

J.A. Oteo. Departamento de Física
Teórica (UVEG). [MMF3-B:2007-8]

TEMA 3: Ecuaciones en derivadas parciales *

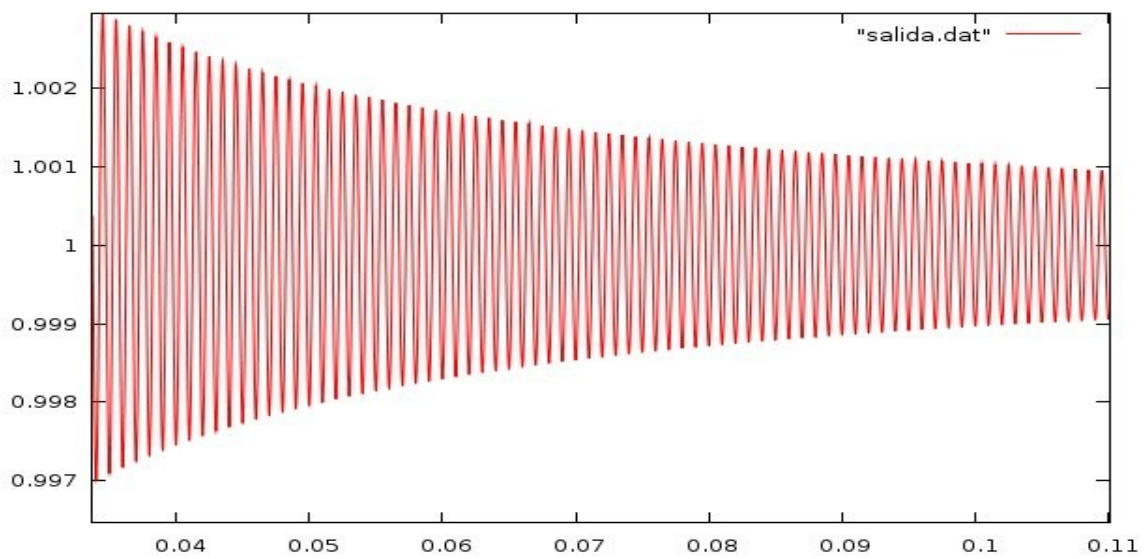
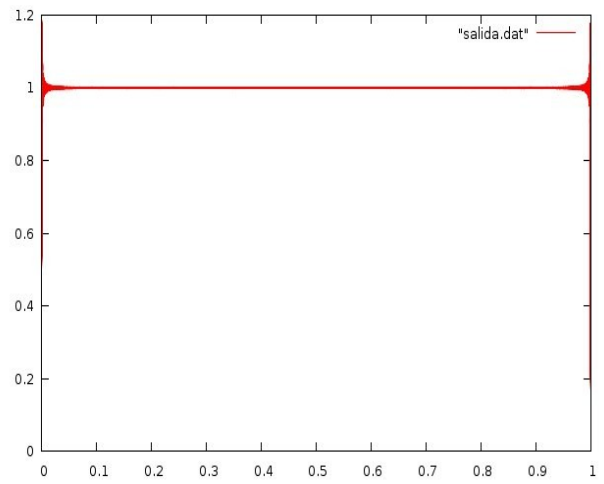
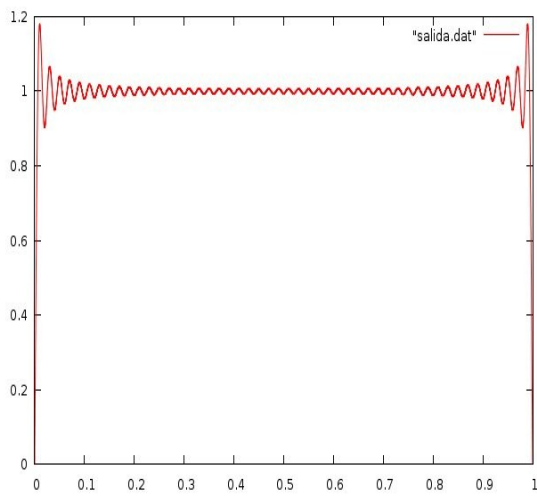
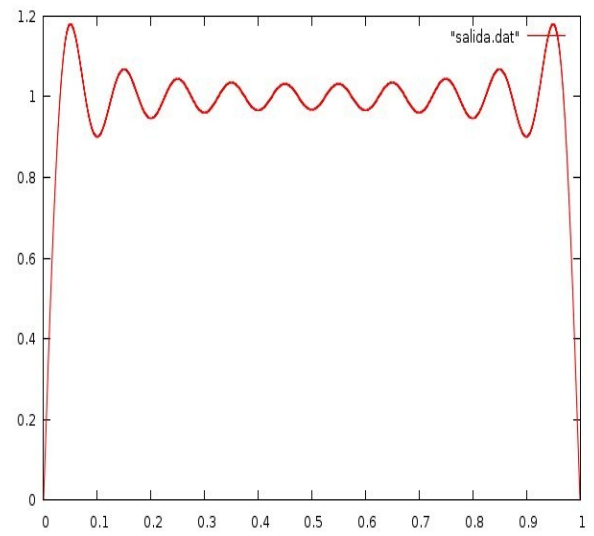
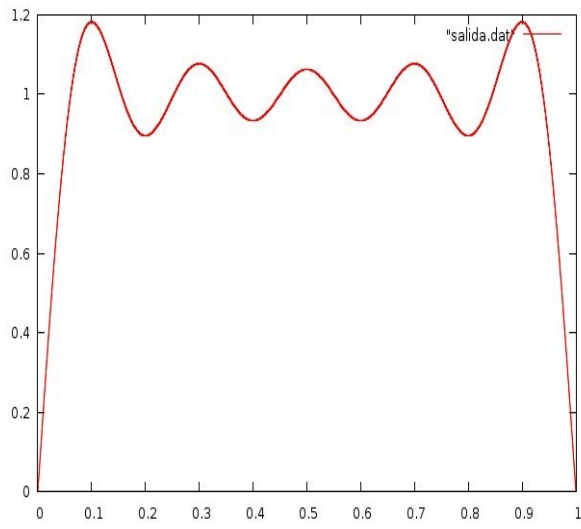
18 de noviembre de 2009

