

## Lea atentamente estas instrucciones antes de comenzar:

- Responda a cada una de las cuestiones prácticas.
- Es necesario aprobar este examen (5 puntos) para aprobar la asignatura.
- No es suficiente este examen para aprobar la asignatura: se deben aprobar los dos trabajos de la parte práctica.

## Sistemas de álgebra computacional: Maxima

1. (1 punto) ¿Qué sintaxis empleamos en Maxima para definir una lista?
2. (1 punto) ¿Qué comando se emplea en Maxima para cargar un “paquete de funciones”? Como p. ej, el paquete “dynamics” que incluye (entre otras) la función rk.
3. (2 punto) El comando rk (Runge-Kutta) del Maxima, ¿sirve para resolver ecuaciones diferenciales ordinarias de orden superior a 1? Explique brevemente cómo.

## Programación en C

1. (2 puntos) Indique qué guarda la siguiente función en el archivo.

```
void comoCSV(char* fn, char** cols, int n, int m, double** vals) {  
    FILE *f=fopen(fn, "wt");  
    int i, j;  
    fprintf(f, "%s", cols[0]);  
    for(j=0; j<m; j++) {  
        fprintf(f, ",%s", cols[j]);  
    }  
    fprintf(f, "\n");  
    for(i=0; i<n; i++) {  
        fprintf(f, "%g", vals[i][0]);  
        for(j=1; j<m; j++) {  
            fprintf(f, ", %g", vals[i][j]);  
        }  
        fprintf(f, "\n");  
    }  
    fclose(f);  
}
```

2. (4 puntos) El método de Newton-Raphson es un método numérico para resolver ecuaciones algebraicas no-lineales. Este método está basado en el hecho de que las soluciones de la ecuación  $h(x) = g(x)$  son las raíces de la función  $f(x) = h(x) - g(x)$ . Para obtener numéricamente estas raíces se utiliza el algoritmo iterativo de Newton-Raphson:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

donde  $f'(x_n)$  es la derivada de la función evaluada en el punto  $x_n$ .

Partiendo de una semilla inicial  $x_0$  la sucesión convergerá (no siempre, dependerá de las características de la ecuación y de la semilla de partida) a una de las raíces de la función  $f(x)$ . En general el algoritmo concluye cuando el error relativo entre dos aproximaciones sucesivas es menor que una cierta tolerancia fijada, que llamaremos  $E$ , que se considera como el error relativo de la aproximación:

$$\frac{|x_{n+1} - x_n|}{|x_{n+1}|} < E$$

El objetivo de este ejercicio es escribir un programa que calcule las soluciones reales de la ecuación  $e^x = x^3$ . La función correspondiente cuyas raíces queremos obtener, y su derivada deberán ser definidas en dos funciones. Los valores de la semilla y del error serán introducidos por línea de comandos (como argumentos de la función “main”) cuando se ejecute el programa. El programa deberá imprimir en pantalla el valor aproximado de la raíz obtenida con esa tolerancia.

# 2017-S1.pdf



**Thegoodking**



**Física Computacional I**



**1º Grado en Física**



**Facultad de Ciencias  
Universidad Nacional de Educación a Distancia**

**WUOLAH + BBVA**

Hazte **cliente de BBVA y...**  
**ahórrate 6 meses**  
**de suscripción**

**BOOM**

**1/6**

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

**NETFLIX**

**Spotify**

**HBOmax**

**Disney+**

**PlayStation Plus**

**DAZN**

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

**Abre tu cuenta**



Hazte cliente de BBVA y ...

**ahórrate 6 meses de suscripción**

**WUOLAH**  
+ BBVA

**NETFLIX**

**Spotify**

**HBOmax**

**Disney+**

**PlayStation.Plus**

**DAZN**

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

**1/6**

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

### Lea atentamente estas instrucciones antes de comenzar:

- Responda a cada una de las cuestiones prácticas.
- Es necesario aprobar este examen (5 puntos) para aprobar la asignatura.
- No es suficiente este examen para aprobar la asignatura: se deben aprobar los dos trabajos de la parte práctica.

## Sistemas de álgebra computacional: Maxima

1. (1 punto) Explique qué se obtiene y qué sale por pantalla después de evaluar este código:

`a : tan(pi)$`

**Respuesta:** Si la variable pi no ha sido definida (recuérdese que el número  $\pi$  se representa en máxima por %pi), el lado derecho del signo de asignación (:) se dejará sin evaluar, como tan(pi) y esta expresión será asignada a la variable a. Como la instrucción finaliza con \$, no se mostrará salida por consola.

2. (1 punto) Después de evaluar este código

`f(x):=block([a],a:3,x^a);`

¿cuál será el valor de la variable a?

