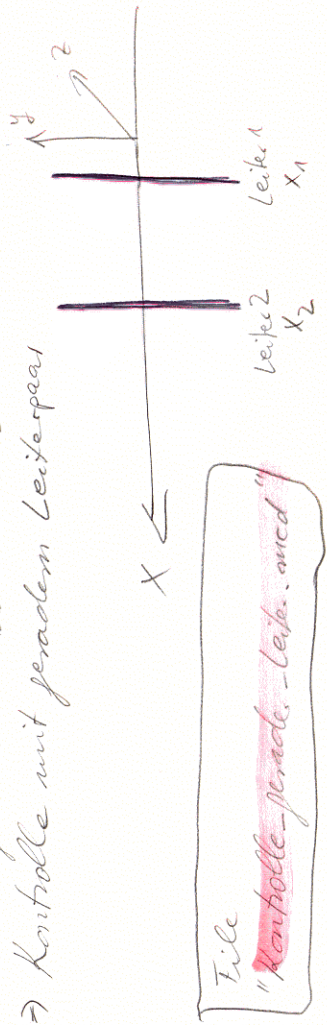




Die Ergebnisse im File "Biot-Savart. med" kommen mir irgendwie zu klein vor. Die Kraft müsste größer sein

→ Kontrolle mit parallelen Leiterpaar



Parametrisierung  $\vec{s}_1(t) = \begin{pmatrix} x_1 \\ \omega_1 t + \varphi \\ 0 \end{pmatrix} \Rightarrow \vec{v}_1 = \dot{\vec{s}}_1 = \begin{pmatrix} 0 \\ \omega_1 \\ 0 \end{pmatrix}$

$\vec{s}_2(t) = \begin{pmatrix} x_2 \\ \omega_2 t + \varphi \\ 0 \end{pmatrix}$

$\Rightarrow \vec{v}_1 \times (\vec{s}_1 - \vec{s}_2) = \begin{pmatrix} 0 \\ \omega_1 (x_2 - x_1) \\ 0 \end{pmatrix}$  und  $|\vec{s}_1 - \vec{s}_2|^3 = (|x_1 - x_2|^2 + |\omega_1 t - \omega_2 t|^2)^{3/2}$

$\Rightarrow dH_z = \frac{\gamma_1}{2\omega_1} \cdot \int_0^1 \frac{\vec{v}_1 \times (\vec{s}_1 - \vec{s}_2)}{|\vec{s}_1 - \vec{s}_2|^3} \cdot \frac{d\varphi}{2\pi} \Rightarrow dH_z = \frac{\gamma_1}{2\omega_1} \cdot \int_0^1 \frac{\omega_1 (x_2 - x_1)}{[|x_1 - x_2|^2 + |\omega_1 t - \omega_2 t|^2]^{3/2}} \cdot \frac{d\varphi}{2\pi} = \frac{1}{4\pi} \frac{A}{m}$

Damit wird die Kraft-Berechnung nach  $d\vec{F}_2 = \begin{pmatrix} 0 \\ \gamma_2 \\ 0 \end{pmatrix}$  für ein infinitesimales Leitersegment von Leiter 2 zu:

$d\vec{F} = \gamma_2 \cdot (d\vec{L}_2 \times \vec{B}) = \mu_0 \gamma_2 \cdot \begin{pmatrix} 0 \\ \gamma_2 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ \gamma_1 \pi \end{pmatrix} = \begin{pmatrix} \frac{1}{8} \cdot \frac{1}{4\pi} \\ 0 \\ 0 \end{pmatrix} \Rightarrow \vec{F}_{\text{ges}} = 8 \cdot d\vec{F} = 4\pi \cdot 10^{-7} \begin{pmatrix} \frac{1}{4\pi} \\ 0 \\ 0 \end{pmatrix} N = 10^{-7} N$

Das ist im wesentlichen richtig,  
evtl. bis auf einen Faktor 2,  
et Tipler S. 1061

Nach dieser Kontrolle ist von der Tauglichkeit  
des Formel ( $x^2$  von S. 42) auszugehen.

Die Kraft bei den Leiterschleifen ist offensichtlich sehr viel geringer als bei geraden parallelen Leitern,  
weil die Elektronen sich auch noch in Leiterrichtung gegenseitig beschleunigen bzw. bremsen (gemäß  
eine Komponente der Lorentzkraft in Richtung des Leiters) Außerdem kompensiert die eine Hälfte eines  
magnetischen Leiterschleife teilweise die andere Hälfte.

⇒ Wir arbeiten also jetzt mit der Kraft-Berechnung gemäß S. 40-42.

"Var-L-002.dpr"

Das programmiere ich jetzt in das File

Geometrische Anordnung nach ( $x^1$  von S. 36) und ( $x^2$  von S. 37), Kraftberechnung nach S. 40-42.

Das funktioniert  
nicht im Moment  
"Var-L-002.dpr"

Das ist die Kraft zwischen 2 Leiterschleifen



```

Program Konverter_mit_variablen_Spule;
{$APPTYPE CONSOLE}
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

Const AnzP=24; {200000} {Anzahl der Zeitschritte zur Lsg. der Dgl.}

Type Feld = Array[0..AnzP] of Double;

Var epo,muo : Double; {Naturkonstanten}
    c : Double; {Lichtgeschwindigkeit}
    ls,bs : Double; {Spulenlänge, Spulenbreite, Schleifen 1 nach *2 von S.40}
    di,da : Double; {Innendurchmesser und Außendurchmesser des Spulenkörpers}
    Dd : Double; {Drahtdurchmesser}
    n,m : LongInt; {Windungszahl, radial und axial der Spule}
    dk,lk : Double; {Spulenkern (Magnetkern) -> Durchmesser und Länge, Schleifen 2 nach *2 von S.40}
    dki : Double; {Hilfsgröße: Innendurchmesser zur Simulation der Magnetkern-Windungen}
    nk,mk : LongInt; {Windungszahl, radial und axial zur Simulation des Magnetkerns}
    dt : Double; {Dauer der Zeitschritte zur Lsg. der Dgl.}
    Abstd : Integer; {Jeder wievielte Punkte soll geplottet werden}
    Ikern : Double; {Strom(DC) zwecks Simulation des Feldes des Magnetkerns}
    Br : Double; {remanentes longitudinales Feld des Magnetkerns}
    i : LongInt; {nur für Testzwecke: Laufvariable}
    Fl,F2 : Feld; {nur für Testzwecke: Arrays für Daten-Output in Excel}
    xkern : Double; {nur für Testzwecke: x-Position des Magnetkerns}

Procedure Wait;
Var Ki : Char;
begin
  Write('<W>'); Read(Ki); Write(Ki);
  If Ki='e' then Halt;
end;

Procedure ExcelAusgabe(Name:String;Spalten:Integer;KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,KK,KL:Feld);
Var fout : Text; {Bis zu 12 Spalten zum Aufschreiben der Ergebnisse}
    lv,j,k : Integer; {Laufvariable}
    Zahl : String; {Die ins Excel zu druckenden Zahlen}
begin
  Assign(fout,Name); Rewrite(fout); {File öffnen}
  For lv:=0 to AnzP do {von "plotanf" bis "plotend"}
  begin
    If (lv mod Abstd)=0 then
    begin
      For j:=1 to Spalten do
      begin {Kolumnen drucken}
        If j=1 then Str(KA[lv]:17:11,Zahl);
        If j=2 then Str(KB[lv]:17:11,Zahl);
        If j=3 then Str(KC[lv]:17:11,Zahl);
        If j=4 then Str(KD[lv]:17:11,Zahl);
        If j=5 then Str(KE[lv]:17:11,Zahl);
        If j=6 then Str(KF[lv]:17:11,Zahl);
        If j=7 then Str(KG[lv]:17:11,Zahl);
        If j=8 then Str(KH[lv]:17:11,Zahl);
        If j=9 then Str(KI[lv]:17:11,Zahl);
        If j=10 then Str(KJ[lv]:17:11,Zahl);
        If j=11 then Str(KK[lv]:17:11,Zahl);
        If j=12 then Str(KL[lv]:17:11,Zahl);
        For k:=1 to Length(Zahl) do
        begin {Keine Dezimalpunkte verwenden, sondern Kommata}
          If Zahl[k]<>'.' then write(fout,Zahl[k]);
          If Zahl[k]='.' then write(fout,',');
        end;
        Write(fout,chr(9)); {Daten-Trennung, Tabulator}
      end;
      Writeln(fout,''); {Zeilen-Trennung}
    end;
  end;
  Close(fout);
end;

Function Fx(r1,r2,l1,l2,x1,x2:Double):Double; {Kraft zwischen den beiden Leiterschleifen (in x-Richtung)}
Var q1,oml,Hy : Double;
Function Fkt(phi:Double):Double; {Die zu integrierende Funktion}
Var Nen : Double;
begin
  Nen:=Sqr(x1-x2)+Sqr(r1*sin(phi)-r2)+Sqr(r1*cos(phi));
  Fkt:=-r1*oml*sin(phi)*(x1-x2)/Sqrt(Nen*Nen*Nen);
end;
Function Simpson(n:LongInt):Double; {Numerische Integration nach dem Simpson-Verfahren}
Var ug,og : Double; {Integralgrenzen}
    i : LongInt; {Laufvariable}
    sum : Double; {Wert des Integrals}
begin
  ug:=0; og:=2*pi; {Untergrenze und Obergrenze der Integration}
  Sum:=Fkt(ug)+Fkt(og);
  For i:=1 to 2*n-1 do

```

```

begin
  If (i mod 2)=0 then Sum:=Sum+2*Fkt(ug+i*(og-ug)/2/n);
  If (i mod 2)=1 then Sum:=Sum+4*Fkt(ug+i*(og-ug)/2/n);
end;
Simpson:=Sum*(og-ug)/2/n/3;
end;
Function Integral:Double;
Var Streif      : LongInt;
    Wert1,Wert2 : Double;
begin
  Streif:=1000;
  Wert2:=Simpson(Streif);
  Repeat
    Wert1:=Wert2;
    Streif:=Streif*2;
    Wert2:=Simpson(Streif);
  Until (Wert2-Wert1)<1E-8;
  Integral:=Wert2;
end;
Function mur:Double;      {Hier kann eine echte Hysterese-Schleife definiert werden !}
Var Wert,Anteil : Double; {Sogar Ummagnetisierungs-Verluste können abgebildet werden.}
begin
  Anteil:=(muo*Hy)/2.00; {Anteil an der Sättigungsmagnetisierung von 2.00 Tesla.}
  Wert:=1+100*Anteil;
{  Wert:=1+0*Wert;      {Falls ich ohne mur arbeiten will.}
  mur:=Wert;
end;
begin
{ For i:=0 to 20 do begin Hy:=i*1E5; Writeln(mu*Hy, ' Tesla => mur = ',mur); end; Wait; Wait; Halt; {Kontrolle
von mur}
  q1:=1; oml:=I1*2*pi/q1;
  Hy:=I1/4/pi/oml*Integral;
  Fx:=-muo*mur*I1*I2/2/oml*r2*Hy;
end;

Function Remanenz(Br:Double):Double; {Simulation als longitudinales Feld in x-Richtung in der Kern-Achse}
Var Bist : Double; {Ist-Wert der magnetischen Induktion}
    Istart : Double; {Strom-Anfangswert zu Berechnungszwecken}
Function H:Double;
Var ax : Double; {Aufpunkt zur Bestimmung der Feldstärke}
    i,j : LongInt;
    Hij,sum,kl1,kl2 : Double;
begin
  ax:=lk; {Aufpunkt zur Bestimmung der Feldstärke: Er liegt am Ende des Magnetkerns}
  Istart:=1; {Ampere} {Strom-Anfangswert zu Berechnungszwecken}
  sum:=0; {Feldstärke}
  For i:=1 to nk do
  begin
    For j:=1 to mk do
    begin
      kl1:=Sqr(dki/2+(i-0.5)*Dd);
      kl2:=Sqr(ax-(j-0.5)*Dd);
      Hij:=Istart*kl1/2/Sqrt((kl1+kl2)*(kl1+kl2)*(kl1+kl2));
      sum:=sum+Hij;
    end;
  end;
  H:=sum;
end;
begin
  Bist:=muo*H;
  Remanenz:=Istart*Br/Bist;
end;

Function Fges(kernpos:Double):Double; {Gesamtkraft zwischen Spule und Magnetkern, x = Position des Magnetkerns}
Var ins,ims : LongInt; {Zählvariablen für alle Windungen der Spule}
    ink,imk : LongInt; {Zählvariablen für alle Windungen des Magnetkerns}
    Fsum : Double; {Kraft-Summe}
    xs,ys,xk,yk : Double; {Positionen der Windungen}
begin
  Fsum:=0;
  For ins:=1 to n do {Spule radial}
  begin
    For ims:=1 to m do {Spule axial}
    begin
      For ink:=1 to nk do {Kern radial}
      begin
        For imk:=1 to mk do {Kern axial}
        begin
          xs:=(ims-0.5)*Dd; {Spule axial}
          ys:=di/2+(ins-0.5)*Dd; {Spule radial, radiale Position entspricht dem halben Durchmesser}
          xk:=kernpos+(imk-0.5)*Dd; {Kern axial}
          yk:=dki/2+(ink-0.5)*Dd; {Kern radial, radiale Position entspricht dem halben Durchmesser}
          Fsum:=Fsum+Fx(ys,yk,1.00,Ikern,xs,xk);
        end;
      end;
    end;
  end;
  Write('.');

```

[illegible]

- Um schneller arbeiten zu können, habe ich einen exemplarischen Kraft-Weg-Daten Satz (Date 0...40) in den Force-code "empbadt" eingebettet  
→ Darstellung in "Var-L-003-Kraftkurve.xls"
- Um ein Gefühl für den Verlauf einer geeigneten Kurve zu bekommen nehme ich das File "Regression.mcd".

Als nichtlineare Fit wähle ich  $y = a_1 \cdot x^{a_2} \cdot e^{-a_3 \cdot x}$

Startwerte für die Parameter  $a_1, a_2, a_3$  (für die Regression) ermittle ich so:

Die ersten 5 Werte um den Kraftnullpunkt gehen in eine Wurzelfit  $y_w = a_1 \cdot x^{a_2}$  weil dort die exponentielle Dämpfung noch nicht viel ausmacht  
Wir logarithmieren die Wurzelfit

$$y_w = a_1 \cdot x^{a_2} \quad | \ln$$

$$\ln y_w = \ln a_1 + a_2 \cdot \ln x \quad \text{und bekommen eine lineare Regression zwischen } \ln y \text{ und } \ln x$$

Damit bekomme ich die Startwerte für die beiden Parameter  $a_1$  und  $a_2$ .

