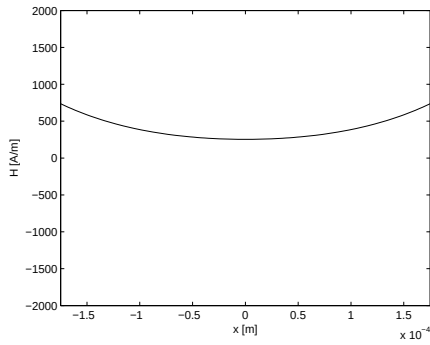


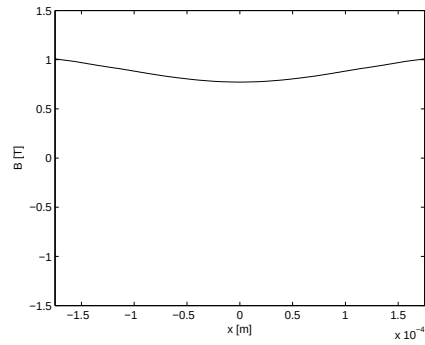
A. Függelék

Szimulációs eredmények: Vezető lemez mágneses térben, 1D

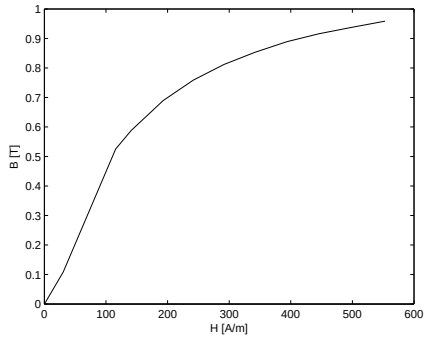
A szimulációs eredmények a mérési eredményekből identifikált Everett-függvénnyel működő skalár Preisach-moddal, $2a = 0,35$ mm széles lemez feltételezése mellett ($a = 0,175$ mm) kerülnek bemutatásra. A lemez szélére Dirichlet-típusú peremfeltételként előírt gerjesztés (mágneses térerősség) frekvenciája $f = 500$ Hz, periódusideje $t = 2$ ms, amplitúdója $H_0 = 2000 \frac{\text{A}}{\text{m}}$ (a vizsgált csomópont az $n = 10$ -es, valamint $\varepsilon = 10^{-3}$ T).



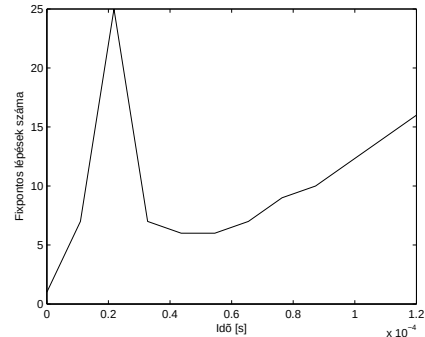
(a) Mágneses térerősség ($t = 0, 12$ ms)



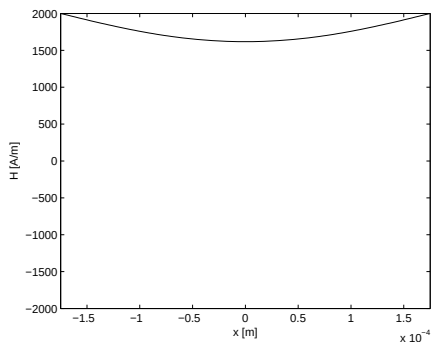
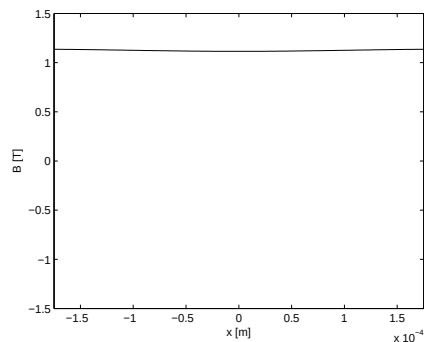
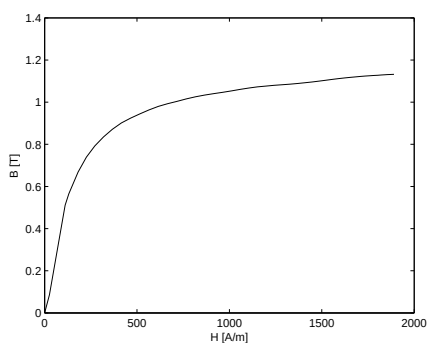
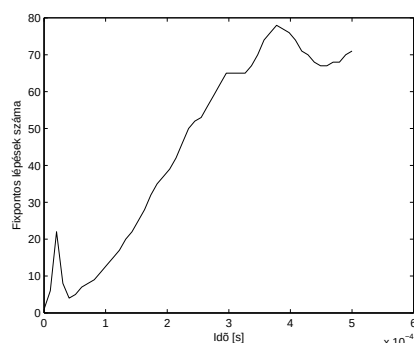
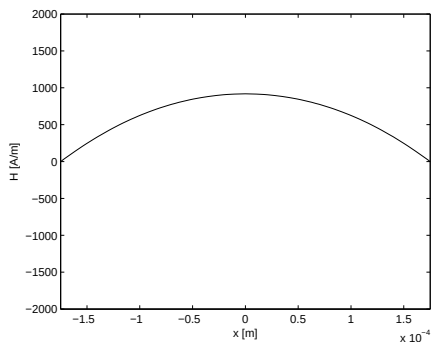
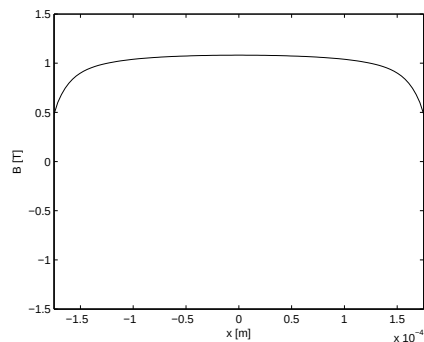
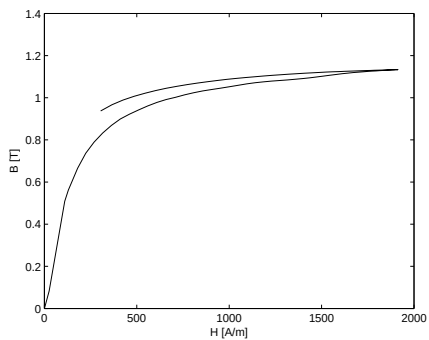
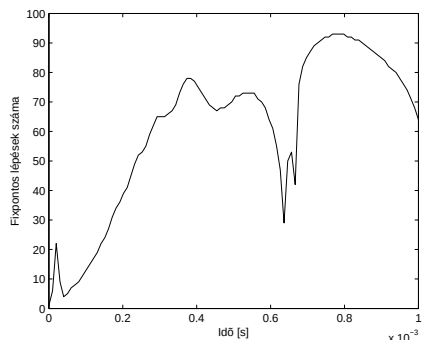
(b) Mágneses indukció ($t = 0, 12$ ms)

