MAC0210 - Exercício Programa 2

Professor: Ernesto G. Birgin
Monitor: Lucas Magno

Arthur Coser Marinho - 7210629
Ludmila Ferreira Vicente e Silva - 7557136

Execução

Para criar as imagens interpoladas, primeiro execute no octave o script

functions

para gerar as malhas $f1.dat$, $f2.dat$, $f3.dat$, $f4.dat$, $f5.dat$ e $f6.dat$.
Em seguida execute no octave o script

ep2 <funcao.dat> <k>

Em que *k* é a taxa de compressão.

Desenvolvimento

Script functions.m:

Dados os inteiros positivos *nx*, *ny*, *ax*, *ay*, *bx*, e *by*, o script *functions.m* cria a malha dos pontos $[xi, yj]$ para sobre a qual iremos interpolar. A partir das seguintes equações:

$xi = ax + i * hx$, para $i = 0, ..., nx$,
$yi = ay + j * hy$, para $j = 0, ..., ny$.

em que

$hx = (bx - ax) / nx$,
$hy = (by - ay) / ny$.

Criamos seis malhas $f1.dat, f2.dat, \ldots, f6.dat$, com $(nx + 1)\, x\, (ny + 1)$ pontos na região $[ax, bx]\, x\, [ay, by]$.

Função constroiv (nx, ny, ax, ay, bx, by, f):

Caso bilinear:

Para fazer a interpolação bilinear é necessário encontrar o valor de *s{i, j}* e para isso precisamos encontrar a matriz dos coeficientes, o que é o mesmo que resolver um sistema linear. Para resolvê-lo de maneira eficiente utilizamos vetorização no octave, que torna a resolução do sistema mais simples e rápida.
Seja *f* a malha que será interpolada e *a*, a matriz dos coeficientes. Para encontrar a matriz a, seguimos o seguinte raciocínio:

Sabemos que

$$s\{i,j\} = [1,\ x - xi\ /\ hx][a00,\ a01;\ a10,\ a11]\ [1,\ y - yj\ /\ hy]$$

e queremos calcular

$$s\{i,\ j+1\},\ s\{i+1,\ j\ \}\ e\ s\{i+1,\ j+1\}$$

como

$$[f\{i,\ j\},\ f\{i,\ j+1\}\ f\{i+1,\ j\}\ f\{i+1,\ j+1\}] = $$
$$= [s\{i,\ j\}\ s\{i,\ j+1\}\ s\{i+1,\ j\}\ s\{i+1,\ j+1\}]$$

então

$$[f\{i,\ j\},\ f\{i,\ j+1\},\ f\{i+1,\ j\},\ f\{i+1,\ j+1\}] = B * [a00,\ a01,\ a10,\ a11]$$

em que a matriz B é uma composição das matrizes $[1,\ x - xi/hx]$ e $[1,\ y - yi/hy]$ e então chegamos ao seguinte resultado

$$a00 = f\{i,\ j)$$
$$a12 = f\{i,\ j+1\} - a11$$
$$a21 = f\{i+1,j\} - a11$$
$$a22 = f\{i+1,j+1\} - a11 - a12 - a21$$

de modo mais fácil, podemos calcular os coeficientes com a seguinte operação matricial:

$$A = B * [f\{i,j\},\ f\{i,j+1\},\ f\{i+1,j\},\ f\{i+1,j+1\}]'$$

e conseguimos encontrar a função v que interpola a função *f*.