Den Startwert für den Parameter  $a_3$  der exponentiellen Dämpfung bekomme ich, indem ich herausfinde, wie stark ich die dämpfende exp-Fkt am letzten Punkt herunterrechnen muß:

$$y = y_w \cdot e^{-a_3 \cdot x} \quad | \cdot \frac{1}{y_w}$$

$$\Rightarrow \frac{y}{y_w} = e^{-a_3 \cdot x} \quad | \ln$$

$$\Rightarrow \ln\left(\frac{y}{y_w}\right) = -a_3 \cdot x \quad | \cdot \frac{-1}{x}$$

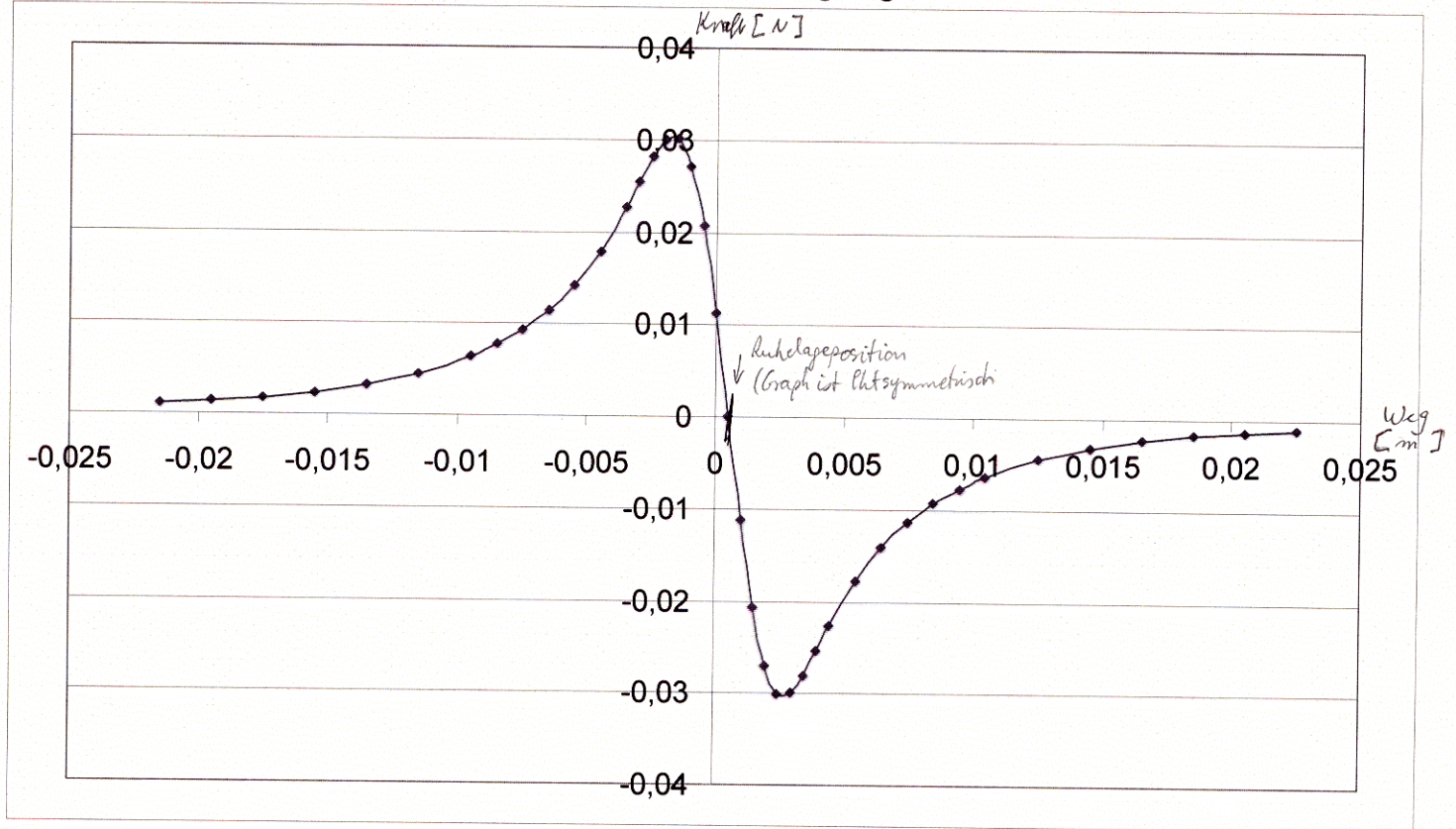
$$\Rightarrow a_3 = \frac{-1}{x} \cdot \ln\left(\frac{y}{y_w}\right) = \frac{1}{x} \cdot \ln\left(\frac{y_w}{y}\right) \quad \text{wo ich für } x \text{ den letzten (äußersten) Punkt einsetze.}$$

$$\Rightarrow a_3 = \frac{1}{x} \cdot (\ln(y_w) - \ln(y)) = \frac{1}{x} \cdot (\ln(a_1) - a_2 \cdot \ln(x))$$

$\uparrow$  im letzten Plot       $\uparrow$  im letzten Plot       $\uparrow$  im letzten Plot

Der Fit läuft jetzt mit nur 3 Parametern  $A13, A23, A33$ , "Var-L-003.dfr"

# Entwicklung einer Kraft-Weg-Regression



Jetzt will ich noch eine Abgabe einbauen, ob an den Spulen- und Magnetem-Messungen etwas verändert wurde. Wenn das der Fall ist, muß ich die Kraft-Weg-Berechnung inkl. Regression neu durchführen. Wenn die Geometrie unverändert ist, benutze ich die zuletzt abge-speicherten Daten als Kraft-Weg-Berechnung und -Regression aus dem letzten Lauf.

→ Das steht in File "Kr-L-004.dpr"  
 dabei werden vorhandene Werte abgespeichert im File "Spule-mech.dat"

So, damit kann ich nun drangehen, den Konverter zu berechnen. → "Var-L-05.dpr"

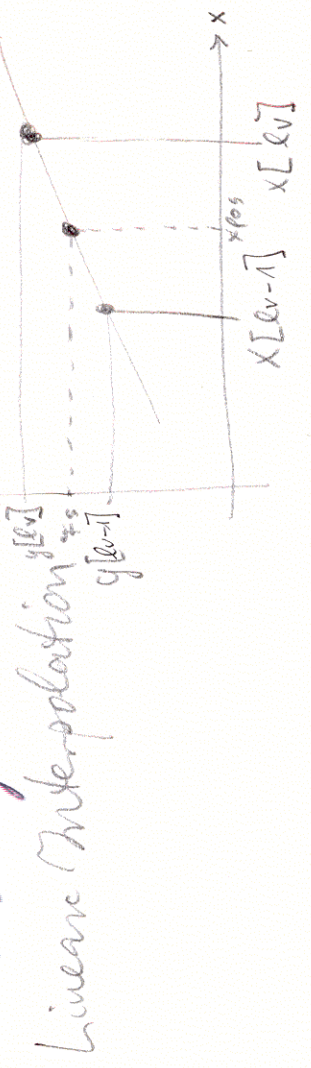
Halt: Bei der Qualitätsüberprüfung merke ich, daß die Regressionen für einige Spulengeometrie-Fälle sehr schlecht wird und die  $F_z$  nicht mehr viel mit der ersten Kraft  $F_{z0}$  zu tun hat.

⇒ Wir verwenden eine einfache lineare Interpolation. Die ist immer noch viel besser als die Regressionen.

Dazu muß ich nochmals weitere Werte eintragen und neu programmieren.  
 Das baue nochmals auf "Kr-L-002.dpr" auf.

"Var-L-006.dpr"

→ Der Algorithmus, zählt jetzt weiter mit



⇒ Kraftberechnung  $F_z(x_{pos})$   
 läuft mit perfekter Genauigkeit

Dokumentation: Files

"Var-L-006-Kraftkurve.xls"

"Var-L-006-Schnellkurve.xls"



24.5.2024  
Program Konverter\_mit\_variablen\_Spule;

{\$APPTYPE CONSOLE}

uses  
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

Const AnzPmax=2000; {Anzahl der Zeitschritte zur Lsg. der Dgl.}

Type Feld = Array[0..AnzPmax] of Double;

Var epo,muo : Double; {Naturkonstanten}  
cl : Double; {Lichtgeschwindigkeit}  
ls,bs : Double; {Spulenlänge,Spulenbreite, Schleifen 1 nach \*2 von S.40}  
L : Double; {Induktivität der Spule 1}  
di,da : Double; {Innendurchmesser und Außendurchmesser des Spulenkörpers}  
Dd : Double; {Drahtdurchmesser}  
n,m : LongInt; {Windungszahl, radial und axial der Spule}  
dk,lk : Double; {Spulenkern (Magnetkern) -> Durchmesser und Länge, Schleifen 2 nach \*2 von S.40}  
dki : Double; {Hilfsgröße: Innendurchmesser zur Simulation der Magnetkern-Windungen}  
nk,mk : LongInt; {Windungszahl, radial und axial zur Simulation des Magnetkerns}  
dt : Double; {Dauer der Zeitschritte zur Lsg. der Dgl.}  
AnzP : LongInt; {Anzahl der tatsächlich berechneten Zeit-Schritte}  
Abstd : Integer; {Jeder wievielte Punkte soll geplottet werden}  
Ikern : Double; {Strom(DC) zwecks Simulation des Feldes des Magnetkerns}  
Br : Double; {remanentes longitudinales Feld des Magnetkerns}  
i,j : LongInt; {nur für Testzwecke: Laufvariable}  
AnzSch,AnzAlt : Integer; {Anzahl der Stützpunkte für die schnelle Kraftberechnung}  
XX,YY,xs,ys : Array[-500..+500] of Double; {Maximal 1000 Stützpunkte für die schnelle Kraftberechnung}  
sw : Double; {Schrittweite für die Kraftermittlung}  
imax : Integer; {An dieser Stelle liegt das Kraftmaximum}  
xkern : Double; {x-Position des Magnetkerns}  
C : Double; {Kapazität des Kondensators}  
rho,R : Double; {Spezifischer und Ohm'scher Widerstand des Spulendrahtes}  
RD,RL : Double; {Drahtwiderstand des Spulendrahtes und Lastwiderstand, beide Ohm'sch}  
Q,Qp,Qpp : Feld; {Ladung auf dem Kondensator als Fkt der Zeit}  
x,xp,xpp : Feld; {Auslenkung jeder einzelnen Kondensatorplatte}  
OMel,OMmech : Double; {elektrische und mechanische Kreisfrequenz der schwingenden Systeme}  
UC,Ul : Double; {Kondensatorspannung}

Procedure Wait;

Var Ki : Char;

begin

Write('<W>'); Read(Ki); Write(Ki);

If Ki='e' then Halt;

end;

Procedure ExcelAusgabe(Name:String;Spalten:Integer;KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,KK,KL:Feld);

Var fout : Text; {Bis zu 12 Spalten zum Aufschreiben der Ergebnisse}

lv,j,k : Integer; {Laufvariable}

Zahl : String; {Die ins Excel zu druckenden Zahlen}

begin

Assign(fout,Name); Rewrite(fout); {File öffnen}

For lv:=0 to AnzP do {von "plotanf" bis "plotend"}

begin

If (lv mod Abstd)=0 then

begin

For j:=1 to Spalten do

begin {Kolumnen drucken}

If j=1 then Str(KA[lv]:17:11,Zahl);

If j=2 then Str(KB[lv]:17:11,Zahl);

If j=3 then Str(KC[lv]:17:11,Zahl);

If j=4 then Str(KD[lv]:17:11,Zahl);

If j=5 then Str(KE[lv]:17:11,Zahl);

If j=6 then Str(KF[lv]:17:11,Zahl);

If j=7 then Str(KG[lv]:17:11,Zahl);

If j=8 then Str(KH[lv]:17:11,Zahl);

If j=9 then Str(KI[lv]:17:11,Zahl);

If j=10 then Str(KJ[lv]:17:11,Zahl);

If j=11 then Str(KK[lv]:17:11,Zahl);

If j=12 then Str(KL[lv]:17:11,Zahl);

For k:=1 to Length(Zahl) do

begin {Keine Dezimalpunkte verwenden, sondern Kommata}

If Zahl[k]<>'.' then write(fout,Zahl[k]);

If Zahl[k]='.' then write(fout,',');

end;

Write(fout,chr(9)); {Daten-Trennung, Tabulator}

end;

Writeln(fout,''); {Zeilen-Trennung}

end;

end;

Close(fout);

end;

Procedure Kraftwerte\_ins\_Excel;

Var fout : Text;

lv,j,k : Integer; {Laufvariablen}

Zahl : String; {Die ins Excel zu druckenden Zahlen}

begin

Assign(fout,'Kraftwerte.dat'); Rewrite(fout); {File öffnen}

For lv:=-AnzSch to AnzSch do {von "plotanf" bis "plotend"}

```

begin
  For j:=1 to 2 do
    begin {Kolumnen drucken}
      If j=1 then Str(XX[lv]*100:17:11,Zahl); {Zentimeter}
      If j=2 then Str(YY[lv]:17:11,Zahl); {Newton}
      For k:=1 to Length(Zahl) do
        begin {Keine Dezimalpunkte verwenden, sondern Kommata}
          If Zahl[k]<>'.' then write(fout,Zahl[k]);
          If Zahl[k]='.' then write(fout,',');
        end;
        Write(fout,chr(9)); {Daten-Trennung, Tabulator}
      end;
      Writeln(fout,','); {Zeilen-Trennung}
    end;
  Close(fout);
end;

Function Fx(r1,r2,I1,I2,x1,x2:Double):Double; {Kraft zwischen den beiden Leiterschleifen (in x-Richtung)}
Var ql,oml,Hy : Double;
Function Fkt(phi:Double):Double; {Die zu integrierende Funktion}
Var Nen : Double;
begin
  Nen:=Sqr(x1-x2)+Sqr(r1*sin(phi)-r2)+Sqr(r1*cos(phi));
  Fkt:=-r1*oml*sin(phi)*(x1-x2)/Sqrt(Nen*Nen*Nen);
end;
Function Simpson(n:LongInt):Double; {Numerische Integration nach dem Simpson-Verfahren}
Var ug,og : Double; {Integralgrenzen}
    i : LongInt; {Laufvariable}
    sum : Double; {Wert des Integrals}
begin
  ug:=0; og:=2*pi; {Untergrenze und Obergrenze der Integration}
  Sum:=Fkt(ug)+Fkt(og);
  For i:=1 to 2*n-1 do
    begin
      If (i mod 2)=0 then Sum:=Sum+2*Fkt(ug+i*(og-ug)/2/n);
      If (i mod 2)=1 then Sum:=Sum+4*Fkt(ug+i*(og-ug)/2/n);
    end;
  Simpson:=Sum*(og-ug)/2/n/3;
end;
Function Integral:Double;
Var Streif : LongInt;
    Wert1,Wert2 : Double;
begin
  Streif:=1000;
  Wert2:=Simpson(Streif);
  Repeat
    Wert1:=Wert2;
    Streif:=Streif*2;
    Wert2:=Simpson(Streif);
  Until (Wert2-Wert1)<1E-8;
  Integral:=Wert2;
end;
Function mur:Double; {Hier kann eine echte Hysterese-Schleife definiert werden !}
Var Wert,Anteil : Double; {Sogar Ummagnetisierungs-Verluste können abgebildet werden.}
begin
  Anteil:=(muo*Hy)/2.00; {Anteil an der Sättigungsmagnetisierung von 2.00 Tesla.}
  Wert:=1+100*Anteil;
  Wert:=1+0*Wert; {Falls ich ohne MUR arbeiten will.}
  mur:=Wert;
end;
begin
{ For i:=0 to 20 do begin Hy:=i*1E5; Writeln(muo*Hy,' Tesla => mur = ',mur); end; Wait; Wait; Halt; {Kontrolle mur}
  ql:=1; oml:=I1*2*pi/ql;
  Hy:=I1/4/pi/oml*Integral;
  Fx:=-muo*mur*I1*I2/2/oml*r2*Hy;
end;

Function Remanenz(Br:Double):Double; {Simulation als longitudinales Feld in x-Richtung in der Kern-Achse}
Var Bist : Double; {Ist-Wert der magnetischen Induktion}
    Istart : Double; {Strom-Anfangswert zu Berechnungszwecken}
Function H:Double;
Var ax : Double; {Aufpunkt zur Bestimmung der Feldstärke}
    i,j : LongInt;
    Hij,sum,kl1,kl2 : Double;
begin
  ax:=lk; {Aufpunkt zur Bestimmung der Feldstärke: Er liegt am Ende des Magnetkerns}
  Istart:=1; {Ampere} {Strom-Anfangswert zu Berechnungszwecken}
  sum:=0; {Feldstärke}
  For i:=1 to nk do
    begin
      For j:=1 to mk do
        begin
          kl1:=Sqr(dki/2+(i-0.5)*Dd);
          kl2:=Sqr(ax-(j-0.5)*Dd);
          Hij:=Istart*kl1/2/Sqrt((kl1+kl2)*(kl1+kl2)*(kl1+kl2));
          sum:=sum+Hij;
        end;
      end;
    end;
  H:=sum;
end;