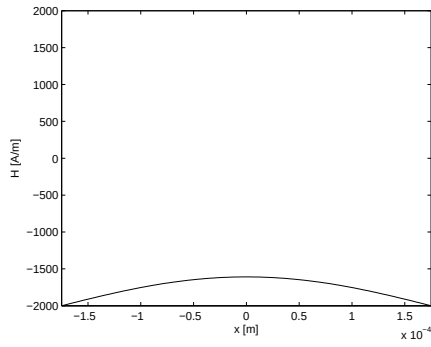
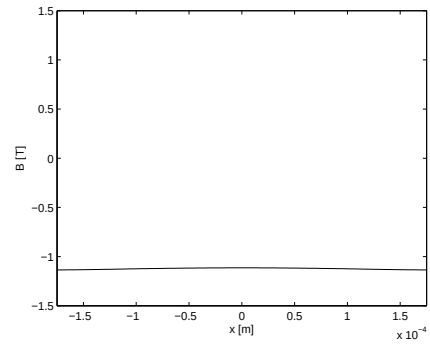
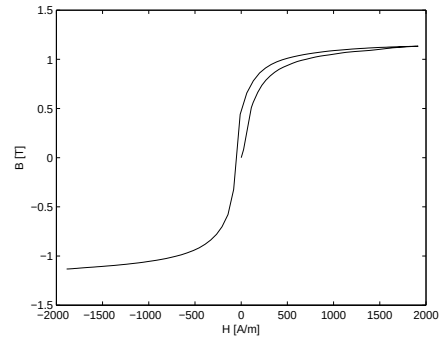
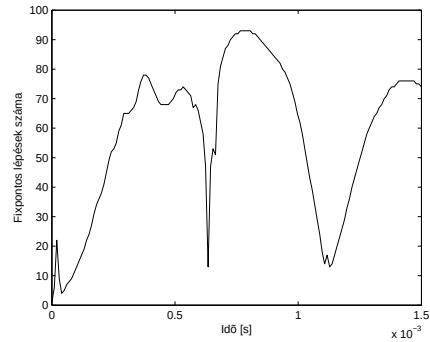
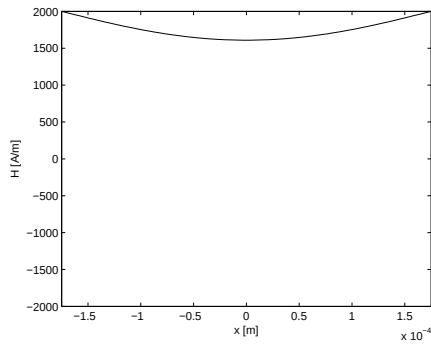
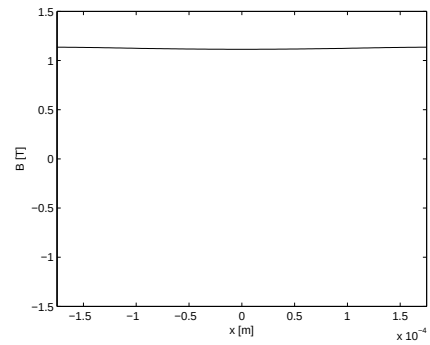
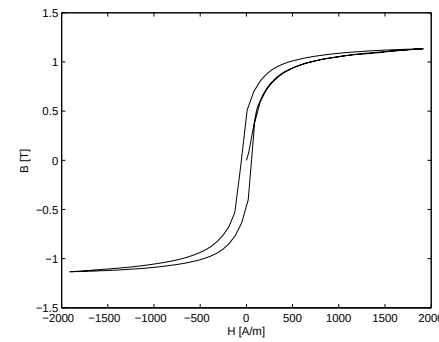
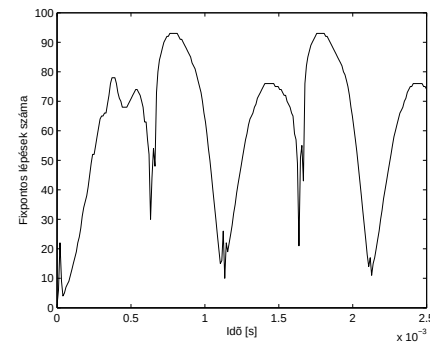


(c) $B - H$ karakterisztika ($t = 0, 12$ ms)



(d) Fixpont-lépések száma ($t = 0, 12$ ms)

(e) Mágneses térerősség ($t = 0,5$ ms)(f) Mágneses indukció ($t = 0,5$ ms)(g) $B - H$ karakterisztika ($t = 0,5$ ms)(h) Fixpontos lépések száma ($t = 0,5$ ms)(i) Mágneses térerősség ($t = 1$ ms)(j) Mágneses indukció ($t = 1$ ms)(k) $B - H$ karakterisztika ($t = 1$ ms)(l) Fixpontos lépések száma ($t = 1$ ms)

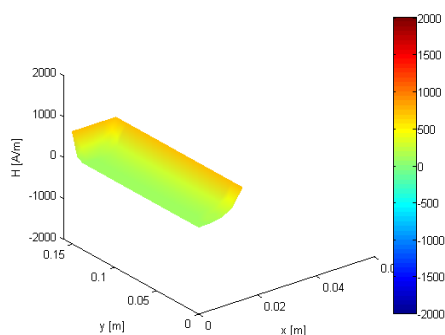
(m) Mágneses térerősség ($t = 1,5$ ms)(n) Mágneses indukció ($t = 1,5$ ms)(o) $B - H$ karakterisztika ($t = 1,5$ ms)(p) Fixpont-lépések száma ($t = 1,5$ ms)(q) Mágneses térerősség ($t = 2,5$ ms)(r) Mágneses indukció ($t = 2,5$ ms)(s) $B - H$ karakterisztika ($t = 2,5$ ms)(t) Fixpont-lépések száma ($t = 2,5$ ms)

A.1. ábra. Szimulációs eredmények

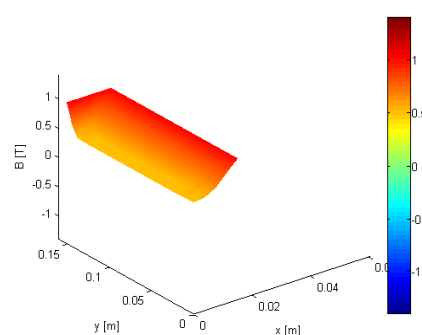
B. Függelék

Szimulációs eredmények: Vezető lemez mágneses térben, 2D

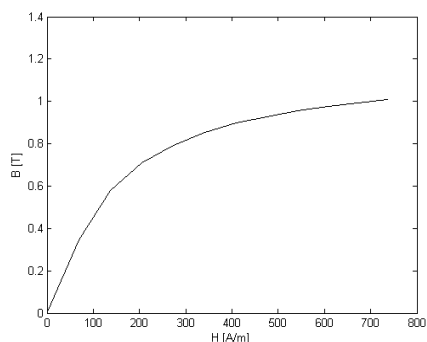
A szimulációs eredmények a mérési eredményekből identifikált Everett-függvénnyel működő skalár Preisach-moddal, $\frac{a}{2} = 0,015$ m lemezszélesség ($a = 0,03$ m) és $\frac{b}{2} = 0,15$ m lemez hosszúság ($b = 0,3$ m) mellett kerülnek bemutatásra. A lemez szélére Dirichlet-típusú peremfeltételként előírt gerjesztés (mágneses térerősség) frekvenciája $f = 1$ Hz, amplitúdója $H_0 = 2000 \frac{\text{A}}{\text{m}}$. Az anyag vezetőképessége $\sigma = 2,22 \cdot 10^6 \frac{\text{A}}{\text{Vm}}$, a vizsgált csomópont az $n = 88$ -as, $\varepsilon = 10^{-3}$ T.



