Física Computacional 4
Integrais numéricos

1. Aplicando c++ aos integrais numéricos
   a. Integral rectangular
   b. A regra do trapézio
   c. Integração de Simpson
   d. Pontos de Gauss
   e. Metropolis Monte-Carlo

bicudo@tecnico.ulisboa.pt

Aplicando aos integrais numéricos

- Aplicando c++ aos integrais numéricos
  - Para calcularmos um integral numéricamente regressamos às origens da integração e aproximamos o integral,

$$\text{integral} = \int_a^b f(x)\,dx$$

  - por uma soma numérica, ou série,

$$\text{integral} = \sum_{i=0,n-1} f(x_i)\,\Delta_i$$

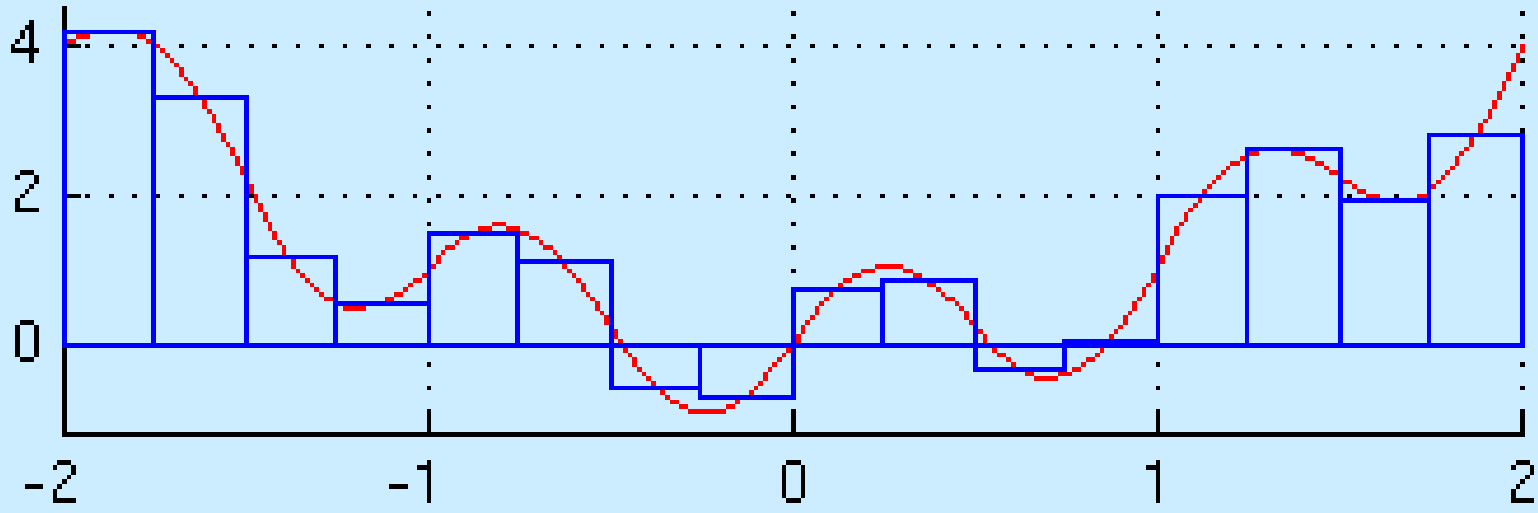
  - que, no limite em que n → ∞, deve convergir para o valor matemático do integral,
  - ora os ciclos em programação (em c++ for, while, do) são ideais para repetir un grande número de vezes cálculos semelhantes, logo para computar integrais numéricos.

- A regra do rectângulo é a mais simples de integração numérica.
  - http://en.wikipedia.org/wiki/Numerical_integration
  - notamos que aqui não iremos dar os fundamentos da integração numérica, mas apenas mostrar algumas regras ou técnicas numéricas, afim de podermos desenvolver códigos numéricos,
  - sendo que a regra do rectângulo consiste em dividir o domínio de integração [a,b] em subintervalos de largura igual e tomar o valor da função no centro de cada subintervalo indexado por i,

$$integral = \sum_{i=0}^{n-1} f(x_i)\Delta$$
$$\Delta = (b-a)/n$$
$$x_i = a + (i+1/2)\Delta$$