```

```

begin
  Bist:=muo*H;
  Remanenz:=Istart*Br/Bist;
end;

Function Fges(kernpos:Double):Double; {Gesamtkraft zwischen Spule und Magnetkern, x = Position des Magnetkerns}
Var ins,ims : LongInt;    {Zählvariablen für alle Windungen der Spule}
    ink,imk : LongInt;    {Zählvariablen für alle Windungen des Magnetkerns}
    Fsum : Double;        {Kraft-Summe}
    xs,ys,xk,yk : Double; {Positionen der Windungen}
begin
  Fsum:=0;
  For ins:=1 to n do {Spule radial}
    begin
      For ims:=1 to m do {Spule axial}
        begin
          For ink:=1 to nk do {Kern radial}
            begin
              For imk:=1 to mk do {Kern axial}
                begin
                  xs:=(ims-0.5)*Dd;          {Spule axial}
                  ys:=di/2+(ins-0.5)*Dd;    {Spule radial, radiale Position entspricht dem halben Durchmesser}
                  xk:=kernpos+(imk-0.5)*Dd; {Kern axial}
                  yk:=dki/2+(ink-0.5)*Dd;   {Kern radial, radiale Position entspricht dem halben Durchmesser}
                  Fsum:=Fsum+Fx(ys,yk,1.00,Ikern,xs,xk);
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  Fges:=Fsum;
end;

Function Fs(xpos:Double):Double;
Var lv : Integer;
    merk : Double;
begin
  lv:=AnzSch;    merk:=0;
  If xkern<=XX[-lv] then merk:=YY[-lv];
  If xkern>=XX[+lv] then merk:=YY[+lv];
  If (xkern>XX[-lv])and(xkern<XX[+lv]) then
    begin
      lv:=-AnzSch-1; Repeat lv:=lv+1; Until XX[lv]>xpos;
      merk:=YY[lv-1]+(xpos-XX[lv-1])/(XX[lv]-XX[lv-1])* (YY[lv]-YY[lv-1]); {Lineare Interpolation}
    end;
  Fs:=merk;
end;

Begin {Hauptprogramm}
{ Initialisierung - Vorgabe der Werte: }          {Wir arbeiten in SI-Einheiten}
Writeln('Leistungsstarker Raumenergie-Konverter: Versuch mit variabler Spule'); Writeln;
{ Allgemeine Werte-Vorgaben -> Input-Parameter:}
epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}
muo:=4*pi*1E-7{Vs/Am};      {Elektrische Feldkonstante}
cl:=Sqrt(1/muo/epo){m/s};    {Lichtgeschwindigkeit} Writeln('Lichtgeschwindigkeit c = ',cl, ' m/s');
{ Spule 1, außen, Bleistift in *1 von S.36:}
m:=4;          {Spule 1: Windungszahl axial, Anzahl der Windungen pro Lage, nebeneinander}
n:=1;          {Spule 1: Windungszahl radial, Anzahl der Lagen, übereinander}
Dd:=1.0E-3; {Meter}      {Drahtdurchmesser (für Spule 1 und ebenso für Magnetkern-Simulation)}
di:=0.02; {Meter}        {Spuleninnendurchmesser, Spule 1}
{ Magnetkern, entspricht Leiterschleifen 2, innen, grün in *1 von S.36:}
mk:=5;          {Magnetkern, Leiter Nr.2: Windungszahl axial, Anzahl der Windungen pro Lage,
nebeneinander}
nk:=1;          {Magnetkern, Leiter Nr.2: Windungszahl radial, Anzahl der Lagen, übereinander}
dk:=di*0.9;    {Magnetkern-Durchmesser -> Der Magnetkern füllt die Spule nicht ganz aus.}
Br:=1.2; {Tesla}      {remanentes longitudinales Feld des Magnetkerns}
{ mur : Die Permeabilität wird über ein Upgm. eingeführt, damit Hysterese-Schleifen abgebildet werden können }
AnzP:=1000;      {Anzahl der Zeitschritte insgesamt}
dt:=0.001; {sec.}    {Größe der Zeitschritte}
Abstd:=1;        {Jeder wievielte Punkte soll geplottet werden}
{ Andere Komponenten (Kondensator und Widerstand):}
C:=200E-6; {Farad}    {Kapazität des Kondensators}
rho:=1.7E-8; {Ohm*m}   {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
RD:=rho*(2*pi*di*n*m)/(pi*(Dd/2)*(Dd/2)){Ohm}; {Drahtwiderstand: Ohm'scher Widerstand des Spulendrahtes}
RL:=1000*RD;          {Lastwiderstand, einstellbar je nach Verbraucher}
R:=RL+RD;             {gesamter dämpfender Widerstand}
{ Abgeleitete Werte, keine Eingabe möglich: Die Werte dienen dem bequemeren Rechnen:}
ls:=m*Dd; bs:=n*Dd;    {Spulenlänge,Spulenbreite}
da:=di+2*bs;           {Spulen-Aussendurchmesser}
lk:=mk*Dd;             {Magnetkern-Länge -> Der Magnetkern füllt die Spule aus.}
dki:=dk-2*nk*Dd;       {Hilfsgröße: Innendurchmesser zur Simulation der Magnetkern-Windungen}
Ikern:=Remanenz(Br);    {Strom(DC) zwecks Simulation des Feldes des Magnetkerns}
L:=n*pi*m*m*(pi*Sqr(da/2))/ls; {Induktivität der zylindrischen Spule, ohne Magnetkern drinnen, also ohne mur}
OMel:=1/Sqrt(L*C);      {Eigen-Kreisfrequenz der harmonischen elektrischen Schwingkreises}
{ Anzeige der Werte:}
Writeln('Spulenlaenge ls = ',ls*100:9:5,' cm, und Spulenkoerper-Breite bs = ',bs*100:9:5,' cm ');
Writeln('Spulen: Innendrm di = ',di*100:9:5,' cm, Aussendrm da = ',da*100:9:5,' cm ');
Writeln('Magnetkern: Durchmesser, dk = ',dk*100:9:5,' cm, Laenge lk = ',lk*100:9:5,' cm ');
Writeln('Innendrm der Magnetkern-Simulations-Wdgn: dki = ',dki*100:9:5,' cm');
Writeln('Magnetkern-Strom Ikern zur Simulation des remanenten Feldes:',Ikern:15:5,' Amp');

```

```

Writeln('Kapazitaet des Kondensators: C = ',C,' Farad');
Writeln('Ohm'sche Widerstaende: R = ',R,' Ohm');
Writeln('Induktivitaet der Spule ohne Magnetkern: L = ',L,' Henry');
Writeln('Eigen-Kreisfrequenz harmon. el. Osz.: OMel = ',OMel,' Hz');

{ Hier beginnt das Rechenprogramm:}
Writeln; Writeln('Vorbereitung: Kraft zwischen Spule und Magnetkern, als Fkt der x-Pos des Kerns. ');
XX[0]:=(ls-lk)/2; YY[0]:=Fges(XX[0]); {Ruhelageposition ist immer bei X[0]}
If Abs(YY[0])>1E-10 then
begin
  Writeln('Stoerung. Kraft in Ruhelageposition nicht Null: ',YY[0]:15:7,' N');
  Wait; Wait; Halt;
end;
sw:=(ls-XX[0])/5; {Erster Einstieg für die Schrittweite der Kraftberechnung}
AnzSch:=8; {Gesamtzahl der Schritte ist 2*AnzSch+1, nämlich von -AnzSch .. +AnzSch}
For i:=1 to AnzSch do
begin
  XX[i]:=XX[0]+i*SW; YY[i]:=Fges(XX[i]);
  XX[-i]:=XX[0]-i*SW; YY[-i]:=Fges(XX[-i]);
end;
Repeat {Hier muß ich falls nötig die Schrittweite verfeinern}
  imax:=0; Repeat imax:=imax+1; Until -YY[imax+1]<-YY[imax]; {An dieser Stelle liegt das Kraftmaximum}
  If imax<7 then {In diesem Fall muß ich zusätzliche Stützpunkte dazwischenschieben}
  begin
    i:=1;
    Repeat
      AnzAlt:=AnzSch;
      For j:=AnzAlt downto i do
      begin
        XX[j+1]:=XX[j]; YY[j+1]:=YY[j];
        XX[-j-1]:=XX[-j]; YY[-j-1]:=YY[-j];
      end;
      XX[i]:=(XX[i-1]+XX[i+1])/2; YY[i]:=Fges(XX[i]);
      XX[-i]:=(XX[-i+1]+XX[-i-1])/2; YY[-i]:=Fges(XX[-i]);
      AnzSch:=AnzSch+1;
      i:=i+2
    Until (i>2*imax+3);
  end;
Until (imax>=7); {Jetzt sollte die Schrittweite fein genug sein.}
imax:=0; Repeat imax:=imax+1; Until -YY[imax+1]<-YY[imax]; {An dieser Stelle liegt jetzt das Kraftmaximum}
Repeat {Hier muß ich auch noch nach außen verlängern}
  AnzSch:=AnzSch+1;
  XX[AnzSch]:=XX[AnzSch-1]+sw; YY[AnzSch]:=Fges(XX[AnzSch]);
  XX[-AnzSch]:=XX[-AnzSch+1]-sw; YY[-AnzSch]:=Fges(XX[-AnzSch]);
Until XX[AnzSch]>5*XX[imax];

{ Kontrolle der Kraft-Weg-Daten:} Writeln;
{ For i:=-AnzSch to +AnzSch do Writeln(i,',',XX[i]*100:15:7,' cm',YY[i]:15:7,' N'); }
Kraftwerte_ins_Excel;

{ Hier kommt ein Test der schnellen Kraftberechnung:}
{ For i:=-500 to +500 do
begin xs[i]:=i/1000*(XX[+AnzSch]-XX[-AnzSch]); ys[i]:=Fs(xs[i]); end;
XX:=xs; YY:=ys; AnzSch:=500; Kraftwerte_ins_Excel; }

{ Nun beginnt die DFEM-Simulation der Systems:}
{ Teil_01: Harmonische Schwingung}
Writeln('Teil_01: Harmonische Schwingung');
UC:=10;{Volt} Q[0]:=C*UC;{Coulomb} {Benötigt eine elektrische Spannung zum Starten}
Qpp[0]:=0; Qp[0]:=0; {Startwerte für die Integrationskonstanten}
{ Writeln(' t/[sec.] | Uc/[V] | '); }
For i:=1 to AnzP do {Zunächst noch ohne Daempfung}
begin
  UC:=Q[i-1]/C; UL:=-UC;
  Qpp[i]:=UL/L;
  Qp[i]:=Qp[i-1]+Qpp[i]*dt;
  Q[i]:=Q[i-1]+Qp[i]*dt;
{ Writeln(i*dt:11:9,' | ',Q[i]/C:7:2,' | '); }
end;
ExcelAusgabe('test_01.dat',3,Q,Qp,Qpp,Q,Qp,Qpp,Q,Qp,Qpp,Q,Qp,Qpp);

{ Teil_02: Gedämpfte Schwingung}
Writeln('Teil_02: Gedaempfte Schwingung (harmonisch, plus dazu Widerstand)');
UC:=10;{Volt} Q[0]:=C*UC;{Coulomb} {Benötigt eine elektrische Spannung zum Starten}
Qpp[0]:=0; Qp[0]:=0; {Startwerte für die Integrationskonstanten}
{ Writeln(' t/[sec.] | Uc/[V] | '); }
For i:=1 to AnzP do {Zunächst noch ohne Daempfung}
begin
  Qpp[i]:=-1/L/C*Q[i-1]-R/2/L*Qp[i-1]; {nach *5 von S.6}
  Qp[i]:=Qp[i-1]+(Qpp[i]-R/2/L*Qp[i-1])*dt; {nach *3 & *4 von S.6}
  Q[i]:=Q[i-1]+Qp[i]*dt;
{ Writeln(i*dt:11:9,' | ',Q[i]/C:7:2,' | '); }
end;
ExcelAusgabe('test_02.dat',3,Q,Qp,Qpp,Q,Qp,Qpp,Q,Qp,Qpp,Q,Qp,Qpp);

{ ----- }
Wait; Wait;
End.

```

so will ich nun die induzierte Spannung berechnen, die das Magnetkern durch seine Bewegung in der Spule induziert. Wir beginnen mit der WW jede einzelnen Windung mit jeder einzelnen Windung. Jede einzelne Windung des Magnetkerns (der ja auch als Spule simuliert ist) erzeugt ein Magnetfeld, welches ich in Analogie zu (\* von 5.4.2) berechnen will (nämlich nach Biot-Savart). Im Unterschied sind aber jetzt die Molit's ausgetauscht, denn das Magnetkern ist Spule Nr. 2, sodass ich die Herleitung von 5.4.0-4.2 jetzt wie folgt umwandeln muß:

Wir beginnen die Analogie in Bild (\* von 5.4.0). Das felderzeugende Spulenelement ist nun an der Position  $\vec{r}_2(t)$ , sodass wir die Parametrisierung der felderzeugenden Schleife aus (\* von 5.4.0) umwandeln in

$$\vec{r}_2(t) = \begin{pmatrix} x_2 \\ r_2 \cdot \sin(\omega_2 t + \varphi) \\ r_2 \cdot \cos(\omega_2 t + \varphi) \end{pmatrix} \quad (*)$$

Da hier existiert sich  $\omega_2$  aus dem Spulenstrom, der die Feldstärke des Spulenstroms annimmt, also  $I_2 = \frac{\omega_2}{2\pi} \cdot q_2$  entsprechend (\* von 5.4.1)

Die Ladung  $q_2$  kann ich dann wie gehabt willkürlich ansetzen, und  $I_2 \cdot \vec{v}_2$  für Biot-Savart (\* von 5.4.1) ergibt sich dann mit  $|\vec{v}_2| = r_2 \cdot \omega_2$  sodass bei großem  $q_2$  das  $\omega_2$  klein wird und umgekehrt bei kleinem  $q_2$  das  $\omega_2$  groß wird. Da gilt also  $I_2 \cdot \vec{v}_2 = q_2 \cdot r_2 \cdot \omega_2 \uparrow \downarrow$  (\*1) (\*2)

Da gilt also  $I_2 \cdot \vec{v}_2 = q_2 \cdot r_2 \cdot \omega_2 = I_2 \cdot 2\pi \cdot r_2 = \text{const}$ , unabhängig von der willkürlichen Wahl des  $q_2$ .

Damit kann ich einsetzen in Biot-Savart entsprechend (\* von 5.4.1) und bekomme durch Umwandlung von (\* von 5.4.1):

$$d\vec{H} = \frac{q_2 \cdot \vec{v}_2 \times (\vec{r}_2 - \vec{r}_1)}{4\pi \cdot |\vec{r}_2 - \vec{r}_1|^3} \cdot \frac{d\varphi}{2\pi} \quad (*)$$

Damit sind  $\vec{r}_2$  und  $\vec{r}_1$  die Orte der in WW stehenden Spulenelemente, und  $\vec{v}_2 = \text{Geschw des Ladungselements in der Magnetkern-Spule}$  bekomme ich durch Ableiten der Parametrisierung (\*) nach der Zeit:

$$\vec{v}_2 = \dot{\vec{r}}_2 = \begin{pmatrix} 0 \\ -r_2 \omega_2 \cdot \cos(\omega_2 t + \varphi) \\ r_2 \omega_2 \cdot \sin(\omega_2 t + \varphi) \end{pmatrix}$$

unnötig

Somit bildet ist das Kreuzprodukt in Zähler des Biot-Savart-Feldelements (\*4 von S. 48) in Analogie zu (\*5 von S. 41) unter Benützung von MathCAD: (File "Unit.mcd")

Kreuzprodukt des Zählers aus Biot-Savart → (\*1)

Setzt den Nenner aus Biot-Savart (\*2)

Sies können wir nun komponentenweise integrieren mit MathCAD zum Zeitpunkt  $t=0!$  (\*3)

Beispiel: Um den magnetischen Fluss, den die Magnethem-Spule im Inneren der Spule 1 erzeugt, bestimmen zu können, muß ich entlang der x-Achse schauen. So wird typischerweise die induzierte Spannung berechnet.

(Ob das ausreicht, weiß ich nicht, eigentlich muß ich die Lorentzkraft auf die Elektronen im Inneren des Leiters der Spule 2 berechnen. Aber ich probiere einfach mal aus.)

Da wir den magnetischen Fluss der Spule 2 durch die Spule 1 berechnen wollen, parametrisieren wir also

$$\vec{r}_1 = \begin{pmatrix} x_1 \\ 0 \\ 0 \end{pmatrix}$$

unnötig

Der langen Rede kurzer Sinn: Man kann das auch alles abkürzen und das Feld eines Leiterschleife auf der x-Achse lt Gerthsen S. 369:

$$H_x = \frac{I_2 \cdot r_2^2}{2(r_2^2 + (x_1 - x_2)^2)^{3/2}}$$

für jede einzelne Leiter-Schleife des Magnethem-Spule Nr. 2

Die y-Komponente spielt da keine Rolle  
→ vgl. Stöcker, Abh. 15.37, S. 449

Somit kann ich nun direkt die in jeder Windung der Spule 1 induzierte Spannung berechnen, und zwar wie folgt: Die Querschnittsfläche der Spule 1 ist (in jeder Windung ihrer Windungen)

$$A = \pi \cdot r_1^2$$

$$\mu_0 \cdot H_x \cdot A = \frac{\mu_0 \cdot \pi \cdot r_1^2 \cdot r_2^2 \cdot I_2^2}{2 \cdot (r_2^2 + (x_1 - x_2)^2)^{3/2}} \quad (*4)$$

⇒ Der magnetische Fluss ebenfalls ist  $\Phi = \int \vec{B} d\vec{A} = \mu_0 \cdot H_x \cdot A$

Die Induzierte Spannung ist dann  $U_{ind} = - \frac{d\Phi}{dt}$

, wobei die zeitliche Änderung des magnetischen Flusses sich aus der Bewegungsgeschwindigkeit  $x_2 = x_2(t)$  des Magnethems ergibt

(Stöcker S. 448)

Hierfür müssen wir im Schnittweise iterativen FEM-Algorithmus  $U_{ind} = - \frac{\Delta \phi}{\Delta t}$  einsetzen (unendlichem  $\Delta \phi$ ).

Also  $U_{ind} = - \frac{\Delta \phi}{\Delta t} = - \frac{1}{\Delta t} \cdot \mu_0 \cdot \Delta H_X \cdot A = \frac{-\mu_0 \pi n_1^2 \cdot n_2^2 \cdot I_2}{2 \cdot \Delta t}$  (\*4 von S. 49)

$\uparrow$  Das ist  $t_i$  im FEM Das ist  $t_{i-1}$  im FEM

$\uparrow$  Das ist  $t_i$  im FEM Das ist  $t_{i-1}$  im FEM

$\Rightarrow x_2(t) = x_2[i]$   $\Rightarrow x_2(t - \Delta t) = x_2[i-1]$

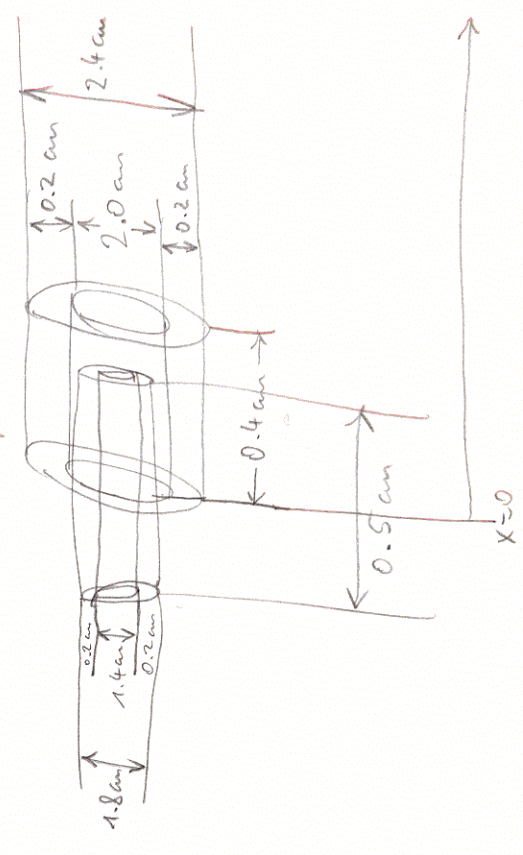
Das ist nun eine feste präfixbare Formel, die ich ins Pascal eintippen kann.  
 $\rightarrow$  Das File heißt dann "K1 - t - 008.dpr"

Zuerst programmiere ich es für die WW  $\rightarrow$  1 Wdg mit 1 Wdg  
 Danach lege ich eine 4fad-Schleife drüber, damit ich "Alle Wdgen mit allen Wdgen" bekommen kann.

