

```
% es 1. Risolvo sistema tridiag con Thomas
```

```
n=100;
[A,f]=creamatrice(n);
u=Thomas(A,f);
```

```
% es 2
```

```
% crea w e v
v=eye(n,1);
coef=(n-1)^2/(4*pi^2);
w=coef*ones(n,1)-A(:,1);
```

```
% Risolvi i due sistemi con A tridiag, in modo efficiente
```

```
Z=Thomas_multi(A,[w,f]);
w1=Z(:,1);
w2=Z(:,2);
theta1=1+v'*w1;
theta2=v'*w2;
```

```
% applica la formula di SM
x = w2-w1*(theta2/theta1);
```

```
% controlla l'accuratezza senza creare B esplicitamente, che costerebbe troppo!
```

```
normSM=norm( f - A*x - w*(v'*x));
disp(' ')
fprintf('Formula di Sherman-Morrison, norma del residuo: %d\n', normSM)
disp(' ')
```

```
% es 3 Facoltativo
```

```
s=5;
W=rand(n,s); V=rand(n,s);
```

```
Z=Thomas_multi(A,[W,f]);
w1=Z(:,1:s);
w2=Z(:,s+1);
theta1=eye(s)+V'*w1;
theta2=V'*w2;
Q=gauss(theta1,theta2);
x = w2-w1*Q;
```

```
normSM=norm( f - A*x - W*(V'*x));
disp(' ')
fprintf('Formula di Sherman-Morrison-Woodbury, norma del residuo: %d\n', normSM)
disp(' ')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function u=Thomas(A,f)
```

```
n=length(f);
```

```
% crea gli elementi, in modo che gli indici corrispondano!
```

```
a=diag(A);b=[0;diag(A,-1)];c=diag(A,1);
```

```
alpha(1)=a(1);
```

```
for i=2:n
    beta(i)=b(i)/alpha(i-1);
    alpha(i)=a(i)-beta(i)*c(i-1);
end
```

```
y(1)=f(1);
for i=2:n
    y(i,1)=f(i) - beta(i)*y(i-1,1);
```

