# Abstrakt

¡+blablabla+¿ Modelom budú (sú) demonštrované základné javy vedenia prúdu cez PN prechod ako sú: driftový prúd, difúzny prúd, vytvorenie vyprázdnenej oblasti pri závernom pólovaní, injekcia minorítných nosičov do opačne dotovaných oblastí pri priepustnom pólovaní, volt-ampérová charateristika a jej zodpovedajúce priestorové rozloženie eletrónov a dier.

# 1 Úvod

# 2 Rovnice polovodičov

| Význam použitých symbolov: | |
|---|---|
| $\varepsilon$ | permitiivta prostredia (kremíku) |
| $\psi$ | elektrický potenciál |
| $\mathbf{E}$ | intenzita elektrického poľa |
| $q$ | náboj elektrónu |
| $p, n$ | koncentrácie dier a elektrónov |
| $N_D, N_A$ | koncentrácie donorov a akceptorov[1] |
| $\mathbf{J_n}, \mathbf{J_p}$ | prúdy elektrónov a dier jednotkou plochy |
| $\mu_n, \mu_p$ | pohyblivosti nosičov |
| $D_n, D_p$ | difúzne konštanty (Fickov zákon) |
| $U_n, U_p$ | miera generácie a rekombinácie elektrónov a dier |

Poissonova rovnica[2]:

$$\nabla \cdot (\varepsilon \nabla \psi) = -q(p - n + N_D - N_A) \tag{1}$$

Prúdy:

$$\mathbf{J_n} = \overbrace{qn\mu_n\mathbf{E}}^{\text{drift}} + \overbrace{qD_n\nabla n}^{\text{difúzia}} \tag{2}$$

$$\mathbf{J_p} = qp\mu_p\mathbf{E} - qD_p\nabla p \tag{3}$$

Rovnice kontinuity (zachovanie resp. časová spojitosť náboja):

$$\frac{\partial n}{\partial t} = \nabla \cdot \mathbf{J_n} + U_n \tag{4}$$

[1] Pre jednoduchosť stačí uvažovať prípad, kde všetky prímesové atómy sú zionizované.

[2] Jedna z Maxwellovych rovníc v diferenciálnom tvare $\nabla \cdot \mathbf{D} = \rho$, kde $\mathbf{D} = \varepsilon E \implies \nabla \cdot (\varepsilon \nabla \psi) = -\rho$

$$\frac{\partial p}{\partial t} = -\nabla \cdot \mathbf{J_p} - U_p \tag{5}$$

Je dobré pripomenúť vzťah $\mathbf{E} = -\nabla\psi$. Prvý sčítanec na pravej strane rovníc (2) a (3) predstavuje driftový prúd (daný pohyblivosťou nábojov a intenzitou elektrického poľa), druhý sčítanec predstavuje difúzny prúd podľa prvého Fickovho zákona, teda úmerný spádu koncentrácie nábojov a difúznej konštante.

Rovnice (1) až (5) tvoria sústavu vzájomne previazaných nelineárnych rovníc s 5 neznámzmi $(\psi, \mathbf{J_n}, \mathbf{J_p}, n, p)$. Dosadením (2) do (4) a (3) do (5) je možné riešiť 3 rovnice s neznýmymi $(\psi, n, p)$. Tento postup bude využitý v stati 4.1.

# 3 MKP, FEniCS a stratégia riešenia

K numerickému riešeniu rovníc polovodičov použijeme diskretizáciu metódou konečných prvkov (MKP) s využitím programových nástrojov akademicko - vedeckej výpočtovej platformy FEniCS [1].

¡+Variacna forma+¿ ¡+Mixed formy vs decoupling+¿ ¡+Casova diskretizacia+¿

# 4 Implementácia

¡++¿

## 4.1 Matematické úpravy

¡++¿ Dosadením (2) do (4) a (3) do (5) a vyžitím vzťahu $\mathbf{E} = -\nabla\psi$ dostávame nasledovnú sústavu rovníc s 3 neznýmymi $(\psi, n, p)$ - tj. Poissonovu rovnicu a rovnice kontinuity elektónov a dier:

$$\nabla \cdot (\varepsilon \nabla \psi) = -q(p - n + N_D - N_A) \tag{6}$$

$$\frac{\partial n}{\partial t} = \nabla \cdot (-n\mu_n \nabla\psi + D_n \nabla n) + U_n \tag{7}$$

$$\frac{\partial n}{\partial t} = \nabla \cdot (-p\mu_p \nabla\psi - D_p \nabla p) + U_p \tag{8}$$

¡+stále sú vzájomne previazané+¿ To je dosť závažná nepríjemnosť.