Diese feste ich jetzt  $\therefore$

$\Delta d = 0.1 \text{ cm} = 1 \text{ mm}$

Spule 1



Jetzt endlich die feste greifbare Formel nach dem Ergebnis (\*1 von S. 50):

$\overset{di}{r1} := 0.2$      $\overset{dk}{r2} := 0.2$      $\overset{unverändert}{I1} := 1.0$      $\overset{I2 := 1.0}{x1} := 0.09$      $\overset{pi := 3.14159265358}{pi} := 3.14159265358$   
 $\overset{dt := 10^{-6} \text{ sec.}}{dt} := 10^{-6} \text{ sec.}$      $\overset{x[17]}{x2i} := 0.10 \text{ m}$      $\overset{x[16]}{x2im1} := 0.15 \text{ m}$      $\overset{muo := 4 \cdot pi \cdot 10^{-7}}{muo} := 4 \cdot pi \cdot 10^{-7}$

$\mu_0$  nicht vergessen! im Pascal-fgm

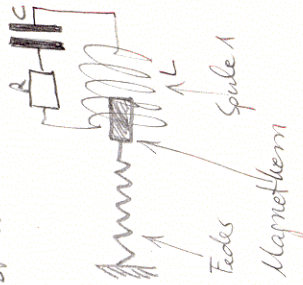
$$U_{ind} := \left( \frac{-\mu_0 \cdot \pi \cdot r1^2 \cdot r2^2 \cdot I2}{2 \cdot dt} \right) \cdot \left[ \frac{1}{[r2^2 + (x1 - x2i)^2]^{1.5}} - \frac{1}{[r2^2 + (x1 - x2im1)^2]^{1.5}} \right] \rightarrow -4.63958 \cdot 10^{-2} \text{ Volt}$$

Das ist nach der Größenordnung sogar sinnvoll. Die Formel ist ok für's Eintippen im Pascal -> DFEM.

so, dann sind alle Vorbereitungen fertig, und jetzt geht's weiter in die Raumenergie-wandelnde Schwingung. S. 51

→ File "Var-L-009.dpr" → Teil-03 im Gm.

Das System sieht nun so aus



R, L, C habe ich aus Teil-02 schon drin

Jetzt nehme ich eine Federkraft auf den Spulen Kern hinzu  $F_{\text{Fed}} = -D(x_{i-1} - x_0)$

In "Var-L-009.dpr" fehlt die reine mechanische Schwingung fast (nur Federkraft)

In "Var-L-010.dpr" ist die Magnetkraft zur Federkraft hinzugenommen.

Das Vorzeichen der Magnetkraft kann ich nach der Polarität des Bauemagnetes ausrichten

In "Var-L-011.dpr" kommt nun die Wkt von Magnetkraft und Federkraft  
auf die Schwingungen der Ladungen mit rein  
(ind. der induzierten Spannung:  $x, \dot{x}, \ddot{x}$  &  $\ddot{x}, \dot{x}, x$ )  
Kombi =  
nation  
beides  
Schwingung  
dunkel

Diese induzierte Spannung in der Spule muß von Anfang an bei jedem einzelnen Schritt mit in die Ladung aufgenommen werden.  
Das geht als drittes bei  $\ddot{Q}$  und folgt stattdessen dem Induktionsgesetz  
(\*3 von S. 5)  $\Rightarrow U_{\text{ind}} = -L \ddot{Q} \Rightarrow \ddot{Q} = -\frac{1}{L} U_{\text{ind}}$  x1

Beobachtung: Solange die abklingende elektrische Schwingung noch ordentlich Amplitude hat, fällt sie Energie an die mechanische Schwingung ab. Danach ist die elektr. Schwingung aufgrund der Dämpfung abgeklungen, und es passiert nichts mehr.

Die mechan. Schwingung macht ein paar Spannungspulse in die elektr. Schwingung, die aber nicht ausreichen, um die elektr. Schwingung dauerhaft anzukregen.

→ Was ich jetzt brauche, ist eine Anpassung des System-Abmessungen (Parameter der Dimensionierung), damit die beiden Schwingungen (med. + el.) zusammenkommen können. Allerdings sieht man bei entsprechendem feiner Auflösung von  $\ddot{Q}, \dot{Q}, Q$  kleine elektrische Schwingungen, die jede dieser Pulse wieder auslöst. Das ist eine Verifikation meiner Rechenmethode: File "Var-L-011-Verifikation.xls"

Somit ist der Magnetkonverter im Algorithmus. Lediglich die Energie-Entnahme fehlt noch.

Was jetzt kommt: Systemgrößen so abstimmen, daß das Bsp sich zur Harmonie-Konversion aufschauelt. → Das ist in "Elektro-L-012.dpr"

- Magneten und Spule größer machen

- Dämpfung des elektr. Schwingkreises minimieren (damit sich's überhaupt aufschaueln kann).
- Feldberechnungsdaten erst von vorangehenden Lauf übernehmen

Indirekte Stelle ist's noch abgespeichert unter "Var-L-011.dpr"

Jetzt geht's los mit der Dimensionierung des Geometrie-Parameter: Jetzt heißt das File endlich "Var-L-012.dpr"

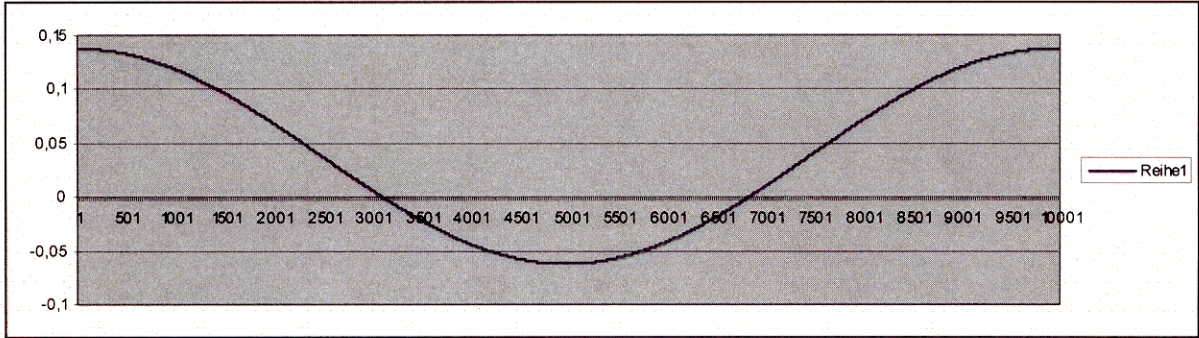
Die endliche Breite der Spalendrähne machen. Aus dem induzierten Spannung → siehe beliebiges Blatt zu S.52  
⇒ zu prob. das ist nicht auflösen (müßte zu wenige Windungen). "m. S. 52.xls"

- Sauer willid auch noch periodische Anregung mit einbauen → File "Var-L-013.dpr"

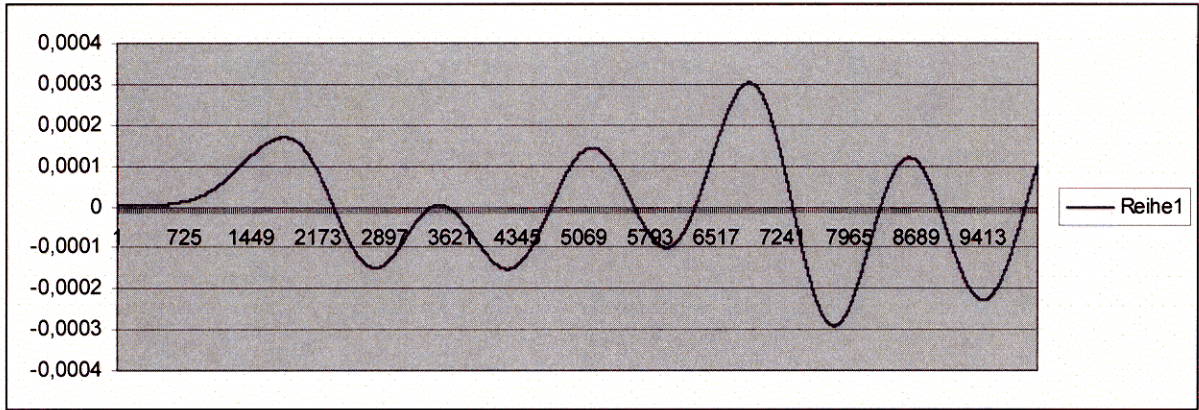
Neu ist das Untergren "l<sub>min</sub> (z.B. LongInt): Double;

Damit muß sich jetzt die mechanische und die elektrische Eigenfrequenz (Schwingung) aufeinander abstimmen und das Bsp anpassen

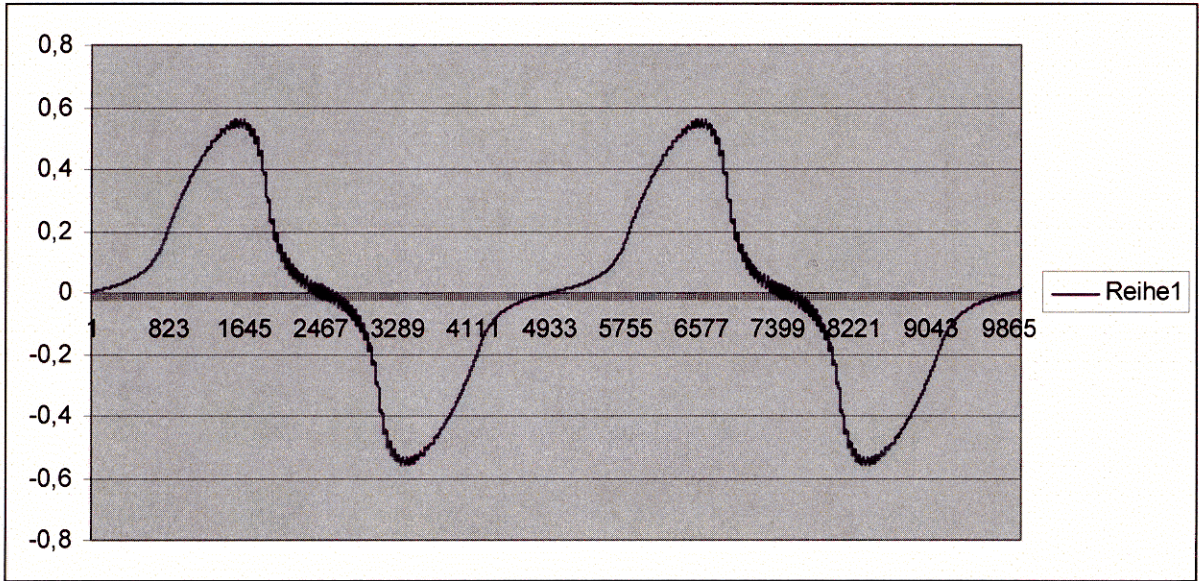
Auslenkung  $x[i]$



Ladung  $q[i]$



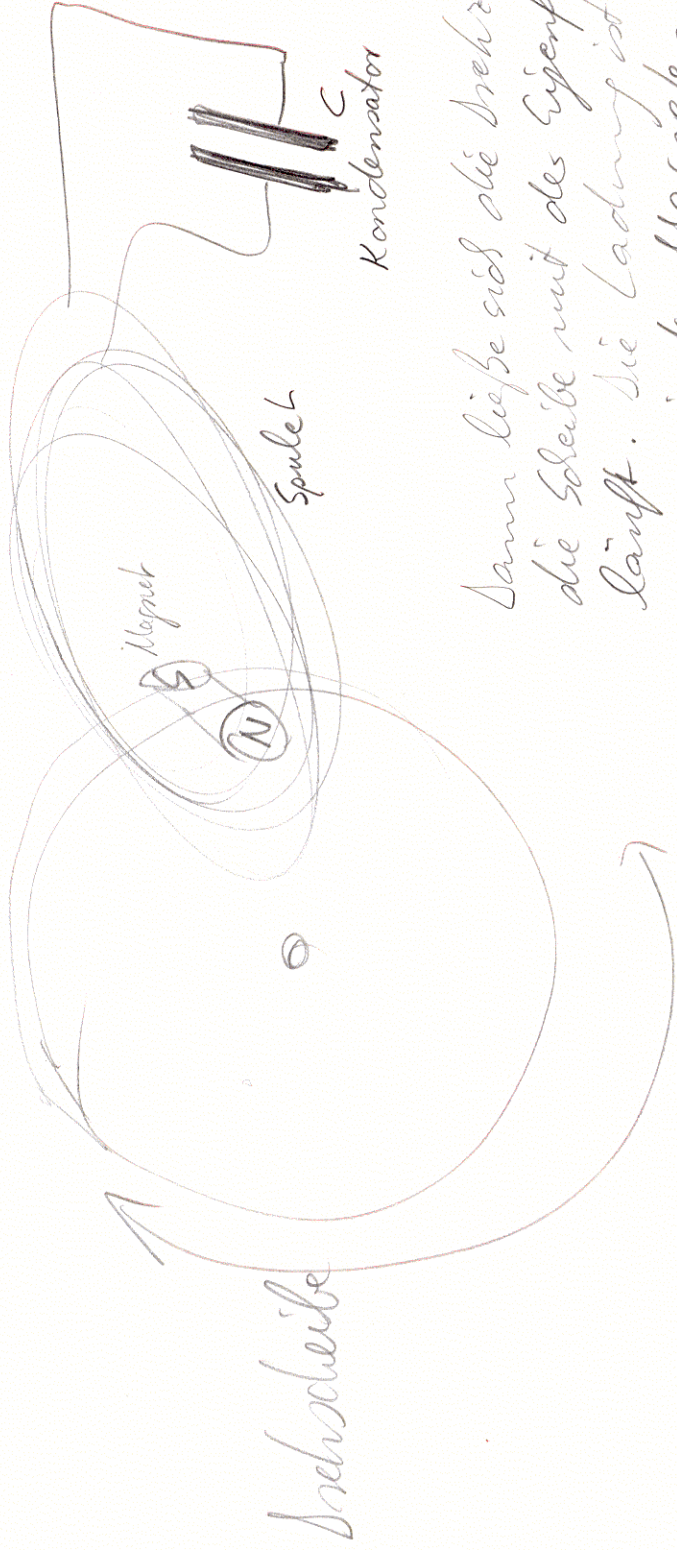
Induzierte Spannung  $-qpp[i]/L$



Ich bekomme die Energie solely aus der mechanischen Bewegung in die Spule bzw aus der Spule in die mechan. Schwingung

Begündung: Der Magnet läuft zu langsam ( $\dot{x}$  ist zu klein)  $\Rightarrow \dot{\Psi}$  ist zu klein, und ordentlich Spannung zu induzieren. (Umgekehrt ist die Feldstärke der Spule zu klein, um den Magneten ordentlich aufzuwickeln.)  $\rightarrow$  Das könnte ich noch durch ein konstantes Spülen mit den Kontakten ausprobieren.

Achille wäre: Wenn ich den Magneten rotieren lasse würde



Dann ließe sich die Drehzahl so einstellen, daß die Scheibe mit der Eigenfrequenz des Schwingkreises läuft. Die Ladung ist dann in der Spule wenn sie den Magneten mechanisch antreibt, und sie wäre im Kondensator, wenn der Magnet Spannung in die Spule induziert.

Fetzt habe ich doch noch eine Idee bekommen und will nicht aufgeben.:

Was bis S. 52 die Extraktion der Energie aus der Spule verhindert, ist die Massenträgheit des oszillierenden Magneten.

- Ist die Feder der Oszillation zu schwach, dann schwingt der Magnet zu langsam, und die niedrige Bewegungsgeschwindigkeit führt dazu, daß zu wenig Spannung in die Spule induziert wird. Damit ist die Aufkopplung an das mechanische Bewegung in die elektr. Bewegung zu schlecht.
- Ist die Feder der Oszillation zu stark, dann wird die mechan. Amplitude des Magneten zu klein und die Aufkopplung klappt deshalb nicht.

→ Abhilfe schaffen könnte die rotierende Maschine von S. 53. Sie hat keine Probleme mit der Massenträgheit, weil der Körper nicht ständig gebremst und beschleunigt werden muß.

Um zu sehen, ob ich mit dieser Idee auf dem richtigen Weg bin, will ich mal die Massenträgheit sehr gering machen, so daß der Magnet mit wenig Kraft schwingen kann.