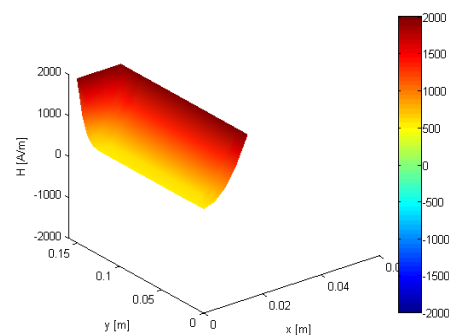
(a) Mágneses térerősség ($t = 60$ ms)



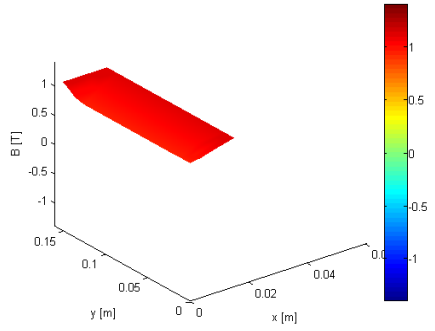
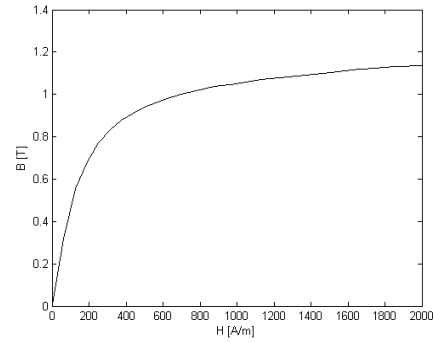
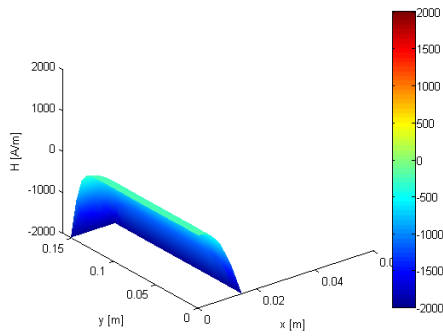
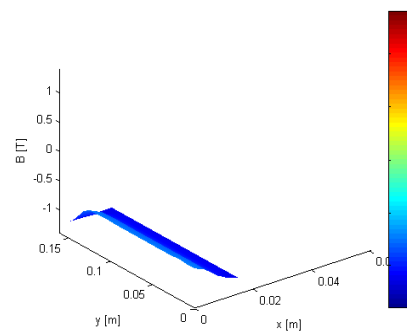
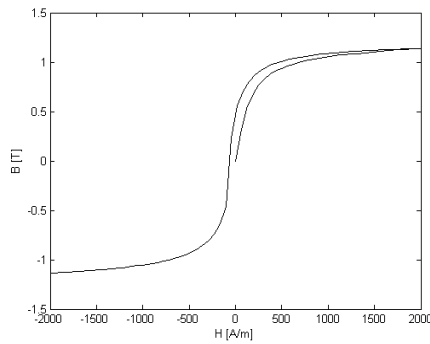
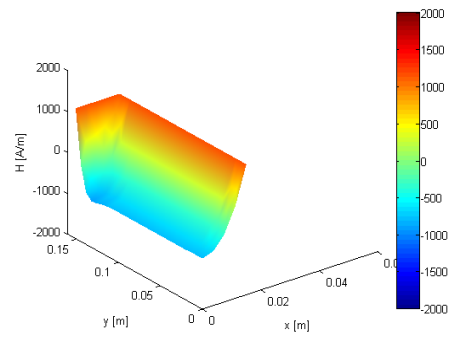
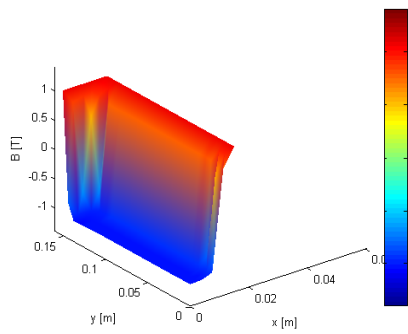
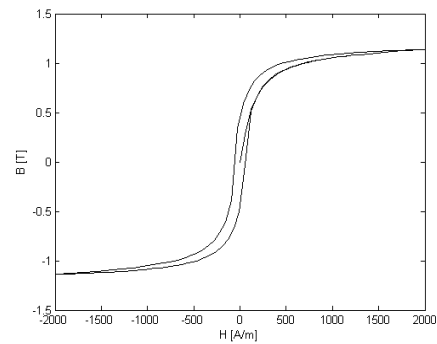
(b) Mágneses indukció ($t = 60$ ms)



(c) $B - H$ karakterisztika ($t = 60$ ms)



(d) Mágneses térerősség ($t = 246,2$ ms)

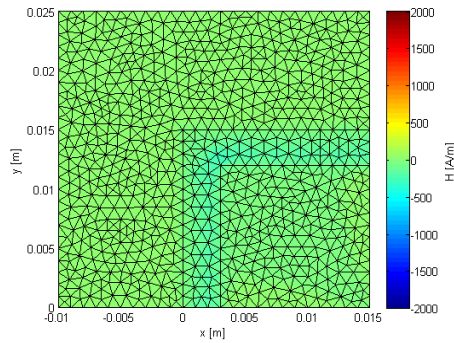
(e) Mágneses indukció ($t = 246,2$ ms)(f) $B - H$ karakterisztika ($t = 246,2$ ms)(g) Mágneses térerősség ($t = 750$ ms)(h) Mágneses indukció ($t = 750$ ms)(i) $B - H$ karakterisztika ($t = 750$ ms)(j) Mágneses térerősség ($t = 1100$ ms)(k) Mágneses indukció ($t = 1100$ ms)(l) $B - H$ karakterisztika ($t = 1100$ ms)

B.1. ábra. Szimulációs eredmények

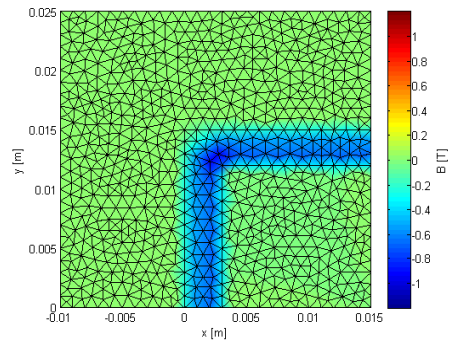
C. Függelék

Szimulációs eredmények: Örvényáramú A-formalizmus, 2D

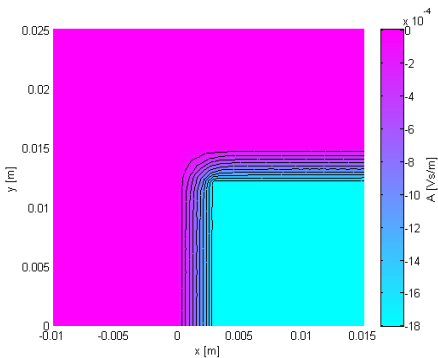
A szimulációs eredmények a mérési eredményekből identifikált Everett-függvénnyel működő skalár Preisach-modellel, 0,003 m széles, 0,015m hosszú lemez feltételezése mellett kerülnek bemutatásra. A lemezek szélésre Neumann-típusú peremfeltételként előírt felületi áram amplitúdója $1500 \frac{\text{A}}{\text{m}}$, frekvenciája 1 Hz, periódusideje 1 másodperc, a lemezek vezetőképessége $\sigma = 2,22 \cdot 10^6 \frac{\text{A}}{\text{Vm}}$.



