MAC 210 – Laboratório de Métodos Numéricos

Primeiro Semestre de 2017

Terceiro Exercício Programa: Interpolação e diferenciação numérica

Sejam n_x e n_y inteiros positivos e $a_x < b_x$ e $a_y < b_y$ reais dados. Definamos $h_x = (b_x - a_x)/n_x$, $h_y = (b_y - a_y)/n_y$ e

$$\begin{aligned} x_i &= a_x + i\, h_x, \text{ para } i = 0, \ldots, n_x, \\ y_j &= a_y + j\, h_y, \text{ para } j = 0, \ldots, n_y. \end{aligned}$$

Desta forma temos que os pontos (x_i, y_j) para $i = 0, \ldots, n_x$ e $j = 0, \ldots, n_y$ formam uma malha com $(n_x + 1) \times (n_y + 1)$ pontos na região $[a_x, b_x] \times [a_y, b_y] \subset \mathbb{R}^2$.

No segundo exercício programa foi estudada a interpolação bicúbica por partes de uma função $f : \mathbb{R}^2 \to \mathbb{R}$. O objetivo foi construir uma função $v : [a_x, b_x] \times [a_y, b_y] \to \mathbb{R}$ que interpolasse f nos $(n_x + 1) \times (n_y + 1)$ pontos definidos acima. Na construção da v supusemos que conhecíamos não apenas os valores de f nos pontos da malha como também os valores das derivadas parciais primeiras e segundas.

Este exercício programa consiste em estender as ideias do exercício programa anterior para o caso em que os valores das derivadas da f não estão disponíveis. Para tanto, os valores das derivadas parciais primeiras e segundas da f nos pontos da malha deverão ser aproximados.

1 O que deve ser feito

O **primeiro passo** consiste em deduzir aproximações $O(h^2)$ para $\partial_x f$, $\partial_y f$ e $\partial^2_{xy} f$ em **todos** os pontos da malha. O relatório deve incluir um passo a passo das deduções com todas as explicações que considere necessárias.

O **segundo passo** é implementar uma função `aproxdf` que recebe n_x, n_y, a_x, b_x, a_y, b_y (parâmetros estes que definem uma malha) e uma matriz com os valores de uma certa f (desconhecida para a função) nos pontos da malha. A função `aproxdf` deve devolver matrizes `dfx`, `dfy` e `d2fxy` com as aproximações de $\partial_x f$, $\partial_y f$ e $\partial^2_{xy} f$ nos pontos da malha.

O **terceiro passo** é implementar uma função `constroiv` que recebe n_x, n_y, a_x, b_x, a_y, b_y (parâmetros estes que definem uma malha) e uma matriz com os valores de uma certa f (desconhecida para a função) nos pontos da malha. A função `constroiv` deve construir a função v (esta função deve usar a função `aproxdf`). Deve também ser implementada uma função `avaliav` que recebe as coordenadas x e y de um ponto $(x, y) \in [a_x, b_x] \times [a_y, b_y]$ e devolve $v(x, y)$.

O **quarto passo** é fazer experimentos com as funções desenvolvidas acima. Os experimentos devem servir para tentar mostrar alguma coisa.

- A primeira coisa que deve ser mostrada é que tudo funciona como deveria e o mais básico é verificar se as aproximações das derivadas de fato aproximam adequadamente as derivadas. Variando h, verifique se o erro teórico $O(h^2)$ verifica-se na prática.
- A segunda coisa é verificar se a interpolação funciona como deveria, quer dizer, se v interpola a f nos pontos da malha. Invente várias funções f. Desenhe f, desenhe v, desenhe $|f - v|$. Observe que nos pontos da malha esta última função vale zero.

- A terceira coisa poderia ser tentar mostrar o que acontece com a v quando variamos a quantidade de pontos na malha. Seria aqui razoável escolher n_x e n_y de forma tal que $h_x = h_y = h$ e fazer experimentos com $h \to 0$. Observe o que acontece com o erro

$$\max_{(x,y)\in[a_x,b_x]\times[a_y,b_y]} \{|f(x,y) - v(x,y)|\}$$

  para diferentes valores de h. Será que os experimentos mostram o comportamento do erro em função de h?

  Bônus: A teoria do caso bivariado não foi abordada em sala de aula. Um bônus neste exercício programa seria, consultando a literatura, descrever o que a teoria fala sobre esse erro e verificar se a prática se comporta conforme descrito na teoria.

- A quarta e última coisa é aplicar o que foi feito à compressão de imagens com perdas. Escolha uma imagem (foto) de sua preferência. Pode ser em preto e branco ou colorida. Faça uma bijeção entre as cores e valores. Isso definirá a sua "função f avaliada numa malha". Se for em preto e branco, você terá uma única função f com a cor (tom de cinza) de cada pixel da imagem. Se for colorida, terá 3 funções f com as quais deverá trabalhar separadamente, cada uma com uma das componentes RGB de cada pixel da imagem.

  Essa imagem que você escolheu é a imagem que desejamos comprimir. Comprimir significa guardar a informação de f (quer dizer, da imagem) para apenas um subconjunto de pontos da malha fina (quer dizer, guardar apenas os valores de f em pontos de uma malha mais "grossa" que esteja contida na malha fina). Tendo feito isso, a imagem original pode ser recuperada interpolando a f nos pontos da malha grossa. A pergunta é: qual é a menor malha grossa que podemos guardar para que a qualidade da imagem recuperada ainda seja "razoável"? Certamente a resposta depende da imagem pois, por exemplo, para uma imagem constante quiçá seja suficiente guardar apenas a informação da imagem nos 4 extremos da imagem. Faça experimentos com imagens diferentes tentando ilustrar a resposta a esta pergunta.

Note que as funções `aproxdf`, `constroiv` e `avaliav` correspondem a uma parte pequena do que deve ser feito neste trabalho. Um relatório bem escrito, completo e detalhado explicando tudo o que foi feito é essencial.