Bei akzeptier ich für den Test auf eine unrealistisch niedrige Dichte des Magnetmaterials.

Falls das einen ordentlichen Betrieb der Maschine erlaubt, dann weiß ich, daß mir die rotierende Maschine nach S. 53 hilft.

Um das zu testen, mache ich das File "Var - L - 014.dpr" und spiele das mit den Parametern.

Falls ich malher ein "rotierender Motor" simulieren will → Tip für die praktische Handhabung: Die Spulendichte machen ein bisschen im Signal. Da muß ich die Kraftberechnung und die Berechnung der induzierten Spannung anders programmieren.

Das ist Scheiße, weil es zu langsam konvergiert und ich nicht ordentlich probieren kann.

Also tete ich die Flucht nach vorne an und programmiere / simuliere den rotierenden Motor. (vgl. S. 53)

Was sieht dann so aus.:

- ① Ein runder Magnet erzeuge ein homogenes Feld, so wie Caesar 500 vor Christus.



Das ist dann leichter und schneller zu rechnen, weil ich den Magneten nicht durch eine Spule simulieren muß

- ② Dieser Magnet rotiert in einer Spule, deren  $n$  Windungen alle an einem einzigen Ort liegen (also nicht nebeneinander). Das ist zwar nur eine Näherung, aber es beschleunigt sehr wesentlich die Rechenzeit (auf dem Computer), weil ich dann nur eine einzige Windung genau rechnen muß und diese dann einfach mit  $n$  hochskalieren kann.

- ③ Da es sich um eine Drehbewegung handelt, arbeiten wir nicht mehr mit einer Translationsbewegung, sondern mit einer Drehbewegung.  $\rightarrow$  keine Kraftberechnung, sondern eine Drehmomentberechnung. Wir betrachten das Drehmoment auf einen magnetischen Dipol im Magnetfeld.

Für einen allerersten Test wollen wir (damit es sich schnell rechnen läßt) auch noch auf Biot-Savart verzichten und nur das Drehmoment einer Spule im Magnetfeld betrachten. Wegen  $actio = reactio$  ist das das selbe wie das Drehmoment eines Magneten (der ein homogenes Feld macht) im Feld der Spule.

opt. Tipler S. 1031  
und mein Physik-Klausuraufsatz S. 233

magnetisches Dipolmoment einer Spule  $\vec{m} = n \cdot I \cdot \vec{A}$ , mit  $I = \text{Strom}$

$n = \text{Windungszahl}$

$\vec{A} = \text{Querschnittsfläche mit Normalenvektor}$

$\vec{m} = \text{Dipolmoment}$

$\vec{M} = \text{Drehmoment}$

und  $\vec{B} = \mu_0 \cdot \vec{H}$  des Dauermagneten



$$\vec{M} = \vec{m} \times \vec{B} = n \cdot I \cdot \vec{A} \times \vec{B}$$

Das Drehmoment ist dann

Damit ist die Kraftberechnung schon fertig, nämlich als Drehmomentberechnung. Bei geeigneter periodischer Anregung läßt sich eine Drehbewegung erzeugen.

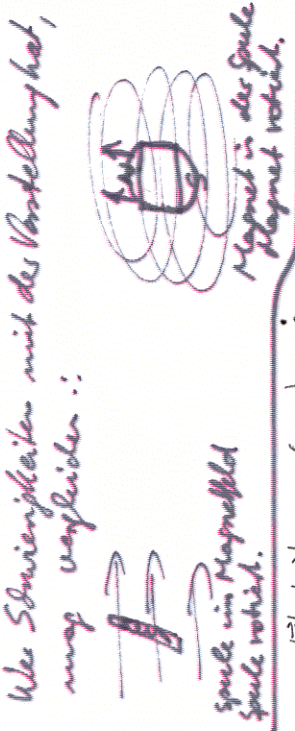
④ Die induzierte Spannung in der Spule muß ich berechnen (aufgrund der Rotation des Dauermagneten). Das geht so:

Der magnetische Fluß  $\Phi$ , den der Dauermagnet in die Spule einbringt, lautet

$$\Phi = \int_A \vec{B} \cdot d\vec{A} = |\vec{B}| \cdot |\vec{A}| \cdot \cos(\varphi) \quad \text{mit } \varphi = \varphi(t)$$

↑  
Magnetfeld des Dauermagneten  
↑  
Querschnittsfläche der Spule

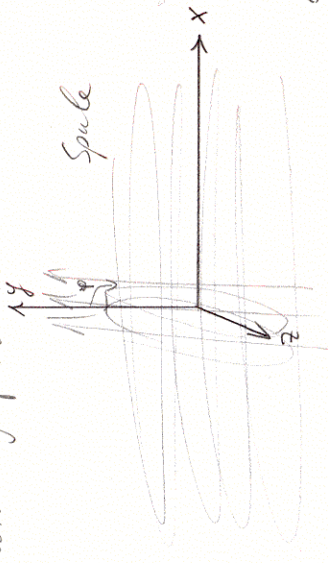
↑  
 $\varphi$  = Winkel zwischen der Magnetfeld-Richtung und der Flächen-Normale.



Die induzierte Spannung ist dann 
$$U_{\text{ind}} = -n \cdot \frac{d\Phi}{dt} = -n \cdot |\vec{B}| \cdot |\vec{A}| \cdot \frac{d}{dt} [\cos(\varphi(t))] \quad \uparrow \text{ Kettenregel}$$

(\*1 von S.55) und (\*1 von S.56) sind zweiganz einfache Formeln, mit denen sich sofort der JFEM-Algorithmus basteln läßt. Den Algo entwickle ich nun in "Kör-L-015: dtr".

Die Anordnung packe ich in kartesische Koordinaten, um das Kreuzprodukt nach (\*1 von S.55) gut rechnen zu können



x-Achse = Rotationsachse, Interessant ist also die x-Komponente von  $\vec{M}$ , d.h.  $M_x$ . Das Feld läuft also in der y-z-Ebene, senkrecht zur x-Achse

Die Spule liegt mit ihren Windungen um die z-Achse, also liegt jede Windung in der xz-Ebene.

Das Kreuzprodukt für (\*1 von S.55) bestimme ich also wie folgt:

$$\vec{M} = n \cdot I \cdot \vec{A} = n \cdot I \cdot A \cdot \vec{e}_y = n \cdot I \cdot A \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\vec{B} = \mu_0 \vec{H} = \mu_0 \cdot |\vec{H}| \cdot \begin{pmatrix} 0 \\ \cos \varphi \\ \sin \varphi \end{pmatrix} \quad \text{mit } \varphi = \text{Winkel gegen die y-Achse}$$

$$\Rightarrow \vec{M} = \vec{m} \times \vec{B} = n I A \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \times \mu_0 |\vec{H}| \cdot \begin{pmatrix} 0 \\ \cos \varphi \\ \sin \varphi \end{pmatrix} = \mu_0 n I A |\vec{H}| \cdot \begin{pmatrix} \sin \varphi \\ 0 \\ 0 \end{pmatrix} \quad \uparrow \text{ Kreuzprodukt}$$

$$\Rightarrow M_x = \mu_0 \cdot n \cdot I \cdot A \cdot \sin(\varphi(t))$$

$$|\vec{B}_0| = \mu_0 \cdot |\vec{H}| = \text{Magnetfeld des Dauermagneten}$$

Die Induktivität der Spule setzen wir des Einfachheit halber an mit  $L = \frac{\mu_0}{4\pi} A m^2$  (L-f. Spule) Störker S.452

Da die Rotation von selbst (ohne Rückstellkraft) in ihren Ausgangspunkt zurückkehrt, ist keine Rückstellfeder und keine harmonische Schwingung nötig. (Die Projektion der Rotation auf die x-Achse und auf die y-Achse ist ja eine harmonische Schwingung.)

Um mit meinen Spln arbeiten zu können, baue ich also das Ganze wieder (dominant) als elektrischen Schwingkreis auf und packe dort die mechanische Drehbewegung mit Kraft/Behmoment (Magnetkraft) und induzierte Spannung hinein.

⇒ Teil 1 und Teil 2 für die elektr. Schwingung kann ich aus "Kv-L-014.dpr" übernehmen. (bis zur gedämpften elektr. Schwingung t, R)

aber Teil 3 hat keine Spl einer Schwingung. Die Drehung hat keine Rückstellfeder. D.h., der Term  $F = -sx$  entfällt.

Die Spl der Bewegung ist also die folgende:  $|\vec{M}| = I \cdot \ddot{\varphi}$  mit  $I = \text{Trägheitsmoment des Magneten (bei Rotation)}$

$$m_T = \text{Träge Masse des Magneten}$$

$$r_m = \text{Radius des Magneten (halber Durchmesser)}$$

$$I = \frac{1}{2} m r^2 \text{ mit } m = 8 \cdot V$$

↑ Vollzylinder, Pubbel, S.B.34.

$$\text{also } I \ddot{\varphi} = M_x = B_0 m V A \cdot \sin(\varphi(t))$$

$$\Rightarrow \text{Spl } \ddot{\varphi} - \frac{B_0 m V A}{I} \cdot \sin(\varphi(t)) = 0 \quad (*)$$

Diese Spl für die mechan. Drehung muß ich jetzt als Teil 3 programmieren

→ Da sieht man, wie die Magnetkraft die Drehung moduliert

"Mag-Dreh-mod.xls"

In Teil 4 packe ich dann die elektr. Schwingung mit der Drehung zusammen → File "Kv-L-016.dpr"

Dazu muß ich  $\varphi, \dot{\varphi}, \ddot{\varphi}$  aus der Drehung nehmen, sowie  $q, \dot{q}, \ddot{q}$  aus der elektr. Schwingung und dazu die induzierte Spannung aufgrund der Drehung des Magneten in der Spule.

Das teste ich und stelle plausible Ergebnisse fest

"Motor mit Kopplung.xls"

Jetzt will ich versuchen, hier mit den Parametern so zu spielen, daß sich die Schwingung (Bewegung) selber aufschaukelt.: File "Var-L-017.dpr"

Dazu müßte ich die Schwingung des elektr. Schwingkreises mit der hochfrequenten des Magneten abstimmen. Das geht entweder durch periodische Bewegung auf elektr. Seite oder durch mechanische Belastung der Bewegung, um die Drehzahl konstant zu halten.

Vorzeichen des Magnetkraft überprüfen: Wenn ich einen Magneten reinste, wird die Bewegung mechanisch gefördert im Vpl zu einem unmagnetischen Eisen, weil man kinetische Energie in elektr. Energie umwandelt. Daraus ergibt sich in Teil-04 das folgende Vorzeichen:

$$\ddot{\varphi}_i = -B \cdot n \cdot \dot{\varphi}_i \cdot \frac{A}{f} \cdot \sin(\varphi)$$

Dieses Minus muß so sein (nicht anders)

$$\ddot{\varphi}_i = \ddot{\varphi}_n + n \cdot B_0 \cdot A \cdot \sin(\varphi) \cdot \ddot{\varphi} \cdot \frac{1}{L}$$

Es könnte auch beide Vorzeichen umdrehen. Das entspräche einem entgegengesetzten Einbau des Magneten. Aber die beiden Vorzeichen müssen immer einander entgegenstehen.

→ Im "Verifikation.xls" sieht man, wie ich langsam Energie aus der mechanischen Bewegung in die Spule herausziehe und diese durch den Widerstand (Ohm'sch) im Spulendraht aufsparen wird.

Und jetzt werde ich, die Bewegung so abstimmen, daß dabei die hochfrequenz des Rotors genau zur Schwingungsfrequenz im LC(R)-Schwingkreis paßt, so daß Raumenergie gewandelt wird.: File "Var-L-018.dpr"

Bsp.: Startanlaufgeschw  $2\pi \frac{\text{rad}}{\text{sa}} = 1 \text{ Umdrehung pro Sekunde}$ . Da ist schnell die Dämpfung so stark, daß das Ding fast stehen bleibt.

Um ein Gefühl für das Ding zu bekommen, arbeite ich zunächst ohne Innenwiderstand des Spulendrahts. Dann mache ich eine Energie-Ermittlung, damit ich schauen kann, wie sich das Ding verhält.

- $K_1 = \frac{1}{2} L I^2 = \text{Spulenenergie}$
- $K_2 = \frac{1}{2} C U^2 = \text{Kondensatorenergie}$
- $K_3 = \frac{1}{2} J \omega^2 = \text{Rotationsenergie}$
- $K_4 = \text{Summe dieser 3 Energien}$

→ File "Energie-Umwand.xls"

Und da merke ich sofort schon ein biphasen Raumenergie-Wandlung  
Da schreibe ich genau diese 4 Energieformen  $K_1, K_2, K_3, K_4$  heraus.

Aus Versuchen habe ich bisher wieder Var-L-018 verwendet.

"Var-L-018.dpr"

Damit probiere ich nun die Parameter-Variation  $\rightarrow$  File

Neue Programmiersprache ist aus die prozentuale Veränderung der Gesamtenergie ein  $\rightarrow$  mit Bildschrim-Ausgabe.

"Ohne Ohm - 89 milliwatt.xls"

Der File speichere ich ab mit einer konstanten Leistung von 0.089 milliwatt und den Ohm'schen Widerstand des Spulendrahtes schaffe ich dabei noch nicht.

$\rightarrow$  Unwahrscheinlich muß ich den Magneten schneller notieren lassen und mit höherer Feldstärke versehen.

Um die bisherigen Parameter nicht zu verlieren, probiere ich das im File "Var-L-019.dpr".

Ich muß mit höherer Drehzahl starten, um die Ohm'schen Widerstände des Spulendrahtes überwinden zu können!

$\rightarrow$  Mit einem <sup>(50 Hz Läufer)</sup> 3000 U/min Läufer habe ich es jetzt geschafft, 830 Milliwatt zu wandeln

"Var-L-019.dpr"

bei erster Berücksichtigung des Widerstands des Spulendrahtes!  $\rightarrow$  Parameter anbei.

"Mit Ohm - 830 milliwatt.xls"

"Var-L-020.dpr"

Damit ist die Parameter nicht verlieren, arbeite ich jetzt mit einer neuen File-Nummer weiter: "Var-L-020.dpr"

bund Verstärken des Magneten auf realistische Werte belomme ich "Mit Ohm - 5 Watt.xls"

Um die Parameter nicht zu verlieren, mache ich wieder mit einem neuen Filenamen weiter: "Var-L-021.dpr"

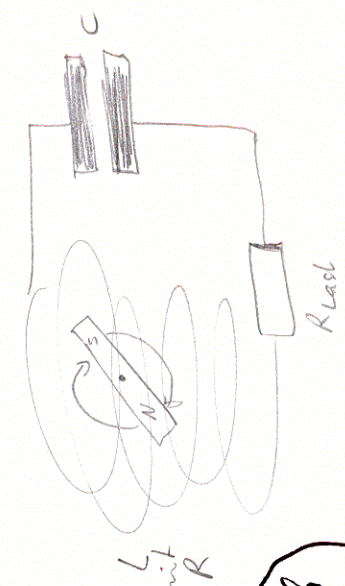
"Gute Zeitauflösung - 4 Watt.xls"

Eine ordentliche Zeitauflösung ist für die Publikation wichtig. Sie habe ich jetzt  $\rightarrow$  Da will ich jetzt einen Lastwiderstand mit der Spule in Reihe schalten, um elektr. Leistung entnehmen zu können. Das muß ich jetzt noch programmieren.

"Var-L-022.dpr"

$\rightarrow$  Leistungsentnahme

J.h.: Ich habe folgenden Aufbau



Bisher habe ich nur das Aufschneiden beobachtet, aber jetzt will ich die Leistung auch herausziehen.

Damit bringe ich in die Schwingungs-Gl der elektr. Schwingung eine zusätzliche Dämpfung ein.  $-L\ddot{Q} + R\dot{Q} + \frac{1}{C}Q = 0$  und  $R$  war der Ohm'sche Widerstand des Spulendrahtes.

Jetzt kommt zusätzlich noch der Lastwiderstand hinzu, also bekommen wir die Gl  $-L\ddot{Q} + (R + R_{Last})\dot{Q} + \frac{1}{C}Q = 0$

Die entnommene Energie am Lastwiderstand ist dann  $P = U_{last} \cdot \dot{Q} = R_{last} \cdot \dot{Q}_i^2$

$U_{last} = R_{last} \cdot \dot{Q}_i$

Den Lastwiderstand bringe ich als Uppm ein, damit ich ihn steuer- und regelbar gestalten kann!

Um 0.1 Watt lasse ich die Leistung im System noch anwachsen, damit demonstriert ist, daß ich nicht aus dem System Leistung entziehe.

