Tutoría 7
Física Computacional I
Grado en Física

Javier Carrasco Serrano, javcarrasco@madrid.uned.es

Prueba evaluable de programación con Maxima

Prueba Maxima - Enunciados

Adobe Acrobat
Document

Prueba Maxima – Ejercicio 1

- Tips sobre orden de los pasos a seguir (¿por qué seguir ese orden de sustitución?), cuál es el input, y cuál es el output:

  https://2019.cursosvirtuales.uned.es/dotlrn/grados/asignaturas/61041094-19/uforums/thread-view?message_id=41387717

“En este caso, dado que el input y output de las funciones pedidas en ambos ejercicios es idéntico, y dado que el ejercicio 1 es (en sentido estricto) un caso particular del ejercicio 2, lo normal es resolver solamente el ejercicio 2, mencionando en un comentario del código que la función del ejercicio 2 también resuelve el ejercicio 1, ya que es un caso particular del 2 (este comentario es imprescindible, para que no puedan acusarnos de haber dejado un ejercicio sin resolver).

Prueba Maxima – Ejercicio 1

Caso general del caso lineal: $\begin{cases} a(x)\cdot f'(x) + b(x)\cdot f(x) + c(x) = 0 \\ f(x_0) - f_0 = 0 \end{cases}$

Utilizando el cambio de variable $x = x(y)$. Esta expresión simplemente quiere decir que damos x en función de y. Por ejemplo, $x = 2y, x = e^y, x = \ln(y)$ son algunos ejemplos de cambios de variable. La notación puede ser un poco confusa, y puede ayudar a verlo escribir el cambio de variable como $x = g(y)$, pero no es nada nuevo, volvemos a dar x en función de y.

Dado un cambio de variable, por ejemplo $x = x(y)$ (ó $x = g(y)$), se puede obtener la expresión del cambio inverso de variable como: $x = x(y) \rightarrow y = x^{-1}(x)$ (ó $x = g(y) \rightarrow y = g^{-1}(x)$). Cuidado porque se está denotando a $x^{-1}(x)$ como $y\ (x)$, es decir $y = y(x)$. Es simplemente notación.

Por ejemplo, para aplicar el cambio de variable a x_0 haríamos $x_0 = x\ (y)$, y despejaríamos la y de la expresión $\mathrm{x}\ (y)$, obteniendo así que $x_0 = x\ (y_0)$. Con la otra notación: $x_0 = g\ (y) \rightarrow y_0 = g^{-1}(x_0)$.

Prueba Maxima – Ejercicio 1

Para sustituir la expresión $f'(x)$ por $f'(x(y))$ [cuidado porque cuando se simplifica la notación y se escribe y en lugar de x, a veces se quiere decir $x(y)$], es decir, para aplicar el cambio de variable a la derivada de la función, hay que aplicar la regla de la cadena que conocemos de las derivadas ($\left(f\left(g(x)\right)\right)' = f'\left(g(x)\right) \cdot g'(x)$) a la función con el cambio de variable:

$$f'(x) = \frac{df(x)}{dx} = \frac{dy(x)}{dx} \cdot \frac{df(y)}{dy} = y'(x) \cdot f'(y)$$

De este desarrollo, cuando estemos programando la solución del ejercicio**, lo único que se debe tener en cuenta es que el lado izquierdo de la expresión es igual al lado derecho** ($f'(x) = y'(x) \cdot f'(y)$). En el último paso de los ejercicios, como se indica, se debe sustituir x por $x(y)$, con lo que quedaría $f'(x) = y'(x) \cdot f'(y) = y'(x(y)) \cdot f'(y)$, que aunque por un tema de notación se pueda escribir como $f'(x) = y'(y) \cdot f'(y)$, en realidad puede llevar a confusión porque la primera y entre paréntesis quiere decir que se ha sustituido x por su cambio de variable en y, y la segunda y entre paréntesis quiere decir efectivamente la variable y. Si resulta confuso, simplemente pensar en la expresión como $f'(x) = y'(x) \cdot f'(y) = y'(x(y)) \cdot f'(y)$, pero sustituir, como indica el Equipo Docente, x por $x(y)$ en el último paso del ejercicio.

