

```

function [A,rhs] = StifMatrix(Tx,Ty,n)

global Fixl Fixr Fixd Fixu ibcX ibcY
global i1 i2 i3 i4 iD

%------%
%           creating the transmissibility matrix           %
%------%

Txeast(2:n(1),:) = Tx(2:n(1),:); Txeast(1,:) = 0;
Tynorth(:,2:n(2)) = Ty(:,2:n(2)); Tynorth(:,1) = 0;
Twest(1:n(1)-1,:) = Tx(2:n(1),:); Twest(n(1),:) = 0;
Tysouth(:,1:n(2)-1) = Ty(:,2:n(2)); Tysouth(:,n(2)) = 0;

%------%
%           preparing and setting the diagonals of the matrix A           %
%------%

Ds = [Tysouth(:) Twest(:) zeros(prod(n),1) Txeast(:) Tynorth(:)];
Ds(:,3) = -sum(Ds,2);
A = spdiags(-Ds,[-n(1),-1,0,1,n(1)],n(1)*n(2),n(1)*n(2));

%------%
%           preparing the right hand side of the equation           %
%------%

rhs = sparse(prod(n),1);

%------%
%           boundary conditions           %
%------%

%------%
% Dirichlet %
%------%

t1 = Tx(1,,:).*ibcX(1,,:);
tr = Tx(n(1)+1,,:).*ibcX(2,,:);
td = Ty(:,1,:).*ibcY(1,1,:);
tu = Ty(:,n(2)+1,:).*ibcY(2,2,:);

tD = [t1(:);tr(:);td(:);tu(:)];
A = A + sparse(iD,iD,tD,prod(n),prod(n));

rhs(i1) = rhs(i1) + t1(:).*Fixl(:);
rhs(i2) = rhs(i2) + tr(:).*Fixr(:);
rhs(i3) = rhs(i3) + td(:).*Fixd(:);
rhs(i4) = rhs(i4) + tu(:).*Fixu(:);

%------%
% Neumann %
%------%

t1 = 1-ibcX(1,,:);
tr = 1-ibcX(2,,:);
td = 1-ibcY(1,1,:);
tu = 1-ibcY(2,2,:);

rhs(i1) = rhs(i1) + t1(:).*Fixl(:);
rhs(i2) = rhs(i2) + tr(:).*Fixr(:);
rhs(i3) = rhs(i3) + td(:).*Fixd(:);
rhs(i4) = rhs(i4) + tu(:).*Fixu(:);

```