

# Podklady ke 3. a 4. přednášce - paraxial approximation

Zakladní konstanty + načtení balíčku, které se budou používat při výpočtech

```
In [1]: qe = 1.602e-19; #Elementarni naboj
me = 9.109e-31; #Hmotnost elektronu
c = 299792458; #Rychlost svetla
eta = sqrt(qe/2/me)
using Plots #Balicek pro grafy
using DifferentialEquations # Balicek pro vypocet diferencialnich rovnic
using SpecialFunctions # Specialni funkce
#using ForwardDiff # Derivace funkcí
using PolyChaos #Balicek pro vypocet hermiteovych polynomu
using Roots # Balicek pro nalezeni korenu funkce
using LaTeXStrings # Latex v grafech
```

WARNING: using SpecialFunctions.eta in module Main conflicts with an existing identifier.

Funkce pro derivace error funkce

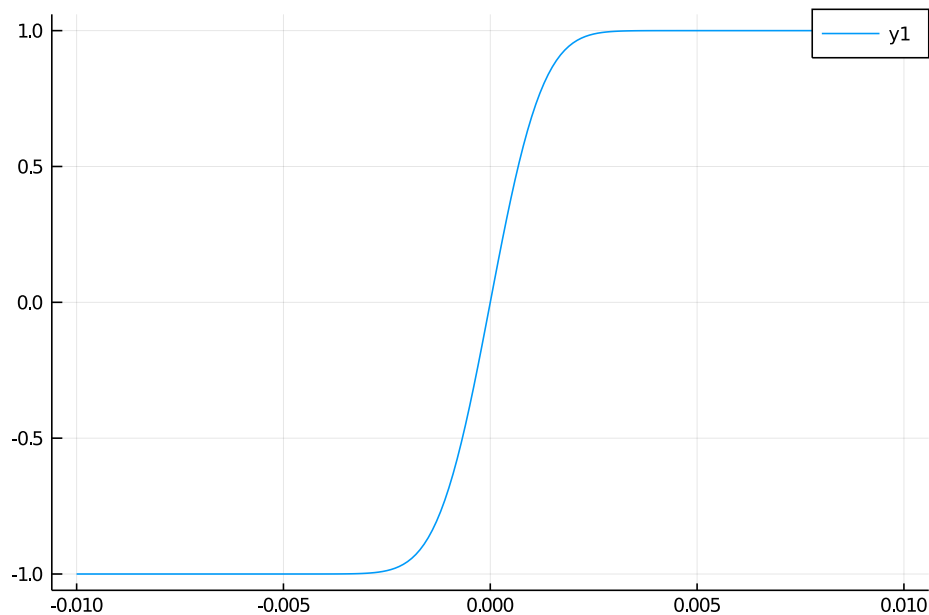
$$\operatorname{erf}^{(n)}(x) = \frac{2(-1)^{(n-1)}}{\sqrt{\pi}} H_{n-1}(x) \exp(-x^2), \quad n > 0$$

$H_n$  Hermiteovy polynomy - viz. Wikipedia

```
In [2]: hrca,hrcb = rm_hermite(100) #Hermite recurrence coefficients (physical monomial)
Hn = (n,x) -> 2^n*PolyChaos.evaluate(n,x,hrca,hrcb) #Hermite polynomial of n-th order
function dnerf(n::Int64,x::Float64) #n stupen derivace n>=0, x - bod v nemz funkci pocita me
    if n==0 #Nulta derivace se musi implementovat zvlast
        return erf(x)
    else
        return 2*(-1)^(n-1)/sqrt(pi)*Hn(n-1,x)*exp(-x^2);
    end
end
;
```

```
In [3]: #Plot error funkce, pripadne jejich derivaci, staci zmenit prvni argument dnerf
z1 = (-1:0.01:1)*1e-2
sigma=1e-3
plot(z1,dnerf.(0,z1/(sqrt(2)*sigma)))
```

Out[3]:



## Pole v systému

Použijeme jednoduchou aproximaci magnetického pole Gausovkou

$$B(z) = B_m \exp\left(-\frac{z^2}{s_B^2}\right)$$

pro výpočet derivaci využijeme derivaci error funkce:

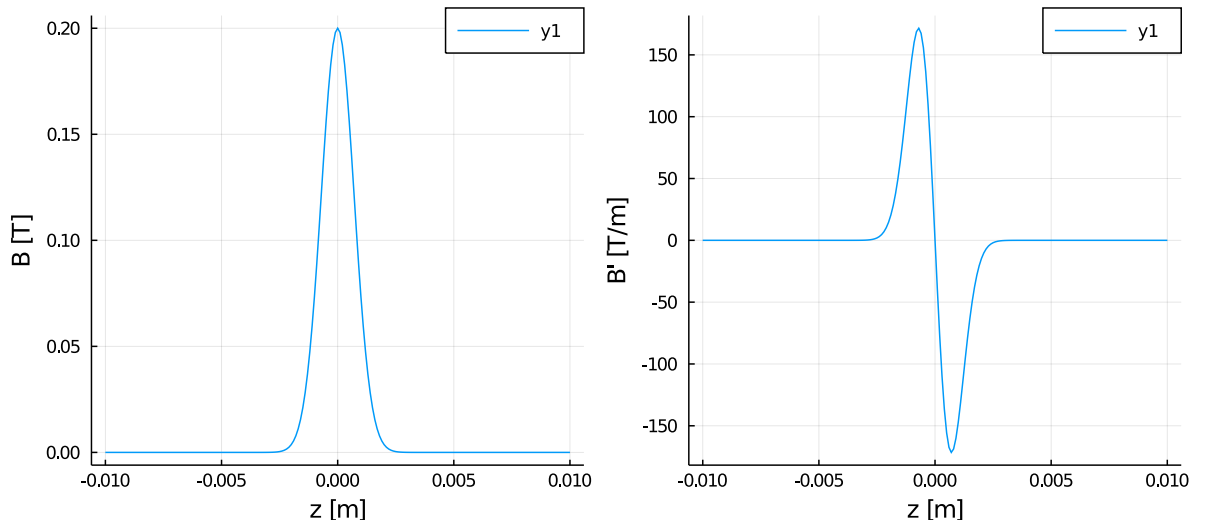
$$\exp(-z^2) = \frac{\pi}{2} \operatorname{erf}'(z)$$

tedy

$$B^{(n)}(z) = B_m \frac{d^n}{dz^n} \exp\left(-\frac{z^2}{s_B^2}\right) = \frac{\sqrt{\pi}}{2s_B^n} \operatorname{erf}^{(n+1)}\left(\frac{z}{s_B}\right)$$

```
In [4]: # Function for Gaussian field n=stupen derivate, z - bod ve kterem funkci pocitam,
# Bm - maximum osoveho pole, sB - parametr urcujici sirku Gausovky
dnB = (n,z,Bm,sB) -> Bm/2*sqrt(pi)/sB^n*dnerf(n+1,z/sB)
z1 = (-1:0.01:1)*1e-2
p=Any[]
Bm=0.2;sB=1e-3
push!(p,plot(z1,dnB.(0,z1,Bm,sB),xlabel="z [m]",ylabel="B [T]"))
push!(p,plot(z1,dnB.(1,z1,Bm,sB),xlabel="z [m]",ylabel="B' [T/m]"))
plot(p...,layout=(1,2),size=(800,350))
```

Out[4]:



Pro elektrostatickou čocku použijeme jednoduché error funkce, Budeme uvažovat tři elektrodovou elektrostatickou cocku, jejich potencial je ve vektoru V. Uvazujeme hladky prechod pomoci error funkce

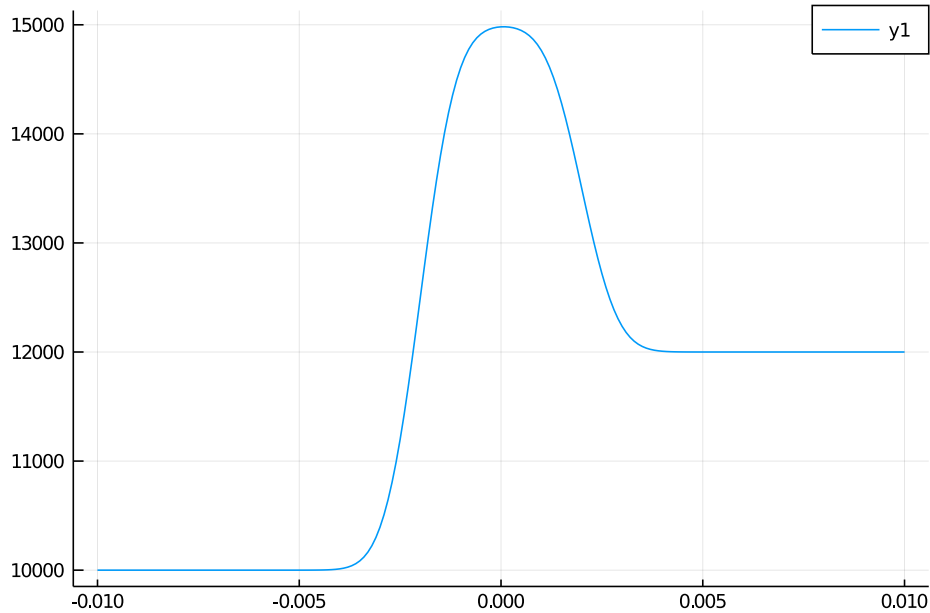
$$\Phi = \frac{1}{2}(\operatorname{erf}((z - z1)/s) + 1)(V[2] - V[1]) + \frac{1}{2}(\operatorname{erf}((z - z2)/s) + 1)(V[3] - V[2]) + V[1]$$

```

In [5]: ze=[-2,2]*1e-3 # z1 a z2 - stredy mezer mezi elektrodami ...
s1=1.0e-3; # Rychlost prechodu
function dPhiV(n::Int64,z::Float64,V::Array{Float64,1})
    if n==0
        dP = (dnerf(n,(z-ze[1])/s1)+1)/2/s1^n*(V[2]-V[1])+(dnerf(n,(z-ze[2])/s1)+1)/2/s1
        ^n*(V[3]-V[2]) +V[1]
    else
        dP = (dnerf(n,(z-ze[1])/s1))/2/s1^n*(V[2]-V[1])+(dnerf(n,(z-ze[2])/s1))/2/s1^n*(
        V[3]-V[2])
    end
    return dP
end
end
#Plot osoveho potencialu, pripadne jeho derivaci
dPhi(n,z) = dPhiV(n,z,[1e4; 1.5e4; 1.2e4])
plot(z1,dPhi.(0,z1))

```

Out[5]:



## Paraxiální rovnice trajektorie

Paraxiální rovnice trajektorie ma tvar:

$$x'' + \frac{\gamma\Phi'}{2\Phi^*}x' + \frac{\gamma\Phi''}{4\Phi^*}x + \frac{\eta B}{\sqrt{\Phi^*}}y' + \frac{\eta B'}{2\sqrt{\Phi^*}}y = 0$$

$$y'' + \frac{\gamma\Phi'}{2\Phi^*}y' + \frac{\gamma\Phi''}{4\Phi^*}y - \frac{\eta B}{\sqrt{\Phi^*}}x' - \frac{\eta B'}{2\sqrt{\Phi^*}}x = 0$$

Pro přepsání do kodu je nutné přejít do soustavy 4 diferencialnich rovnic 1. radu

$$x' = dx$$

$$y' = dy$$

$$dx' = -\frac{\gamma\Phi'}{2\Phi^*}dx - \frac{\gamma\Phi''}{4\Phi^*}x - \frac{\eta B}{\sqrt{\Phi^*}}dy - \frac{\eta B'}{2\sqrt{\Phi^*}}y$$

$$dy' = -\frac{\gamma\Phi'}{2\Phi^*}dy - \frac{\gamma\Phi''}{4\Phi^*}y + \frac{\eta B}{\sqrt{\Phi^*}}dx + \frac{\eta B'}{2\sqrt{\Phi^*}}x$$

```

In [13]: dPhi(n,z) = dPhiV(n,z,[10000.0,10000.0,10000.0]);
dBax(n,z) = dnB(n,z,0.3,1e-3);
Phir(Phi) = Phi*(1+qe*Phi/(2*me*c^2))
#dBax(n,z) = 0.0;
function f(u,p,z)
    Ph = dPhi(0,z); dPh = dPhi(1,z); d2Ph = dPhi(2,z)
    Phr = Phir(Ph)
    B = dBax(0,z); dB = dBax(1,z)
    du = zeros(Float64,4)
    du[1] = u[3]
    du[2] = u[4]
    du[3] = -dPh/(2*Phr)*u[3]-d2Ph/(4*Phr)*u[1]-eta*B/sqrt(Phr)*u[4]-eta*dB/(2*sqrt(Phr))
    du[4] = -dPh/(2*Phr)*u[4]-d2Ph/(4*Phr)*u[2]+eta*B/sqrt(Phr)*u[3]+eta*dB/(2*sqrt(Phr))
    return du
end

```

Out[13]: f (generic function with 1 method)

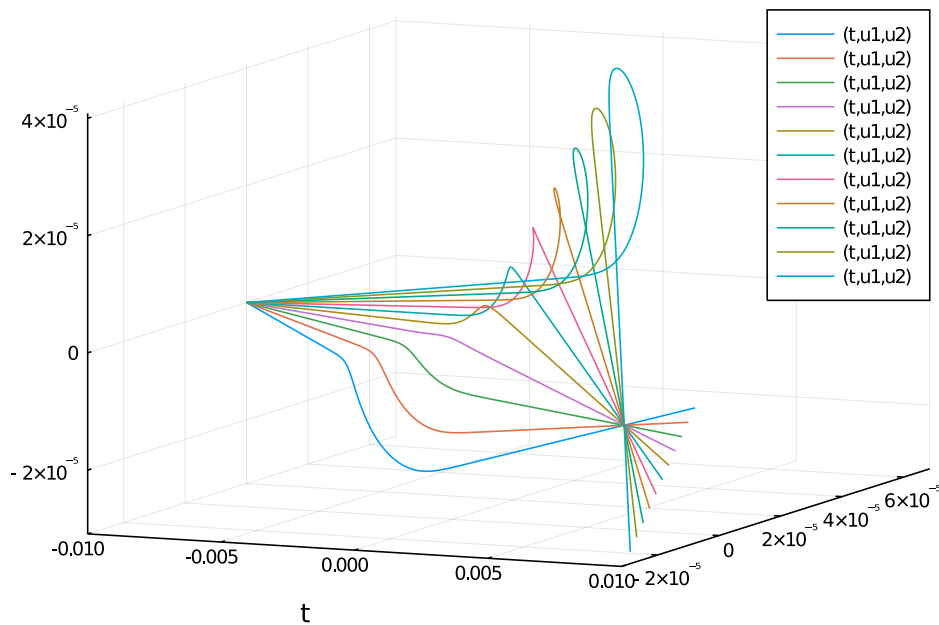
zobrazime nekolik paprsku a) pomoci primeho vypoctu (Vypocet paraxialni rovnice pro kazdou pocatecni podminku zvlast)

```

In [14]: thx=(-1:0.2:1)*5e-3;
zspan = (-1.0e-2,1.0e-2)
x0=2e-5; y0=0;
p=plot()
for i=1:length(thx)
    u0 = [x0,y0,thx[i],0]
    prob = ODEProblem(f,u0,zspan)
    sol = solve(prob,Tsit5(), reltol=1e-10, abstol=1e-12)
    plot!(p,sol,vars=(0,1,2))
end
plot(p)

