UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE SÃO CARLOS

EXERCÍCIO PRATICO II

COMPARAÇÃO DOS MÉTODOS DE RESOLUÇÃO DE SISTEMAS LINEARES

RAUL BORTOLOTTO GHIZZI, 7171220

CAIO BISULLI SANT ANA, 7171290

SÃO CARLOS

2011

Para efeito de comparação foram usados três métodos para se resolverem diversos sistemas lineares do tipo *Ax=b*, onde *A* é uma matriz quadrada n por n, *x* e *b* são vetores n por 1. Os métodos empregados foram, Decomposição L.U, Gauss-Seidel e Gratientes Conjugados. Para se compararem os três métodos foram usadas 100 matrizes de tamanho variando de 10 por 10 até 1000 por 1000, com um almento de tamanho de 10 em 10. Todos os métodos foram implementados no MatLab. Os algoritimos são os seguintes:

**Gauss-Seidel:**

%Programa de Solução de sistemas lineares pelo método de Gauss-Seidel%

%Raul Bortolotto Ghizzi, nº 7171220%

%Caio Bisulli Sant Ana, nº 7171290%

%Turma: Mecatrônica%

for cont=10:10:1000

cont2 = int2str(cont);

arquivo = strcat('matrizes/','Ab',cont2,'.txt');

tam(cont/10,1) = cont;

t1=tic;

fid = fopen(arquivo); %Abre o arquivo de entrada% Ax=b

%M é a matrix A

n = fscanf(fid,'%f',1); %Pega o tamanho n da matriz% %b é o vetor b.

A = zeros(n,n); %Define a matriz A

temp = zeros(n,1);

tempo = zeros(200,1);

tempo2 = zeros(200,1);

mx = zeros(n,1);

%Pega as Matrizes do arquivo.

A = fscanf(fid,'%f',[n,n]); %Pega A do arquivo.

M = A'; %Transpõe A para ficar na ordem certa.

B = fscanf(fid,'%f',[1,n]); %Pega o vetor B de Resultados.

b1 = B'; %Transpõe B e forma o vetor R de resultado (Ax=R)

x = zeros(n,1);

x2 = zeros(n,1);

xg = linsolve(M,b1); %Resolve o sistema para checar o gabarito.

%Decompõe A em L, D e R

L1 = zeros(n,n); %Matriz inferior de M

D = zeros(n,n); %Matriz diagonal de M

R1 = zeros(n,n); %Matriz superior de M

for i=1:n

for j=1:n

if(i==j)

D(i,j) = M(i,j);

elseif(i>j)

L1(i,j) = M(i,j);

elseif(j>i)

R1(i,j) = M(i,j);

end

end

end

D2 = inv(D);

L = D2\*L1;

R = D2\*R1

b = D2\*b1;

%M2=L+R+eye(n);

M2=L+R;

%erro > 10^(-13)

erro = 1;

nit=0; %Número de Iterações

clc;

while(erro>10^(-14))

t2=tic;

x2=x;

for i=1:n

temp = sum(M2(i,1:n)\*x(1:n,1));

x(i,1)= b(i,1)-temp;

end

erro = (norm(x-x2)/norm(x));

nit=nit+1;

tempo(nit,1)=toc(t1);

tempo2(nit,1)=toc(t2);

end

fclose('all');

clc;

time = toc(t1);

numiter(cont/10,1)=nit;

tempo\_total(cont/10,1) = toc(t1);

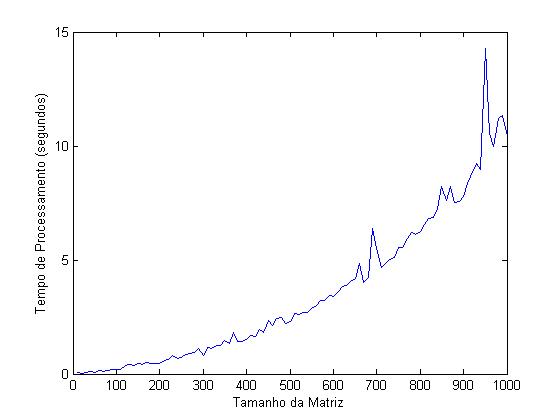
end

Os resoltados obitdos foram:

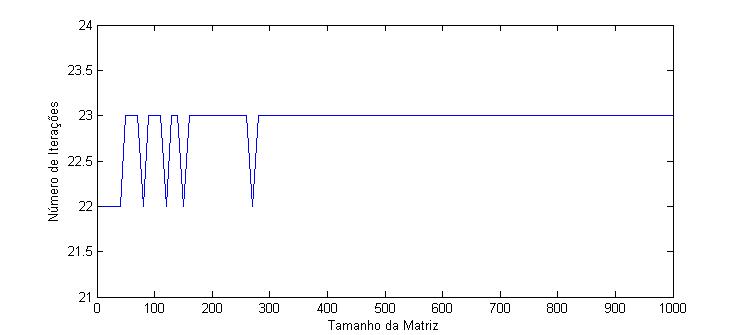
|  |  |  |
| --- | --- | --- |
| Tamanho da Matriz | Tempo de Processamento (s) | Número de Iterações |
| 10 | 0,063934708 | 22 |
| 20 | 0,018808122 | 22 |
| 30 | 0,078271037 | 22 |
| 40 | 0,10481974 | 22 |
| 50 | 0,066439949 | 23 |
| 60 | 0,137529623 | 23 |
| 70 | 0,112607738 | 23 |
| 80 | 0,174678547 | 22 |
| 90 | 0,185747554 | 23 |
| 100 | 0,207049673 | 23 |
| 110 | 0,214887042 | 23 |
| 120 | 0,377698327 | 22 |
| 130 | 0,426515072 | 23 |
| 140 | 0,393782214 | 23 |
| 150 | 0,454442756 | 22 |
| 160 | 0,403809529 | 23 |
| 170 | 0,503137294 | 23 |
| 180 | 0,478202699 | 23 |
| 190 | 0,461225725 | 23 |
| 200 | 0,458082565 | 23 |
| 210 | 0,583697021 | 23 |
| 220 | 0,620270595 | 23 |
| 230 | 0,794515553 | 23 |
| 240 | 0,687160265 | 23 |
| 250 | 0,707115176 | 23 |
| 260 | 0,870804395 | 23 |
| 270 | 0,908349269 | 22 |
| 280 | 0,947242329 | 23 |
| 290 | 1,112320307 | 23 |
| 300 | 0,814841972 | 23 |
| 310 | 1,17115899 | 23 |
| 320 | 1,099101926 | 23 |
| 330 | 1,256208596 | 23 |
| 340 | 1,256709644 | 23 |
| 350 | 1,451478507 | 23 |
| 360 | 1,342644028 | 23 |
| 370 | 1,834590108 | 23 |
| 380 | 1,431756767 | 23 |
| 390 | 1,421825262 | 23 |
| 400 | 1,491992519 | 23 |
| 410 | 1,676620169 | 23 |
| 420 | 1,635654481 | 23 |
| 430 | 1,93138477 | 23 |
| 440 | 1,809652092 | 23 |
| 450 | 2,332458821 | 23 |
| 460 | 2,114453061 | 23 |
| 470 | 2,431238121 | 23 |
| 480 | 2,471808348 | 23 |
| 490 | 2,196386393 | 23 |
| 500 | 2,308645596 | 23 |
| 510 | 2,661397108 | 23 |
| 520 | 2,605547604 | 23 |
| 530 | 2,709163362 | 23 |
| 540 | 2,688252305 | 23 |
| 550 | 2,891771178 | 23 |
| 560 | 3,002680729 | 23 |
| 570 | 3,217628874 | 23 |
| 580 | 3,233721071 | 23 |
| 590 | 3,447903713 | 23 |
| 600 | 3,420469744 | 23 |
| 610 | 3,556344686 | 23 |
| 620 | 3,833979074 | 23 |
| 630 | 3,860768769 | 23 |
| 640 | 4,067293441 | 23 |
| 650 | 4,199708952 | 23 |
| 660 | 4,840058131 | 23 |
| 670 | 4,014353927 | 23 |
| 680 | 4,230683987 | 23 |
| 690 | 6,372365221 | 23 |
| 700 | 5,51370232 | 23 |
| 710 | 4,673258992 | 23 |
| 720 | 4,840124123 | 23 |
| 730 | 5,036151716 | 23 |
| 740 | 5,104659405 | 23 |
| 750 | 5,569797704 | 23 |
| 760 | 5,566656988 | 23 |
| 770 | 5,933624088 | 23 |
| 780 | 6,226283074 | 23 |
| 790 | 6,121801603 | 23 |
| 800 | 6,252733034 | 23 |
| 810 | 6,54964384 | 23 |
| 820 | 6,83019555 | 23 |
| 830 | 6,866267588 | 23 |
| 840 | 7,253787923 | 23 |
| 850 | 8,231908011 | 23 |
| 860 | 7,62369923 | 23 |
| 870 | 8,226247878 | 23 |
| 880 | 7,504864321 | 23 |
| 890 | 7,567487994 | 23 |
| 900 | 7,793249001 | 23 |
| 910 | 8,345892775 | 23 |
| 920 | 8,883059786 | 23 |
| 930 | 9,242235466 | 23 |
| 940 | 8,972377343 | 23 |
| 950 | 14,27413585 | 23 |
| 960 | 10,53525863 | 23 |
| 970 | 9,992086863 | 23 |
| 980 | 11,20617516 | 23 |
| 990 | 11,35691485 | 23 |
| 1000 | 10,47291945 | 23 |