Avaliação da função v

$f = x^2 + y^2$

- Verificação de interpolação nos pontos da malha

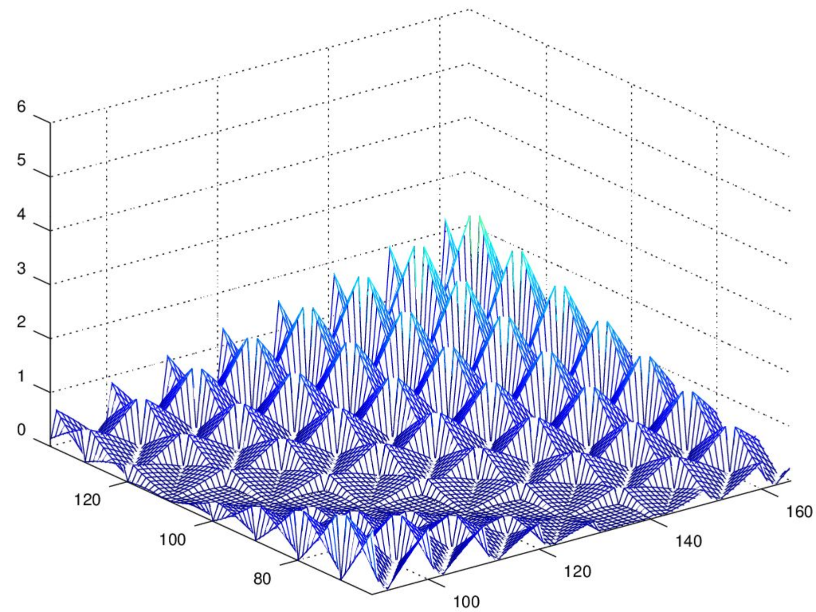


Gráfico de |f - v| para $n_x = n_y = 20$

Interpolando a função em 200 pontos e fazendo o gráfico de |f - v|. é possível observar que nos pontos da malha, a função vale zero.

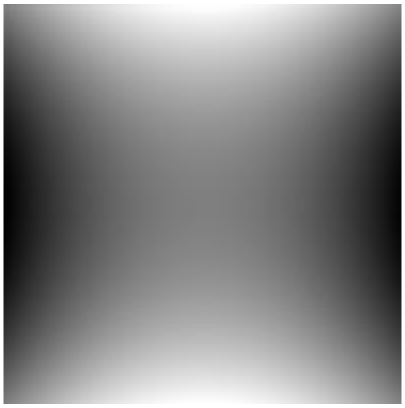
- Variando o número de pontos na malha

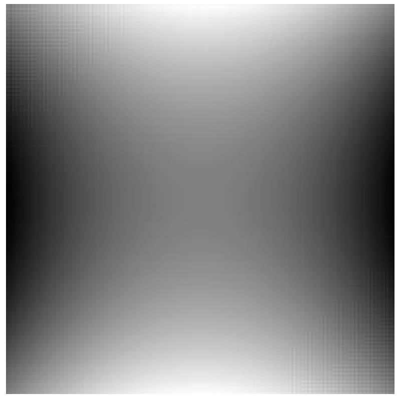
Com $a_x = a_y = -4$ e $b_x = b_y = 4$

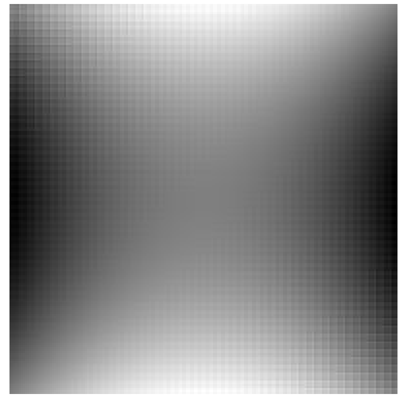
h	0.8	0.4	0.16	0.08
Erro máximo	10.914	5.4576	1.8768	0.63200

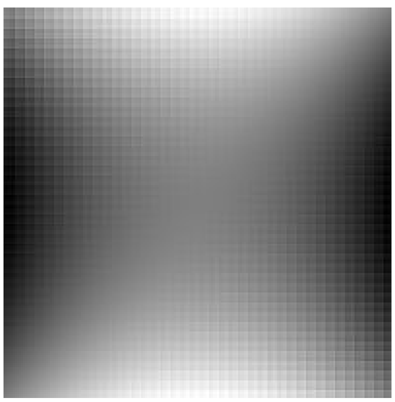
Podemos observar que o erro máximo é diretamente proporcional a h, diminuindo conforme h tende a zero.