```

Out[14]:



b) pomoci linearni kombinace reseni. Vsechny řešené lineární diferenciální rovnice tvoří vektorový prostor. Jako bazi zvolíme řešení:

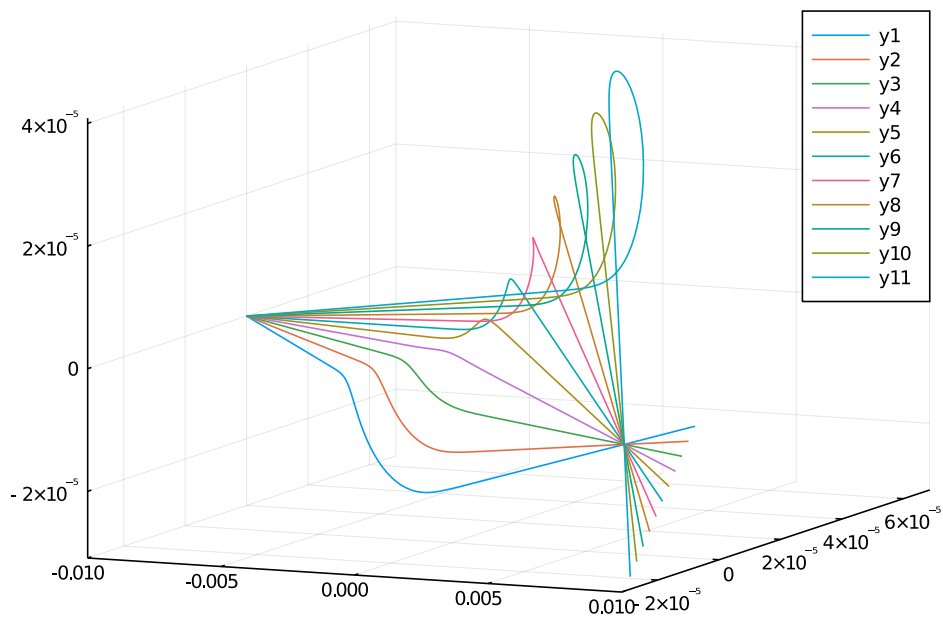
$$\begin{aligned}
 \vec{q}_x : \vec{q}_x(z_0) &= [1, 0], \vec{q}'_x(z_0) = [0, 0] \\
 \vec{q}_y : \vec{q}_y(z_0) &= [0, 1], \vec{q}'_y(z_0) = [0, 0] \\
 \vec{h}_x : \vec{q}_x(z_0) &= [0, 0], \vec{h}'_x(z_0) = [1, 0] \\
 \vec{h}_y : \vec{q}_y(z_0) &= [0, 0], \vec{h}'_y(z_0) = [0, 1]
 \end{aligned}$$

```
In [15]: u0=[1,0,0,0]
prob = ODEProblem(f,u0,zspan)
solgx = solve(prob,Tsit5(),reltol=1e-10,abstol=1e-12);
u0=[0,1,0,0]
prob = ODEProblem(f,u0,zspan)
solgy = solve(prob,Tsit5(),reltol=1e-10,abstol=1e-12);
u0=[0,0,1,0]
prob = ODEProblem(f,u0,zspan)
solhx = solve(prob,Tsit5(),reltol=1e-10,abstol=1e-12);
u0=[0,0,0,1]
prob = ODEProblem(f,u0,zspan)
solhy = solve(prob,Tsit5(),reltol=1e-10,abstol=1e-12);
```

Je výhodné přejít do komplexních souřadnic  $w = x + iy$ ,  $\bar{w} = x - iy$

```
In [16]: z1 = -0.01:0.0001:0.01
p=plot();
w = (z,wo,dwo) -> sum((real(wo)*solgx(z) + imag(wo)*solgy(z) + real(dwo)*solhx(z) + imag
(dwo)*solhy(z))[1:2]).*
[1;lim])
for i=1:length(thx)
w1 = w.(z1,x0+lim*y0,thx[i])
plot!(p,z1,real(w1),imag(w1))
end
plot(p)
xlims!((-0.01,0.01))
```

Out[16]:



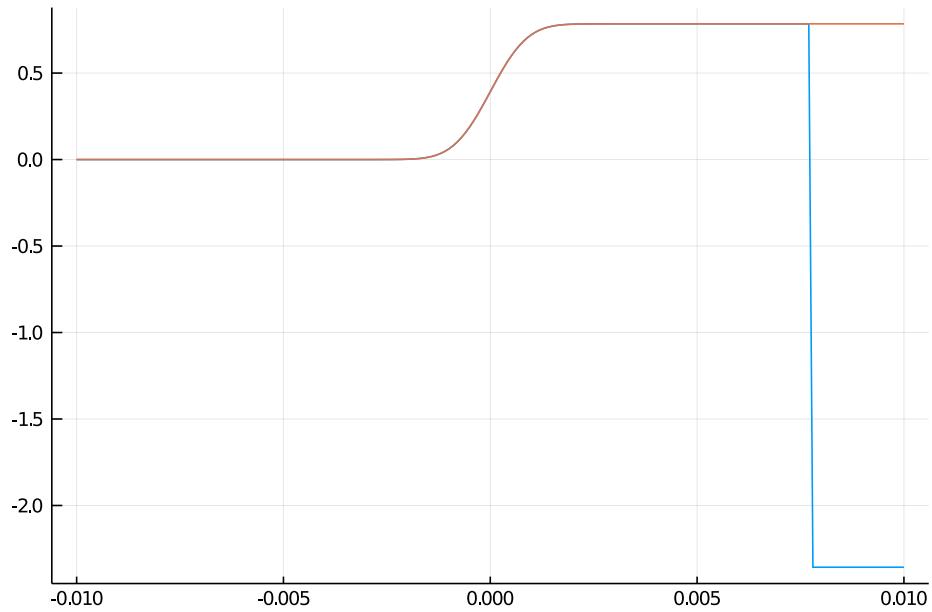
Rotace svazku v laboratorních souřadnicích

$$\theta(z) = \int_{z_0}^z \frac{\eta^2 B(z)}{2\Phi^{*\frac{1}{2}}} dz$$

```
In [19]: # Integral pocitame pomoci reseni diferencialni rovnice (neni to moc efektivni ...)
prob3=ODEProblem((u,p,z)->eta*dBax(θ,z)/(2*sqrt(Phir(dPhi(θ,z)))),0.0,(-0.01,0.01))
thr = solve(prob3,Tsit5(),reltol=1e-8, abstol=1e-10);

plot(z1,angle.(w.(z1,θ,1e-3)),legend=false) #Vypocet rotace z trajektorii
plot!(z1,thr.(z1),legend=false) #Vypoctena rotace (behem integrace par. rovnice)
```

