Curso de Métodos Numéricos

DEMAT, Universidad de Guanajuato

Clase 0: **Descripción del curso**

- Objetivos
- Temario
- Criterios de evaluación
- Motivación

MAT–251

Dr. Joaquín Peña Acevedo
CIMAT A.C.
e-mail: joaquin@cimat.mx

Descripción general del curso

- Para cada tema se da una introducción y se plantea el problema que se quiere resolver.
- Revisamos la teoría y vemos la manera de deducir uno o varios algoritmos de solución.
- Mediante las tareas, se programa algunos de los algoritmos vistos en clase, con el próposito de evaluar su desempeño.
- Hay que tomar en cuenta que hay una secuencia en los temas porque para desarrollar algunos algoritmos se se requiere de parte de la teoría vista anteriormente y de los algoritmos asociados.
- Pasamos al siguiente tema y repetimos.

Objetivos del curso

- Deducir, analizar y aplicar los métodos numéricos tradicionales en la solución computacional de problemas científicos.
- Tener la capacidad de implementar los algoritmos en computadoras.
- Tener la capacidad de análisis de las técnicas numéricas, con especial atención al tratamiento de los errores de truncamiento, y los de redondeo inherentes a su implementación en computadoras.
- Tener la capacidad de seleccionar los métodos numéricos más apropiados para resolver un problema planteado.
- Tener la capacidad de obtener de información para resolver problemas que requieren cómputo numérico.

Temario

Unidad 1. **Preliminares**

1.1. Introducción del curso
1.2. Representación de números en la computadora
1.3. Errores numéricos y precisión de la máquina
1.4. Propagación de errores y estimación del error

Representación de un número en la computadora

1	10100111	10100100101100110001011
Signo	Exponente	Mantisa

$$x \in \mathbb{R} \quad \longmapsto \quad \mathtt{fl(x)} = x + \epsilon$$

Temario

Unidad 2. **Solución de ecuaciones no lineales**

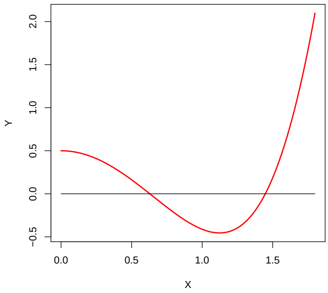
2.1. Método de bisección
2.2. Método de punto fijo.
2.3. Aspectos de programación: Apuntadores a funciones, lectura y escritura de datos. Graficación.
2.4. Método de Newton-Raphson.
2.5. Método de la secante.

Dada una función de una variable, $f:\mathbb{R}\rightarrow\mathbb{R}$, idealmente quisieramos encontrar los valores $x\in\mathbb{R}$ tal que

$$f(x)=0.$$

Veremos que en realidad el problema que plateamos es encontrar un valor x tal que

$$|f(x)|<\epsilon.$$



Temario

Unidad 3. **Solución de sistemas de ecuaciones lineales**

3.1. Normas vectoriales y matriciales

3.2. Aspectos de programación: Memoria dinámica y uso de la librería GSL o de la librería Numpy y de algebra lineal numérica de Python.

3.3. Métodos directos de solución:

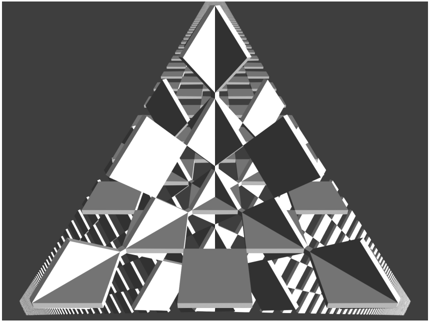
- 3.3.1. Eliminación Gaussiana y pivoteo.
- 3.3.2. Sistemas con matrices diagonales y triangulares
- 3.3.3. Solución para matrices tridiagonales.
- 3.3.4. Factorización LU.
- 3.3.5. Factorización de Cholesky y LDL^t

3.4. Métodos iterativos de solución:

- 3.4.1. Método iterativo de Jacobi.
- 3.4.2. Método de Gauss-Seidel.

Unidad 3. Solución de sistemas de ecuaciones lineales

Hay que solver el sistema $\boldsymbol{Ax} = \boldsymbol{0}$ con $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ de manera eficiente, considerando si $\boldsymbol{A}$ tiene ciertas características, estructura o tamaño.