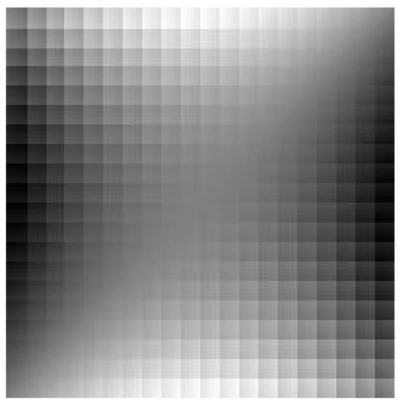
- Compressão de imagens

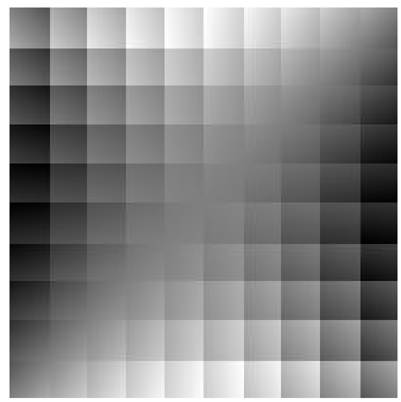

Sem compressão


50% de compressão


75% de compressão


80% de compressão


90% de compressão


95% de compressão

Podemos observar imagens satisfatórias com bem menos pontos na malha em relação à original. Mas claro que para uma compressão alta demais perdemos muita informação.

f = sin(x) + cos(y);

- Verificação de interpolação nos pontos da malha

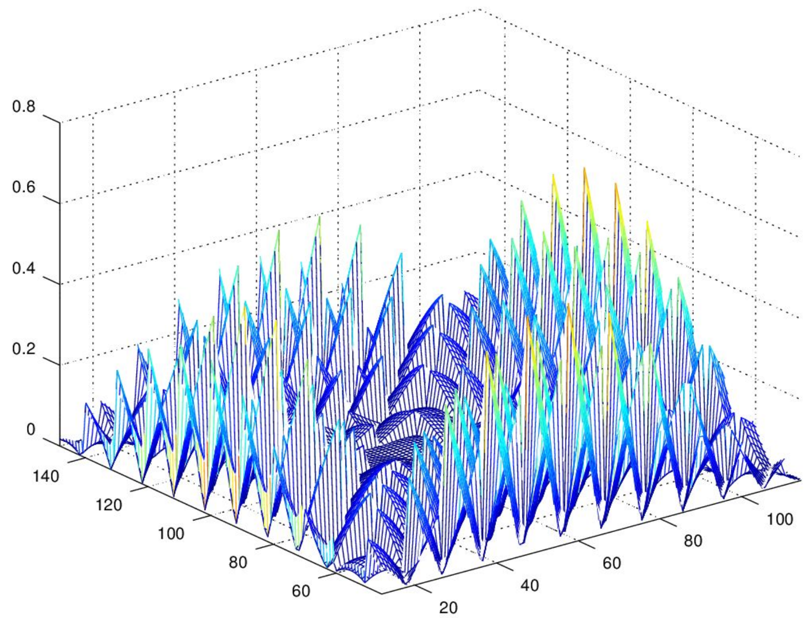


Gráfico de |f - v| para $n_x = n_y = 20$

Interpolando a função em 200 pontos e fazendo o gráfico de |f - v|. é possível observar que nos pontos da malha, a função vale zero.