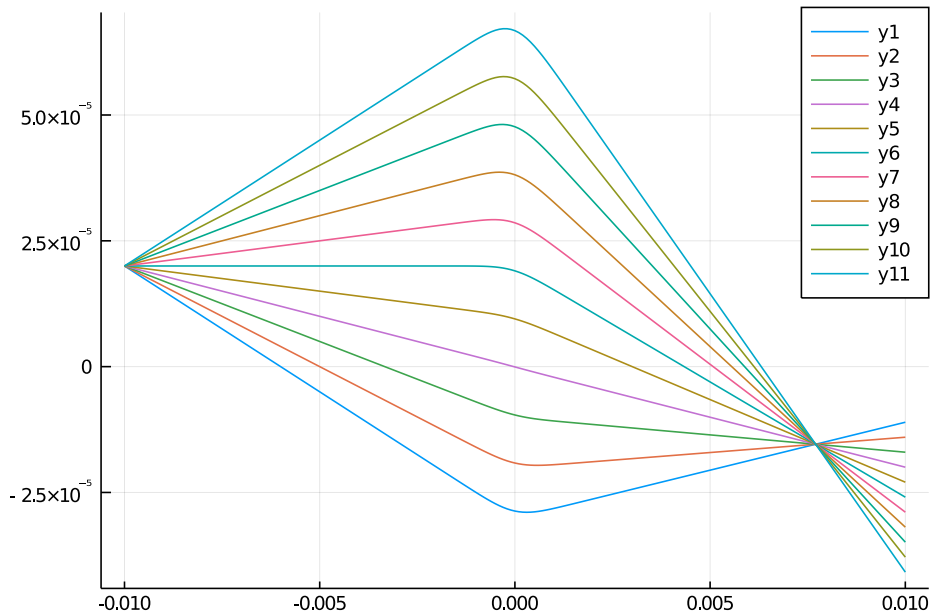
Out[19]:



Prechod do rotacnich soradnic

```
In [20]: p=plot()
for i=1:length(thx)
    plot!(p,z1,real(w.(z1,x0,thx[i]).*exp.(-lim*thr.(z1))))
end
plot(p)
```

Out[20]:



## Paraxialni aproximace v rotačních souřadnicích - magnetická čočka

$$u'' + \frac{\gamma\Phi'}{2\Phi^*}u' + \frac{\gamma\Phi'' + \eta^2 B^2}{4\Phi^*}u = 0$$

V případě čistě magnetické čočky dostaneme

$$u'' + \frac{\eta^2 B^2}{4\Phi^*}u = 0$$

Pro osově magnetické pole zvolíme jednoduchou aproximaci gausovkou  $B(z) = B_{max} \exp(-z^2/s_B^2)$  a rovnici trajektorie převedem na dvě ODE prvního řádu

$$\begin{aligned}u' &= du \\ du' &= -\frac{\eta^2 B(z)}{4\Phi^*}u\end{aligned}$$

$$B_{max} = 0.3T \text{ a } \Phi = 8 \text{ keV}$$

```
In [22]: Bm =0.3; sB = 1e-3
Bax(z) = dnB(0,z,Bm,sB); Ph = 8000; Phr = Ph*(1+qe*Ph/(2*me*c^2))
f2(u,p,z) = [u[2], -eta^2*Bax(z)^2/(4*Phr)*u[1]];
```

Jako bázi vektoru řešení parxiální rovnice zvolíme řešení  $g(z) : g(z_o) = 1, g'(z_o) = 0, h(z) : h(z_o) = 0, h'(z_o) = 1$ . Pozice predmětu  $z_o = -0.01$  m. Libovolný poprsek je pak dán polohou a směrnici v predmětu:

$$u(z) = u_o g(z) + u'_o h(z)$$

a jeho směrnice

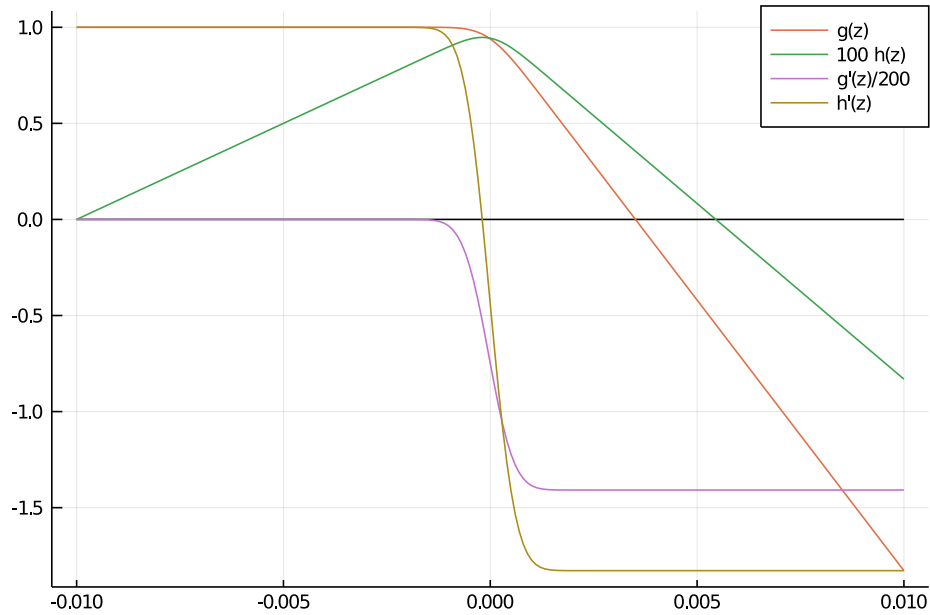
$$u'(z) = u_o g'(z) + u'_o h'(z)$$

```
In [23]: zo = -0.01;
zspan = (-1.0e-2, 2.0e-2) #Hranice integrace
u0=[1,0] #Pocatecni podminka pro trajektorii g
#Vypocet trajektorie g
prob2 = ODEProblem(f2,u0,zspan)
solg = solve(prob2,Tsit5(), reltol=1e-8, abstol=1e-10);
u0 = [0,1] #Pocatecni podminky pro trajektorii h
#Vypocet trajektorie h
prob2 = ODEProblem(f2,u0,zspan)
solh = solve(prob2,Tsit5(), reltol=1e-8, abstol=1e-10);
h = z->solh(z)[1] #funkce trajektorie h
dh = z->solh(z)[2] #funkce trajektorie h
g = z->solg(z)[1] #funkce trajektorie g
dg = z->solg(z)[2] #funkce trajektorie g
```

Out[23]: #25 (generic function with 1 method)

```
In [24]: #Plot paraxialnich trajektorii a jejich derivaci
#(je treba je naskalovat aby se vlezly do jednoho grafu ...)
z1 = (zo:1e-4:0.01)
plot(z1,z1*0,label=false,color="black")
plot(z1,g.(z1),label="g(z)")
plot(z1,h.(z1)*100,label="100 h(z)")
plot(z1,dg.(z1)/200,label="g'(z)/200")
plot(z1,dh.(z1),label="h'(z)")
```