¡+Poissonova rovnica - varacna forma atd[3]+¿

[3] $dx$ predstavuje vektor takého rozmeru, ako je rozmer priestorových súradníc riešenej úlohy

Časová diskretizácia rovnice (7):

$$\frac{n_i - n_{i-1}}{dt} = \nabla(-\mu_n n_{i-1} \nabla \psi_i) + \nabla \cdot (D_n \nabla n_i) + U_n \tag{9}$$

$$n_i - \nabla \cdot (dt D_n \nabla n_i) = n_{i-1} - \nabla(dt \mu_n n_{i-1} \nabla \psi_i) + dt U_n \tag{10}$$

Z toho bilineárna a lineárna forma pre MKP $a(n, v) = L(v)$ (neznámu $n_i$ budeme ďalej označovať jednoducho $n$) získaná vynásobením oboch strán (10) testovacou funkciou $v$ a následným integrovaním podľa $\mathrm{d}x$:

$$\begin{aligned} &\int_\Omega nv \,\mathrm{d}x + \int_\Omega dt D_n \nabla n \nabla v \,\mathrm{d}x = \\ &= \int_\Omega n_{i-1} v \,\mathrm{d}x - \int_\Omega dt \mu_n n_{i-1} \nabla \psi_i \nabla v \,\mathrm{d}x + \int_\Omega dt U_n v \,\mathrm{d}x \end{aligned} \tag{11}$$

(8)

## 4.2 Okrajové podmienky

¡++¿

## 4.3 Scaling

Pokiaľ majú výsledky - predovšetkým elektrických veličín ako potenciál či prúdové hustoty - kvantitatívne zodpovedať realistickým hodnotám, je nutné aplikovať realistické hodnoty materiálových i rozmerových konštánt. Tým však vzniknú obrovské rozdiely v rádoch jednotlivých veličín, čo má za následok jednak zbytočne vysokú výpočtovú cenu (resp. čas výpočtov), jednak možnú stratu numerickej riešiteľnosti. Preto sa rovnice násobia ešte pred výpočtom škálovacími konštantami [?] [?] a následne sa spätne „zrozmerňujú“ až výsledky.

Účelom tohto článku je odvodenie a zostavenie výpočtového modelu s následným kvalitatívnym overením jeho funkčnosti. Pre jednoduchosť bude preto tento model bezrozmerný, tj. konštanty budú jednotkové.

## 4.4 Výpočtové prevednie (Python)

```
# Mesh and function space
nx = 100
ny = 1
mesh = RectangleMesh(Point(0.0, 0.0), Point(x_length, y_length)
    , nx, ny, 'crossed')
V = FunctionSpace(mesh, "CG", 1)
Vn=V
Vp=V
```

```
####################
## Poisson
####################
rho = Expression("q * (p - n + Nd - Na)", q=q, p=p0, n=n0, Nd=Nd
    , Na=Na, degree=1)
Psi = TrialFunction(V)
v_psi = TestFunction(V)
a = -inner(grad(Psi), grad(v_psi)) * dx
L = -rho/eps_Si*v_psi*dx

Psi_result = Function(V)
#solve(a==L, Psi_result, [bc_psi1, bc_psi2])
```

```
####################
## n, p Continuity
####################
n_i1 = Function(Vn)
n_i1.assign(n0)
n = TrialFunction(Vn)
vn = TestFunction(Vn)
#an = n*vn*dx + dt*Dn*inner(grad(n), grad(vn))*dx
an = n*vn*dx + dt*D_n*inner(grad(n), grad(vn))*dx
Ln = n_i1*vn*dx + dt*mob_n*n_i1*inner(grad(Psi_result), grad(vn)
    )*dx

n_result = Function(Vn)
#solve(an==Ln, n_result, [bc_n1, bc_n2])

p_i1 = Function(Vp)
p_i1.assign(p0)
p = TrialFunction(Vp)
vp = TestFunction(Vp)
ap = p*vp*dx + dt*D_p*inner(grad(p), grad(vp))*dx
Lp = p_i1*vp*dx - dt*mob_p*p_i1*inner(grad(Psi_result), grad(vp)
    )*dx

p_result = Function(Vp)
#solve(ap==Lp, p_result, [bc_p1, bc_p2])
```

```
Psi_result = Function(V)
n_result = Function(Vn)
p_result = Function(Vp)
#solve(a==L, Psi_result, [bc_psi1, bc_psi2])
#solve(an==Ln, n_result, [bc_n1, bc_n2])
#solve(ap==Lp, p_result, [bc_p1, bc_p2])
```

