la investigación y/o la extensión, explotando las ventajas que tiene el apoyarse en sistemas de cómputo y lenguajes de alto nivel para el estudio de un problema en particular.

Al terminar el curso, el estudiante será capaz de:

- Resolver sistemas de ecuaciones lineales haciendo uso de matrices, así como resolver problemas de valores y vectores propios, mediante el empleo de técnicas de cómputo (simbólico, numérico y/o de visualización).
- Hacer uso de métodos computacionales para la solución de ecuaciones diferenciales ordinarias, tanto para problemas de valores iniciales como de valores a la frontera.
- Resolver ecuaciones diferenciales de orden superior y sistemas de ecuaciones diferenciales mediante el uso de herramientas computacionales, tanto para problemas de valores iniciales como valores a la frontera.

## 3. Temario

- 1) Introducción.
- 2) Sistemas de ecuaciones lineales y estrategias de pivoteo. Matrices. Inversión de matrices. Determinantes. Factorización de matrices.
- 3) Valores y vectores propios de matrices.
- 4) Problemas de valor inicial para ecuaciones diferenciales. Teoría elemental.
- 5) Métodos de Euler y de Taylor de orden superior.
- 6) Métodos de Runge-Kutta. Métodos multipasos.
- 7) Ecuaciones de orden superior y sistemas de ecuaciones diferenciales.
- 8) Método de disparo lineal. Método de disparo para problemas no lineales.

## 4. Estrategias didácticas

Como parte de las estrategias didácticas de esta asignatura se sugiere:

1. procurar que el curso se desarrolle centrandó el proceso de aprendizaje en el alumno, promoviendo la participación activa de los estudiantes con especial atención al desarrollo de habilidades de carácter general así como específicas de los métodos computacionales involucrados.
2. desarrollar el curso en forma de taller bajo la modalidad de problemas dirigidos donde se involucren las herramientas requeridas (aprendizaje basado en proyectos).
3. tener acceso a un laboratorio de estaciones de trabajo Linux, donde se encuentren sistemas de cómputo simbólico, numérico y de visualización instalados, para que el estudiante pueda trabajar en forma individual, buscando desarrollar el curso utilizando recursos de software libre y de fuente abierta.
4. contar con una página en Internet donde se encuentren los materiales para apoyar las actividades del curso, donde cada estudiante pueda acceder a los materiales y actividades en línea para el trabajo en forma individual.
5. tener acceso a sistemas de cómputo de alto rendimiento con sus herramientas (Fortran90, C, MPI, etc).
6. promover la investigación bibliográfica sobre los aspectos teóricos del curso.

## 5. Estrategias para la evaluación

La evaluación del curso será con base en los productos desarrollados a lo largo del curso, donde el estudiante ira mostrando su avance en la incorporación de las herramientas de cómputo simbólico, numérico y de visualización como un método auxiliar en simulación y resoluciones de problemas.

## 6. Bibliografía

La bibliografía, recursos en Internet y herramientas sugeridas para este curso son los siguientes:

1. Advanced Mathematical Methods with Maple, Derek Richards, Cambridge University Press, November 2001, ISBN 0521779812
2. Advanced Mathematics and Mechanics Applications Using Matlab, Howard B Wilson, Louis H Turcotte, David Halpern, CRC Press, September 2002, ISBN 158488262X
3. Mathematical Computing, David Betounes, Mylan Redfern, Springer Verlag, January 2002, ISBN 0387953310
4. Experimentation in Mathematics, Jonathan M Borwein, A K Peters, Ltd, April 2004, ISBN 1568811365
5. Computer Algebra and Symbolic Computation, Joel S. Cohen, A K Peters, Ltd, January 2003, ISBN 1568811594