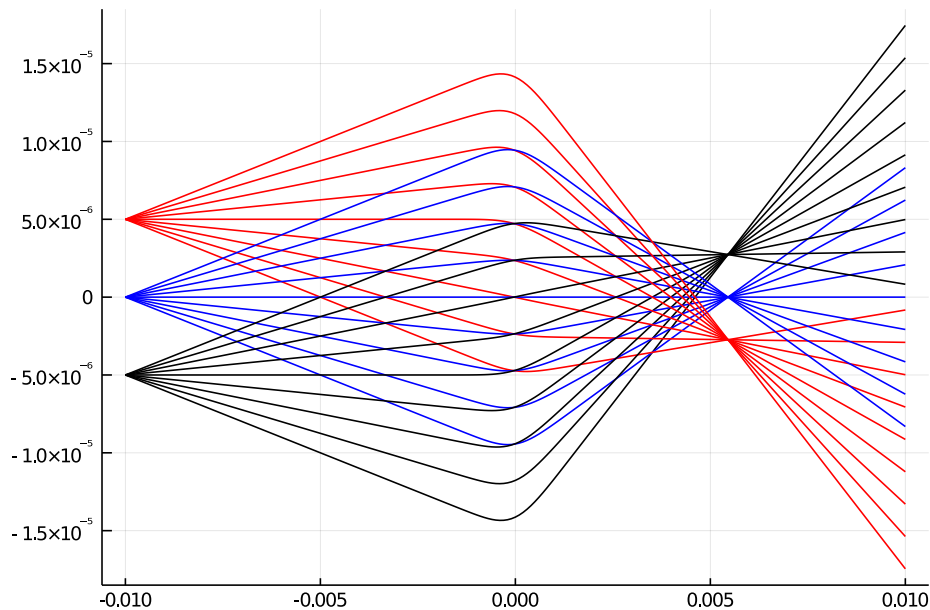
Out[24]:



Vykreslení několika trajektorií

```
In [25]: z1 = (zo:1e-4:0.01)
dxo = (-1:0.25:1)*1e-3
xo = [-1, 0, 1]*5e-6
tcol = ["black", "blue", "red"]
p=plot()
for i=1:length(dxo)
    for j=1:length(xo)
        plot!(p,z1,xo[j]*g.(z1)+dxo[i]*h.(z1),color=tcol[j],legend=false)
    end
end
plot(p)
```

Out[25]:





Pozice obrazu je daná podmínkou  $h(z_i) = 0$ , proč?

```
In [26]: zi = find_zero(h,[0,0.1],Bisection())
```

```
Out[26]: 0.005452623179240787
```

(Příčné zvětšení) je pak  $M = g(z_i)$ , a úhlové zvětšení  $M_a = dh(z_i)$

```
In [27]: M = g(zi); Ma = dh(zi)
print("M = ", M, "\nMa = ", Ma, "\n")
```

```
M = -0.5470826808671513
Ma = -1.8278772665045115
```

Použití trajektorií  $g$  a  $h$  je analog přechodových matic, jak je znáte za základního kurzu optiky. Přechodová matice udává zobrazení mezi  $z_1$  a  $z$ , pokud je  $z_1 = z_o$  je přechodová matice rovna

$$\begin{pmatrix} x(z) \\ x'(z) \end{pmatrix} = \hat{M}(z_o, z) \begin{pmatrix} x_o \\ x'_o \end{pmatrix}$$
$$\hat{M}(z_o, z) = \begin{pmatrix} g(z) & h(z) \\ g'(z) & h'(z) \end{pmatrix}$$

v případě zobrazení mezi předmětem a obrazem pak je přechodová matice

$$\hat{M}(z_o, z_i) = \begin{pmatrix} M & 0 \\ g'(z_i) & M_a \end{pmatrix}$$

Můžeme také nalézt základní charakteristiky čočky, jako jsou polohy ohnisek a hlavní roviny. Pro tyto účely je vhodné zavést tzv.

**princiální trajektorie:**  $u_\pi$ , která jde z mínus nekonečna s nulovou směrnici s osou systému a  $u_{\bar{\pi}}$ , která jde z nekonečna s nulovou směrnici s osou  $z$ , tj.

$$u_\pi(-\infty) = 1, \quad u'_\pi(-\infty) = 0$$
$$u_{\bar{\pi}}(\infty) = 1, \quad u'_{\bar{\pi}}(\infty) = 0$$

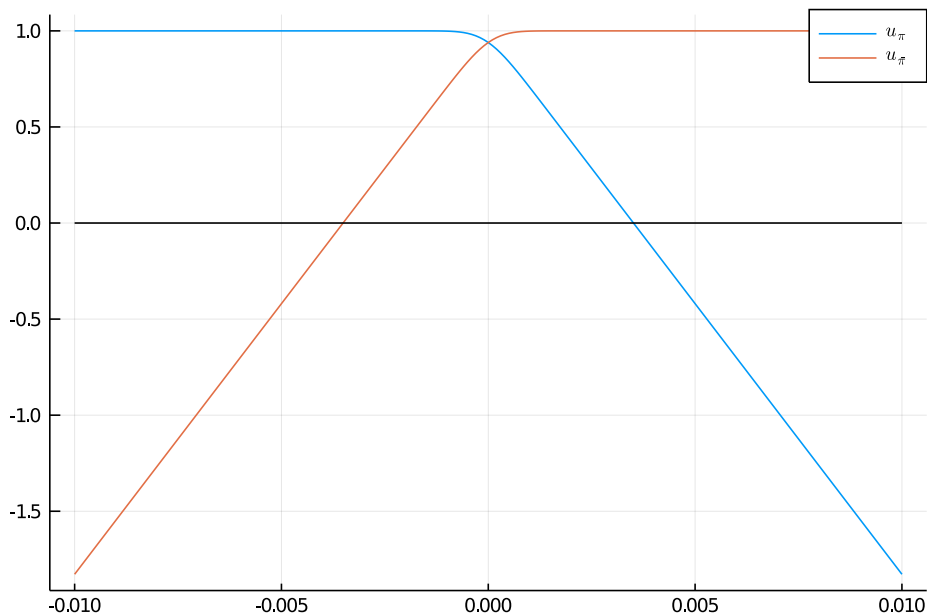
(Jelikož v našem případě je pole v předmětu zanedbatelné, je dána průsečíkem  $g(z)$  s osou.) Obrazové ohnisko je dané průsečíkem  $u_\pi$  s osou a předmětové ohnisko průsečíkem  $u_{\bar{\pi}}$  s osou. Ohniskové vzdálenosti jsou pak dané směrnicí těchto paprsků:  $\frac{1}{f_i} = -u'_\pi(\infty)$ ,

$\frac{1}{f_o} = u'_\pi(-\infty)$  (vzdálenost od ohniska k rovině, kde se asymptoticky potkají prodloužené paprsky z ohniska s prodlouženým paprskem z nekonečna - pozice hlavní roviny)

```
In [28]: zFi = find_zero(g,[0,0.01],Bisection())
print("zFi = ", zFi,"m \n")
fi = -1/dg(0.1)
print("fi = ", fi,"m \n")
u0=[1,0]; zspan2 = (1e-2,-1e-2)
prob2 = ODEProblem(f2,u0,zspan2)
solbpi = solve(prob2,Tsit5(),reltol=1e-8, abstol=1e-10);
ubpi = z -> solbpi(z)[1]
dubpi = z -> solbpi(z)[2]
zFo = find_zero(ubpi,[-1e-2,0],Bisection())
fo = 1/dubpi(-0.1)
print("zFo = ", zFo,"m\n")
print("fo = ", fo,"m\n")
gr()
plot(z1,g.(z1),label=L"u_{\pi}")
plot!(z1,ubpi.(z1),label=L"u_{\bar{\pi}}")
plot!(z1,z1*0,color="black",label=false)
```

```
zFi = 0.0035102421503962556m
fi = 0.0035504341423596973m
zFo = -0.003510242150391164m
fo = 0.003550434142350055m
```

