MAC0210 - Exercício-Programa 3

Vinicius Agostini - 4367487

Julho 2020

1 Computando Trabalho

A primeira tarefa do EP é aproximar a quantidade de trabalho exercida estimando a integral $\int_{x_0}^{x_n} F(x)\cos(\theta(x))dx$, primeiro interpolando os valores fornecidos. O método de interpolação escolhido foi o método de Lagrange. Então, devemos usar esses valores para estimar a integral usando a regra do trapézio composto e a regra de Simpson composto.

Para fazer a interpolação, decidi dividir o intervalo $[0, 30]$ em $n-1$ intervalos e calculei os n pontos resultantes através de uma função que executa a interpolação pelo método de Lagrange.

Resultados:

	$n = 7$	$n = 19$	$n = 61$	$n = 100$	$n = 500$
Trapezoidal	119.089250	117.347276	117.151007	117.138741	117.131902
Simpson	117.127167	117.069079	117.131612	117.069079	117.123476

Inicialmente achei que deveria utilizar valores de n tal que $n = 6k + 1$ para obter melhores aproximações já que dessa forma garantiríamos que os pontos originais estariam presentes no conjunto de pontos interpolados e que isso seria uma coisa boa, porém, conforme n aumenta, isso não se mostrou um fator tão importante. No caso da regra de Simpson, o erro relacionado a utilizar uma quantidade par ou ímpar de intervalos também se torna negligenciável conforme n aumenta.

Uso do programa:
`gcc trabalho.c -o trabalho`
`./trabalho a b n`, para calcular $\int_a^b F(x)\cos(\theta(x))dx$ com n pontos, interpolando a partir da tabela do enunciado.
Exemplo:
`./trabalho 0 30 19` (1 ponto interpolado entre cada ponto dado).

2 Integração por Monte Carlo

2.1 Integrais unidimensionais

Nesta seção do EP devemos calcular uma aproximação para integrais unidimensionais como $I = \int_0^1 g(x)dx$ escolhendo pontos aleatórios U_i no intervalo $[0, 1]$ e calculando $\hat{I} = \sum_{i=1}^{n} \frac{g(U_i)}{n}$, que, pela Lei Forte dos Grandes Números, se aproxima de I quando $n \to \infty$. No caso de integrais em intervalos arbitrários $[a, b]$ é possível realizar uma troca de variável para resolver uma integral no intervalo $[0, 1]$ que tenha resultado equivalente.

Integrais a serem resolvidas e trocas de variável utilizadas:

$$\int_0^1 \sin(x)dx \tag{1}$$

$$\int_3^7 x^3dx = \int_0^1 4\cdot(4y+3)^3dy, \qquad \text{com } y = \frac{x-3}{4} \tag{2}$$

$$\int_0^\infty e^{-x}dx = \int_0^1 \frac{e^{1-\frac{1}{y}}}{y^2}dy, \qquad \text{com } y = \frac{1}{1+x} \tag{3}$$

Resultados:

	10^4	10^5	10^6	10^7	Valor esperado
$\int_0^1 \sin(x)dx$	0.456837	0.459729	0.459457	0.459850	$\approx$ 0.45970
$\int_0^1 4\cdot(4y+3)^3dy$	575.820989	580.472969	580.176466	580.232532	580
$\int_0^1 \frac{e^{1-\frac{1}{y}}}{y^2}dy$	0.996641	0.999422	0.999720	1.000108	1

Para esta parte foi implementada uma função única para o método de Monte Carlo que recebe uma função como parâmetro com assinatura **monte_carlo(int iters, function func)** onde **iters** é a quantidade de iterações a serem realizadas e **func** uma função que recebe um double e retorna um double. A definição do tipo **function** foi criada a partir do comando **typedef double (*function)(double)**.

Uso do programa:
```
gcc monte_carlo_1d.c -o monte_carlo_1d
./monte_carlo_1d iters
```
Exemplo:
```
./monte_carlo_1d 1000
```

2.2 Integrais multidimensionais

Aqui o objetivo é encontrar uma estimativa para o valor de π. O método utilizado é considerar a parte do círculo unitario que está no primeiro quadrante e calcular sua área por Monte Carlo da seguinte forma, como definido no enunciado:

$$g(x,y) = \begin{cases} 1, & \text{se } x^2+y^2 \leq 1 \\ 0, & \text{caso contrário} \end{cases}$$

$$\pi = 4\cdot \int_0^1 \int_0^1 g(x,y)dxdy$$

Resultados:

	$\approx \pi$
$n = 10^4$	3.142400
$n = 10^5$	3.144680
$n = 10^6$	3.140356
$n = 10^7$	3.141540

Uso do programa:
```
gcc pi.c -o pi
./pi iters
```
Exemplo:
```
./pi 1000
```