CÁLCULO NUMÉRICO

davide.barbieri@uam.es

www.uam.es/davide.barbieri/docencia.html

TUTORÍAS
Jueves de 10 a 12
Despacho 305

T1. INTRODUCCIÓN
- Coma flotante

T2. SOLUCIÓN DE ECUACIONES $\mathbb{R} \to \mathbb{R}$

T3. SOLUCIONES DE SISTEMAS LINEALES

$$Ax = b \,, \quad A \in \mathbb{R}^{n\times m} \quad (GL_m(\mathbb{R}))$$

PARCIAL 1

UNIVERSIDAD AUTONOMA DE MADRID

Facultad de Ciencias

Departamento de Matemáticas

LABORATORIO DE CÁLCULO (M17-101)

En la parte de abajo de esta hoja encontrarás tu nombre de usuario y contraseña para el examen que vas a realizar. Sólo sirven para el examen y quedarán anulados en cuanto termine.

1. Al comienzo del examen, en ocasiones después de unos cinco minutos, se cortará el acceso a Internet y la posibilidad de montar pinchos USB, pero los pinchos que ya estuvieran conectados seguirán funcionando.
2. Además, una vez dentro de la cuenta no puedes salir porque estarán bloqueadas. Si por accidente ocurre debes comunicarlo al profesor a cargo del examen.
3. Debes deshabilitar el *salvapantallas* para que no bloquee tu cuenta durante el examen. Para eso debes ir a

   `Sistema>Preferencias>Salvapantallas`

   y *desmarcar* la casilla *Bloquear pantalla cuando el salvapantallas esté activo.*
4. Si durante el examen vas a utilizar Sage, debes arrancarlo en una terminal

   `Aplicaciones>Herramientas del sistema>Terminal de MATE`

   ejecutando en la ventana de la terminal `arrancar-jupyter-ex.sh`.
   Debes mover el archivo con los enunciados, que estará en el escritorio, a la carpeta

   `ENTREGA_EXAMEN_alejandro.santorum-numidt-ex1`

   y después verás el archivo desde dentro de `jupyter`.
5. En el escritorio de tu cuenta encontrarás una carpeta llamada

   `ENTREGA_EXAMEN_alejandro.santorum-numidt-ex1`

   en la que debes colocar todos los archivos con las soluciones a los ejercicios del examen y nada más. Por favor, antes de salir de la cuenta y dar por terminado tu examen comprueba cuidadosamente que la carpeta indicada contiene todos los archivos con tus soluciones.

Madrid, 16 de marzo de 2018

Nombre y apellidos: ALEJANDRO SANTORUM VARELA

DNI: 44090946-S

Firma:

`alejandro.santorum-numidt-ex1`

`5gu9y9Y`

Práctica de Calculo Numérico I

Primer Examen Parcial - Grupo 720, 2º Doble Gr.

20 Marzo 2018

Ejercicio 1 (4 puntos)

Escribir una `MATLAB function` que implemente el método de Newton para resolver un problema `f(x)` = 0, dejando que el usuario pueda elegir la tolerancia para el error absoluto de la solución `x`, y devuelva como output también el número de iteraciones. Usarla en un `MATLAB script` para encontrar una solución positiva al problema

$$\frac{1}{(1+e^{-x})} = x^2 - 1$$

con una tolerancia `1e-6`.

Ejercicio 2 (4 puntos)

Escribir una `MATLAB function` que implemente el método de bisección para resolver un problema `f(x)` = 0, dejando que el usuario pueda elegir la tolerancia para el error absoluto de la solución `x`, y devuelva como output también el número de iteraciones. Usarla en un `MATLAB script` para resolver

$$x^{13} - 230x^{12} + 0.3x^9 - 77x^8 + 0.15x^5 - 33x^4 + 0.05x - 12 = 0$$

tomando como intervalo inicial $I = [100, 600]$ y como tolerancia `1e-6`.

Ejercicio 3 (2 puntos)

Usar las dos `MATLAB functions` de los ejercicio anteriores para resolver $f(x) = 0$, donde f es la función del Ejercicio 2, según el siguiente esquema, que depende de dos parámetros enteros N y M a elegir en $\{1, \ldots, 10\}$.

1. Se empieza haciendo N pasos de bisección a partir de un intervalo I_0 definido por el usuario, para acercarse a la solución. Se encuentra así un intervalo I_1.

2. Se escoge el punto medio del intervalo obtenido en 1. para encontrar la solución con el método de Newton.

3. Si el método de Newton no resultara convergente en M pasos, se vuelve a hacer 1. a partir del intervalo I_1. Se genera así un intervalo I_2 y un nuevo punto medio, y este se usa nuevamente como en 2. para hacer hasta M pasos de Newton.

4. Se continua con esta repetición de las dos técnicas hasta cuando el error absoluto sobre `c` es menor o igual a `1e-6`.

Usar este esquema con $M = 3$, $N = 3$ y $I_0 = [100, 600]$.

Observación: el comando `isfinite` devuelve un output lógico que vale 1 si su argumento es un número finito, y 0 si su argumento es infinito o `NaN`. Ver también `help isfinite`.

ALEJANDRO SANTORUM VARELA

Práctica de Calculo Numérico I

Segundo Examen Parcial - Grupo 220

8 de Mayo de 2018

Ejercicio 1 (6 puntos)

Escribir una MATLAB `function` llamada `lagrangepol` que, dado un function handle f, unos nodos $\{x_j\}_{j=1}^N \subset \mathbb{R}$, y un punto $x \in \mathbb{R}$, calcule el valor en el punto x del polinomio en la forma de Lagrange interpolador de f en esos nodos.

Nota: recordamos que el polinomio interpolador en la forma de Lagrange es dado por

$$p(x) = \sum_{k=1}^{N} f(x_k) L_k(x)$$

donde $L_k(x) = \prod_{j=1}^{N}{}_{j \neq k} \frac{x - x_j}{x_k - x_j}$ por $k = 1, \ldots, N$.

Ejercicio 2 (4 puntos)

Sea p el polinomio interpolador de

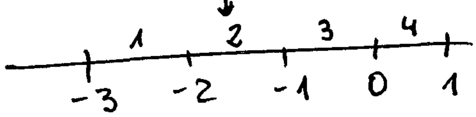
$$f(x) = \frac{e^x - 1}{1 + x^2}$$

en los puntos $x_1 = 0; x_2 = -1; x_3 = -3; x_4 = 1; x_5 = -2.7$.

Escribir un MATLAB `script` (**no una** `function`) que use la `function` del apartado anterior para calcular

$$\int_{-3}^{1} p(x)dx$$

con la fórmula del punto medio compuesta en 4 subintervalos de igual tamaño.



$I^{M}_{Comp}[f] =$

$= \sum I^{M}_{Simple}[f] = \sum p\left(\frac{a+b}{2}\right)(b-a)$

$\sum_{i=-3}^{0} p\left(\frac{i+(i+1)}{2}\right)\cdot(i+1-i) = \sum_{i=-3}^{0} p\left(\frac{i+(i+1)}{2}\right)\cdot 1 = \sum_{i=-3}^{0} p\left(\frac{i+(i+1)}{2}\right)$

donde p es el polinomio interpolador de $f(x) = \frac{e^x-1}{1+x^2}$ en los puntos (nodos) $\begin{cases} x_1 = 0 \\ x_2 = -1 \\ x_3 = -3 \\ x_4 = 1 \\ x_5 = -2.7 \end{cases}$