Out [28]:



V optice se často používají i jiné báze v prostoru řešení paraxiální rovnice. Význačná je totiž pozice finální apertury  $z_a$  - někdy je vhodné definovat svazek pomocí jeho pozice v předmětu a apertuře. Zavedou se trajektorie  $s : s(z_o) = 1, s(z_a) = 0$ ,  $t : t(z_o) = 0, t(z_a) = 1$  a libovolný paprsek je pak dostaneme pomocí jeho pozice v předmětu a apertuře ve tvaru:

$$\begin{aligned} u(z) &= s(z)u_o + t(z)u_a \\ u'(z) &= s'(z)u_o + t'(z)u_a \\ u'(z_o) &= s'(z_o)u_o + t'(z_o)u_a \end{aligned}$$

Trajektorie  $s$  a  $t$  můžeme lehce spočítat z trajektorií  $g$  a  $h$ . Víme totiž, že  $s$  i  $t$  jsou lineární kombinace  $g$  a  $h$ :

$$\begin{aligned} s(z) &= ag(z) + bh(z) \\ t(z) &= cg(z) + dh(z) \end{aligned}$$

Pro trajektorii  $s$  pak dostaneme:

$$1 = s(z_o) = ag(z_o) + bh(z_o) = a, 0 = s(z_a) = g(z_a) + bh(z_a) \Rightarrow b = -\frac{g(z_a)}{h(z_a)}$$

a v případě trajektorie  $t$ :

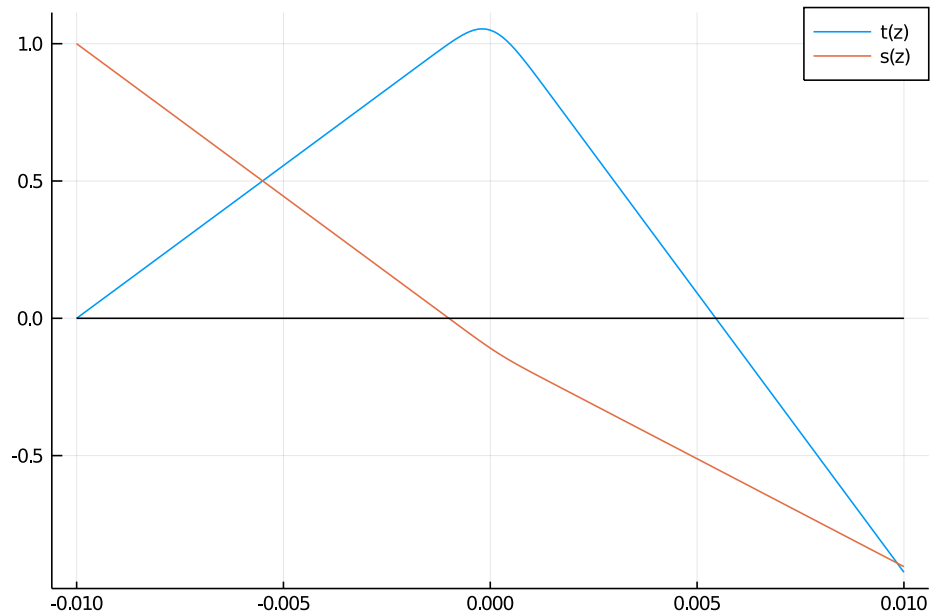
$$0 = t(z_o) = cg(z_o) + dh(z_o) = c, 1 = t(z_a) = dh(z_a) \Rightarrow d = \frac{1}{h(z_a)}$$

tedy:

$$\begin{aligned} s(z) &= g(z) - \frac{g(z_a)}{h(z_a)}h(z) \\ t(z) &= \frac{h(z)}{h(z_a)} \end{aligned}$$

```
In [29]: za = -1e-3
s(z) = g(z) - g(za)/h(za)*h(z)
t(z) = h(z)/h(za)
plot(z1,t.(z1),label="t(z)")
plot!(z1,s.(z1),label="s(z)")
plot!(z1,z1*0.0,color="black",label=false)
```

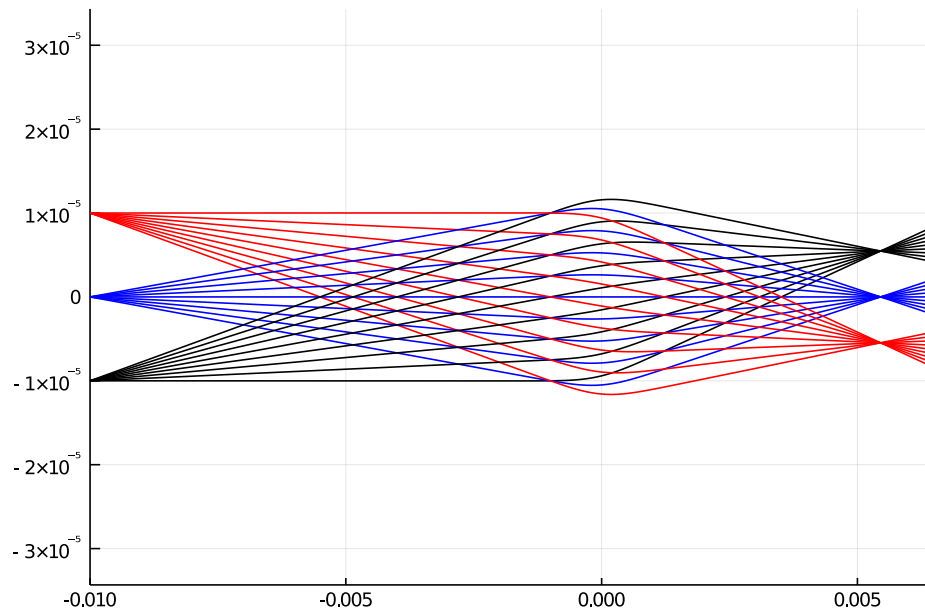
Out[29]:



Ted vykreslime nekolik svazku pomoci této parametrizace ...

```
In [30]: z1 = (zo:1e-4:0.015)
xa = (-1:0.25:1)*1e-5
xo = [-1, 0, 1]*10e-6
tcol = ["black", "blue", "red"]
p=plot()
for i=1:length(xa)
    for j=1:length(xo)
        plot!(p,z1,xo[j]*s.(z1)+xa[i]*t.(z1),color=tcol[j],legend=false)
    end
end
xlims!((zo,zi+1e-3))
plot(p)
```

Out[30]:



Pozice v aperture není zrovna ideální parametr, v elektronové optice je lepší používat úhly. Proto se často místo trajektorie  $t$  používá přímo trajektorie  $h$ , tj.

$$u(z) = s(z)u_o + h(z)\alpha_o$$

Jaký je význam úhlu  $\alpha_o$ ?

$$u'(z) = s'(z)u_o + h'(z)\alpha_o$$

$$u'(z_o) = s'(z_o)u_o + \alpha_o$$

$s(z)u_o$  je paprsek, který má v předmětu polohu  $u_o$  a protíná osu v rovině apertury.  $s'(z_o)u_o$  je pak jeho směrnice v předmětu.  $\alpha_o$  tedy udává úhlovou odchylku v předmětu dané trajektorie od trajektorie parrsku  $s(z)u_o$ .