Apartir destes dados foram construidos os seguintes gráficos:

Tempo de Processamento x Tamanho da Matriz



Número de Iterações x Tamanho da Matriz



Verificou-se que o tempo de processamento almenta de forma exponencial aproximada de f (x) = 0.4819 \*exp(0.003211 \*x), sendo x o tamanho da matriz.

O número de iterações sofreu poucas variações, sendo a maioria dos processos terminados com 23 iterações e alguns, para matrizes com tamanho menores que 300 por 300, terminaram com 22 iterações.

**Gradientes Conjugados:**

%Programa de Solução de sistemas lineares pelo método dos Gradientes Conjugados%

%Raul Bortolotto Ghizzi, nº 7171220%

%Caio Bisulli Sant Ana, nº 7171290%

%Turma: Mecatrônica%

for h = 10:10:1000

h2 = int2str(h);

arquivo = strcat('matrizes/','Ab',h2,'.txt');

arquivo2 = strcat('Resultado',h2,'.txt');

tam(h/10,1) = h;

t1=tic;

fid = fopen(arquivo); %Abre o arquivo de entrada% Ax=b

%M é a matrix A

n = fscanf(fid,'%f',1); %Pega o tamanho n da matriz% %b é o vetor b.

A = zeros(n,n); %Define a matriz A

temp = zeros(n,1);

%Pega as Matrizes do arquivo.

A = fscanf(fid,'%f',[n,n]); %Pega A do arquivo.

M = A'; %Transpõe A para ficar na ordem certa.

B = fscanf(fid,'%f',[1,n]); %Pega o vetor B de Resultados.

b1 = B'; %Transpõe B e forma o vetor R de resultado (Ax=R)

% x = zeros(n,1);

nit = 1

tempo = zeros(1000,1);

xg = linsolve(M,b1); %Resolve o sistema para checar o gabarito.

v = zeros(n,1000);

p = zeros(n,1000);

r = zeros(n,1000);

alfa = zeros(1,1000);

q = zeros(1,1000);

t3=tic;

r(1:n,1)=-B;

p(1:n,2)=-r(1:n,1);

q(1,2)=(dot(r(1:n,1),r(1:n,1)))/(dot(M\*r(1:n,1),r(1:n,1)));

v(1:n,2) = v(1:n,1) - q(1,2)\*r(1:n,1);

% r(1:n,2) = r(1:n,1) + q(1,2)\*M\*p(1:n,2);

% tempo(1,1) = toc(t3);

for k=2:999

% t2=tic;

r(1:n,k) = r(1:n,k-1) + q(1,k)\*M\*p(1:n,k);

alfa(1,k) = (dot(r(1:n,k),r(1:n,k)))/(dot(r(1:n,k-1),r(1:n,k-1)));

p(1:n,k+1)= alfa(1,k)\*p(1:n,k) - r(1:n,k);

q(1,k+1)=(dot(r(1:n,k),r(1:n,k)))/(dot(M\*p(1:n,k+1),p(1:n,k+1)));

v(1:n,k+1) = v(1:n,k) + q(1,k+1)\*p(1:n,k+1);

erro = (norm(v(1:n,k) - v(1:n,k-1))/norm(v(1:n,k)))

nit = nit + 1;

% tempo(nit,1) = toc(t2);

if (erro<10^(-13))

x = v(1:n,k+1);

break;

end

end

fclose('all');

clc;

tempo\_total(h/10,1) = toc(t1);

numiter(h/10,1)=nit;