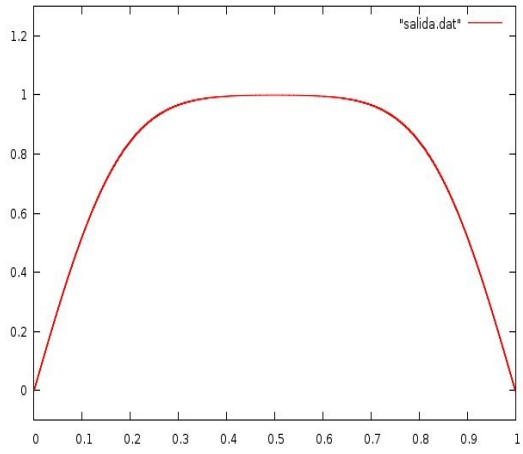
- **Página 2:** Representación gráfica de la descomposición de la función $f(x) = 1$, $0 < x < 1$, en serie de Fourier. Las 4 primeras gráficas son para sumas parciales incrementando el número de términos. La gráfica #5 es un zoom del origen.
- **Página 3:** Representación gráfica de la solución de la EDP de difusión del calor para diferentes instantes.

EDP	$u_t = u_{xx}$
CC	$u(0, t) = 0$ $(0 < t < \infty)$ $u(1, t) = 0$
CI	$u(x, 0) = 1$ $(0 < x < 1)$

- **Página 4:** Programa en C que permite obtener las gráficas anteriores.