Také se často používají trajektorie spojené s rovinou obrazu:  $u_\alpha : u_\alpha(z_i) = 0, u'_\alpha(z_i) = 1$  a  $u_\gamma : u_\gamma(z_i) = 1, u'_\gamma(z_i) = 0$ , pak:

$$u(z) = \gamma u_\gamma(z) + \alpha u_\alpha(z)$$

jaký je význam  $\gamma$  a  $\alpha$ ? Jak získáme  $u_\gamma(z)$  a  $u_\alpha(z)$  z trajektorií  $s(z)$  a  $h(z)$ ?

## Obecné vlastnosti paraxiálního aproximace

### Wroskian

I bez numerických výpočtů můžeme říct některé základní vlastnosti paraxiální aproximace. Vyjdeme z obecné rovnice pro osově symetrický systém v rotačních souřadnicích.

$$u'' + \frac{\gamma\Phi'}{2\Phi^*}u' + \frac{\gamma\Phi'' + \eta^2 B^2}{4\Phi^*}u = 0$$

Tu lze přepsat do tvaru

$$\Phi^{*\frac{1}{2}} \frac{d}{dz} (\Phi^{*\frac{1}{2}} u') + \left( \frac{\gamma_0}{4} \Phi'' + \frac{e}{8m_e} B^2 \right) u = 0$$

Mějme dvě nezávislá řešení  $u_1$  a  $u_2$ , která tvoří bazi vektorového prostoru všech řešení. Můžeme pro ně psát

$$\Phi^{*\frac{1}{2}} \frac{d}{dz} (\Phi^{*\frac{1}{2}} u'_1) + \left( \frac{\gamma_0}{4} \Phi'' + \frac{e}{8m_e} B^2 \right) u_1 = 0$$

$$\Phi^{*\frac{1}{2}} \frac{d}{dz} (\Phi^{*\frac{1}{2}} u'_2) + \left( \frac{\gamma_0}{4} \Phi'' + \frac{e}{8m_e} B^2 \right) u_2 = 0$$

Pokud první rovnici vynásobíme  $u_2$ , druhou rovnici vynásobíme  $u_1$  a sečteme, po krátké úpravě dostaneme rovnici

$$\Phi^{*\frac{1}{2}} \frac{d}{dz} (\Phi^{*\frac{1}{2}} (u_1 u'_2 - u_2 u'_1)) = 0$$

kteřá po integraci vede k zákonu zachování Wronskiánu

$$W = \Phi^{*\frac{1}{2}} (u_1 u'_2 - u_2 u'_1) = \text{const}$$

### Lagrange-Helmholtz Relations

V případě báze  $g(z)$  a  $h(z)$  dostaneme:

$$W = \Phi^{*\frac{1}{2}} (gh' - hg') = \Phi^{*\frac{1}{2}}(z_o)(g(z_o)h'(z_o) - h(z_o)g'(z_o)) = \Phi^{*\frac{1}{2}}(z_i)(g(z_i)h'(z_i) - h(z_i)g'(z_i))$$

Což nám určuje vztah mezi úhlovým a příčným zvětšením:

$$\Phi^{*\frac{1}{2}}(z_o) = \Phi^{*\frac{1}{2}}(z_i) M M_a$$

Užitím zákona zachování Wronskiánu na principiální paprsky a roviny  $z = -\infty, z = \infty$  dostaneme:

$$\Phi^{*\frac{1}{2}}_{-\infty} (u_\pi(-\infty)u'_\pi(-\infty) - u_\pi(-\infty)u'_\pi(-\infty)) = \Phi^{*\frac{1}{2}}_{\infty} (u_\pi(\infty)u'_\pi(\infty) - u_\pi(\infty)u'_\pi(\infty))$$

což se využitím definice principiálních trajektorií redukuje na:

$$\Phi^{*\frac{1}{2}}_{-\infty} u'_\pi(-\infty) = -\Phi^{*\frac{1}{2}}_{\infty} u'_\pi(\infty)$$

Pak dostaneme vztah mezi předmětovou a obrazovou ohniskovou vzdáleností

$$\frac{\bar{f}}{f} = \sqrt{\frac{\Phi^*_{-\infty}}{\Phi^*_\infty}}$$

V případě námi vypočtené magnetické čočky:

```
In [31]: print("M*Ma = ", M*Ma, "\n")
print("fi/f_o = ", fi/fo, "\n")
```

```
M*Ma = 0.9999999952554085
fi/f_o = 1.0000000000027158
```

## Longitudiální zvětšení

Při změně předmětové roviny  $z_o \rightarrow z_o + dz_o$  se také posune rovina obrazu  $z_i \rightarrow z_i + dz_i$ , pokud je tato změna dostatečně malá, je změna obrazové roviny přímo úměrná změně roviny předmětu, kde konstantu úměrnosti nazýváme longitudiálním zvětšením. Pokud tedy posuneme rovinu předmětu změní se i trajektorie  $h \rightarrow \tilde{h}$ . Jelikož se také jedná o řešení paraxiální rovnice trajektorie lze ji psát jako lineární kombinaci původních charakteristických trajektorií

$$\tilde{h}(z) = ah(z) + bg(z)$$

víme, že trajektorie  $\tilde{h}(z)$  v  $z_o + dz_o$  splňuje:

$$0 = \tilde{h}(z_o + dz_o) = ah(z_o + dz_o) + bg(z_o + dz_o) = adz_o + b$$

$$1 = \tilde{h}'(z_o + dz_o) = ah'(z_o + dz_o) + bg'(z_o + dz_o) = a(h'(z_o) + h''(z_o)dz_o) + b(g'(z_o) + g''(z_o)dz_o) = a(1 + h''(z_o)dz_o)$$

Pokud uvažujeme, že  $z_o$  je mimo pole, tak druhé derivace jsou nulové a z druhé rovnice dostaneme  $a = 1$ . Následně pak z první

$b = -dz_o$ . Trajektorie pak má tvar

$$\tilde{h}(z) = h(z) - dz_o g(z)$$

Pokud vyjádříme tuto trajektorii v novém fokusu

$$0 = \tilde{h}(z_i + dz_i) = h(z_i + dz_i) + dz_o g(z_i + dz_i)$$

po rozvoji do mocnin v  $dz_i$  a zanedbání kvadratických členů v  $dz_i$  a  $dz_o$  dostaneme

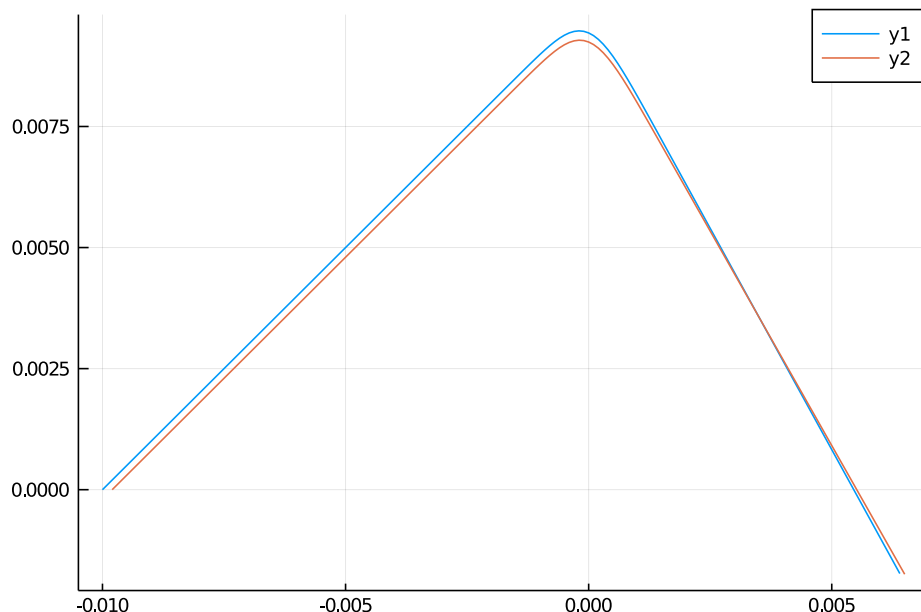
$$dz_i = dz_o M^2 \sqrt{\frac{\Phi_i^*}{\Phi_o^*}}$$