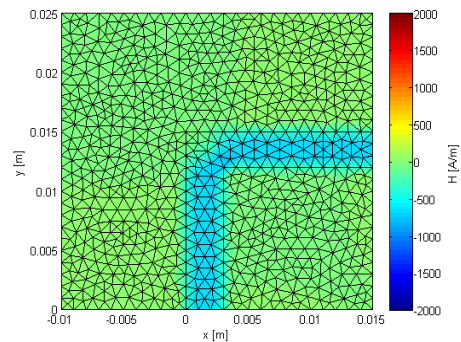
(a) Mágneses térerősség ($t = 60$ ms)



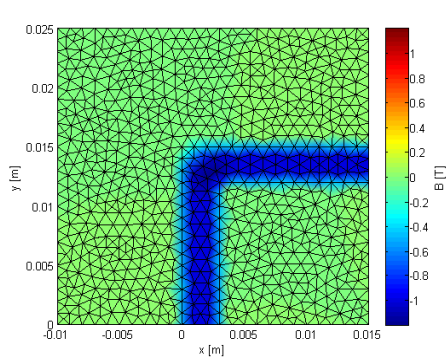
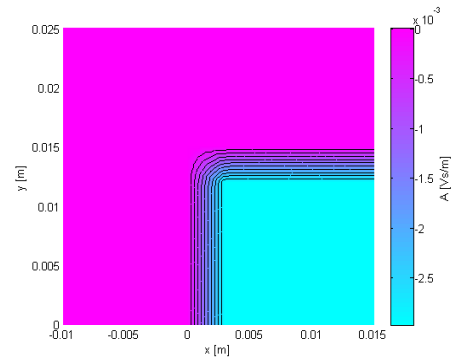
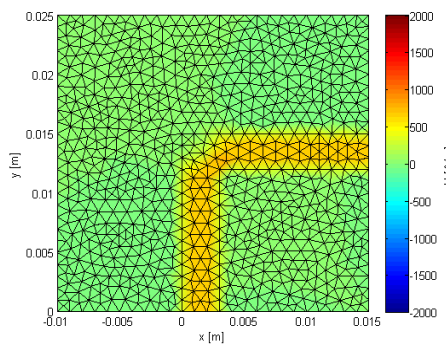
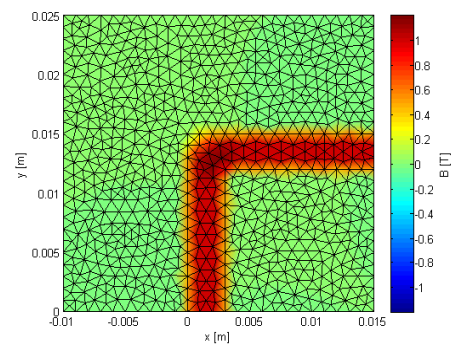
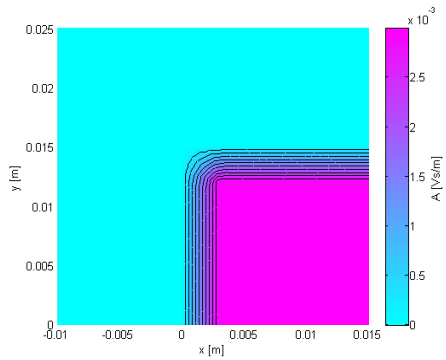
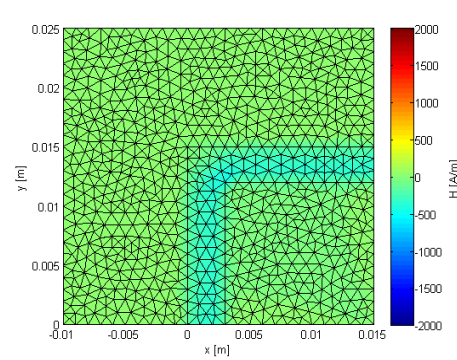
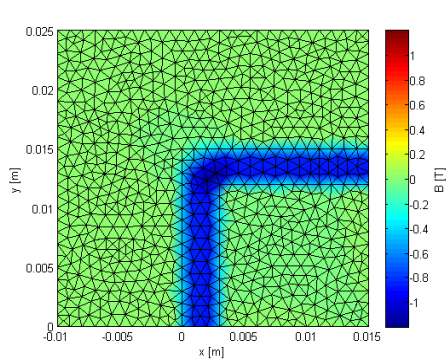
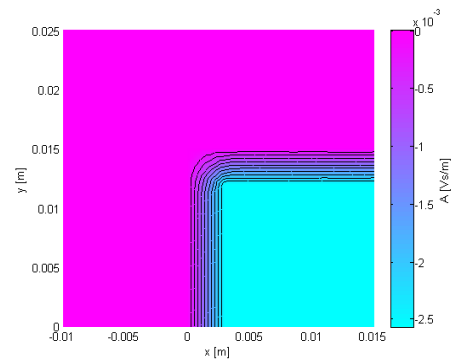
(b) Mágneses indukció ($t = 60$ ms)



(c) Mágneses vektorpotenciál ($t = 60$ ms)



(d) Mágneses térerősség ($t = 250$ ms)

(e) Mágneses indukció ($t = 250$ ms)(f) Mágneses vektorpotenciál ($t = 250$ ms)(g) Mágneses térerősség ($t = 750$ ms)(h) Mágneses indukció ($t = 750$ ms)(i) Mágneses vektorpotenciál ($t = 750$ ms)(j) Mágneses térerősség ($t = 1,1$ s)(k) Mágneses indukció ($t = 1,1$ s)(l) Mágneses vektorpotenciál ($t = 1,1$ s)

C.1. ábra. Szimulációs eredmények

D. Függelék

Preisach-modell

D.1. Preisach.m

```
function [Bk,Lkm1,Inck] = Preisach(alpha,beta,E,mu_0,Mmax,Hmax,Lkm1,Inck,Hkm1,Hk)

hkm1 = Hkm1 / Hmax; hk = Hk / Hmax;
Lk = Lkm1; Inckm1 = Inck; n = size(Lk,2);

if abs(hk) >= 1 %Teljes kivezérlés esete

    Lk = [1;-1];

elseif hk > hkm1 %Növekvő térerősség

    if Inckm1 == 1 %Eddig is növekedett a H

        Inck = 1;

    elseif Inckm1 == 0 %Eddig csökkent a H

        if n == 1 %Csak (1;-1) pont van tárolva

            Lk = [Lk,[abs(hkm1);-abs(hkm1)]];

        else %Már nem csak az (1;-1) pont van tárolva

            Lk = [Lk,[Lk(1,n);hkm1]]; %Az \alpha-koordináta fog ekkor megegyezni

        end

    end

end

%Növekvő mágneses térerősségnél azokat a pontokat kell törölni
%a lépcsős függvényből, melyek \alpha-koordinátája nagyobb vagy
%egyenlő az aktuális hk-val

j = 1;
while j <= size(Lk,2)

    if hk > Lk(1,j) || abs(hk-Lk(1,j)) < 1e-3

        Lk(:,j) = [];

    else

        j = j+1;

    end

end

Inck = 1;
```

```

elseif hk < hkm1 %Csökkenő térerősség

    if Inckm1 == 0 %Eddig is csökkent a H

        Inck = 0;

    elseif Inckm1 == 1 %Eddig növekedett a H

        if n == 1 %Csak (1;-1) pont van tárolva

            Lk = [Lk,[abs(hkm1);-abs(hkm1)]];

        else %Már nem csak az (1;-1) pont van tárolva

            Lk = [Lk,[hkm1;Lk(2,n)]]; %A \beta-koordináta fog ekkor megegyezni

        end

    end

end

%Csökkenő mágneses térerősségnél azokat a pontokat kell törölni
%a lépcsős függvényből, melyek \beta-koordinátája kisebb vagy
%egyenlő az aktuális hk-val

j = 1;
while j <= size(Lk,2)

    if hk < Lk(2,j) || abs(hk-Lk(2,j)) < 1e-3

        Lk(:,j) = [];

    else

        j = j+1;

    end

end

end

Inck = 0;

end
%Dinamikus L-kezelés
if hk >= 1 %Teljes kivezérlés esete

elseif size(Lk,2) == 1 %Nincs tárolva fordulópont

    Lk = [Lk,[abs(hk);-abs(hk)]];

elseif size(Lk,2) > 1 %Már van fordulópont

    if hk > hkm1 %Növekszik a térerősség értéke

        Lk = [Lk,[hk;Lk(2,size(Lk,2))]]; %A \alpha-koordináta változik

    elseif hk < hkm1 %Csökken a térerősség értéke

        Lk = [Lk,[Lk(1,size(Lk,2));hk]]; %A \beta-koordináta változik

    end

end

end

%Számítás

if size(Lk,2) == 2 %Első mágnesezési görbe

    if hk > hkm1

        sign = 1;