Prueba Maxima – Ejercicio 1

Una vez obtenida la expresión $f'(x) = y'(x) \cdot f'(y)$, simplemente hay que sustituirla, así como sustituir x por $x(y)$, en la ecuación diferencial inicial, quedando:

$$\begin{cases} a(x(y)) \cdot f'(y) \cdot y'(x(y)) + b(x(y)) \cdot f(x(y)) + c(x(y)) = 0 \\ f(x(y_0)) - f_0 = 0 \end{cases}$$

Ojo porque en la expresión anterior se podría "relajar" la notación y escribir una y en cada una de las $x(y)$ para indicar que esas funciones están dadas en la nueva variable y, pero NO es un cambio directo (hay que usar el cambio de variable).

Matemáticamente, puede haber problemas si el cambio de variable que se utiliza no tiene solución (inversa), o tiene varias soluciones. El enunciado indica que se asume que el cambio de variable tiene una solución (inversa) única. Esto quiere decir que cuando tengáis hecho el programa, y estéis probando un ejemplo para ver si funciona bien, NO podéis meter un ejemplo de cambio de variable como $x = y^2, x = \sin(x)\,, x = \cos(y)$ ...

Prueba Maxima – Ejercicio 1

Ejemplo: $\begin{cases} e^x \cdot f'(x) + e^x \cdot f(x) + e^x = 0 \\ f(0) - 1 = 0 \end{cases}$

Utilizamos el cambio de variable $x = x(y) = \ln(y)$. Así, $x = \ln(y) \to y = y(x) = e^x$. Para sustituir $f'(x)$ por $f'(x(y))$ necesitamos aplicar la expresión deducida antes de la regla de la cadena.

En primer lugar necesitamos calcular la derivada del cambio de variable en y:

$y = y(x) = e^x \to y' = y'(x) = e^x$, por lo que $f'(x) = y'(x) \cdot f'(y) = e^x \cdot f'(y)$.

Ahora hay que sustituir en la Ecuación Diferencial los términos $f'(x)$ por $e^x \cdot f'(y)$, $f(x)$ por $f(\ln(y))$, y x por $x(y)$, en el orden que indica el enunciado:

$e^x \cdot f'(x) + e^x \cdot f(x) + e^x = 0 \to [f'(x) = e^x \cdot f'(y)]$

$e^x \cdot e^x \cdot f'(y) + e^x \cdot f(x) + e^x = 0 \to [f(x) = f(\ln(y))]$

$e^x \cdot e^x \cdot f'(y) + e^x \cdot f(\ln(y)) + e^x = 0 \to [x = \ln(y)]$

$y^2 \cdot f'(y) + y \cdot f(\ln(y)) + y = 0$

Y para la condición inicial: $f(0) - 1 = 0 \to f(x_0 = 0) - 1 = 0 \to x_0 = 0 \leftrightarrow \ln(y) = 0 \to y_0 = e^0 = 1$. Por lo que la nueva condición inicial es $f(1) - 1 = 0 \to f(y_0 = 1) - 1 = 0$

Prueba Maxima – Ejercicio 2

Caso general: $\begin{cases} \mathrm{E}(f'(x), f(x), x) \; = 0 \\ \quad f(x_0) - f_0 = 0 \end{cases}$

Todo aplica exactamente igual, únicamente hay que tener en cuenta que el input será una ecuación no necesariamente lineal, expresada por ejemplo como $\mathrm{E}(f'(x), f(x), x) \; = 0$, y el output debe tener la misma forma una vez hecho el cambio de variable y los ajustes anteriores, es decir, la salida tendrá la forma: $\mathrm{E}(f'(y) \cdot y'(x(y)), f(x(y)), y) \; = 0$