Damit bringe ich 5.6 Watt elektr. heraus

"Var-L-022.dpr"

"Entnahme - 5 Watt.xls"

## 830 Milli Watt mit (!) Ohm'schem Widerstand der Spulenwicklung

```

Begin {Hauptprogramm}
{ Initialisierung - Vorgabe der Werte: }           {Wir arbeiten in SI-Einheiten}
Writeln('Raumenergie-Konverter mit Rotation.');
```

{ Vorgabe der Werte -> Input-Parameter:}

```

epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}
muo:=4*pi*1E-7{Vs/Am};      {Elektrische Feldkonstante}
lichtgesch:=Sqrt(1/muo/epo){m/s};  Writeln('Lichtgeschwindigkeit c = ',lichtgesch, ' m/s');
{ Spule, Magnet, Kondensator:}
n:=1000;      {Windungszahl der Spule}
di:=0.09;     {Spulenkörper-Durchmesser}
Dd:=0.0010;   {Drahtdurchmesser}
Bo:=0.0067;   {Tesla} {Magnetfeld (Amplitude) des Dauermagneten}
ls:=0.01;     {Meter} {Länge des zylindrischen Spulenkörpers}
C:=6.9E-6;    {Farad} {Kapazität des Kondensators}
rm:=0.035;    {Meter} {Radius des zylindrischen Dauermagneten}
dm:=0.01;     {Meter} {Dicke des zylindrischen Dauermagneten}
rhom:=7.8E3;   {Dichte des Magnet-Materials, Eisen, Kohlrausch Bd.3}
{ Abgeleitete Parameter, keine Eingabe möglich:}
A:=di*di; {Meter * Meter} {Querschnittsfläche der Spule}
L:=muo*a*n*n/ls; {Induktivität der Spule}
omo:=1/Sqrt(L*C); {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
T:=2*pi/omo;     {Schwingungsdauer des elektrischen Schwingkreises}
rho:=1.7E-8; {Ohm*m} {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
R:=rho*(2*pi*di*n)/(pi*(Dd/2)*(Dd/2)); {Ohm} {Ohm'scher Widerstand des Spulendrahtes}
{ Sonstige:}
UmAn:=50;      {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}
AnzP:=AnzPmax; {Anzahl der Zeitschritte insgesamt}
dt:=0.005; {sec.} {Größe der Zeitschritte}
Abstd:=1;      {Jeder wievielte Punkte soll geplottet werden}
mt:=pi*rm*rm*dm*rhom; {Träge Masse des zylindrischen Dauermagneten}
J:=1/2*mt*rm*rm; {Trägheitsmoment des zylindrischen Dauermagneten}
{ Anzeige der Werte:}
Writeln('Induktivitaet der Luft-Spule: L = ',L,' Henry');
Writeln('Eigen-Kreisfreq harmon.el.Osz.: omo = ',omo:8:4,' Hz => T = ',T:15,'sec.');
```

Writeln('Laenge des Spulendrahts: ',(2\*pi\*di\*n),' m');

Writeln('Ohm'scher Widerstand des Spulendrahts: R = ',R:8:2,' Ohm');

Writeln('Traege Masse des zylindrischen Dauermagneten: mt = ',mt,' kg');

Writeln('Traegheitsmoment des Dauermagneten: J = ',J,' kg\*m^2');

Writeln('Gesamtdauer der Betrachtung: ',AnzP\*dt,' sec.');

## Mit Ohm, 5 Watt

```
Begin {Hauptprogramm}
{ Initialisierung - Vorgabe der Werte: }           {Wir arbeiten in SI-Einheiten}
Writeln('Raumenergie-Konverter mit Rotation.');
```

Writeln('Vorgabe der Werte -> Input-Parameter:');

```
epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}
muo:=4*pi*1E-7{Vs/Am};      {Elektrische Feldkonstante}
lichtgesch:=Sqrt(1/muo/epo){m/s}; Writeln('Lichtgeschwindigkeit c = ',lichtgesch, ' m/s');
```

{ Spule, Magnet, Kondensator:}

```
n:=1000;           {Windungszahl der Spule}
di:=0.09;          {Spulenkörper-Durchmesser}
Dd:=0.0010;        {Drahtdurchmesser}
Bo:=0.2000;        {Tesla} {Magnetfeld (Amplitude) des Dauermagneten}
ls:=0.01;          {Meter}  {Länge des zylindrischen Spulenkörpers}
C:=6.9E-6;         {Farad}  {Kapazität des Kondensators}
rm:=0.035;         {Meter}  {Radius des zylindrischen Dauermagneten}
dm:=0.01;          {Meter}  {Dicke des zylindrischen Dauermagneten}
rhom:=7.8E3;        {Dichte des Magnet-Materials, Eisen, Kohlrausch Bd.3}
```

{ Abgeleitete Parameter, keine Eingabe möglich:}

```
A:=di*di; {Meter * Meter} {Querschnittsfläche der Spule}
L:=muo*a*n*n/ls; {Induktivität der Spule}
omo:=1/Sqrt(L*C); {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
T:=2*pi/omo;      {Schwingungsdauer des elektrischen Schwingkreises}
rho:=1.7E-8; {Ohm*m} {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
R:=rho*(2*pi*di*n)/(pi*(Dd/2)*(Dd/2)); {Ohm} {Ohm'scher Widerstand des Spulendrahtes}
```

{ Sonstige:}

```
UmAn:=50;           {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}
AnzP:=AnzPmax;      {Anzahl der Zeitschritte insgesamt}
dt:=0.001; {sec.}   {Größe der Zeitschritte}
Abstd:=1;           {Jeder wievielte Punkte soll geplottet werden}
mt:=pi*rm*rm*dm*rhom; {Träge Masse des zylindrischen Dauermagneten}
J:=1/2*mt*rm*rm;    {Trägheitsmoment des zylindrischen Dauermagneten}
```

{ Anzeige der Werte:}

```
Writeln('Induktivitaet der Luft-Spule: L = ',L,' Henry');
```

Writeln('Eigen-Kreisfreq harmon.el.Osz.: omo = ',omo:8:4,' Hz => T = ',T:15,'sec.');

Writeln('Laenge des Spulendrahts: ',(2\*pi\*di\*n),' m');

Writeln('Ohm'scher Widerstand des Spulendrahts: R = ',R:8:2,' Ohm');

Writeln('Traege Masse des zylindrischen Dauermagneten: mt = ',mt,' kg');

Writeln('Traegheitsmoment des Dauermagneten: J = ',J,' kg\*m^2');

Writeln('Gesamtdauer der Betrachtung: ',AnzP\*dt,' sec.');

## "Var L 021.dpr"

*Das bringt zwar nur 4 Werte,  
aber dafür mit ordentlicher Zeitauflösung*

```
Program Konverter_mit_variablen_Spule;
{$APPTYPE CONSOLE}
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

Const AnzPmax=10000; {Anzahl der Zeitschritte zur Lsg. der Dgl.}

Type Feld = Array[0..AnzPmax] of Double;

Var epo,muo      : Double; {Naturkonstanten}
    lichtgesch   : Double; {Lichtgeschwindigkeit}
    n            : LongInt; {Windungszahl der Spule}
    A            : Double; {Querschnittsfläche der Spule}
    Bo           : Double; {Magnetfeld (Amplitude) des Dauermagneten}
    ls           : Double; {Länge der zylindrischen Spule}
    di           : Double; {Durchmesser des Spulenkörpers}
    Dd           : Double; {Drahtdurchmesser}
    rm           : Double; {Radius des Dauermagneten}
    L            : Double; {Induktivität der Spule}
    C            : Double; {Kapazität des Kondensators}
    R            : Double; {Ohm'scher Widerstand des Spulendrahtes}
    rho          : Double; {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
    phi,phip,phipp : Feld; {Drehwinkel und dessen Ableitungen}
    Q,Qp,Qpp     : Feld; {Ladung und deren Ableitungen}
    i            : LongInt; {Laufvariable}
    AnzP         : LongInt; {Anzahl der tatsächlich berechneten Zeit-Schritte}
    dt           : Double; {Dauer der Zeitschritte zur Lsg. der Dgl.}
    Abstd        : Integer; {Jeder wievielte Punkte soll geplottet werden}
    omo          : Double; {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
    T            : Double; {Schwingungsdauer des elektrischen Schwingkreises}
    UC,UL        : Double; {Kondensatorspannung und Spulenspannung}
    rhom         : Double; {Dichte des Magnetmaterials}
    dm           : Double; {Dicke des zylindrischen Dauermagneten}
    mt           : Double; {Träge Masse des zylindrischen Dauermagneten}
    J            : Double; {Trägheitsmoment des zylindrischen Dauermagneten}
    K0,K1,K2,K3,K4,K5 : Feld; {Kontroll-Felder für Plot-Zwecke}
    EmA,EmE,siA,siE : Double; {Energie: Mittelwerte und Sigma "Anfang" und "Ende"}
    delE,sigdelE    : Double; {Veränderung der Energie-Mittelwerte "Anfang" zu "Ende"}
    UmAn           : Double; {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}

Procedure Wait;
Var Ki : Char;
begin
  Write('<W>'); Read(Ki); Write(Ki);
  If Ki='e' then Halt;
end;

Procedure ExcelAusgabe(Name:String;Spalten:Integer;KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,KK,KL:Feld);
Var fout : Text; {Bis zu 12 Spalten zum Aufschreiben der Ergebnisse}
    lv,j,k : Integer; {Laufvariable}
    Zahl : String; {Die ins Excel zu druckenden Zahlen}
begin
  Assign(fout,Name); Rewrite(fout); {File öffnen}
  For lv:=0 to AnzP do {von "plotanf" bis "plotend"}
  begin
    If (lv mod Abstd)=0 then
    begin
      For j:=1 to Spalten do
      begin {Kolumnen drucken}
        If j=1 then Str(KA[lv]:19:14,Zahl);
        If j=2 then Str(KB[lv]:19:14,Zahl);
        If j=3 then Str(KC[lv]:19:14,Zahl);
        If j=4 then Str(KD[lv]:19:14,Zahl);
        If j=5 then Str(KE[lv]:19:14,Zahl);
        If j=6 then Str(KF[lv]:19:14,Zahl);
        If j=7 then Str(KG[lv]:19:14,Zahl);
        If j=8 then Str(KH[lv]:19:14,Zahl);
        If j=9 then Str(KI[lv]:19:14,Zahl);
        If j=10 then Str(KJ[lv]:19:14,Zahl);
        If j=11 then Str(KK[lv]:19:14,Zahl);
        If j=12 then Str(KL[lv]:19:14,Zahl);
        For k:=1 to Length(Zahl) do
        begin {Keine Dezimalpunkte verwenden, sondern Kommata}
          If Zahl[k]<>'.' then write(fout,Zahl[k]);
          If Zahl[k]='.' then write(fout,',');
        end;
        Write(fout,chr(9)); {Daten-Trennung, Tabulator}
      end;
      Writeln(fout,''); {Zeilen-Trennung}
    end;
  end;
  Close(fout);
end;
```

```

Begin {Hauptprogramm}
{ Initialisierung - Vorgabe der Werte: }           {Wir arbeiten in SI-Einheiten}
Writeln('Raumenergie-Konverter mit Rotation.');
```

20.5.59 d

```

{ Vorgabe der Werte -> Input-Parameter:}
epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}
muo:=4*pi*1E-7{Vs/Am};      {Elektrische Feldkonstante}
lichtgesch:=Sqrt(1/muo/epo){m/s}; Writeln('Lichtgeschwindigkeit c = ',lichtgesch, ' m/s');
{ Spule, Magnet, Kondensator:}
n:=1000;      {Windungszahl der Spule}
di:=0.09;     {Spulenkörper-Durchmesser}
Dd:=0.0010;   {Drahtdurchmesser}
Bo:=0.5000;   {Tesla} {Magnetfeld (Amplitude) des Dauermagneten}
ls:=0.01;     {Meter} {Länge des zylindrischen Spulenkörpers}
C:=1.60E-6;   {Farad} {Kapazität des Kondensators}
rm:=0.035;    {Meter} {Radius des zylindrischen Dauermagneten}
dm:=0.01;     {Meter} {Dicke des zylindrischen Dauermagneten}
rhom:=7.8E3;  {Dichte des Magnet-Materials, Eisen, Kohlrusch Bd.3}
{ Abgeleitete Parameter, keine Eingabe möglich:}
A:=di*di; {Meter * Meter} {Querschnittsfläche der Spule}
L:=muo*a*n*n/ls; {Induktivität der Spule}
omo:=1/Sqrt(L*C); {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
T:=2*pi/omo;     {Schwingungsdauer des elektrischen Schwingkreises}
rho:=1.7E-8; {Ohm*m} {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrusch,T193}
R:=rho*(2*pi*di*n)/(pi*(Dd/2)*(Dd/2)); {Ohm} {Ohm'scher Widerstand des Spulendrahtes}
{ Sonstige:}
UmAn:=50;      {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}
AnzP:=AnzPmax; {Anzahl der Zeitschritte insgesamt}
dt:=0.0001; {sec.} {Größe der Zeitschritte}
Abstd:=1;      {Jeder wievielte Punkte soll geplottet werden}
mt:=pi*rm*rm*dm*rhom; {Träge Masse des zylindrischen Dauermagneten}
J:=1/2*mt*rm*rm; {Trägheitsmoment des zylindrischen Dauermagneten}
{ Anzeige der Werte:}
Writeln('Induktivitaet der Luft-Spule: L = ',L,' Henry');
Writeln('Eigen-Kreisfreq harmon.el.Osz.: omo = ',omo:8:4,' Hz => T = ',T:15,'sec.');
```

Writeln('Laenge des Spulendrahts: ',(2\*pi\*di\*n),' m');

Writeln('Ohm'scher Widerstand des Spulendrahts: R = ',R:8:2,' Ohm');

Writeln('Traege Masse des zylindrischen Dauermagneten: mt = ',mt,' kg');

Writeln('Traegheitsmoment des Dauermagneten: J = ',J,' kg\*m^2');

Writeln('Gesamtdauer der Betrachtung: ',AnzP\*dt,' sec.');

{ Hier beginnt das Rechenprogramm.}

```

{ Teil_04: Mechanische Drehung}
Writeln(' -> Teil_04: Mechanisch und elektrisch gekoppelt');
UC:=0;{Volt} Q[0]:=C*UC;      Qpp[0]:=0; Qp[0]:=0; {Elektrische Startwerte}
phi[0]:=0;   phip[0]:=UmAn*2*pi; phipp[0]:=0;      {Mechanische Startwerte}
K0[0]:=0;
K1[0]:=1/2*L*Sqr(Qp[0]); {Spulen-Energie}
K2[0]:=1/2*C*Sqr(Q[0]/C); {Kondensator-Energie}
K3[0]:=1/2*J*Sqr(phip[0]); {Rotations-Energie}
K4[0]:=K1[0]+K2[0]+K3[0]; {Gesamt-Energie}
For i:=1 to AnzP do
begin
  Qpp[i]:=-1/L/C*Q[i-1]-R/2/L*Qp[i-1];      {nach *5 von S.6}
  Qpp[i]:=Qpp[i]+n*Bo*A*sin(phi[i-1])*phip[i-1]/L; {Induzierte Spannung in Spule bringen.}
  Qp[i]:=Qp[i-1]+(Qpp[i]-R/2/L*Qp[i-1])*dt; {nach *3 & *4 von S.6}
  Q[i]:=Q[i-1]+Qp[i]*dt;
  phipp[i]:=-Bo*n*Qp[i]*A/J*sin(phi[i-1]); {Mechanisches Drehmoment, x-Komponente}
  phip[i]:=phip[i-1]+phipp[i]*dt;
  phi[i]:=phi[i-1]+phip[i]*dt;
  K0[i]:=0; K1[i]:=0; K2[i]:=0; K3[i]:=0; K4[i]:=0; K5[i]:=0;
  K1[i]:=1/2*L*Sqr(Qp[i]); {Spulen-Energie}
  K2[i]:=1/2*C*Sqr(Q[i]/C); {Kondensator-Energie}
  K3[i]:=1/2*J*Sqr(phip[i]); {Rotations-Energie}
  K4[i]:=K1[i]+K2[i]+K3[i]; {Gesamt-Energie}
end;
{ Gesamt-Energie-Bilanz und Anzeige:}
EmA:=0; EmE:=0; siA:=0; siE:=0;
For i:=1 to 10 do EmA:=EmA+K4[i]/10; {Mittelwert zu Beginn}
For i:=AnzP-9 to AnzP do EmE:=EmE+K4[i]/10; {Mittelwert am Ende}
For i:=1 to 10 do siA:=siA+Sqr(EmA-K4[i]); {Varianz zu Beginn}
For i:=AnzP-9 to AnzP do siE:=siE+Sqr(EmE-K4[i]); {Varianz am Ende}
siA:=Sqrt(siA)/10; siE:=Sqrt(siE)/10; {Standardabweichungen}
Writeln('Energie-Werte: E_Anfang = (' ,EmA:11:7,' +/- ' ,siA:11:7,' ) Joule');
Writeln(' E_End = (' ,EmE:11:7,' +/- ' ,siE:11:7,' ) Joule');
delE:=EmE-EmA; sigdelE:=Sqrt(Sqr(siE)+Sqr(siA));
Writeln('=> Veraenderung: delta_E = (' ,delE:11:7,' +/- ' ,sigdelE:11:7,' ) Joule');
Writeln('=> Konvertierte Leistung = (' ,delE/(AnzP*dt):11:7,' +/- ' ,sigdelE/(AnzP*dt):11:7,' ) Watt');
ExcelAusgabe('test_04.dat',11,Q,Qp,Qpp,phi,phip,phipp,K0,K1,K2,K3,K4,K0);

Wait; Wait;
End.
```