```

```

else
    sign = -1;
end

Mk = sign * interp2(alpha,beta,E,Lk(1,2),Lk(2,2),'spline');
else %Nem az első mágnesezési görbén vagyunk
    if Lk(1,3) > -Lk(2,3) % Az alpha = -beta-egyenes felett vagyunk
        sum = 0;
        for k = 3:size(Lk,2)
            if k == 3
                sum = sum + (interp2(alpha,beta,E,Lk(1,k-1),Lk(2,k-1),'spline') ...
                    - interp2(alpha,beta,E,Lk(1,k),Lk(2,k),'spline'));
            elseif Lk(2,k) == Lk(2,k-1)
                sum = sum + interp2(alpha,beta,E,Lk(1,k),Lk(2,k),'spline');
            elseif Lk(1,k) == Lk(1,k-1)
                sum = sum - interp2(alpha,beta,E,Lk(1,k),Lk(2,k),'spline');
            end
        end
        Mk = -interp2(alpha,beta,E,Lk(1,2),Lk(2,2),'spline') + 2*sum;
    elseif Lk(1,3) < -Lk(2,3) %Az alpha = -beta egyenes alatt vagyunk
        sum = 0;
        for k = 3:size(Lk,2)
            if k == 3
                sum = sum + (interp2(alpha,beta,E,Lk(1,k-1),Lk(2,k-1),'spline') ...
                    - interp2(alpha,beta,E,Lk(1,k),Lk(2,k),'spline'));
            elseif Lk(1,k) == Lk(1,k-1)
                sum = sum + interp2(alpha,beta,E,Lk(1,k),Lk(2,k),'spline');
            elseif Lk(2,k) == Lk(2,k-1)
                sum = sum - interp2(alpha,beta,E,Lk(1,k),Lk(2,k),'spline');
            end
        end
        Mk = interp2(alpha,beta,E,Lk(1,2),Lk(2,2),'spline') - 2*sum;
    end
end

Bk = mu_0 * (Mk*Mmax + hk*Hmax);

%Ideiglenes pont törlése
Lk(:,size(Lk,2)) = [];

Lkm1 = Lk; n = size(Lk,2);

```

E. Függelék

Megoldók

E.1. vezető_lemez_FEM_Preisach.m

```
%Vezető lemez mágneses térben
%Skalár Preisach-modellel

clear;
clc;

%Geometria, végelelemes felosztás

a = 0.175e-3; %-a...+a
N = 100;
x = [-a:(2*a)/N:a];

%Anyagparaméterek /Lsd: script.m/

mu_min = 6.6480e-05;
mu_max = 0.0052;
mu0 = (mu_min + mu_max) / 2;

sigma = 2.22e6;

%Everett-függvény

alpha = load('alpha');
beta = load('beta');
E = load('E');

%Mért B-H karakterisztika

BH = load('BH3000m.lvm'); Hmert = BH(:,1); Bmert = BH(:,2);

%Maximum

mu_0 = (4*pi)*1e-7; Mmax = (max(Bmert)/mu_0) - max(Hmert); Hmax = max(Hmert);

%Gerjesztés

H0 = 2000;
f = 500;
t = linspace(0,(2/f),140);
dt = t(2) - t(1);

Hperem = H0 * sin(2*pi*f*t);

%Inicializálás

%Lépcsős függvény
for i = 1:(N+1)
    eval(['Lkm1_' num2str(i) '=[1;-1];']);
end
```

```

VecInck = ones(1,N+1);
%%
Gaussp = [-1/sqrt(3) 1/sqrt(3)];
Gausss = [1 1];
th = 0.5;

K = zeros(N+1);
Blin = zeros(N+1);
Bres = zeros(N+1);

for i = 1:N

    x1 = x(i);
    x2 = x(i+1);

    gN1 = -1/(x2-x1);
    gN2 = 1/(x2-x1);

    Ke = ([gN1;gN2]*[gN1 gN2]) * (x2-x1);

    Be = zeros(2);

    for j = 1:length(Gaussp)

        xgp = (((x2-x1)*Gaussp(j))+(x2+x1))/2;

        N1 = (-xgp+x2)/(x2-x1);
        N2 = (xgp-x1)/(x2-x1);

        Be = Be + ([N1;N2]*[N1 N2]) * Gausss(j) * ((x2-x1)/2);
    end

    K(i,i) = K(i,i) + th * Ke(1,1) * dt / sigma + Be(1,1) * mu0;
    K(i,i+1) = K(i,i+1) + th * Ke(1,2) * dt / sigma + Be(1,2) * mu0;
    K(i+1,i) = K(i+1,i) + th * Ke(2,1) * dt / sigma + Be(2,1) * mu0;
    K(i+1,i+1) = K(i+1,i+1) + th * Ke(2,2) * dt / sigma + Be(2,2) * mu0;

    Blin(i,i) = Blin(i,i) - (1-th) * Ke(1,1) * dt / sigma + Be(1,1) * mu0;
    Blin(i,i+1) = Blin(i,i+1) - (1-th) * Ke(1,2) * dt / sigma + Be(1,2) * mu0;
    Blin(i+1,i) = Blin(i+1,i) - (1-th) * Ke(2,1) * dt / sigma + Be(2,1) * mu0;
    Blin(i+1,i+1) = Blin(i+1,i+1) - (1-th) * Ke(2,2) * dt / sigma + Be(2,2) * mu0;

    Bres(i,i) = Bres(i,i) + Be(1,1);
    Bres(i,i+1) = Bres(i,i+1) + Be(1,2);
    Bres(i+1,i) = Bres(i+1,i) + Be(2,1);
    Bres(i+1,i+1) = Bres(i+1,i+1) + Be(2,2);

end

%K peremf.

K(1,:) = 0;
K(1,1) = 1;
K(N+1,:) = 0;
K(N+1,N+1) = 1;

iK = inv(K);

Hregi = zeros(N+1,1);
Rregi = zeros(N+1,1);
R = zeros(N+1,1);

for i = 1:length(t)

    hiba = 1;
    step = 0;

    while (hiba > 1e-2)