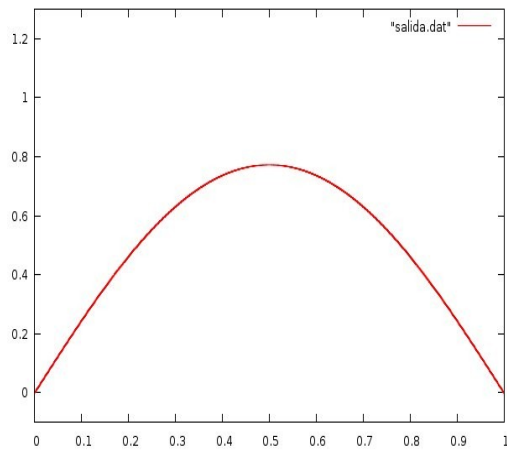
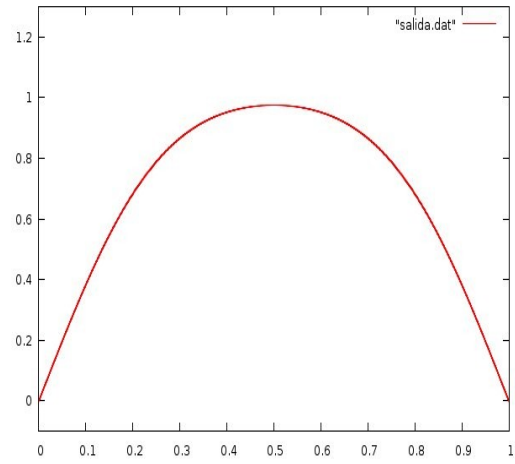
*Material preparado por: Torroba





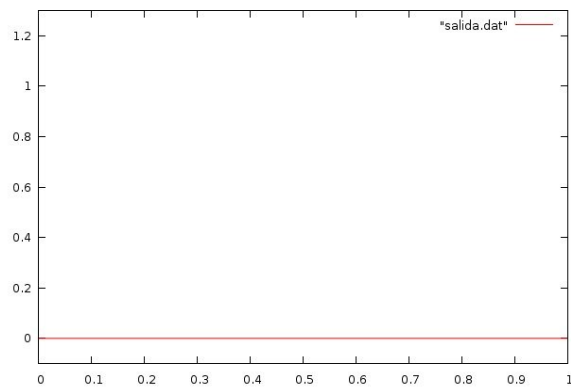
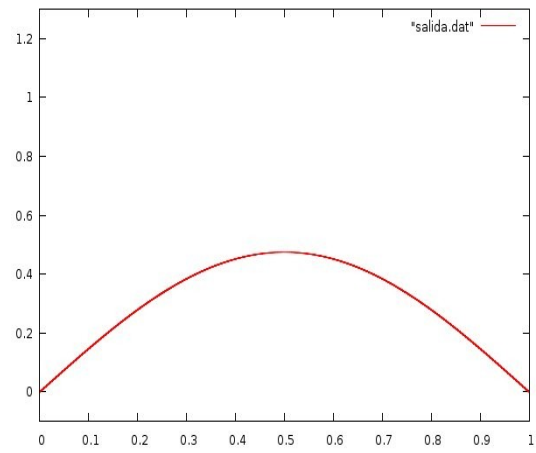
<t=1

t=2>



<t=5

t=10>



<t=1000

/* Representacion de:

EDP : $U_t = A^2 U_{xx}$

CC : $U_x(0,t) = 0$

$U_x(1,t) = 0 \quad 0 \leq t < +\infty$

CI : $U(x,0) = 1 \quad ; 0 < x < 1$

Compilado en linux con "gcc -o flujocalor flujocalor.cc"

Realizado por Torroba*/

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <cmath>
```

```
#define PI 3.141692
```

```
#define T 1000 //tiempo
```

```
//En T=0 se observa serie de Fourier hasta n=N que aproxima f(x)=1
```

```
#define ALFA 0.1
```

```
//Cuanto mas bajo sea ALFA mas lento se acerca a 0
```

```
#define N 50
```

```
#define epsilon 0.00001
```

```
// Cuanto menor sea epsilon mas suave sera la grafica
```

```
// pero aumentará el tamaño del archivo de salida
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    ofstream fout ("salida.dat");
```

```
// Procesar en gnuplot con {plot [[-0.1:1.4]"salida.dat" with lines}
```

```
    float y,k;
```

```
    for(float x=0;x<=1;x=x+epsilon)
```

```
    {
```

```
        y=0;
```

```
        for(int i=0;i<N;i++)
```

```
        {
```

```
            k=-((2*i+1)*PI*ALFA)*((2*i+1)*PI*ALFA)*T;
```

```
            y=y+exp(k)*4/PI*sin((2*i+1)*PI*x)/(2*i+1);
```

```
        }
```

```
        fout << x << " " << y << endl;
```

```
    }
```

```
    return 0;
```

```
}
```