V případě námi počítané magnetické čočky:

```
In [32]: dzo = 2e-4;
zo2 = zo+dzo
zspan = (zo2,1e-2);
u0 = [0,1]#Pocatecni podminky pro trajektorii h
#Vypocet trajektorie h
prob2 = ODEProblem(f2,u0,zspan)
solh2 = solve(prob2,Tsit5(),reftol=1e-8, abstol=1e-10);
h2 = z->solh2(z)[1] #funkce trajektorie h
zi2 = find_zero(h2,[0,0.01],Bisection())
print("zo = ", zo, " m, zo2 = ",zo2, " m, dzo = ", zo2-zo," m\n")
print("zi = ", z1, " m, zi2 = ",zi2, " m, dzi = ", zi2-zi," m\n")
print("dzi_c = ", M^2*dzo)
z1 = zo:1e-4:zi+1e-3; z2 = zo2:1e-4:zi2+1e-3;
p=plot()
plot!(p,z1,h.(z1))
plot!(p,z2,h2.(z2))
plot(p)
```

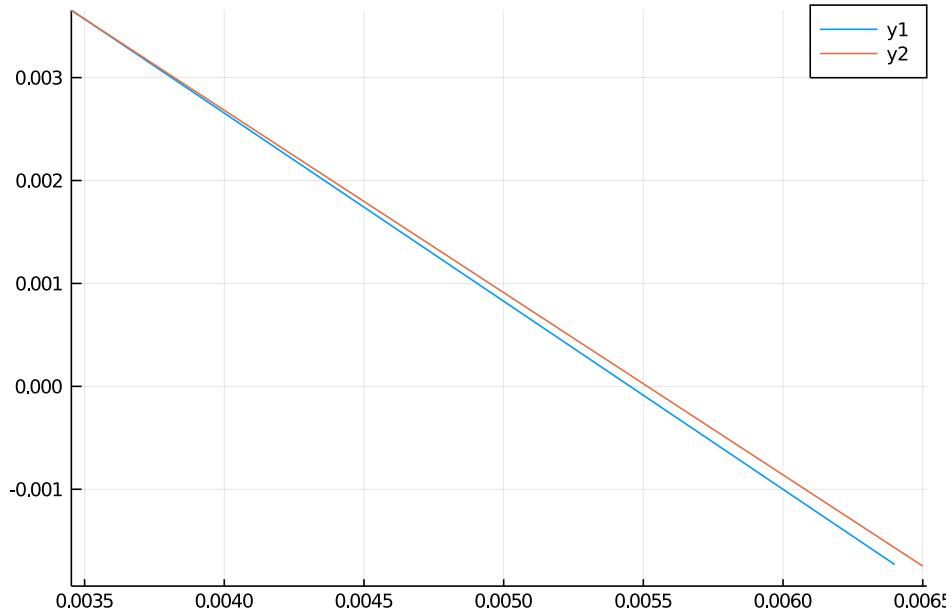
```
zo = -0.01 m, zo2 = -0.0098 m, dzo = 0.000200000000000000052 m
zi = -0.01:0.0001:0.015 m, zi2 = 0.0055143864785419236 m, dzi = 6.176329930113623e-5 m
dzi_c = 5.985989194095786e-5
```

Out[32]:



```
In [34]: plot(p)
          xlims!(zi-2e-3, zi2+1e-3)
          ylims!(h(zi2+1e-3), h(zi-2e-3))
```

Out[34]:



## Aproximace tenkou čočkou

V tomto případě zhomogenizovanou rovnici ještě dále upravíme pomocí Pichtovy transformace

$$u(z) = \Phi^{*-1/4} v(z)$$

na tvar

$$v'' + G(z)v = 0$$

kde koeficient

$$G(z) = \frac{3}{16} \frac{\Phi'^2}{\Phi^{*2}} \left( 1 + \frac{4}{3} \epsilon \Phi^* \right) + \frac{eB^2}{8m_e \Phi^*} + \frac{\Phi_1 \bar{\Phi}_1}{8\Phi}$$

je vždy kladný. Předpokládejme, že čočka je tenká, můžeme v ní tedy zanedbat změnu souřadnice  $v$ , změní se pouze její směrnice

$$v' = - \int_{-\infty}^{\infty} G(z)v(z)dz \approx -v_0 \int_{-\infty}^{\infty} G(z)dz$$

pokud tento vztah aplikujeme na principiální paprsky  $v_\pi, v_{\bar{\pi}}$ , použijeme vztah  $u = v\Phi^{*-1/4}$  a uvážíme že v nekonečnách je osový potenciál konstantní, tj  $u'_\pi(\infty) = \Phi_\infty^{*-1/4} v'_\pi(\infty)$ ,  $u'_\pi(-\infty) = \Phi_{-\infty}^{*-1/4} v'_\pi(-\infty)$  můžeme psát

$$v'_\pi(\infty) = u'_\pi(\infty)\Phi_\infty^{*1/4} = -v_\pi(-\infty) \int_{-\infty}^{\infty} G(z)dz = -\Phi_{-\infty}^{*1/4} u_\pi(-\infty) \int_{-\infty}^{\infty} G(z)dz$$

$$v'_\pi(-\infty) = u'_\pi(-\infty)\Phi_{-\infty}^{*1/4} = v_\pi(\infty) \int_{-\infty}^{\infty} G(z)dz = \Phi_\infty^{*1/4} u_\pi(\infty) \int_{-\infty}^{\infty} G(z)dz$$

Pro ohniskové dálky pak můžeme psát:

$$\frac{1}{f} = \frac{\Phi_{-\infty}^{*1/4}}{\Phi_\infty^{*1/4}} \int_{-\infty}^{\infty} G(z)dz, \quad \frac{1}{\bar{f}} = \frac{\Phi_\infty^{*1/4}}{\Phi_{-\infty}^{*1/4}} \int_{-\infty}^{\infty} G(z)dz,$$

tedy:

$$\frac{f}{\bar{f}} = \frac{\Phi_\infty^{*1/2}}{\Phi_{-\infty}^{*1/2}}$$

Pro naši magnetickou čočku vychází:

In [35]:

```
G(z)=qe*Bax(z)^2/(8*me*Phr)
prob3=ODEProblem((u,p,z)->G(z),0.0,(-0.01,0.01))
iG = solve(prob3,Tsit5(),reltol=1e-8, abstol=1e-10);
fo2 = 1/iG(0.01)
print("Ohniskova vzdalenost z rovnice trajektorie: ", fo,"\n")
print("Ohniskova vzdalenost z aproximace tenkou cockou: ", fo2,"\n")
```

Ohniskova vzdalenost z rovnice trajektorie: 0.003550434142350055

Ohniskova vzdalenost z aproximace tenkou cockou: 0.0032514106373007713