```

```

step = step + 1;
b = Blin * Hregi + Bres * (R-Rregi);

b(1) = Hperem(i);
b(N+1) = Hperem(i);

H = iK * b;
Rutolso = R;

%Modell-function meghívása

for j = 1:(N+1)

    Lkm1 = eval(['Lkm1_' num2str(j)]);
    Inck = VecInck(j);
    Hkm1 = Hregi(j); Hk = H(j);

    [B(j,1),tmp_Lkm1,VecInck(j)] = Preisach(alpha,beta,E,mu_0,Mmax, ...
        Hmax,Lkm1,Inck,Hkm1,Hk);

    eval(['Lkm1_' num2str(j) '=tmp_Lkm1;']);

end

R = B - mu0 * H;

hiba = norm(R-Rutolso)

end

Hregi = H;
Rregi = R;

disp([step hiba]);

figure(4)
axis square;
plot(x,H);
axis([-a,a,-H0,H0]);
xlabel('x [m]');
ylabel('H [A/m]');

figure(2)
axis square;
plot(x,B);
axis([-a,a,-1.5,1.5]);
xlabel('x [m]');
ylabel('B [T]');

figure(1)
hold on;
axis square;
plot(H(10),B(10),'r. ');
xlabel('H [A/m]');
ylabel('B [T]');

end

```

E.2. vezeto_lemez_ketdim.m

```

%Lemez, 2D, FEM, Preisach-modellel

clear;
clc;

load('Nodes.txt');
load('Connect.txt');
load('Dirichlet.txt');

```

```

% figure(1)
% axis square;
% hold all;
% pdeplot(Nodes',[],Connect','mesh','on');
% xlabel('x [m]'); ylabel('y [m]'); axis square;
% axis([0 0.06 0 0.18]);
%%
%Anyagjellemzők

mu_min = 6.6480e-05;
mu_max = 0.0052;
mu0 = (mu_min + mu_max) / 2;

sigma = 2.22e6;

%Gerjesztés

H0 = 2000;
f = 1;
t = linspace(0,(1/f),200);
dt = t(2) - t(1);

Hperem = H0 * sin(2*pi*f*t);

%Everett-függvény

alpha = load('alpha');
beta = load('beta');
E = load('E');

%Mért B-H karakterisztika

BH = load('BH3000m.lvm'); Hmert = BH(:,1); Bmert = BH(:,2);

%Maximum

mu_0 = (4*pi)*1e-7; Mmax = (max(Bmert)/mu_0) - max(Hmert); Hmax = max(Hmert);

%Lépcsős függvény
for i = 1:(size(Nodes,1))
    eval(['Lkm1_' num2str(i) '= [1;-1];']);
end

VecInck = ones(1,size(Nodes,1));

Gauss = [1/3 1/3 1/3];
Gaussp = [2/3 1/6 1/6; 1/6 1/6 2/3; 1/6 2/3 1/6];
th = 1;

K = zeros(size(Nodes,1));
Blin = zeros(size(Nodes,1));
Bres = zeros(size(Nodes,1));

for i = 1:size(Connect,1)

    p1 = Connect(i,1); p2 = Connect(i,2); p3 = Connect(i,3);

    x1 = Nodes(p1,1); x2 = Nodes(p2,1); x3 = Nodes(p3,1);
    y1 = Nodes(p1,2); y2 = Nodes(p2,2); y3 = Nodes(p3,2);

    T = 0.5 * det( [1 x1 y1; 1 x2 y2; 1 x3 y3] );

    %Gradiensek

    gN1x = (0.5*(y2-y3))/T; gN2x = (0.5*(y3-y1))/T; gN3x = (0.5*(y1-y2))/T;
    gN1y = (0.5*(x3-x2))/T; gN2y = (0.5*(x1-x3))/T; gN3y = (0.5*(x2-x1))/T;

    Ke = ( [gN1x;gN2x;gN3x;]*[gN1x,gN2x,gN3x] + [gN1y;gN2y;gN3y;]*[gN1y,gN2y,gN3y] ) * T;

    %Gauss-kvadratura

```

```

a = x2-x1; b = x3-x1; c = x1;
d = y2-y1; e = y3-y1; f = y1;

Be = zeros(3);

for j = 1:3

    kszi = Gaussp(j,2); eta = Gaussp(j,3);

    x = a*kszi + b*eta + c; %Interpolált x
    y = d*kszi + e*eta + f; %Interpolált y

    T1 = 0.5 * det( [ 1 x y; 1 x2 y2; 1 x3 y3 ] ) ;
    T2 = 0.5 * det( [ 1 x1 y1; 1 x y; 1 x3 y3 ] ) ;
    T3 = 0.5 * det( [ 1 x1 y1; 1 x2 y2; 1 x y ] ) ;

    N1 = T1/T;
    N2 = T2/T;
    N3 = T3/T;

    Be = Be + ( [N1;N2;N3] * [N1,N2,N3] ) * Gausss(j) * T/3;
end

K(p1,p1) = K(p1,p1) + th * Ke(1,1) * dt / sigma + Be(1,1) * mu0; K(p1,p2) ...
= K(p1,p2) + th * Ke(1,2) * dt / sigma + Be(1,2) * mu0; K(p1,p3) = K(p1,p3) + ...
th * Ke(1,3) * dt / sigma + Be(1,3) * mu0;
K(p2,p1) = K(p2,p1) + th * Ke(2,1) * dt / sigma + Be(2,1) * mu0; K(p2,p2) ...
= K(p2,p2) + th * Ke(2,2) * dt / sigma + Be(2,2) * mu0; K(p2,p3) = K(p2,p3) + ...
th * Ke(2,3) * dt / sigma + Be(2,3) * mu0;
K(p3,p1) = K(p3,p1) + th * Ke(3,1) * dt / sigma + Be(3,1) * mu0;
K(p3,p2) = K(p3,p2) + th * Ke(3,2) * dt / sigma + Be(3,2) * mu0;
K(p3,p3) = K(p3,p3) + th * Ke(3,3) * dt / sigma + Be(3,3) * mu0;

Blin(p1,p1) = Blin(p1,p1) - (1-th) * Ke(1,1) * dt / sigma + Be(1,1) * mu0; Blin(p1,p2) ...
= Blin(p1,p2) - (1-th) * Ke(1,2) * dt / sigma + Be(1,2) * mu0; Blin(p1,p3) = Blin(p1,p3) ...
- (1-th) * Ke(1,3) * dt / sigma + Be(1,3) * mu0;
Blin(p2,p1) = Blin(p2,p1) - (1-th) * Ke(2,1) * dt / sigma + Be(2,1) * mu0; Blin(p2,p2) ...
= Blin(p2,p2) - (1-th) * Ke(2,2) * dt / sigma + Be(2,2) * mu0; Blin(p2,p3) = Blin(p2,p3) ...
- (1-th) * Ke(2,3) * dt / sigma + Be(2,3) * mu0;
Blin(p3,p1) = Blin(p3,p1) - (1-th) * Ke(3,1) * dt / sigma + Be(3,1) * mu0;
Blin(p3,p2) = Blin(p3,p2) - (1-th) * Ke(3,2) * dt / sigma + Be(3,2) * mu0;
Blin(p3,p3) = Blin(p3,p3) - (1-th) * Ke(3,3) * dt / sigma + Be(3,3) * mu0;

Bres(p1,p1) = Bres(p1,p1) + Be(1,1); Bres(p1,p2) = Bres(p1,p2) + Be(1,2);
Bres(p1,p3) = Bres(p1,p3) + Be(1,3);
Bres(p2,p1) = Bres(p2,p1) + Be(2,1); Bres(p2,p2) = Bres(p2,p2) + Be(2,2);
Bres(p2,p3) = Bres(p2,p3) + Be(2,3);
Bres(p3,p1) = Bres(p3,p1) + Be(3,1); Bres(p3,p2) = Bres(p3,p2) + Be(3,2);
Bres(p3,p3) = Bres(p3,p3) + Be(3,3);
end

Hregi = zeros(size(Nodes,1),1);
Rregi = zeros(size(Nodes,1),1);

R      = zeros(size(Nodes,1),1);

for i = 1:length(t)

    hiba = 1;
    step = 0;

    while ( hiba > 1e-3 )

        step = step + 1;
        b = Blin * Hregi + Bres * (R-Rregi);

        %Peremfeltétel

        for j = 1:length(Dirichlet)
            K(Dirichlet(j),:) = K(Dirichlet(j),:) * 0;
            K(Dirichlet(j),Dirichlet(j)) = 1;