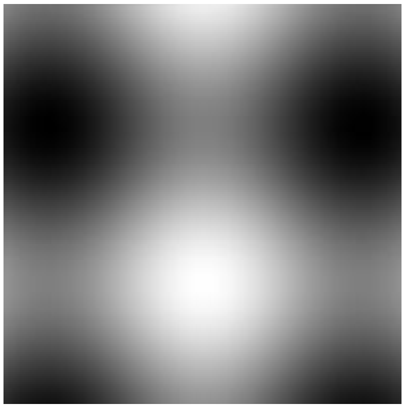
- Variando o número de pontos na malha

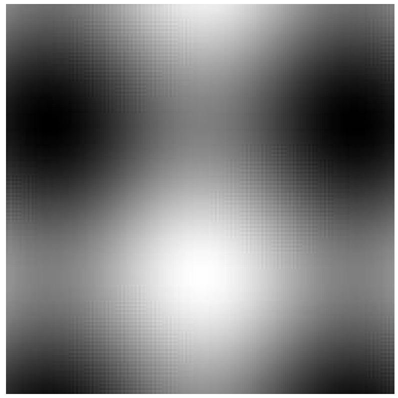
Com $a_x = a_y = -4$ e $b_x = b_y = 4$

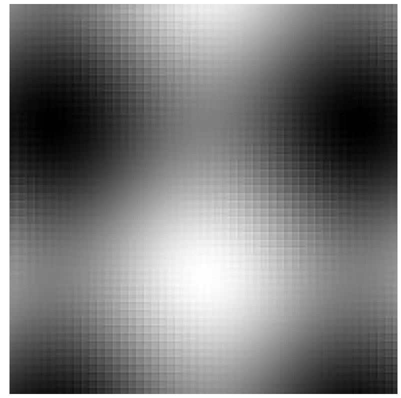
h	0.8	0.4	0.16	0.08
Erro máximo	1.4597	0.77570	0.23965	0.079984

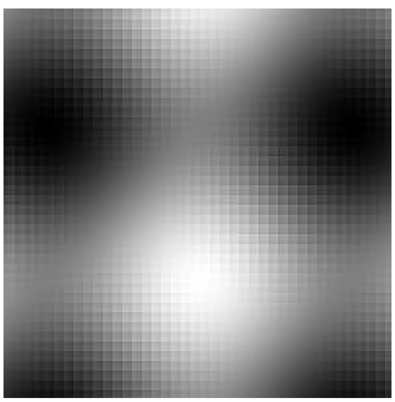
Podemos observar que o erro máximo é diretamente proporcional a h, diminuindo conforme h tende a zero.

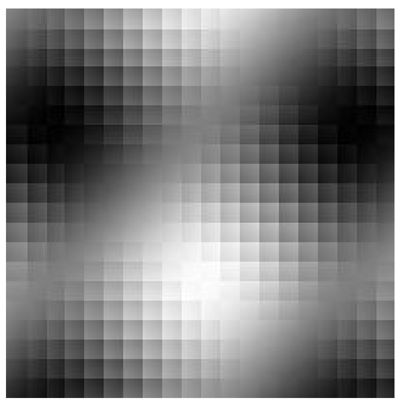
- Compressão de imagens

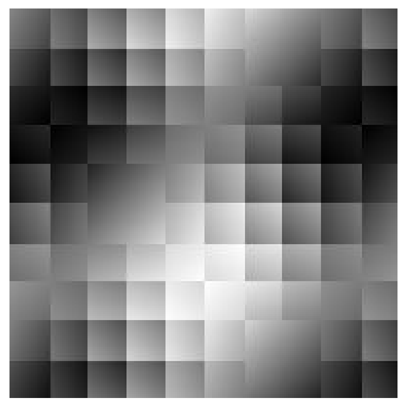

Sem compressão


50% de compressão


75% de compressão


80% de compressão


90% de compressão


95% de compressão

Podemos observar imagens satisfatórias com bem menos pontos na malha em relação à original. Mas claro que para uma compressão alta demais perdemos muita informação.

f = 3 x sin(x) - 4 y cos(x);