# 5 Príklad riešenia - PN prechod; rovnovážny stav, napäťové okrajové podmienky

¡++¿

## 5.1 Interpretácia výsledkov

¡++¿

### 5.1.1 Koncentrácie voľných nosičov

¡++¿

### 5.1.2 Driftový a difúzny prúd

¡++¿

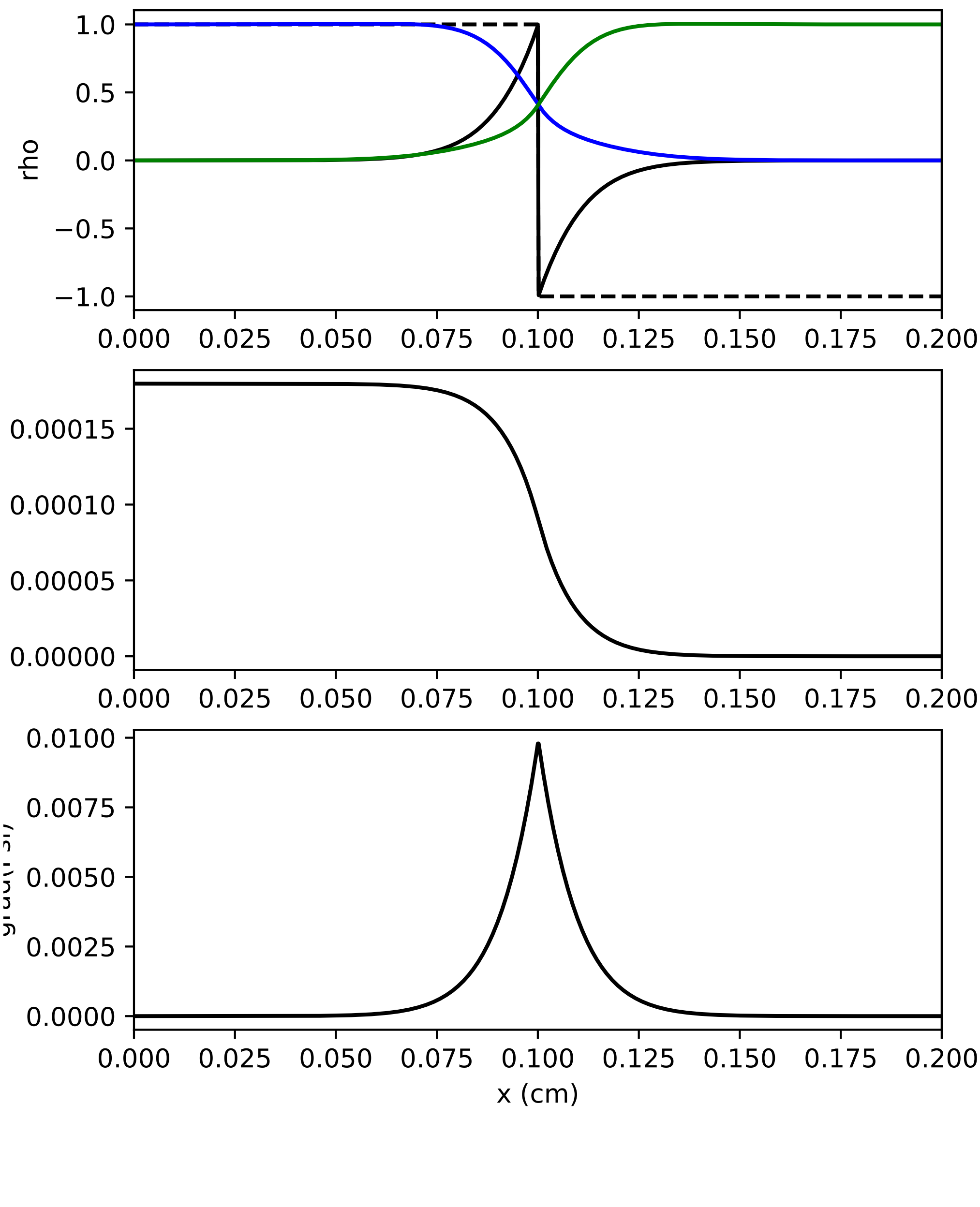
# Literatúra

[1] FEniCS Project. `https://fenicsproject.org`.