## Var L 022.dpr mit Leistungsentnahme 5.6 Watt

```
Program Konverter_mit_variabler_Spule;
{$APPTYPE CONSOLE}
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

Const AnzPmax=10000; {Anzahl der Zeitschritte zur Lsg. der Dgl.}

Type Feld = Array[0..AnzPmax] of Double;

Var epo,muo      : Double; {Naturkonstanten}
    lichtgesch   : Double; {Lichtgeschwindigkeit}
    n             : LongInt; {Windungszahl der Spule}
    A             : Double; {Querschnittsfläche der Spule}
    Bo           : Double; {Magnetfeld (Amplitude) des Dauermagneten}
    ls           : Double; {Länge der zylindrischen Spule}
    di           : Double; {Durchmesser des Spulenkörpers}
    Dd           : Double; {Drahtdurchmesser}
    rm           : Double; {Radius des Dauermagneten}
    L            : Double; {Induktivität der Spule}
    C            : Double; {Kapazität des Kondensators}
    R            : Double; {Ohm'scher Widerstand des Spulendrahtes}
    rho          : Double; {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
    phi,phip,phipp : Feld; {Drehwinkel und dessen Ableitungen}
    Q,Qp,Qpp     : Feld; {Ladung und deren Ableitungen}
    i            : LongInt; {Laufvariable}
    AnzP         : LongInt; {Anzahl der tatsächlich berechneten Zeit-Schritte}
    dt           : Double; {Dauer der Zeitschritte zur Lsg. der Dgl.}
    Abstd        : Integer; {Jeder wievielte Punkte soll geplottet werden}
    omo          : Double; {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
    T            : Double; {Schwingungsdauer des elektrischen Schwingkreises}
    UC,UL        : Double; {Kondensatorspannung und Spulenspannung}
    rhom         : Double; {Dichte des Magnetmaterials}
    dm           : Double; {Dicke des zylindrischen Dauermagneten}
    mt           : Double; {Träge Masse des zylindrischen Dauermagneten}
    J            : Double; {Trägheitsmoment des zylindrischen Dauermagneten}
    K0,K1,K2,K3,K4,K5 : Feld; {Kontroll-Felder für Plot-Zwecke}
    EmA,EmE,siA,siE : Double; {Energie: Mittelwerte und Sigma "Anfang" und "Ende"}
    delE,sigdelE   : Double; {Veränderung der Energie-Mittelwerte "Anfang" zu "Ende"}
    UmAn          : Double; {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}
    Eent          : Double; {Entnommene Energie, elektrisch}

Procedure Wait;
Var Ki : Char;
begin
  Write('<W>'); Read(Ki); Write(Ki);
  If Ki='e' then Halt;
end;

Procedure ExcelAusgabe(Name:String;Spalten:Integer;KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,KK,KL:Feld);
Var fout : Text; {Bis zu 12 Spalten zum Aufschreiben der Ergebnisse}
    lv,j,k : Integer; {Laufvariable}
    Zahl : String; {Die ins Excel zu druckenden Zahlen}
begin
  Assign(fout,Name); Rewrite(fout); {File öffnen}
  For lv:=0 to AnzP do {von "plotanf" bis "plotend"}
  begin
    If (lv mod Abstd)=0 then
    begin
      For j:=1 to Spalten do
      begin {Kolumnen drucken}
        If j=1 then Str(KA[lv]:19:14,Zahl);
        If j=2 then Str(KB[lv]:19:14,Zahl);
        If j=3 then Str(KC[lv]:19:14,Zahl);
        If j=4 then Str(KD[lv]:19:14,Zahl);
        If j=5 then Str(KE[lv]:19:14,Zahl);
        If j=6 then Str(KF[lv]:19:14,Zahl);
        If j=7 then Str(KG[lv]:19:14,Zahl);
        If j=8 then Str(KH[lv]:19:14,Zahl);
        If j=9 then Str(KI[lv]:19:14,Zahl);
        If j=10 then Str(KJ[lv]:19:14,Zahl);
        If j=11 then Str(KK[lv]:19:14,Zahl);
        If j=12 then Str(KL[lv]:19:14,Zahl);
        For k:=1 to Length(Zahl) do
        begin {Keine Dezimalpunkte verwenden, sondern Kommata}
          If Zahl[k]<>'.' then write(fout,Zahl[k]);
          If Zahl[k]='.' then write(fout,',');
        end;
        Write(fout,chr(9)); {Daten-Trennung, Tabulator}
      end;
      Writeln(fout,''); {Zeilen-Trennung}
    end;
  end;
  Close(fout);
end;

Function Rlast:Double;
begin
```

Rlast:=21;  
end;

Begin (Hauptprogramm)  
{ Initialisierung - Vorgabe der Werte: } {Wir arbeiten in SI-Einheiten}  
Writeln('Raumenergie-Konverter mit Rotation.');

{ Vorgabe der Werte -> Input-Parameter:}  
epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}  
muo:=4\*pi\*1E-7{Vs/Am}; {Elektrische Feldkonstante}  
lichtgesch:=Sqrt(1/muo/epo){m/s}; Writeln('Lichtgeschwindigkeit c = ',lichtgesch, ' m/s');

{ Spule, Magnet, Kondensator:}  
n:=1000; {Windungszahl der Spule}  
di:=0.09; {Spulenkörper-Durchmesser}  
Dd:=0.0010; {Drahtdurchmesser}  
Bo:=0.5000; {Tesla} {Magnetfeld (Amplitude) des Dauermagneten}  
ls:=0.01; {Meter} {Länge des zylindrischen Spulenkörpers}  
C:=1.641E-6; {Farad} {Kapazität des Kondensators}  
rm:=0.035; {Meter} {Radius des zylindrischen Dauermagneten}  
dm:=0.01; {Meter} {Dicke des zylindrischen Dauermagneten}  
rhom:=7.8E3; {Dichte des Magnet-Materials, Eisen, Kohlrusch Bd.3}

{ Abgeleitete Parameter, keine Eingabe möglich:}  
A:=di\*di; {Meter \* Meter} {Querschnittsfläche der Spule}  
L:=muo\*a\*n\*n/ls; {Induktivität der Spule}  
omo:=1/Sqrt(L\*C); {Kreis-Eigenfrequenz des elektrischen Schwingkreises}  
T:=2\*pi/omo; {Schwingungsdauer des elektrischen Schwingkreises}  
rho:=1.7E-8; {Ohm\*m} {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrusch, T193}  
R:=rho\*(2\*pi\*di\*n)/(pi\*(Dd/2)\*(Dd/2)); {Ohm} {Ohm'scher Widerstand des Spulendrahtes}

{ Sonstige:}  
UmAn:=50; {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}  
AnzP:=AnzPmax; {Anzahl der Zeitschritte insgesamt}  
dt:=0.0001; {sec.} {Größe der Zeitschritte}  
Abstd:=1; {Jeder wievielte Punkte soll geplottet werden}  
mt:=pi\*rm\*rm\*dm\*rhom; {Träge Masse des zylindrischen Dauermagneten}  
J:=1/2\*mt\*rm\*rm; {Trägheitsmoment des zylindrischen Dauermagneten}

{ Anzeige der Werte:}  
Writeln('Induktivitaet der Luft-Spule: L = ',L,' Henry');  
Writeln('Eigen-Kreisfreq harmon.el.Osz.: omo = ',omo:8:4,' Hz => T = ',T:15,'sec.');

Writeln('Laenge des Spulendrahts: ',(2\*pi\*di\*n),' m');  
Writeln('Ohm'scher Widerstand des Spulendrahts: R = ',R:8:2,' Ohm');  
Writeln('Traege Masse des zylindrischen Dauermagneten: mt = ',mt,' kg');  
Writeln('Traegheitsmoment des Dauermagneten: J = ',J,' kg\*m^2');  
Writeln('Gesamtdauer der Betrachtung: ',AnzP\*dt,' sec.');

{ Hier beginnt das Rechenprogramm.}

{ Teil\_04: Mechanische Drehung}  
Writeln(' -> Teil 04: Mechanisch und elektrisch gekoppelt');  
UC:=0;{Volt} Q[0]:=C\*UC; Qpp[0]:=0; Qp[0]:=0; {Elektrische Startwerte}  
phi[0]:=0; phip[0]:=UmAn\*2\*pi; phipp[0]:=0; {Mechanische Startwerte}  
Eent:=0; {Reset für: Entnommene Energie, elektrisch}  
K0[0]:=0;  
K1[0]:=1/2\*L\*Sqr(Qp[0]); {Spulen-Energie}  
K2[0]:=1/2\*C\*Sqr(Q[0]/C); {Kondensator-Energie}  
K3[0]:=1/2\*J\*Sqr(hiph[0]); {Rotations-Energie}  
K4[0]:=K1[0]+K2[0]+K3[0]; {Gesamt-Energie}  
For i:=1 to AnzP do  
begin  
Qpp[i]:=-1/L/C\*Q[i-1]-(R+Rlast)/2/L\*Qp[i-1]; {nach \*5 von S.6}  
Qpp[i]:=Qpp[i]+n\*Bo\*A\*sin(phi[i-1])\*hiph[i-1]/L; {Induzierte Spannung in Spule bringen.}  
Qp[i]:=Qp[i-1]+(Qpp[i]-R/2/L\*Qp[i-1])\*dt; {nach \*3 & \*4 von S.6}  
Q[i]:=Q[i-1]+Qp[i]\*dt;  
phipp[i]:=-Bo\*n\*Qp[i]\*A/J\*sin(phi[i-1]); {Mechanisches Drehmoment, x-Komponente}  
hiph[i]:=hiph[i-1]+phipp[i]\*dt;  
phi[i]:=phi[i-1]+hiph[i]\*dt;  
K0[i]:=0;  
K1[i]:=1/2\*L\*Sqr(Qp[i]); {Spulen-Energie}  
K2[i]:=1/2\*C\*Sqr(Q[i]/C); {Kondensator-Energie}  
K3[i]:=1/2\*J\*Sqr(hiph[i]); {Rotations-Energie}  
K4[i]:=K1[i]+K2[i]+K3[i]; {Gesamt-Energie}  
K5[i]:=Rlast\*Sqr(Qp[i]); {Am Lastwiderstand entnommene Leistung}  
Eent:=Eent+K5[i]\*dt; {Am Lastwiderstand entnommene Energie}  
end;

{ Gesamt-Energie-Bilanz und Anzeige:}  
EmA:=0; EmE:=0; siA:=0; siE:=0;  
For i:=1 to 10 do EmA:=EmA+K4[i]/10; {Mittelwert zu Beginn}  
For i:=AnzP-9 to AnzP do EmE:=EmE+K4[i]/10; {Mittelwert am Ende}  
For i:=1 to 10 do siA:=siA+Sqr(EmA-K4[i]); {Varianz zu Beginn}  
For i:=AnzP-9 to AnzP do siE:=siE+Sqr(EmE-K4[i]); {Varianz am Ende}  
siA:=Sqr(siA)/10; siE:=Sqr(siE)/10; {Standardabweichungen}  
Writeln('Energie-Werte: E Anfang = (' ,EmA:11:7,' +/- ',siA:11:7,' ) Joule');  
Writeln(' E Ende = (' ,EmE:11:7,' +/- ',siE:11:7,' ) Joule');  
delE:=EmE-EmA; sigdelE:=Sqr(Sqr(siE)+Sqr(siA));  
Writeln('=> Veraenderung: delta\_E = (' ,delE:11:7,' +/- ',sigdelE:11:7,' ) Joule');  
Writeln('=> Konvertierte Leistung = (' ,delE/(AnzP\*dt):11:7,' +/- ',sigdelE/(AnzP\*dt):11:7,' ) Watt');  
Writeln('An Rlast entnom. Leistung = ',Eent/(AnzP\*dt):11:7,' Watt');  
ExcelAusgabe('test\_04.dat',12,Q,Qp,Qpp,phi,hiph,phipp,K0,K1,K2,K3,K4,K5);

Wait; Wait;  
End.

Nach versuche mal, ein bisschen mit den Parametern zu spielen und probiere, ob's möglich ist, die entnommene Leistung noch zu erhöhen. Wenn man die ist das File "Var - L - 023. dpr" → "Entnahme - 74 kVA.xls"

Das habe ich für den 50 Hz - Läufer auf eine Leistungsentnahme von 7.1 kVA optimiert.

Nach probiere aber auch noch einen 100 Hz Läufer → File "Var - L - 024. dpr" → "Entnahme - 227 kVA.xls"

"Var - L - 025. dpr" → "Entnahme - 1070 kVA.xls"

Die Ströme sind am Ende schon bei über 15 Ampere (bis 18 Ampere) (Wohlerstrom).  
 ← Noch stärkeres darf ich das Ding nicht belasten, sonst brennen mir die Leitungen durch.  
 Deshalb habe ich mich entschieden, bei gut 1 kW aufzuhören, die Parameter zu optimieren.

Die experimentell größte Herausforderung wird sein:

Im File "Entnahme - 1070 kVA.xls" sehen wir, wieviel Ladung fließt:

Maximal  $Q = 0.015 \text{ Coulomb!}$

$$\text{bei } C = \frac{Q}{U} \Rightarrow U = \frac{Q}{C} \Rightarrow U = \frac{0.015}{0.23 \cdot 10^{-6}} \quad V = 65218 \text{ Volt}$$

mit  $C = 0.23 \mu F$

Man muß eine Kondensatorbank machen, die das aushält!

## Var L 025.dpr → 1.07 KiloWatt Leistungsentnahme

```
Program Magnetkonverter_mit_realer_Leistungsentnahme;
{$APPTYPE CONSOLE}
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

Const AnzPmax=10000; {Anzahl der Zeitschritte zur Lsg. der Dgl.}

Type Feld = Array[0..AnzPmax] of Double;

Var epo,muo      : Double;    {Naturkonstanten}
    lichtgesch   : Double;    {Lichtgeschwindigkeit}
    n            : LongInt;    {Windungszahl der Spule}
    A            : Double;    {Querschnittsfläche der Spule}
    Bo          : Double;    {Magnetfeld (Amplitude) des Dauermagneten}
    ls          : Double;    {Länge der zylindrischen Spule}
    di          : Double;    {Durchmesser des Spulenkörpers}
    Dd          : Double;    {Drahtdurchmesser}
    rm          : Double;    {Radius des Dauermagneten}
    L           : Double;    {Induktivität der Spule}
    C           : Double;    {Kapazität des Kondensators}
    R           : Double;    {Ohm'scher Widerstand des Spulendrahtes}
    rho         : Double;    {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
    phi,phip,hipp : Feld;    {Drehwinkel und dessen Ableitungen}
    Q,Qp,Qpp    : Feld;    {Ladung und deren Ableitungen}
    i           : LongInt;    {Laufvariable}
    AnzP        : LongInt;    {Anzahl der tatsächlich berechneten Zeit-Schritte}
    dt          : Double;    {Dauer der Zeitschritte zur Lsg. der Dgl.}
    Abstd       : Integer;    {Jeder wievielte Punkte soll geplottet werden}
    omo         : Double;    {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
    T           : Double;    {Schwingungsdauer des elektrischen Schwingkreises}
    UC,UL       : Double;    {Kondensatorspannung und Spulenspannung}
    rhom        : Double;    {Dichte des Magnetmaterials}
    dm          : Double;    {Dicke des zylindrischen Dauermagneten}
    mt          : Double;    {Träge Masse des zylindrischen Dauermagneten}
    J           : Double;    {Trägheitsmoment des zylindrischen Dauermagneten}
    K0,K1,K2,K3,K4,K5 : Feld; {Kontroll-Felder für Plot-Zwecke}
    EmA,EmE,siA,siE : Double; {Energie: Mittelwerte und Sigma "Anfang" und "Ende"}
    delE,sigdelE   : Double; {Veränderung der Energie-Mittelwerte "Anfang" zu "Ende"}
    UmAn          : Double; {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}
    Eent          : Double; {Entnommene Energie, elektrisch}
    Rlast         : Double; {Ohm'scher Lastwiderstand}

Procedure Wait;
Var Ki : Char;
begin
  Write('<W>'); Read(Ki); Write(Ki);
  If Ki='e' then Halt;
end;

Procedure ExcelAusgabe(Name:String;Spalten:Integer;KA,KB,KC,KD,KE,KF,KG,KH,KI,KJ,KK,KL:Feld);
Var fout : Text; {Bis zu 12 Spalten zum Aufschreiben der Ergebnisse}
    lv,j,k : Integer; {Laufvariable}
    Zahl : String; {Die ins Excel zu druckenden Zahlen}
begin
  Assign(fout,Name); Rewrite(fout); {File öffnen}
  For lv:=0 to AnzP do {von "plotanf" bis "plotend"}
  begin
    If (lv mod Abstd)=0 then
    begin
      For j:=1 to Spalten do
      begin {Kolumnen drucken}
        If j=1 then Str(KA[lv]:19:14,Zahl);
        If j=2 then Str(KB[lv]:19:14,Zahl);
        If j=3 then Str(KC[lv]:19:14,Zahl);
        If j=4 then Str(KD[lv]:19:14,Zahl);
        If j=5 then Str(KE[lv]:19:14,Zahl);
        If j=6 then Str(KF[lv]:19:14,Zahl);
        If j=7 then Str(KG[lv]:19:14,Zahl);
        If j=8 then Str(KH[lv]:19:14,Zahl);
        If j=9 then Str(KI[lv]:19:14,Zahl);
        If j=10 then Str(KJ[lv]:19:14,Zahl);
        If j=11 then Str(KK[lv]:19:14,Zahl);
        If j=12 then Str(KL[lv]:19:14,Zahl);
        For k:=1 to Length(Zahl) do
        begin {Keine Dezimalpunkte verwenden, sondern Kommata}
          If Zahl[k]<>'.' then write(fout,Zahl[k]);
          If Zahl[k]='.' then write(fout,',');
        end;
        Write(fout,chr(9)); {Daten-Trennung, Tabulator}
      end;
      Writeln(fout,''); {Zeilen-Trennung}
    end;
  end;
  Close(fout);
end;
```

zu S.60

```

Begin {Hauptprogramm}
{ Initialisierung - Vorgabe der Werte: }           {Wir arbeiten in SI-Einheiten}
  Writeln('Raumenergie-Konverter mit Rotation.');
```