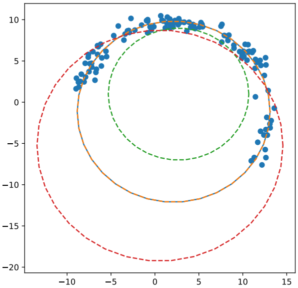


Temario

Unidad 4. **Mínimos cuadrados lineales**

Se tiene un conjunto de datos $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$, donde $\boldsymbol{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$.
Se quiere ajustar un modelo $m(\boldsymbol{x}; p1, ..., p_s)$ a los datos, que depende de los parámetros $p1, ..., p_s$. Para esto, se resuelve el problema

$$\min_{p1,...,p_s} \sum_{i=1}^{m} [m(\boldsymbol{x}_i; p1, ..., p_s) - y_i]^2$$



Temario

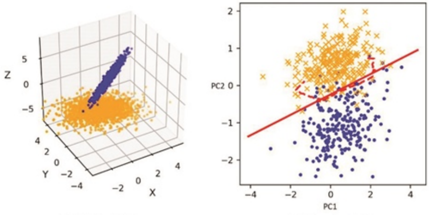
Unidad 5. **Cálculo de eigenvalores, eigenvectores y valores singulares**

5.1. Método de la potencia directo e inverso.
5.2. Método QR.
5.3. Método de ortogonalizacion de Gram-Schmidt.
5.4. Descomposicion SVD.
5.5. Número de condición de las matrices.

Dada la matriz $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, queremos encontrar los vectores $\boldsymbol{x}$ y los escalares λ tales que

$$\boldsymbol{Ax} = \lambda \boldsymbol{x}.$$

Esto nos lleva a la descomposición espectral de una matriz. Después vemos una descomposición más general, SVD, basada en valores singulares.

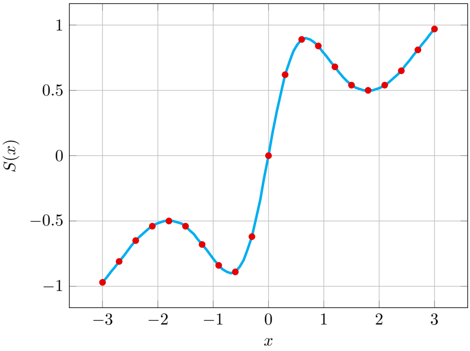


(a) Original data (b) Regular PCA

Unidad 6. Interpolación

Se tiene un conjunto de datos $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, donde $\boldsymbol{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$. Queremos encontrar un función tal que , $f : \mathbb{R}^n \to \mathbb{R}$, tal que

$$f(\boldsymbol{x}_i) = y_i \quad \text{para} \;\; i = 1, \ldots, m.$$



Temario

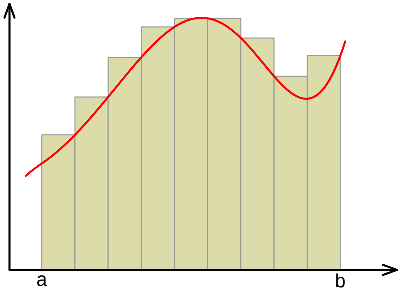
Unidad 7.	Integración numérica

7.1. Regla del trapecio.
7.2. Método de Romberg.
7.3. Regla de Simpson.
7.4. Cuadratura Gaussiana.
7.5. Integrales impropias.
7.6. Integrales múltiples.

Dada una función $f : \mathbb{R} \to \mathbb{R}$ continua a pedazos, queremos calcular

$$\int_a^b f(x)\,dx$$

En realidad solo obtenemos una aproximación.

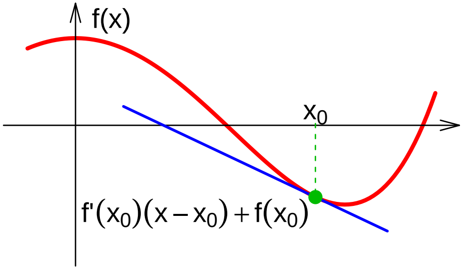


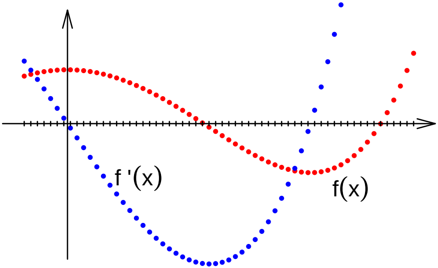
Temario

Unidad 8.	Diferenciación numérica

8.1. Aproximación de primeras derivadas.

8.2. Aproximación de segundas derivadas.

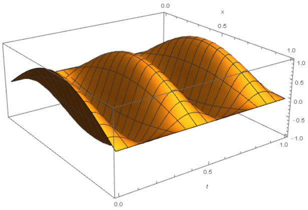


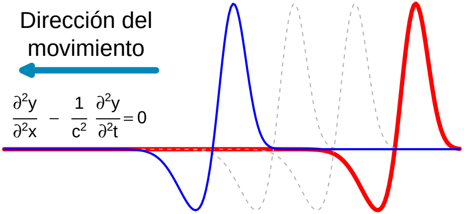


Temario

Unidad 9. **Solución numérica de ecuaciones diferenciales**

9.1. Conceptos sobre convergencia, estabilidad y consistencia.
9.2. Método de Euler.
9.3. Métodos de Runge-Kutta.
9.4. Métodos predictor-corrector.
9.5. Problemas de valores en la frontera y el método de disparo.
9.6. Ecuaciones de orden mayor a 1 y sistemas de ecuaciones diferenciales.
9.7. Método de diferencias finitas.
9.8. Introducción a ecuaciones diferenciales parciales.