```

```

        b(Dirichlet(j)) = Hperem(i);
    end

    H = K \ b;

    Rutolso = R;

    %Modell-function meghívása

    for j = 1:size(Nodes,1)

        Lkm1 = eval(['Lkm1_' num2str(j)]);
        Inck = VecInck(j);
        Hkm1 = Hregi(j); Hk = H(j);

        [B(j,1),tmp_Lkm1,VecInck(j)] = Preisach(alpha,beta,E,mu_0, ...
            Mmax,Hmax,Lkm1,Inck,Hkm1,Hk);

        eval(['Lkm1_' num2str(j) '=tmp_Lkm1;']);

    end

    R = B - mu0 * H;

    hiba = norm(R-Rutolso)

end

disp([step hiba]);

%Értékátadás

Hregi = H;
Rregi = R;

figure(1)
axis square;
pdeplot(Nodes',[],Connect','xydata',H,'zdata',H, 'colormap','jet');
axis([0 0.06 0 0.16 -H0 H0]);
caxis([-H0 H0]);
xlabel('x [m]');
ylabel('y [m]');
zlabel('H [A/m]');

figure(2)
axis square;
pdeplot(Nodes',[],Connect','xydata',B,'zdata',B, 'colormap','jet');
caxis([-1.4 1.4]);
axis([0 0.06 0 0.16 -1.4 1.4]);
xlabel('x [m]');
ylabel('y [m]');
zlabel('B [T]');

end

```

E.3. epstein_nemlin_Preisach.m

```

%Epstein-keret, 2D, örvényáramú A, nemlineáris, Preisach-modell

clear;
clc;

tic;

Nodes    = load('Nodes.txt') / 1000;
Connect  = load('Connect.txt');

```

```

Dirichlet = load('Dirichlet.txt');
Exc = load('Exc.txt');
%%
%Mesh

% figure(1)
% axis square;
% pdeplot(Nodes',[],Connect(:,2:4)', 'mesh', 'on');
% xlabel('x [m]');
% ylabel('y [m]');

%Anyagjellemzők

mu_min = 0;
mu_max = 0.007418265782302;
mu0 = (mu_min + mu_max) / 2;

nu_max = 1 / mu_min;
nu_min = 1 / mu_max;
nu_lev = 1 / (4*pi*1e-7);

sigma_vas = 2.22e6;
sigma_lev = 0;

% nu_opt = (2*nu_max*nu_min) / (nu_max+nu_min);

nu_opt = 2 / mu_max + mu_min;

%Lépcsős függvény
for i = 1:(size(Connect,1))
    eval(['Lkm1x_' num2str(i) '= [1;-1];']);
end

for i = 1:(size(Connect,1))
    eval(['Lkm1y_' num2str(i) '= [1;-1];']);
end

VecInckx = ones(1,size(Connect,1));
VecIncky = ones(1,size(Connect,1));

%Everett-függvény

alpha = load('alpha');
beta = load('beta');
E = load('E');

%Mért B-H karakterisztika

BH = load('BH3000m.lvm'); Hmert = BH(:,1); Bmert = BH(:,2);

%Maximum

mu_0 = (4*pi)*1e-7; Mmax = (max(Bmert)/mu_0) - max(Hmert); Hmax = max(Hmert);

%Gerjesztés

freq = 1;
t = linspace(0,(1/freq),150);
dt = t(2) - t(1);

K0 = 1500;
Kexc = K0 * sin(2*pi*freq*t);

Gausss = [1/3 1/3 1/3];
Gaussp = [2/3 1/6 1/6; 1/6 1/6 2/3; 1/6 2/3 1/6];

K = zeros(size(Nodes,1));
B = zeros(size(Nodes,1));

for i = 1:size(Connect,1)

    p1 = Connect(i,2); p2 = Connect(i,3); p3 = Connect(i,4);

```

```

x1 = Nodes(p1,1); x2 = Nodes(p2,1); x3 = Nodes(p3,1);
y1 = Nodes(p1,2); y2 = Nodes(p2,2); y3 = Nodes(p3,2);

T = 0.5 * abs(det([1 x1 y1; 1 x2 y2; 1 x3 y3]));

%Gradiensek

gN1x = (0.5*(y2-y3))/T; gN2x = (0.5*(y3-y1))/T; gN3x = (0.5*(y1-y2))/T;
gN1y = (0.5*(x3-x2))/T; gN2y = (0.5*(x1-x3))/T; gN3y = (0.5*(x2-x1))/T;

if Connect(i,1) == 1 %Levegő-tartomány

    nu    = nu_lev;
    sigma = sigma_lev;

elseif Connect(i,1) == 2 %Vas-tartomány

    nu    = nu_opt;
    sigma = sigma_vas;

end

K_1 = nu * ( [gN1x;gN2x;gN3x;]*[gN1x,gN2x,gN3x] + [gN1y;gN2y;gN3y;]*[gN1y,gN2y,gN3y] ) * T;

%Gauss-kvadratúra

a = x2-x1; b = x3-x1; c = x1;
d = y2-y1; e = y3-y1; f = y1;

K_2 = zeros(3);

for j = 1:3

    kszi = Gaussp(j,2); eta = Gaussp(j,3);

    x = a*kszi + b*eta + c;
    y = d*kszi + e*eta + f;

    T1 = 0.5 * abs(det( [1 x y; 1 x2 y2; 1 x3 y3] ));
    T2 = 0.5 * abs(det( [1 x1 y1; 1 x y; 1 x3 y3] ));
    T3 = 0.5 * abs(det( [1 x1 y1; 1 x2 y2; 1 x y] ));

    N1 = T1/T;
    N2 = T2/T;
    N3 = T3/T;

    K_2 = K_2 + sigma /dt * ( [N1;N2;N3]*[N1 N2 N3] ) * Gausss(j) * T;

end

K(p1,p1) = K(p1,p1) + K_1(1,1) + K_2(1,1); K(p1,p2) = K(p1,p2) + K_1(1,2)...
+ K_2(1,2); K(p1,p3) = K(p1,p3) + K_1(1,3) + K_2(1,3);
K(p2,p1) = K(p2,p1) + K_1(2,1) + K_2(2,1); K(p2,p2) = K(p2,p2) + K_1(2,2)...
+ K_2(2,2); K(p2,p3) = K(p2,p3) + K_1(2,3) + K_2(2,3);
K(p3,p1) = K(p3,p1) + K_1(3,1) + K_2(3,1); K(p3,p2) = K(p3,p2) + K_1(3,2)...
+ K_2(3,2); K(p3,p3) = K(p3,p3) + K_1(3,3) + K_2(3,3);

B(p1,p1) = B(p1,p1) + K_2(1,1); B(p1,p2) = B(p1,p2) + K_2(1,2); B(p1,p3) = B(p1,p3) + K_2(1,3);
B(p2,p1) = B(p2,p1) + K_2(2,1); B(p2,p2) = B(p2,p2) + K_2(2,2); B(p2,p3) = B(p2,p3) + K_2(2,3);
B(p3,p1) = B(p3,p1) + K_2(3,1); B(p3,p2) = B(p3,p2) + K_2(3,2); B(p3,p3) = B(p3,p3) + K_2(3,3);

end

b_exc = zeros(size(Nodes,1),1);
b_Adt = zeros(size(Nodes,1),1);
b      = zeros(size(Nodes,1),1);
bIx    = zeros(size(Nodes,1),1);
bIy    = zeros(size(Nodes,1),1);

Aold   = zeros(size(Nodes,1),1);