{ Vorgabe der Werte -> Input-Parameter:}

```

  epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}
  muo:=4*pi*1E-7{Vs/Am};      {Elektrische Feldkonstante}
  lichtgesch:=Sqrt(1/muo/epo){m/s};  Writeln('Lichtgeschwindigkeit c = ',lichtgesch, ' m/s');
```

{ Spule, Magnet, Kondensator:}

```

  n:=1600;      {Windungszahl der Spule}
  di:=0.09;     {Spulenkörper-Durchmesser}
  Dd:=0.0010;   {Drahtdurchmesser}
  Bo:=0.700;    {Tesla} {Magnetfeld (Amplitude) des Dauermagneten}
  ls:=0.01;     {Meter} {Länge des zylindrischen Spulenkörpers}
  C:=0.23E-6;   {Farad} {Kapazität des Kondensators}
  rm:=0.039;    {Meter} {Radius des zylindrischen Dauermagneten}
  dm:=0.01;     {Meter} {Dicke des zylindrischen Dauermagneten}
  rhom:=7.8E3;  {Dichte des Magnet-Materials, Eisen, Kohlrausch Bd.3}
```

{ Abgeleitete Parameter, keine Eingabe möglich:}

```

  A:=di*di; {Meter * Meter} {Querschnittsfläche der Spule}
  L:=muo*a*n*n/ls; {Induktivität der Spule}
  omo:=1/Sqrt(L*C); {Kreis-Eigenfrequenz des elektrischen Schwingkreises}
  T:=2*pi/omo;     {Schwingungsdauer des elektrischen Schwingkreises}
  rho:=1.7E-8; {Ohm*m} {Spez. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
  R:=rho*(2*pi*di*n)/(pi*(Dd/2)*(Dd/2)); {Ohm} {Ohm'scher Widerstand des Spulendrahtes}
```

{ Sonstige:}

```

  UmAn:=100;      {Startwert: Umdrehungen pro Sekunde bei Anlaufen des Rotors}
  Rlast:=28;      {Ohm'scher Lastwiderstand}
  AnzP:=AnzPmax;  {Anzahl der Zeitschritte insgesamt}
  dt:=0.0001; {sec.} {Größe der Zeitschritte}
  Abstd:=1;       {Jeder wievielte Punkte soll geplottet werden}
  mt:=pi*rm*rm*dm*rhom; {Träge Masse des zylindrischen Dauermagneten}
  J:=1/2*mt*rm*rm; {Trägheitsmoment des zylindrischen Dauermagneten}
```

{ Anzeige der Werte:}

```

  Writeln('Induktivität der Luft-Spule: L = ',L,' Henry');
  Writeln('Eigen-Kreisfreq harmon.el.Osz.: omo = ',omo:8:4,' Hz => T = ',T:15,'sec.');
```

{ Laenge des Spulendrahts: ',(2\*pi\*di\*n),' m');}

```

  Writeln('Ohm'scher Widerstand des Spulendrahts: R = ',R:8:2,' Ohm');
```

{ Traege Masse des zylindrischen Dauermagneten: mt = ',mt,' kg');}

```

  Writeln('Traegheitsmoment des Dauermagneten: J = ',J,' kg*m^2');
```

{ Gesamtdauer der Betrachtung: ',AnzP\*dt,' sec.');

{ Hier beginnt das Rechenprogramm.}

```

  Writeln('Mechanisch und elektrisch gekoppelte Schwingung.');
```

UC:=0;{Volt} Q[0]:=C\*UC; Qpp[0]:=0; Qp[0]:=0; {Elektrische Startwerte}

```

  phi[0]:=0;  phip[0]:=UmAn*2*pi; phipp[0]:=0; {Mechanische Startwerte}
  Eent:=0;    {Reset für: Entnommene Energie, elektrisch}
```

K0[0]:=0;

```

  K1[0]:=1/2*L*Sqr(Qp[0]); {Spulen-Energie}
  K2[0]:=1/2*C*Sqr(Q[0]/C); {Kondensator-Energie}
  K3[0]:=1/2*J*Sqr(hiph[0]); {Rotations-Energie}
  K4[0]:=K1[0]+K2[0]+K3[0]; {Gesamt-Energie}
  K5[0]:=0;
```

For i:=1 to AnzP do

```

begin
  Qpp[i]:=-1/L/C*Q[i-1]-(R+Rlast)/2/L*Qp[i-1]; {nach *5 von S.6}
  Qpp[i]:=Qpp[i]+n*Bo*A*sin(phi[i-1])*hiph[i-1]/L; {Induzierte Spannung in Spule bringen.}
  Qp[i]:=Qp[i-1]+(Qpp[i]-R/2/L*Qp[i-1])*dt; {nach *3 & *4 von S.6}
  Q[i]:=Q[i-1]+Qp[i]*dt;
  phipp[i]:=-Bo*n*Qp[i]*A/J*sin(phi[i-1]); {Mechanisches Drehmoment, x-Komponente}
  phip[i]:=hiph[i-1]+phipp[i]*dt;
  phi[i]:=phi[i-1]+hiph[i]*dt;
  K0[i]:=0;
  K1[i]:=1/2*L*Sqr(Qp[i]); {Spulen-Energie}
  K2[i]:=1/2*C*Sqr(Q[i]/C); {Kondensator-Energie}
  K3[i]:=1/2*J*Sqr(hiph[i]); {Rotations-Energie}
  K4[i]:=K1[i]+K2[i]+K3[i]; {Gesamt-Energie}
  K5[i]:=Rlast*Sqr(Qp[i]); {Am Lastwiderstand entnommene Leistung}
  Eent:=Eent+K5[i]*dt; {Am Lastwiderstand entnommene Energie}
end;
```

{ Gesamt-Energie-Bilanz und Anzeige:}

```

  EmA:=0; EmE:=0; siA:=0; siE:=0;
  For i:=1 to 10 do EmA:=EmA+K4[i]/10; {Mittelwert zu Beginn}
  For i:=AnzP-9 to AnzP do EmE:=EmE+K4[i]/10; {Mittelwert am Ende}
  For i:=1 to 10 do siA:=siA+Sqr(EmA-K4[i]); {Varianz zu Beginn}
  For i:=AnzP-9 to AnzP do siE:=siE+Sqr(EmE-K4[i]); {Varianz am Ende}
  siA:=Sqrt(siA)/10; siE:=Sqrt(siE)/10; {Standardabweichungen}
  Writeln('Energie-Werte: E_Anfang = (' ,EmA:11:7,' +/- ',siA:11:7,' ) Joule');
```

Writeln(' E\_End = (' ,EmE:11:7,' +/- ',siE:11:7,' ) Joule');

```

  delE:=EmE-EmA; sigdelE:=Sqr(Sqr(siE)+Sqr(siA));
  Writeln('=> Veraenderung: delta_E = (' ,delE:11:7,' +/- ',sigdelE:11:7,' ) Joule');
```

Writeln('=> Konvertierte Leistung = (' ,delE/(AnzP\*dt):11:7,' +/- ',sigdelE/(AnzP\*dt):11:7,' ) Watt');

```

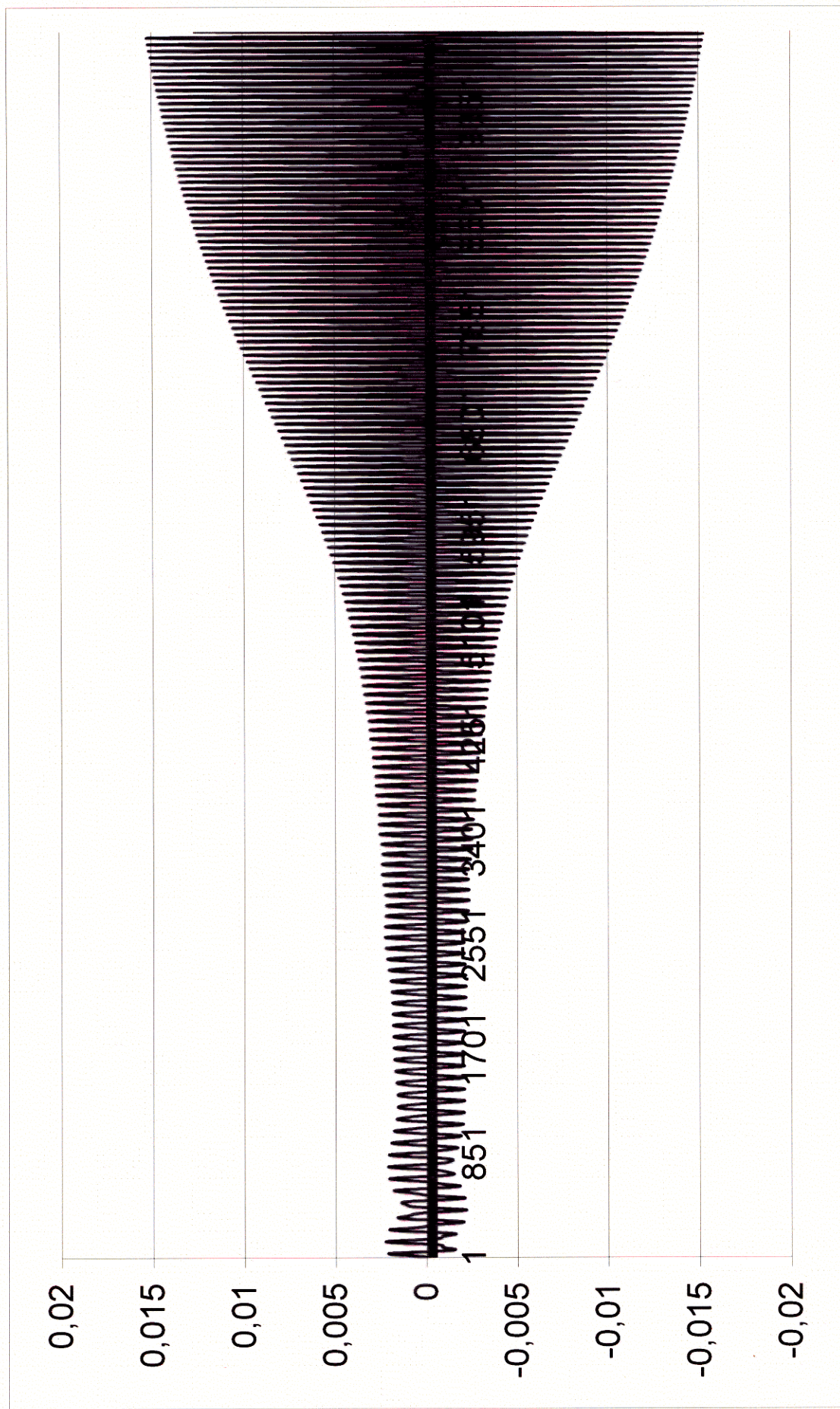
  Writeln('An Rlast entnom. Leistung = ',Eent/(AnzP*dt):11:7,' Watt');
```

ExcelAusgabe('test\_04.dat',12,Q,Qp,Qpp,phi,hiph,phipp,K0,K1,K2,K3,K4,K5);

```

Wait; Wait;
End.
```

Ladung als Funktion der Zeit aus „Entnahme\_1070Watt.xls“



Es folgen Fragen eines von mir sehr hoch geschätzten Kollegen, die Fragen in blauer Farbe, meine Antworten in roter Farbe.

Habe ich das richtig verstanden, dass sich die Drehzahl des Magneten bis zu einem maximalen Wert einstellt, ab dem eine weitere Erhöhung der Drehzahl für eine Verschlechterung der Abstimmung der Systemparameter sorgen würde?

Ja, in meinem Beispiel ist es so. Andere Raumenergie-Konverter arbeiten anders, aber in diesem Beispiel überdreht die Maschine nicht von alleine.

Ist nun diese Maximaldrehzahl abhängig von der angelegten Last bzw. wie stark wirken sich Laständerungen auf die Drehzahl aus?

Natürlich spielt die Last mit. Wie sich die eingestellte Drehzahl als Funktion der Last ändert, kann man sehr leicht herausfinden, wenn man in meinem Algorithmus verschiedene Lastwiderstände eingibt.

Leistung und Drehzahl hängen sicher zusammen. Ist das ein linearer Zusammenhang, wie ich mal vermuten würde?

Die Leistung steigt extrem stark mit der Drehzahl, viel mehr als linear, vielleicht sogar über-exponentiell. Man braucht ja nur in meinem Algorithmus verschiedene Drehzahlen eingeben, das System dann immer wieder justieren (optimieren) und schauen, was passiert. Bei 3000 U/min konnte ich gerade eben 7 Watt herausholen.

Könnte man die Drehzahl - eben die maximal, bis zu der wieder die Verstimmung beginnt - irgendwie einstellen?

Man gibt die Drehzahl vor justiert dann das System, indem man die Kapazität des Kondensators so einstellt, daß der elektr. Schwingkreis zur Rotationsfrequenz des Magneten paßt. Das macht man zunächst ohne Ohm'schen Widerstand. Wenn man den guten Arbeitspunkt gefunden hat, dann fährt man langsam den Ohm'schen Widerstand hoch und gleicht immer wieder die Zahl der Spulenwindungen und den Kondensator an.

Was ich immer noch nicht kapiert habe, ist das Phänomen, dass in den verwendeten Gleichungen zur Berechnung der gekoppelten Schwingkreise die Raumenergie nirgends implizit/explicit auftritt bzw. irgendwie in Parametern von Anfang an eingeführt ist. Wie erklärst Du das? Oder willst Du das künftig noch näher ausarbeiten?

Das Verhalten liegt außerhalb des mathematischen Funktionsbegriffs, deshalb kann man die Funktion nicht in analytischer Form explicit angeben. Deshalb geht eben nur die numerische Lösung der Differentialgleichungen. Das Problem, daß man eine Funktion nicht in analytischer Form explicit angeben kann, gibt es hin- und wieder in der Mathematik, z.B. auch bei der Integration der Dichtefunktion der Gauß-Verteilung. Das liegt einfach nur daran, daß der klassische Funktionsbegriff in der Mathematik nicht alle denkbar möglichen Kurvenverläufe umfassen kann.

Nach versuche mal, ein bisschen mit den Parametern zu spielen und probiere, ob's möglich ist, die entnommene Leistung noch zu erhöhen. Dann mache ich das File "Var-L-023.dpr" → "Entnahme-744W.xls". Das habe ich für den 50Hz-Läufer auf eine Leistungsentnahme von 7.1Watt angepasst.

Nach probiere aber auch noch einen 100 Hz Läufer → File "Var-L-024.dpr" → "Entnahme-227Watt.xls".  
 "Var-L-025.dpr" → "Entnahme-1070Watt.xls".

Die Ströme sind am Ende schon bei über 15 Ampere (bis 18 Ampere) (Wohrstrom).  
 ← Noch stärker darf es das Ding nicht belasten, sonst brennen mir die Leitungen durch.  
 Deshalb habe ich mich entschieden, bei gut 1kW aufzuhören, die Parameter zu optimieren.

Die experimentell größte Herausforderung wird sein:

Im File "Entnahme-1070Watt.xls" sehen wir, wieviel Ladung fließt:

Maximal  $Q = 0.015 \text{ Coulomb!}$

bei  $C = \frac{Q}{U} \Rightarrow U = \frac{Q}{C} \Rightarrow U = \frac{0.015}{0.23 \cdot 10^{-6}} V = 65218 \text{ Volt}$   
 mit  $C = 0.23 \mu F$

Man muß eine Kondensatorbank machen, die das aushält!

Achille sollte so sehen: L ablesen, Rest anpassen  
 Dort könnte man rechnen: (Mit meinem Algo "Var-L-025.dpr")  
 $n = 50 \Rightarrow U \approx 2200 \text{ Volt}, \uparrow \uparrow, R_{\text{last}} \text{ anpassen}, C \approx 9 \mu F \dots 10 \mu F$   
 $n = 20 \Rightarrow U \approx 800 \text{ Volt}, \uparrow \uparrow \uparrow, R_{\text{last}} \text{ anpassen}, C \approx 20 \mu F$   
 $n = 10 \Rightarrow U \approx 400 \text{ Volt}, \uparrow \uparrow \uparrow, R_{\text{last}} \text{ anpassen}, C \approx 40 \mu F$