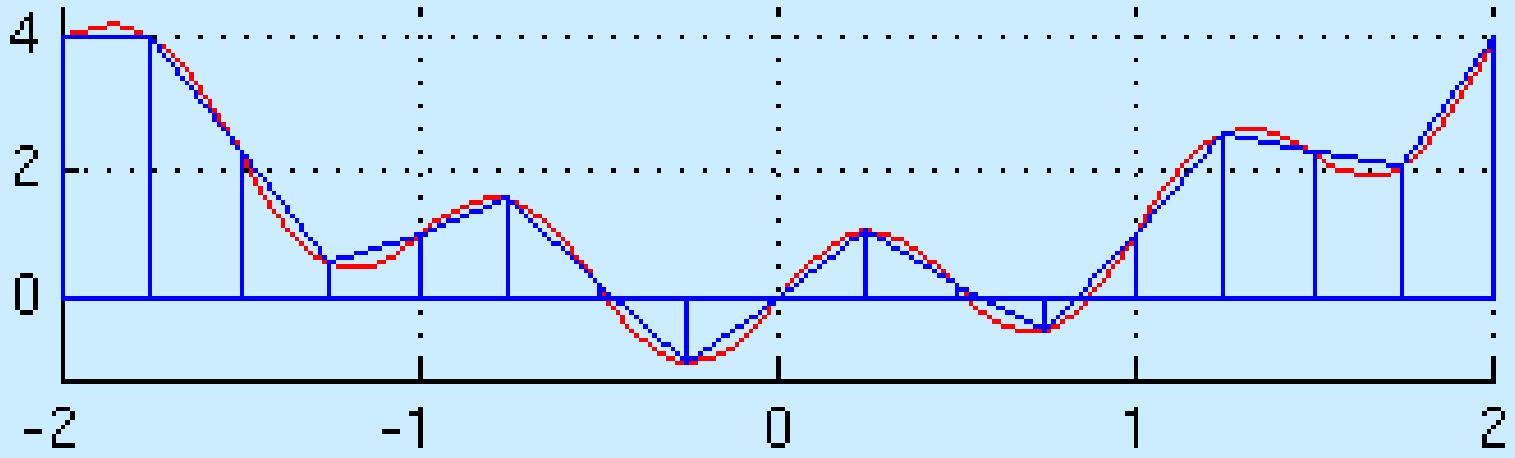
Aplicando aos integrais numéricos

- gráficamente, a regra do rectânculo consiste em se aproximar o integral por uma soma de rectângulos



Aplicando aos integrais numéricos

- A regra do trapézio é um melhoramento da regra anterior,
  - http://en.wikipedia.org/wiki/Trapezoidal_rule
  - obviamente podemos melhorar a regra do rectângulo com uma melhor interpolação da função a calcular,
  - gráficamente o método do trapézio consiste em ligarmos os pontos da curva f(x) com segmentos de recta,



Aplicando aos integrais numéricos

- notamos que a área dum trapézio é dada por
- base x(h1 +h2)/2
- de forma que se somarmos sobre os (n+1) pequenos trapézios, passamos a incluir os pontos inicial e final no sumatório, e somamos duas vezes sobre as alturas intermédias f(xi), obtendo