```

```

Ixold = zeros(size(Connect,1),1);
Iyold = zeros(size(Connect,1),1);

Preisachx = zeros(size(Connect,1),1);
Preisachy = zeros(size(Connect,1),1);

Hxold = zeros(size(Connect,1),1);
Hyold = zeros(size(Connect,1),1);

for i = 1:length(t)

    %Gerjesztés megadása

    for j = 1:size(Exc,1)

        if(Exc(j,1) == 3) %y irányú, pozitív gerjesztésű

            sign = 1; CoordIndex = 2;

        elseif(Exc(j,1) == 4) %y irányú, negatív gerjesztésű

            sign = -1; CoordIndex = 2;

        elseif(Exc(j,1) == 5) %x irányú, pozitív gerjesztésű

            sign = 1; CoordIndex = 1;

        elseif(Exc(j,1) == 6) %x irányú, negatív gerjesztésű

            sign = -1; CoordIndex = 1;

        end

        p1 = Exc(j,2); p2 = Exc(j,3);

        Coord1 = Nodes(p1,CoordIndex); Coord2 = Nodes(p2,CoordIndex);

        dl = abs(Coord2 - Coord1);

        b_exce = [1;1;] * Kexc(i) * 0.5 * dl * sign;

        b_exc(p1) = b(p1) + b_exce(1); b_exc(p2) = b(p2) + b_exce(2);

    end

    %Előző időlépés A-ja alapján

    b_Adt = B * Aold;

    %Indul a fixpont-móka

    eps = 4;

    step = 0;

    while(eps > 3)

        step = step + 1;

        bfp = b_exc + b_Adt - (bIx+bIy);

        %Dirichlet-perem

        for j = 1:length(Dirichlet)

            K(Dirichlet(j),:) = K(Dirichlet(j),:) * 0;
            K(Dirichlet(j),Dirichlet(j)) = 1;

            bfp(Dirichlet(j)) = 0;

        end

    end

```

```

%A számítása

A = K \ bfp;

%Bx és By számítása (rotA)

p1 = Connect(:,2); p2 = Connect(:,3); p3 = Connect(:,4);

x1 = Nodes(p1,1); x2 = Nodes(p2,1); x3 = Nodes(p3,1);
y1 = Nodes(p1,2); y2 = Nodes(p2,2); y3 = Nodes(p3,2);

D = 0.5 * abs( (x1-x3).*(y3-y2) - (y1-y3).*(x3-x2) );

gN1x = 0.5 * (y2-y3) ./ D; gN1y = 0.5 * (x3-x2) ./ D;
gN2x = 0.5 * (y3-y1) ./ D; gN2y = 0.5 * (x1-x3) ./ D;
gN3x = 0.5 * (y1-y2) ./ D; gN3y = 0.5 * (x2-x1) ./ D;

Bx = A(p1,:).*gN1y + A(p2,:).*gN2y + A(p3,:).*gN3y;
By = -A(p1,:).*gN1x - A(p2,:).*gN2x - A(p3,:).*gN3x;

%Hx és Hy saccolása Ixold és Iyold segítségével

Hx = nu_opt * Bx + Ixold;
Hy = nu_opt * By + Iyold;

%Modell-function meghívása (x-irányra)

for j = 1:size(Connect,1)

    Lkm1 = eval(['Lkm1x_' num2str(j)]);
    Inck = VecInckx(j);
    Hkm1 = Hxold(j); Hk = Hx(j);

    [Preisachx(j,1),tmp_Lkm1x,VecInckx(j)] = Preisach(alpha,beta,E,mu_0,...
        Mmax,Hmax,Lkm1,Inck,Hkm1,Hk);

    eval(['Lkm1x_' num2str(j) '=tmp_Lkm1x;']);

end

%Modell-function meghívása (y-irányra)

for j = 1:size(Connect,1)

    Lkm1 = eval(['Lkm1y_' num2str(j)]);
    Inck = VecIncky(j);
    Hkm1 = Hyold(j); Hk = Hy(j);

    [Preisachy(j,1),tmp_Lkm1y,VecIncky(j)] = Preisach(alpha,beta,E,mu_0,...
        Mmax,Hmax,Lkm1,Inck,Hkm1,Hk);

    eval(['Lkm1y_' num2str(j) '=tmp_Lkm1y;']);

end

%Ix és Iy saccolása

Ix = Hx - nu_opt * Preisachx;
Iy = Hy - nu_opt * Preisachy;

%bIx és bIy feltöltése

bIx = zeros(size(Nodes,1),1);
bIy = zeros(size(Nodes,1),1);

for j = 1:size(Connect,1)

    p1 = Connect(j,2); p2 = Connect(j,3); p3 = Connect(j,4);

    x1 = Nodes(p1,1); x2 = Nodes(p2,1); x3 = Nodes(p3,1);

```

```

y1 = Nodes(p1,2); y2 = Nodes(p2,2); y3 = Nodes(p3,2);

T = 0.5 * abs(det([1 x1 y1; 1 x2 y2; 1 x3 y3]));

%Gradiensek

gN1x = (0.5*(y2-y3))/T; gN2x = (0.5*(y3-y1))/T; gN3x = (0.5*(y1-y2))/T;
gN1y = (0.5*(x3-x2))/T; gN2y = (0.5*(x1-x3))/T; gN3y = (0.5*(x2-x1))/T;

bIxelem = [gN1y;gN2y;gN3y;] * Ix(j) * T;
bIyelem = -[gN1x;gN2x;gN3x;] * Iy(j) * T;

bIx(p1) = bIx(p1) + bIxelem(1); bIx(p2) = bIx(p2) + bIxelem(2); bIx(p3) =...
    bIx(p3) + bIxelem(3);
bIy(p1) = bIy(p1) + bIyelem(1); bIy(p2) = bIy(p2) + bIyelem(2); bIy(p3) =...
    bIy(p3) + bIyelem(3);

end

eps = abs(max(abs(Ix)) - max(abs(Ixold)))

Ixold = Ix;
Iyold = Iy;

end

Aold = A;
Hxold = Hx;
Hyold = Hy;

figure(1)
axis square;
pdeplot(Nodes',[],Connect(:,2:4)', 'xydata', Bx+By, 'colormap', 'jet', 'mesh', 'on');
caxis([-1.2 1.2]);
xlabel('x [m]');
ylabel('y [m]');
ylabel(colorbar, 'B [T]');

figure(2)
axis square;
pdeplot(Nodes',[],Connect(:,2:4)', 'xydata', Hx+Hy, 'colormap', 'jet', 'mesh', 'on');
caxis([-2000 2000]);
xlabel('x [m]');
ylabel('y [m]');
ylabel(colorbar, 'H [A/m]');

figure(3)
hold on;
plot(Hx(1468), Bx(1468), 'r+');

% figure(3)
% axis square;
% pdeplot(Nodes',[],Connect(:,2:4)', 'xydata', A, 'contour', 'on');
% xlabel('x [m]');
% ylabel('y [m]');
% ylabel(colorbar, 'A [Vs/m]');

end

toc;

```