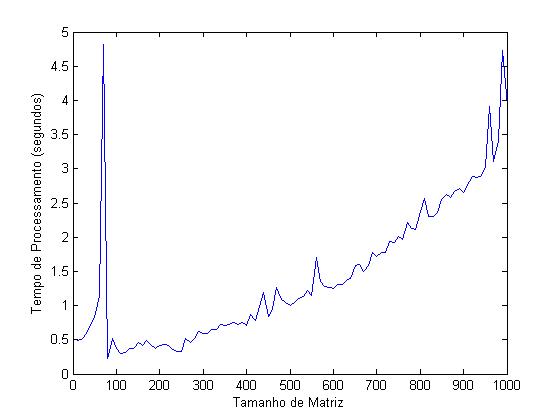
end

Os resultados obitidos foram:

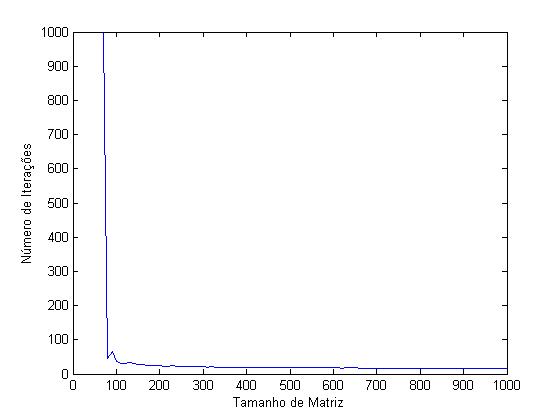
|  |  |  |
| --- | --- | --- |
| Tamanho da Matriz | Tempo de Processamento (s) | Número de Iterações |
| 10 | 0,489275128 | 999 |
| 20 | 0,504235689 | 999 |
| 30 | 0,587879917 | 999 |
| 40 | 0,72502581 | 999 |
| 50 | 0,821005107 | 999 |
| 60 | 1,147324746 | 999 |
| 70 | 4,824402946 | 999 |
| 80 | 0,228916876 | 45 |
| 90 | 0,511707415 | 66 |
| 100 | 0,368865705 | 36 |
| 110 | 0,297906059 | 30 |
| 120 | 0,307138053 | 32 |
| 130 | 0,377812224 | 34 |
| 140 | 0,37493743 | 30 |
| 150 | 0,466527546 | 28 |
| 160 | 0,411327205 | 27 |
| 170 | 0,493994756 | 26 |
| 180 | 0,423070305 | 24 |
| 190 | 0,368943918 | 24 |
| 200 | 0,423910111 | 24 |
| 210 | 0,434883796 | 23 |
| 220 | 0,409865123 | 23 |
| 230 | 0,35143901 | 24 |
| 240 | 0,326212585 | 22 |
| 250 | 0,322167539 | 22 |
| 260 | 0,518314895 | 22 |
| 270 | 0,46615506 | 21 |
| 280 | 0,516193384 | 21 |
| 290 | 0,628641275 | 22 |
| 300 | 0,589794653 | 21 |
| 310 | 0,598898574 | 20 |
| 320 | 0,645755121 | 21 |
| 330 | 0,645194925 | 20 |
| 340 | 0,721783907 | 20 |
| 350 | 0,711742415 | 20 |
| 360 | 0,722606115 | 20 |
| 370 | 0,753106498 | 20 |
| 380 | 0,726011286 | 20 |
| 390 | 0,754319279 | 20 |
| 400 | 0,708215037 | 20 |
| 410 | 0,868241473 | 19 |
| 420 | 0,777752196 | 19 |
| 430 | 0,981264225 | 19 |
| 440 | 1,188948884 | 19 |
| 450 | 0,837395 | 18 |
| 460 | 0,964571259 | 19 |
| 470 | 1,259778503 | 19 |
| 480 | 1,094903877 | 19 |
| 490 | 1,043883499 | 19 |
| 500 | 0,999531703 | 19 |
| 510 | 1,038873996 | 18 |
| 520 | 1,108712761 | 18 |
| 530 | 1,129333455 | 18 |
| 540 | 1,220496825 | 18 |
| 550 | 1,143774392 | 18 |
| 560 | 1,697861674 | 18 |
| 570 | 1,355859965 | 18 |
| 580 | 1,276832712 | 18 |
| 590 | 1,263652459 | 18 |
| 600 | 1,246282468 | 18 |
| 610 | 1,315653425 | 18 |
| 620 | 1,307086237 | 17 |
| 630 | 1,364594332 | 18 |
| 640 | 1,39758969 | 18 |
| 650 | 1,580575369 | 18 |
| 660 | 1,603375744 | 17 |
| 670 | 1,491834628 | 17 |
| 680 | 1,590175939 | 17 |
| 690 | 1,777326916 | 17 |
| 700 | 1,712308966 | 17 |
| 710 | 1,773016925 | 17 |
| 720 | 1,769360008 | 17 |
| 730 | 1,93181005 | 17 |
| 740 | 1,926671007 | 17 |
| 750 | 2,010208182 | 17 |
| 760 | 1,969084602 | 17 |
| 770 | 2,212059175 | 17 |
| 780 | 2,129603776 | 17 |
| 790 | 2,119198597 | 17 |
| 800 | 2,379657547 | 17 |
| 810 | 2,568389392 | 17 |
| 820 | 2,306848178 | 17 |
| 830 | 2,304611304 | 17 |
| 840 | 2,367175829 | 17 |
| 850 | 2,545087969 | 17 |
| 860 | 2,62602116 | 17 |
| 870 | 2,584915179 | 17 |
| 880 | 2,665648928 | 16 |
| 890 | 2,706218177 | 17 |
| 900 | 2,659620709 | 17 |
| 910 | 2,768901488 | 17 |
| 920 | 2,8865681 | 17 |
| 930 | 2,867912003 | 17 |
| 940 | 2,89072411 | 17 |
| 950 | 3,03612092 | 17 |
| 960 | 3,91484774 | 16 |
| 970 | 3,100231118 | 17 |
| 980 | 3,412782933 | 16 |
| 990 | 4,732919394 | 17 |
| 1000 | 3,918206962 | 17 |