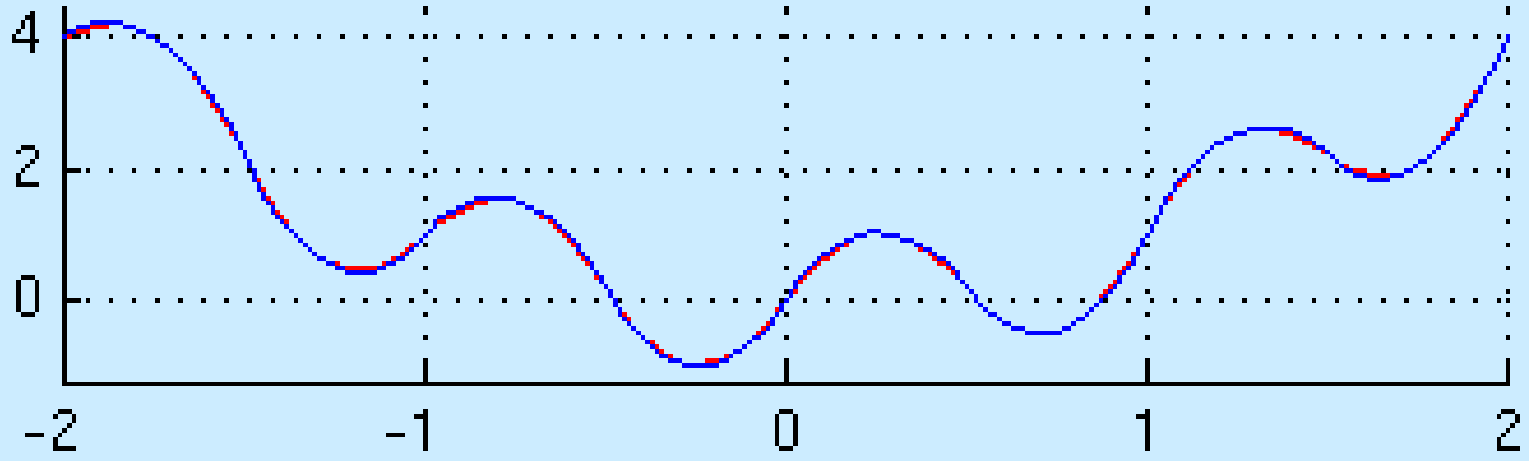
$$integral = [f(a) + f(b)]\Delta/2 + \sum_{i=1}^{n-1} f(x_i)\Delta$$

$$\Delta = (b-a)/n$$

$$x_i = a + i\Delta$$

Aplicando aos integrais numéricos

- A regra de Simpson consiste numa interpolação ainda melhor que a do trapézio,
  - http://en.wikipedia.org/wiki/Simpson's_rule



Aplicando aos integrais numéricos

- que considera a média de 2/3 da regra do rectângulo e de 1/3 da regra do trapézio, ou seja considera o dobro de pontos das regras anteriores,

$$\text{integral} = (2/3)\,\text{integral}_{\text{rectângulo}} + (1/3)\,\text{integral}_{\text{trapézio}}$$

$$= \left[f(a) + f(b) \right] \Delta / 6 + \sum_{i=1,n-1} f(x_i)\,\Delta / 3$$

$$+ \sum_{i=0,n-1} f(y_i)\,\Delta\,(2/3)\,,$$

$$\Delta = (b - a)/n\,,$$

$$x_i = i\,\Delta + a\,,$$

$$y_i = (i + 1/2)\,\Delta + a\,.$$

Aplicando aos integrais numéricos

- Uma técnica aperfeiçoada analícitamente que em certos casos tem uma enorme precisão é a técnica da quadratura de Gauss,
  - http://en.wikipedia.org/wiki/Gaussian_quadrature
  - onde a quadratura é um tema que já interessava a antiguidade.
  - Gauss percebeu que é possível calcular exactamente os integrais de quaisquer polinónimos de graun n num intervalo, sempre com os mesmos e apenas os mesmos n pontos de integração

$$\text{integral} = \int_{-1}^{1} f(x)\,dx = \sum_{i=0,n-1} f(x_i)\,\Delta\,,$$

$$\Delta = (2)/n\,,$$

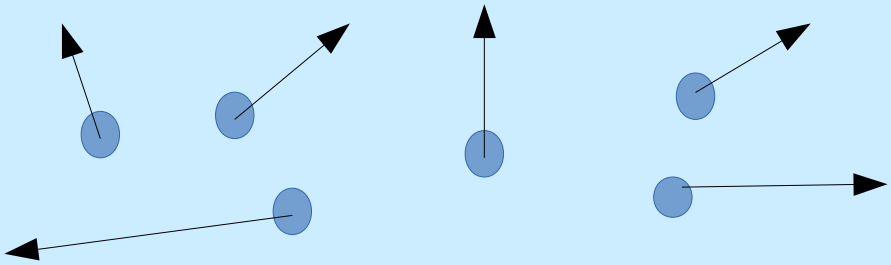
$$x_i = \text{pontos de Gauss}\,,$$

  - em seguida esta técnica foi aperfeiçoada por outros autores, levando às quadraturas de Gauss-Kronrod e de Gauss-Lobatto.
  - São técnicas muito eficientes, pois basta ter uma tabela de pontos de gauss x_i para calcular integrais. Mas não iremos aqui detalhar os x_i.