Referencias

- Análisis numérico
  Richard L. Burden, J. Douglas Faires. Thompson Learning
- Numerical Recipes in C; The Art of Scientific Computing
  William H.Press, Brian P. Flannery, et. al. Cambridge University Press
- Numerical Mathematics
  Alfio Quarteroni.
  Springer Verlag. 2000.
- Numerical Python: A practical techniques approach for industry.
  Robert Johansson.
  Apress 2015.

Pre-requisitos

- Conocimientos de programación en C, C++ o Python.
- Conocimientos básicos de álgebra lineal.
- Conocimientos básicos de cálculo diferencial e integral.

Apoyo para las tareas de programación en C/C++

- Repaso de memoria dinámica, apuntadores a funciones y lectura y escritura de datos.
- Graficación usando Gnuplot.
- A lo largo del curso, usaremos la librería de métodos numéricos GSL, la cual tiene implementados una gran variedad de algoritmos.

Apoyo para las tareas de programación en Python

- Referencias sobre la sintaxis del lenguaje.
- Información para leer y escribir datos en archivos.
- Graficación usando Matplotlib.
- A lo largo del curso, usaremos algunos de los módulos que tienen implementados algoritmos para cálculo numérico.

Clases

- Las clases son los lunes y miércoles de 11:00 a 12:20.
- Usaremos Google Classroom para organizar y proporcionar algunos materiales, como las diapositivas de los temas vistos, videos con la explicación de algunos temas y los archivos de las tareas.
- La dinámica de las clases será la siguiente:
  (a) Iniciamos cada clase con una videoconferencia usando Google Meet, recapitulando los temas vistos previamente y mencionando los temas que se verán durante la clase.
  (b) Continuamos con la revisión un video corto que explica uno de los temas.
  (c) Al terminar el video, continuamos con la videoconferencia, para comentarlo, ampliar la información y responder las dudas.
  (d) Repetimos los incisos (a) y (b) hasta casi agotar el tiempo de la clase.
  (e) Los últimos minutos de la clase los empleramos para dar indicaciones de la tarea, revisar sus comentarios u otros pendientes.

Clases

Algunas consideraciones:

- Los estudiantes deben ingresar a la videoconferencia con el micrófono apagado, y sólo activarlo cuando quieran responder una pregunta o comentar algo.
- Durante la clase puede usar el chat de Meet para escribir las dudas o comentarios que se tengan.
- El propósito de los videos es que puedan escuchar la explicación de un tema sin ruidos o cortes que se pueden generar durante la trasmisión en vivo.
- El tiempo para ver un video será el que corresponda a su duración más dos minutos adicionales. Durante este tiempo, los estudiantes pueden salir de la conferencia para no consumir sus recursos, y regresar la videoconferencia cuando haya trascurrido ese tiempo.
- El material que se desarrolle en los espacios entre las reproducciones de los videos no será proporcionado, por lo que se recomienda tomar notas.

Criterios de evaluación

El curso se evaluará mediante:

- **Tareas**.
  Generalmente habrá una tarea por semana.
  Las tareas pueden incluir ejercicios teóricos, en los que hay que escribir la solución del problema, y ejercicios prácticos, en los hay que programar un algoritmo y probarlo.
  La tarea que se encarga en una semana y se entrega a más tardar el domingo de la siguiente semana.

  Fecha de asignación →

Lu	Ma	Mi	Ju	Vi	Sa	Do
Lu	Ma	Mi	Ju	Vi	Sa	Do
Lu	Ma	Mi	Ju	Vi	Sa	Do

  Fecha de ← entrega

  Nota: Se puede entregar la tarea después de la fecha estipulada, pero se resta 1 punto por cada día de retraso y no puede haber más de 4 días de retraso.

Criterios de evaluación

- **Participación en clase**.
  Se llevará el registro de los estudiantes que participen explicando un resumen de la clase anterior, respondiendo correctamente a las preguntas que se formulen, haciendo comentarios oportunos, etc.
- **Dos exámenes parciales**.
  Debido a la situación actual, los exámenes serán prácticos principalmente, con un cuestionario rápido para evaluar la teoría. Los ejercicios de programación serń para resolver un problema usando los códigos desarrollados en las tareas.
- **Un proyecto al final del curso**.
  Se darán opciones de temas, aunque los estudiantes también pueden proponer el tema.