Apartir dos dados obitidos foram construidos os seguintes gráficos:

Tempo de Processamento x Tamanho da Matriz



Número de iteraçoes x Tamanho da Matriz



Foi verificado que o tempo de processamento almenta com relação ao tamanho da matriz na forma exponencial aproximada por f (x) = 0.3315 \*exp(0.0024 \*x), sendo x o tamanho da matriz. Ocorreram dois picos no tempo de processamento nas matrizes de tamanhos 60 e 70. O número de iterações para as matrizes de tamanhos 10 até 70 foi de 999, número máximo para os testes realizados, para as demais matrizes houve pouca variação no número de iterações, as matrizes criadas pelo programa gerador de matrizes utilizado, de tamanho menores que 70 apresentaram uma tendencia a demorar mais para convergir, foram realizados outros cinco testes com matrizes diferentes e o padrão se manteve, essa demora porém se deve ao erro que foi colocado como base para os métodos iterativos que foi da ordem de 10^(-13), ordem de grandeza do erro obitido com a decomposição L.U, enquanto o erro para tais matrizes ficou na ordem de 10^(-12). A matriz de tamanho 50 colocada no site convergiu normalmente, com cerca de 25 iterações.

Em um teste posterior com uma matriz de tamanho 50, gerada aleatóriamente pelo programa, com o mesmo método, porém dessa vez sem restrição de iterações, foi possivél se obiter um erro de 1,3070e-12 após 42689 iterações, porém não atingiu a ordem de 10^(-13).

**Decomposição L.U:**

%Programa de Solução de sistemas lineares pelo método da Decomposição L.U%

%Raul Bortolotto Ghizzi, nº 7171220%

%Caio Bisulli Sant Ana, nº 7171290%

%Turma: Mecatrônica%

for cont=10:10:1000

cont2 = int2str(cont);

arquivo = strcat('matrizes/','Ab',cont2,'.txt');

arquivo2 = strcat('Resultado',cont2,'.txt');

tam(cont/10,1) = cont;

t1 = tic;

fid = fopen(arquivo); %Abre o arquivo de entrada%

n = fscanf(fid,'%f',1); %Pega o tamanho n da matriz%

A = zeros(n,n); %Define a matriz A

temp = zeros(1,n);

%Pega as Matrizes do arquivo.

A = fscanf(fid,'%f',[n,n]); %Pega A do arquivo.

M = A'; %Transpõe A para ficar na ordem certa.

B = fscanf(fid,'%f',[1,n]); %Pega o vetor B de Resultados.