Aplicando aos integrais numéricos

Aplicando aos integrais numéricos

- Um algoritmo com uma filosofia completamente diferente, por se basear não na determinação analítica dos pontos de integração, mas sim na sua geração aleatória, é o algoritmo de Metropolis Monte Carlo.
  - Notamos que em geral qualquer processo aleatório é chamado de Monte Carlo devido à famosa roleta do Casino.
  - Este método é poderosíssimo para integrais com muitíssimas dimensões como na distribuição de Maxwell-Boltzmann da física estatística, ou para o integral de caminho em teoria quântica do campo,
  - http://xbeams.chem.yale.edu/~batista/vaa/node42.html
  - http://en.wikipedia.org/wiki/Metropolis-Hastings_algorithm
  -



~NA moléculas com posições e velocidades diferentes

Aplicando aos integrais numéricos

- Este método, que utiliza uma cadeia aleatória de Markov, aplica-se quando temos uma distribuição de probabilidade (contínua), e quando calculamos um integral do tipo,

$$\text{integral} = \int f(x)\,\rho(x)\,dx$$

- que pretendemos representar por um conjunto discreto de pontos, cuja densidade seja dada por $\rho(x)$, número de pontos
- **$\rho(x)$ = número de pontos / unidade de comprimento**

- onde $\rho(x)$ é uma distribuição de probabilidade (contínua) e onde o integral tanto pode ser definido como indefinido.
- A questão fundamental que se pretende resolver com o método de Metropolis Monte-Carlo é a de representar a densidade de probabilidade $\rho(x)$ com uma núvem de pontos, com a mesma concentração distribuida no espaço dos x que a densidade $\rho(x)$.
- Com o algoritmo geramos um conjunto de n configurações x_0, x_1, ... x_{n-1} do sistema tal que, a probabilidade dum intervalo *int* é dada por $\rho(int) = n(int) / n$
- sendo n o número de pontos x_i pertencentes ao intervalo *int*.

Aplicando aos integrais numéricos

As configurações x i são geradas da seguinte forma,

1. é gerado aleatóriamente um ponto ξ do domínio de integração
2. dando um salto com um passo aleatório, gerado aleatóriamente num intervalo [-r,r] (random walk) geramos um segundo ponto ξ' do domínio de integração dado por ξ mais o passo aleatório,
3. em seguida comparamos as probabilidade dos pontos, $\rho(\xi)$ e $\rho(\xi')$,
4. notando que se pretendessemos apenas determinar o ponto mais provável escoleríamos de entre ξ e ξ' o que tivesse probabilidade mais alta, mas o que pretendemos é um conjunto de n pontos que reproduzam a densidade $\rho(x)$, geramos aleatóriamente um número al num intervalo [0,1] e com base no valor de al,

   seleccionamos ξ' *se* $\rho(\xi')/\rho(\xi) > al$

   senão mantemos o ponto ξ
5. repetimos este processo de escolha um determinado número de vezes m, e com esta repetição ficamos com uma cadeia de pontos denominada cadeia de Markov, ao fim do qual atribuimos à nossa primeira configuração x0 o valor finalmente seleccionado.

6. Notamos que o objectivo das m repetições da cadeia de Markov é o pronto final x0 ser independente da escolha do primeiro de todos os pontos ξ.
7.Agora partimos deste ponto x0 , e repetimos os passos dados nos pontos 2. a 5. vamos gerar um novo ponto x1 e depois ainda um outro ponto x2 e dai em diante até termos todas as nossas n configuraçoes geradas.
8.Notamos que o passo aleatório r deve ser tal que o rácio de aceitações do ponto 4. deve ser (para gausseanas a 1 dimensão) da ordem de 0.5.
9.Assim, o imput do nosso código deve ser,

- intervalo onde se gera o ponto inicial ξ
- passo aleatório r
- número de passos intermédios m para termos uma cadeia de Markov
- número de configurações n

Finalmente o valor numérico do integral é simplesmente,

$$integral = \int f(x)\rho(x)dx = \sum_{i=0}^{n-1} f(x_i)/n$$

Exemplos, ilustrando o caminho de Markov e a forma como as distribuições de probabilidades escolhidas representam $\rho(r)$