Criterios de evaluación

La calificación de cada una de las cuatro activides descritas anteriormente está entre 0 y 10.

Para calcular la calificación final del curso, se suman las calificaciones de cada actividad multiplicada por la ponderación indicada en la siguiente tabla:

Actividad	Ponderación
Tareas:	**0.45**
Primer examen parcial:	**0.15**
Segundo examen parcial:	**0.15**
Proyecto final:	**0.15**
Participación en clase:	**0.10**

Tutorías

- No se si habrá un ayudante para el curso.
- Para consultas con el profesor, mandar un correo electrónico a `joaquin@cimat.mx` o un mensaje mediante Google Classroom.

Herramientas de programación

Para los que quieran usar C/C++:

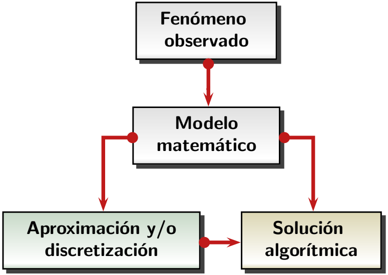
- Usaremos el compilador gcc, o en Windows, MinGW (`http://www.mingw.org/`).
- Para las tareas prácticas, usaremos Code::Blocks como ambiente integrado de programación.

Para los que quieran usar Python:

- Se puede instalar Anaconda en una computadora personal y disponer de diferentes herramientas para programar en Python.
- Otra alternativa es usar Google Colab, el cual ofrece un ambiente gratuito para escribir y ejecutar código Python a través de un navegador web.

Panorama general

El cómputo científico es una disciplina relacionada con el desarrollo y estudio de algoritmos numéricos para resolver problemas matemáticos que tienen su origen en varias áreas.

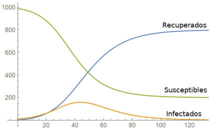


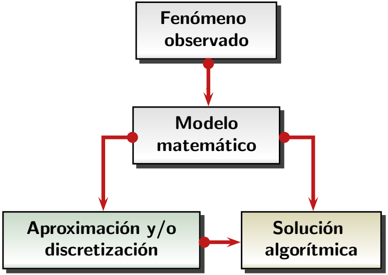
Panorama general

El cómputo científico es una disciplina relacionada con el desarrollo y estudio de algoritmos numéricos para resolver problemas matemáticos que tienen su origen en varias áreas.

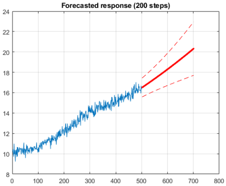
Modelo basado en ecuaciones diferenciales:

$$
\begin{aligned}
\frac{dS}{dt} &= -\beta S I \\
\frac{dI}{dt} &= \beta S I - \gamma I \\
\frac{dR}{dt} &= \gamma I
\end{aligned}
$$





Predicción mediante un modelo para una serie de tiempo



Algoritmos de solución

Características deseadas de los algoritmos:

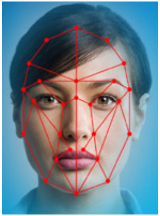
Eficiencia **Exactitud** **Robustez**

Algoritmos de solución

Características deseadas de los algoritmos:

Eficiencia | **Exactitud** | **Robustez**

Ejemplo:



Uso factible en diferentes tipos de dispositivos

Puede requerir varios intentos para lograr el reconocimiento

El resultado depende de varios factores o cambios en las personas

Algoritmos de solución

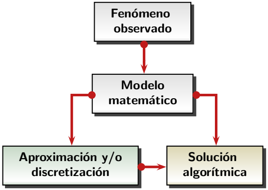
Características deseadas de los algoritmos:

Eficiencia | **Exactitud** | **Robustez**

Factores que intervienen en el desarrollo y desempeño de los algoritmos:

Ambiente de programación | **Estructuras de datos** | **Arquitectura de cómputo**

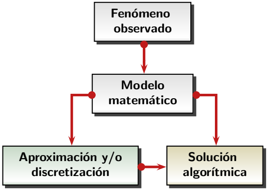
Fuentes de error



Todo proceso computacional que trata con la solución de problema matemático involucra errores:

- Errores en las observaciones del fenómeno.
- Errores de modelación del problema.
- Errores al usar aproximaciones matemáticas.
- Errores debidos a la representación de los números en la computadora y la aritmética.
- Errores de programación.

Fuentes de error



Todo proceso computacional que trata con la solución de problema matemático involucra errores:

- Errores en las observaciones del fenómeno.
- Errores de modelación del problema.
- Errores al usar aproximaciones matemáticas.
- Errores debidos a la representación de los números en la computadora y la aritmética.
- Errores de programación.

En este momento, estamos interesados en analizar el penúltimo punto de la lista.