R = B'; %Transpõe B e forma o vetor R de resultado (Ax=R)

%Mbackup = M; %Guarda a Matriz M inalterada para imprimir depois.

xg = linsolve(M,R); %Resolve o sistema para checar o gabarito.

if (det(M) == 0) %Verifica se o determinante é igual a zero, no caso se o sistema tem uma única solução.

erro = 'O sistema não tem solução única';

%Imprime o erro

disp(erro);

else

%Decomposição L.U

U = M; %Matriz U da decomposição.

L = eye(n); %Matriz L da decomposição.

Utemp = U;

for i=1:n-1 %Coluna a ser analisada.

for j=i+1:n %Percorre a Coluna a ser analisada.

%Pivoteamento

lpivo = i;

for h=i:n-1

if (abs(M(i,h+1))>abs(M(i,h)))

lpivo = h+1; %Pega o maior elemento da coluna, no caso a linha que será trocada no piteamento.

end

end

temp(1,:) = M(i,:); %Armazena a linha que tem o fator pequeno da matriz em um vetor temporário. Usei a matriz A pois a leitura do MatLab lê a matriz coluna por coluna e não linha por linha.

M(i,:) = M(lpivo,:); %Troca as linhas a serem pivotadas.

M(lpivo,:)=temp(1,:);

% Fim do Pivoteamento

p = (U(j,i)/U(i,i)); %p é o elemento pivo.

L(j,i) = p;

for k=i+1:n

U(j,k) = U(j,k) - (p\*U(i,k));

end

R(j,1) = R(j,1) - (p\*R(i,1));

end

end

for i=1:n

for j=i+1:n

U(j,i) = 0;

end

end

[Lg,Ug,P] = lu(M); %Usei a função L.U do MatLab para saber se o programa estava funcionando.

%Começa resolver o sistema agora, L\*U\*x = R, U\*x = y => L\*y = R.

%Acha o y

y = zeros(n,1);

for i=1:n

soma = 0; %O valor de "soma" é a soma dos produto de L por y, esse valor muda a cada iteração quando um novo valor de y é encontrado.

for j=1:n

soma = soma + L(j,i)\*y(j,1);

end

y(i,1) = R(i,1) - soma;

end

%Acha o x: U\*x=y -> x=inv(U)\*y

x = inv(U)\*y;

end

fclose('all');

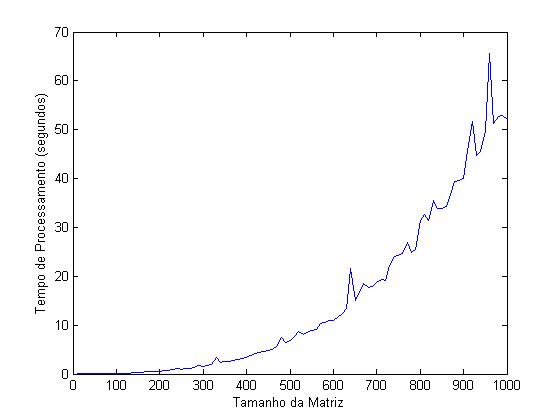
tempo\_total(cont/10,1) = toc(t1);

end

Os resultados obitidos foram:

|  |  |
| --- | --- |
| Tamanho da Matriz | Tempo de Processamento (s) |
| 10 | 0,094926852 |
| 20 | 0,023164062 |
| 30 | 0,033409395 |
| 40 | 0,057657676 |
| 50 | 0,061584915 |
| 60 | 0,09155932 |
| 70 | 0,106687061 |
| 80 | 0,113439233 |
| 90 | 0,10726681 |
| 100 | 0,150009385 |
| 110 | 0,144644505 |
| 120 | 0,188666831 |
| 130 | 0,200549246 |
| 140 | 0,222623712 |
| 150 | 0,293055914 |
| 160 | 0,363826384 |
| 170 | 0,445108109 |
| 180 | 0,412461285 |
| 190 | 0,481822955 |
| 200 | 0,609844396 |
| 210 | 0,688280168 |
| 220 | 0,698852527 |
| 230 | 0,882949799 |
| 240 | 1,057465567 |
| 250 | 0,918050048 |
| 260 | 1,067530034 |
| 270 | 1,14512209 |
| 280 | 1,269695343 |
| 290 | 1,834879005 |
| 300 | 1,553364794 |
| 310 | 1,830872576 |
| 320 | 1,963461621 |
| 330 | 3,438080727 |
| 340 | 2,427990841 |
| 350 | 2,513294149 |
| 360 | 2,551771706 |
| 370 | 2,825438283 |
| 380 | 2,916557658 |
| 390 | 3,131022842 |
| 400 | 3,401150797 |
| 410 | 3,759089745 |
| 420 | 4,134644564 |
| 430 | 4,459081239 |
| 440 | 4,664648298 |
| 450 | 4,833627119 |
| 460 | 5,079160214 |
| 470 | 5,568264252 |
| 480 | 7,397718252 |
| 490 | 6,360061436 |
| 500 | 6,899702402 |
| 510 | 7,772664481 |
| 520 | 8,602128257 |
| 530 | 8,067904475 |
| 540 | 8,592288162 |
| 550 | 8,841889767 |
| 560 | 9,02201043 |
| 570 | 10,30501214 |
| 580 | 10,52242984 |
| 590 | 11,02679654 |
| 600 | 11,01152704 |
| 610 | 11,65002601 |
| 620 | 12,36071304 |
| 630 | 13,44571817 |
| 640 | 21,68780649 |
| 650 | 14,94795211 |
| 660 | 16,87348806 |
| 670 | 18,54432735 |
| 680 | 17,60608629 |
| 690 | 17,82707634 |
| 700 | 18,62626117 |
| 710 | 19,35148203 |
| 720 | 19,21311944 |
| 730 | 22,21213934 |
| 740 | 24,1095335 |
| 750 | 24,35483978 |
| 760 | 24,6999387 |
| 770 | 26,98841479 |
| 780 | 24,93664797 |
| 790 | 25,54043678 |
| 800 | 31,43106703 |
| 810 | 32,61723449 |
| 820 | 31,51833396 |
| 830 | 35,47342734 |
| 840 | 33,96061616 |
| 850 | 33,81408967 |
| 860 | 34,2184198 |
| 870 | 36,69558814 |
| 880 | 39,47218083 |
| 890 | 39,62002276 |
| 900 | 39,92538099 |
| 910 | 45,68891662 |
| 920 | 51,68807975 |
| 930 | 44,73309635 |
| 940 | 45,47055108 |
| 950 | 49,74842744 |
| 960 | 65,51722481 |
| 970 | 51,18925929 |
| 980 | 52,6061733 |
| 990 | 53,0067204 |
| 1000 | 52,00589575 |

Com os dados acima foi montado um gráfico do Tempo de Processamento x Tamanho da Matriz



O almento do tempo de processamento em função do tamanho da matriz se da na forma exponencial, aproximada por f(x) = 1.067\*exp(0.004056\*x), sendo x o tamanho da matriz.

Para os testes feitos foram utilizadas matrizes geradas atrazés do seguinte algoritmo:

%Programa Gerador de Matrizes diagonalmente dominantes%

%Raul Bortolotto Ghizzi, nº 7171220%

%Turma: Mecatrônica%

for n=100:100:1000 %Tamanho da menor matriz:Passo:Tamanho da maior matriz

n2 = int2str(n);

arquivo = strcat('Ab',n2,'.txt');

file = fopen(arquivo,'a');

M=rand(n)\*10; %Cria uma matriz

soma = sum(M);

for j=1:n %Transforma a Matriz em uma Matriz diagonalmente dominate

somaparcial = soma(j) - M(j,j);

M(j,j) = M(j,j) + somaparcial;

end

x = rand(1,n)\*10;

A(1:n,1:n) = M;

A(n+1,1:n) = x;

%Salva a matriz em um arquivo

save(arquivo,'n','-ascii');

save(arquivo,'M','-ascii','-append');

save(arquivo,'x','-ascii','-append');

fclose('all');

end

fclose('all');

**Conclusão:**

Ambos os métodos iterativos convergiram em todos os testes. O método que convergiu mais rápdo foi o método dos gradientes conjugados, seguido pelo método de Gauss-Seidel e por último a decomposição L.U, sendo em alguns casos o método iterativo dos Gradientes Conjugados até dez vezes mais rápido que o método estacionário da Decomposição L.U, e até tres vezes mais rápido que o método iterativo de Gauss-Seidel, sendo que a diferença entre estes é que o método dos gradientes conjugados utiliza-se de processos de relaxação que aceleram a convergência do método.