**Respuesta:** Este código define una función, por lo tanto, en tanto no se evalúe, no afectará al valor de ninguna variable. De todos modos, aún si se ejecutase como

`f(x);`

la variable a sería una variable local (se incluye en el corchete que va como primer argumento de block()), con valor sólo dentro de la función, por lo que al ejecutar ésta, a seguirá valiendo lo que valía antes de ejecutarla.

3. (2 puntos) La identidad  $e^{i\pi} + 1 = 0$ , debida a Euler, contiene todos los números importantes de las matemáticas. Escriba una instrucción condicional en Maxima que diga 'Euler tenía razón', en caso de que la igualdad anterior sea cierta, o 'Euler estaba equivocado', en caso de que, para Maxima, no sea así. **Nota:** para emitir los mensajes, utilice la función disp('cadena').

**Respuesta:** En Maxima las función condicional es de la forma

`if condicion then consecuencia else alternativa`

así que escribiríamos

`if exp(%i*pi)+1=0 then disp("Euler tenia razon") else disp("Euler estaba equivocado")$`



**WUOLAH**  
+ BBVA

**WUOLAH**

## Programación en C

1. (6 puntos) El objetivo de este ejercicio es diseñar un programa que calcule el centro de masas de un sistema discreto de partículas.
  - El programa debe estar compuesto de una única función “main”.
  - Supondremos que los datos de las  $N$  partículas se encuentran en un archivo con cuatro columnas: las tres primeras columnas corresponden a las coordenadas  $(x_i, y_i, z_i)$  de cada partícula  $i$ , mientras que la cuarta columna corresponde a la masa  $m_i$ . Aunque no conocemos exactamente el número de partículas que componen el sistema, sabemos que es menor que 1000. El nombre del archivo que contiene los datos deberá ser introducido por línea de comandos (como argumento de la función “main”) cuando se ejecute el programa.
  - Para leer y almacenar los datos de las partículas, se deberá definir una estructura denominada “Punto” que representa una partícula con su posición y masa. Después, en la función “main” de deberá declarar un “array” de “Puntos”, es decir, un array de estas estructuras, para almacenar en él la información de todo el sistema de partículas.
  - El programa debe escribir en pantalla la posición del centro de masas del sistema

$$\mathbf{r}_{CM} = \frac{\sum_{i=1}^N m_i \mathbf{r}_i}{\sum_{i=1}^N m_i}$$

```
Solución: #include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 1000

struct punto {
    double x, y, z, masa;
};

typedef struct punto Punto;

int main(int argc, char** argv) {
    int i, n, npts;
    double x, y, z, masa, masaCM;
    Punto P[N];
    Punto PCM;
    FILE *fin;

    fin=fopen(argv[1], "r");
    npts=0;

    do {
        n=fscanf(fin, "%lf_%lf_%lf_%lf", &x, &y, &z, &masa);
        if (n==4) {
            P[npts].x=x;
            P[npts].y=y;
            P[npts].z=z;
            P[npts].masa=masa;
            npts++;
        }
    } while (n==4);

    // Cálculo del centro de masas
    for (i=0; i<npts; i++) {
        masaCM += P[i].masa;
        x += P[i].x * P[i].masa;
        y += P[i].y * P[i].masa;
        z += P[i].z * P[i].masa;
    }
    masaCM /= npts;
    x /= masaCM;
    y /= masaCM;
    z /= masaCM;

    printf("Centro de masas: (%f, %f, %f)\n", x, y, z);
    return 0;
}
```

```

    }
} while (n==4);

fclose ( fin );

PCM.x=0;
PCM.y=0;
PCM.z=0;
masaCM=0;

for ( i=0; i<npts; i++) {
    PCM.x+=(P[ i ].masa)*(P[ i ].x);
    PCM.y+=(P[ i ].masa)*(P[ i ].y);
    PCM.z+=(P[ i ].masa)*(P[ i ].z);
    masaCM+=P[ i ].masa;
}

printf ("La_posicion_del_Centro_de_Masas_es_(%g,_%g,_%g)\n",
        PCM.x/masaCM, PCM.y/masaCM, PCM.z/masaCM);

return 0;
}

```

# 2017-S2.pdf



**Thegoodking**



**Física Computacional I**



**1º Grado en Física**



**Facultad de Ciencias  
Universidad Nacional de Educación a Distancia**

**WUOLAH + BBVA**

Hazte **cliente de BBVA y...**  
**ahórrate 6 meses**  
**de suscripción**

**BOOM**

**1/6**

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

**NETFLIX**

**Spotify**

**HBOmax**

**Disney+**

**PlayStation Plus**

**DAZN**

Promoción sólo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

**Abre tu cuenta**





# Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH  
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

## Lea atentamente estas instrucciones antes de comenzar:

- Responda a cada una de las cuestiones prácticas.
- Es necesario aprobar este examen (5 puntos) para aprobar la asignatura.
- No es suficiente este examen para aprobar la asignatura: se deben aprobar los dos trabajos de la parte práctica.