6. A First Course in Scientific Computing: Symbolic, Graphic, and Numerical Modeling Using Maple, Java, Mathematica, and Fortran90., Rubin H. Landau, Princeton University Press, 2005, ISBN 0.691-12183-4
7. An Introduction to Computational Physics, Tao Pang, Cambridge University Press, September 1997, ISBN 0521485924
8. Computational physics. Fortran Version. Steven E. Koonin, Dawn C. Meredith. Westview press, 1990, ISBN: 0-201-38623-2
9. Computational Physics: An introduction, Franz Vesely, Franz J. Vesely. Second edition. Kluwer Academic/Plenum Publishers, 1994, ISBN: 0-306-46631-7
10. Computational Methods in Physics and Engineering, Samuel S M Wong, World Scientific, April 2003, ISBN 9810230435
11. Computational Physics, J M Thijssen, Cambridge University Press, June 1999, ISBN 0521575885
12. Numerical Recipes in Fortran 77, The Art of Scientific Computing, Vol. 1 of Fortran Numerical Recipes, Cambridge University Press, Second Edition, 2001 (<http://library.lanl.gov/numerical/bookfpdf.html>)
13. Numerical Recipes in Fortran90, Vol. 2 of Fortran Numerical Recipes, Cambridge University Press, Second Edition, 2002 (<http://library.lanl.gov/numerical/bookf90pdf.html>)
14. Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press, Second Edition, 2002. (<http://library.lanl.gov/numerical/bookcpdf.html>)
15. The Math Forum on Numerical Analysis (<http://mathforum.org/advanced/numerical.html>)
16. Michael Wester, A Critique of the Mathematical Abilities of CA Systems ([http://www.math.unm.edu/~wester/cas\\_review.html](http://www.math.unm.edu/~wester/cas_review.html))
17. Sistema Axiom (<http://savannah.nongnu.org/projects/axiom>). Axiom BSD, es un sistema de álgebra computacional diseñado para cómputo científico.
18. Sistema Maxima (<http://maxima.sourceforge.net/>). Maxima, es un sistema de álgebra computacional desarrollado en Lisp para realizar cómputo simbólico. El sistema está basado en MACSYMA y es licencia GPL.
19. Sistema Yacas (<http://yacas.sourceforge.net/>). El sistema YACAS GPL, es un sistema de álgebra computacional de propósito general. Está diseñado para tanto para cómputo simbólico, como numérico.
20. Sistema Octave (<http://www.octave.org/>). El sistema GNU Octave, es un lenguaje de alto nivel diseñado principalmente para cómputo numérico. El lenguaje Octave es compatible con Matlab.
21. Sistema Rlabplus (<http://rlabplus.sourceforge.net/>). Rlabplus es un ambiente de programación científica interpretado. Proporciona un lenguaje de alto nivel diseñado para el desarrollo rápido de programas y prototipos, así como para el procesamiento y visualización de datos.
22. Gnuplot (<http://www.gnuplot.info/>). Herramienta potente para graficación de datos y funciones.
23. PtPlot (<http://ptolemy.eecs.berkeley.edu/java/ptplot/>). Herramienta para graficación de datos y funciones desarrollada en Java.
24. Grace (<http://plasma-gate.weizmann.ac.il/Grace/>). Herramienta de graficación WYSIWYG desarrollada para el sistema de ventana X.

25. SourceForge.Net (<http://sourceforge.net>). Repositorio de Software de Libre y de Fuente Abierta..
26. Netlib.Org (<http://www.netlib.org/>). Repositorio de software matemático.
27. Jscience (<http://jscience.org/>). Repositorio de herramientas Java para aplicaciones científicas.

La bibliografía complementaria sugerida es la siguiente:

- 1) Richard L. Burden, J. Douglas Faires. *Análisis Numérico*, Séptima Edición, Thomson Learning, (2002). ISBN: 0-534-38216-9.
- 2) David Kincaid, Ward Cheney. *Numerical analysis. Mathematics of scientific computing*. Third edition. Thomson Brooks/Cole (2002). ISBN: 0-534-38905-8.
- 3) Melvin J. Maron, Robert J. Lopez. *Análisis Numérico. Un enfoque práctico*, Tercera edición. CECSA (1995). ISBN: 9-682-61251-9.
- 4) Brice Carnahan, H.A. Luther, James O. Wilkes. *Applied numerical methods*. John Wiley and Son (1969). ISBN: 0-471-13507-0.
- 5) David Kahaner, Cleve Moler, Stephen Nash. *Numerical Methods and Software*. Prentice Hall (1988). ISBN: 0-136-27258-4.
- 6) William H Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in Fortran 77. The Art of Scientific Computing*, Second edition. Cambridge University Press (1992). ISBN: 0-521-43064-X.
- 7) William H Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in Fortran 90. The Art of Scientific Computing*, Second edition. Cambridge University Press (1996). ISBN: 0-521-57439-0.

## 7. Perfil docente

El profesor que imparte esta materia deberá poseer una formación sólida en el campo de la física, así como experiencia en el uso del cómputo simbólico, numérico y sistemas de visualización científica. Deberá además tener la capacidad de dirigir al estudiante para que este adquiera habilidades en Física Computacional. Se requiere que el profesor o facilitador del curso, tenga fluidez de trabajo en ambientes de cómputo científico en plataformas Linux/UNIX, ambientes gráficos y en aplicaciones via Internet; además de poseer un dominio completo del temario del curso, que le permita trascender su contenido con base en sus opiniones y comentarios.