- Verificação de interpolação nos pontos da malha

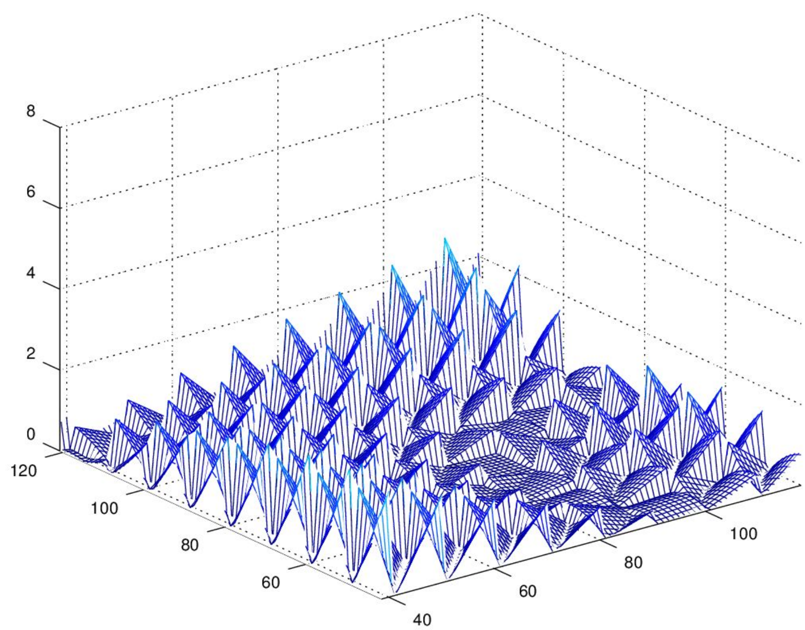


Gráfico de |f - v| para $n_x = n_y = 20$

Interpolando a função em 200 pontos e fazendo o gráfico de |f - v|. é possível observar que nos pontos da malha, a função vale zero.

- Variando o número de pontos na malha

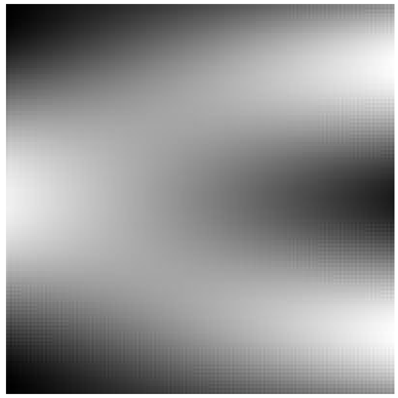
Com $a_x = a_y = -4$ e $b_x = b_y = 4$

h	0.8	0.4	0.16	0.08
Erro máximo	14.485	7.9642	2.8798	0.96749

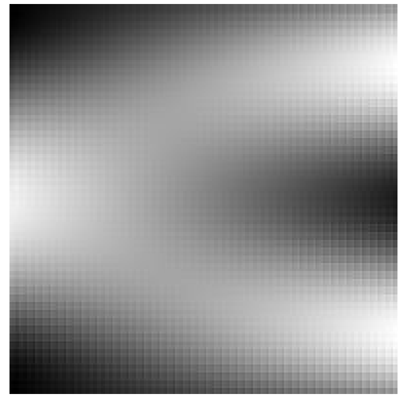
Podemos observar que o erro máximo é diretamente proporcional a h, diminuindo conforme h tende a zero.

- Compressão de imagens

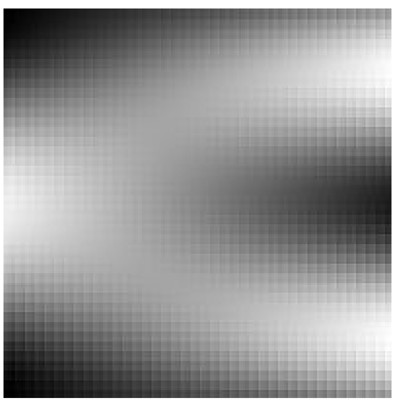
Sem compressão



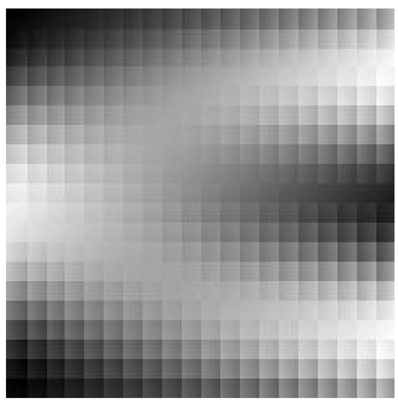
50% de compressão



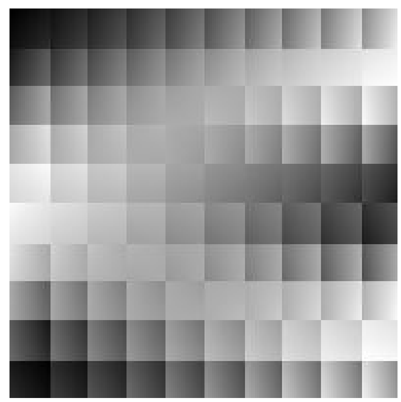
75% de compressão



80% de compressão



90% de compressão



95% de compressão

Podemos observar imagens satisfatórias com bem menos pontos na malha em relação à original. Mas claro que para uma compressão alta demais perdemos muita informação.