## Sistemas de álgebra computacional: Maxima

1. (1 punto) ¿Qué caracteres empleamos en Maxima para definir una lista?

**Respuesta:** Se emplean "[", "]" y ",". Los dos primeros indican el comienzo y final de lista, respectivamente, mientras que la coma se emplea para separar los elementos de la lista.

2. (1 punto) Explique qué se obtiene y qué sale por pantalla después de evaluar este código:

```
a : tan(%pi)$  
b : a * c^2;
```

**Respuesta:** La primera instrucción asigna a la variable a el valor 0, ya que éste es el valor de la tangente de  $\pi$ ; esto es, evalúa el lado derecho del operador de asignación y asigna ese valor a la variable que figura al lado izquierdo. Como la instrucción está terminada por un \$, no produce ninguna salida en la consola.

La segunda instrucción asigna a la variable b el valor resultante de evaluar  $a \cdot c^2$ . Dado que en la instrucción anterior se asignó a la variable a el valor 0, el resultado será 0. Ahora, como la instrucción finaliza con un ;, se mostrará el valor en la consola.

3. (2 puntos) Escriba una función en Maxima que reciba como argumentos una lista de coeficientes (como p.ej.  $[a_0, a_1, a_2, \dots, a_n]$ ) y una variable (como p.ej.  $x$ ) y que construya a partir de este input el polinomio en la variable dada

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

**Respuesta:** Llamemos a la función Pa(as) donde as es la lista de coeficientes; el número de coeficientes (que es el grado del polinomio más uno, vale length(as)). La forma más sencilla de calcular el polinomio es mediante la función sum() aplicada a los monomios  $a_i x^i$ :

```
Pa(as):=sum( as[i]*x^i, i, 0, length(as)-1 );
```

## Programación en C

1. (2 puntos) Escribir una función que, dada una lista de números complejos en la forma de un array de estructuras

```
struct complexf {  
    float re, im;  
};
```

WUOLAH



devuelva el número complejo resultante de multiplicar todos ellos.

**Solución:**

```
struct complexf complex_prod(int N, struct complexf x[]) {
    int i;
    float aux;
    struct complexf p;
    p.re=1.0;
    p.im=0.0;
    for(i=0; i < N; i++) {
        aux=p.re;
        p.re=p.re*x[i].re-p.im*x[i].im;
        p.im=aux*x[i].im+p.im*x[i].re;
    }
    return p;
}
```

2. (4 puntos) Supongamos que tenemos un archivo de datos "datos.dat" con  $N$  pares de puntos que representan el comportamiento de una función  $f(x)$  en un intervalo dado. La primera columna del archivo corresponde a los valores  $x_i$  (que están ordenados de menor a mayor) mientras que la segunda muestra los valores de la función  $f(x_i)$ .

Escribir un programa que lea esos puntos y obtenga los valores de  $x$  en los que la función muestra los mínimos y máximos relativos dentro del intervalo. El valor de  $N$  debe ser introducido por línea de comandos como argumento de la función "main" cuando se ejecuta el programa. El programa debe mostrar en pantalla el resultado del siguiente modo (es un ejemplo):

'La función tiene tres mínimos relativos en los puntos  $x = 0.1, 1.2, 4.5$ '

'La función tiene dos máximos relativos en los puntos  $x = 0.9, 2.9$ '

Del mismo modo el programa también deberá indicar si la función no tiene mínimos ni máximos relativos.

**Solución:**

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int N, i, nmin, nmax;
    N=atoi(argv[1]);
    FILE *fin;
    double x[N], y[N], xmin[N], xmax[N];

    fin=fopen("datos.dat", "r");
    for (i=0; i<N; i++) {
        fscanf(fin, "%f_%f", &x[i], &y[i]);
    }
    fclose(fin);

    nmin=nmax=0;

    for(i=1; i<N-1; i++) {
        if (y[i-1]>y[i] && y[i+1]>y[i]) {
            xmin[nmin]=x[i];
            nmin++;
        }
    }
}
```

```

    }
    if (y[i-1]<y[i] && y[i+1]<y[i]) {
        xmax[nmax]=x[i];
        nmax++;
    }
}

if (nmin==0)
    printf("La_funcion_no_tiene_ningun_minimo_relativo\n");
else {
    printf("La_funcion_tiene_%d_minimos_relativos "
           "en_los_puntos_x=", nmin);
    for (i=0; i<nmin; i++)
        printf("%g, ", xmin[i]);
    printf("%n");
}

if (nmax==0)
    printf("La_funcion_no_tiene_ningun_maximo_relativo\n");
else {
    printf("la_funcion_tiene_%d_maximos_relativos "
           "en_los_puntos_x=", nmax);
    for (i=0; i<nmax; i++)
        printf("%g, ", xmax[i]);
}

return 0;
}

```