Prueba Maxima – Ejercicio 2

Ejemplos de EDOs de orden 1:

- $f'(x) = \cos\left(f(x)\right)$
- $f'(x) = e^{f(x)}$
- $f'(x) = \ln\left(f(x)\right)$
- $f'(x) = \left(f(x)\right)^2$

Prueba Maxima – Ejercicio 3

Caso general: $\begin{cases} \mathrm{E}(f''(x), f'(x), f(x), x) = 0 \\ f(x_0) - f_0 = 0 \\ f'(x_0) = v_0 \end{cases}$

La única diferencia respecto al caso de orden 1 es que ahora tenemos que sustituir $f''(x)$ por una expresión equivalente en términos de $f''(x(y))$, y que debemos transformar la nueva condición inicial ($f'(x_0) = v_0$). Para hallar la expresión de $f''(x)$ al aplicar el cambio de variable, se deriva la expresión anterior$f'(x) = y'(x) \cdot f'(y)$ y se aplica le regla de la derivada del producto y nuevamente la regla de la cadena:

$$f''(x) = \frac{d^2 f(x)}{dx^2} = \frac{d^2 y(x)}{dx^2} \cdot \frac{df(y)}{dy} + \left(\frac{dy(x)}{dx}\right)^2 \cdot \frac{d^2 f(y)}{dy^2} = y''(x) \cdot f'(y) + (y'(x))^2 \cdot f''(y)$$

Nuevamente, cuando estemos programando la solución del ejercicio**, lo único que se debe tener en cuenta es que el lado izquierdo de la expresión es igual al lado derecho** ($f''(x) = y''(x) \cdot f'(y) + (y'(x))^2 \cdot f''(y)$).

Prueba Maxima – Ejercicio 3

La última diferencia del caso de orden 2 respecto a la EDO de orden 1 es la transformación que hay que aplicar a la segunda condición inicial (la que incluye el valor inicial de la función derivada). En este caso, no habrá que sustituir directamente, sino que habrá que sustituir primero la función derivada en x, por la función derivada en $x(y)$ con la expresión que ya hemos utilizado antes: $f'(x) = y'(x) \cdot f'(y)$

Por lo que si la segunda condición inicial es $f'(x_0) - v_0 = 0$, la nueva condición sería, utilizando que $x_0 = x(y_0)$:

$$f'(x) = y'(x) \cdot f'(y) \rightarrow f'(x_0) = y'(x(y_0)) \cdot f'(x(y_0)) \rightarrow v_0 = y'(x(y_0)) \cdot f'(x(y_0))$$
$$\rightarrow f'(x(y_0)) - \frac{v_0}{y'(x(y_0))} = 0$$

Prueba Maxima – Ejercicio 3

Ejemplos:

- $f''(x) = \cos\big(f(x)\big) + \sin(f'(x))$
- $f''(x) = \cos\big(f'(x)\big) + \sin(f(x))$
- $f''(x) = e^{f'(x)} + e^{f(x)}$
- $f''(x) = \ln\big(f'(x)\big)$
- $f''(x) = \big(f'(x)\big)^2 + \big(f(x)\big)^4$

Prueba Maxima – Ejercicio 3

- Tips sobre orden de los pasos a seguir, pero añadiendo como primer paso la sustitución de $f''(x)$ por $y''\big(x(y)\big) \cdot f'(y) + \left(y'\big(x(y)\big)\right)^2 \cdot f''(y)$:

  https://2019.cursosvirtuales.uned.es/dotlrn/grados/asignaturas/61041094-19/uforums/thread-view?message_id=41387717

Prueba Maxima – Ejercicio 4

- Tips sobre rango de la función que se debe graficar:

  https://2019.cursosvirtuales.uned.es/dotlrn/grados/asignaturas/61041094-19/uforums/thread-view?message_id=41566212

Tema 10: Programas informáticos: qué son y cómo se construyen

10

Introducción

- Programa informático: conjunto de datos e instrucciones que ejecuta un ordenador.
- C: lenguaje de programación de sistemas operativos (UNIX escrito en C, Linux basado en UNIX…).
- C estructura el código en funciones (agrupaciones de instrucciones).
- Facilita el acceso a la memoria y disco de la máquina (uso de punteros).
- Usos:
  - Cálculos con cadenas de símbolos: máquinas o autómatas celulares, para simulaciones.
  - Tratamiento de imágenes.
  - Tratamiento de datos que hay en un dispositivo del laboratorio (periférico).
- Manuales:
  - *Apuntes de Física Computacional*, Equipo Docente, Ud 10-15.
  - *El lenguaje de programación C*, Kernigan and Ritchie.
  - *The GNU C reference manual*, https://www.gnu.org/software/gnu-c-manual/
  - Aprenda lenguaje ANSI C como si estuviera en Primero,

  http://www4.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/ansic/leng_c.pdf

- Linux: Geany (editor) + GCC (compilador)
- Windows:

  Adaptación de GCC para Windows: Mingwin (compilador)

  http://www.mingw.org/ → 32 bits

  https://mingw-w64.org/ → 64 bits

  Code::Blocks (IDE)

  http://www.codeblocks.org/ → la instalación incluye mingwin → descargar *binary*

  → *codeblocks-17.12mingw-setup.exe*

  Alternativa: utilizar el notepad2 como editor, y ejecutar desde terminal

  http://www.flos-freeware.ch/notepad2.html

10.2. Compilación, enlazado y ejecución de programas

Etapas compilación fichero:

- Preprocesamiento: interpreta directivas, sirve para cargar determinadas funciones del sistema, como por ejemplo printf (imprimir en pantalla) → **#include** <stdio.h>

- Interpretación de la sintaxis: lee palabras clave (*int, return, {}, main, printf,…*).

- Compilación: una vez entendida la estructura, funciones, variables… lo traduce a lenguaje máquina para que lo interprete el procesador.

- Enlazador: llama a funciones (guardadas en objetos O) de las que haga uso nuestro programa.

10.2. Compilación, enlazado y ejecución de programas

Proyecto en Code::Blocks. Se utiliza un proyecto para dividir un programa grande en pequeños programas (archivos).

- File → New → Project → Console application → C → “HolaMundo” → “HolaMundo.cpb” (una carpeta por proyecto).
- Al elegir *console application* crea una estructura por defecto.
- File → Save file as… → “helloworld0.c” → sobre el panel de la izquierda (*workspace*), sobre “main.c” → remove file from Project → add files → “helloworld0.c”
- Añadimos *system(“PAUSE”)* al programa para que no se cierre al ejecutarse desde consola (o al abrir el .exe, que es lo mismo). Este comando deja abierta la consola hasta que se pulse alguna tecla. *stdlib.h* hace que se cargue esta función.
- Dos posiblidades:
    - Build → Compile current file → crea un objeto (O) del módulo, y ejecuta sólo esta parte del programa; rápido para detectar errores.
    - Build → Build → crea un objeto (O) por cada módulo, ejecuta todo el programa y crea el ejecutable.
    - Build and run, Run, o doble click sobre el .exe que se ha generado para correr el programa.

10.3. Aplicaciones auxiliares

- Se construyen programas pequeños para cada tarea dentro de programas mayores.

- Comando para acceder a la documentación de una función: *man*

  man atoi (la función *atoi* interpreta cadenas como números enteros)

- Comando para buscar una función (cuando se sabe qué hace pero no su nombre): *apropos*

  apropos print → *man 3 printf*

- Representación gráfica (mismo paquete que Maxima): *gnuplot*, comando *plot*

  plot

  help plot

  http://www.gnuplot.info → debería estar instalado de wxMaxima

Gracias!