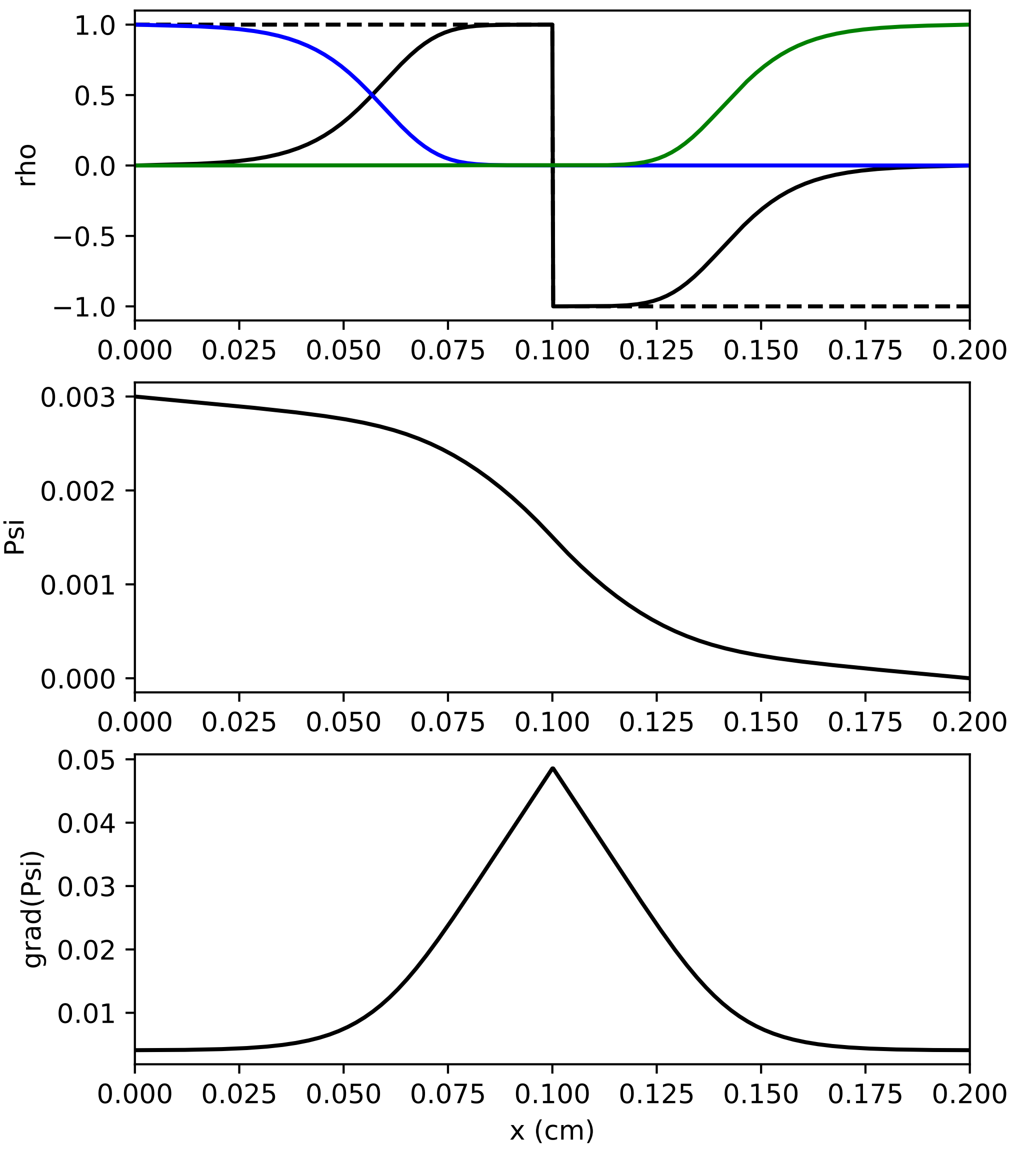
[2] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

# A Zdrojový kód (Python)

¡++¿

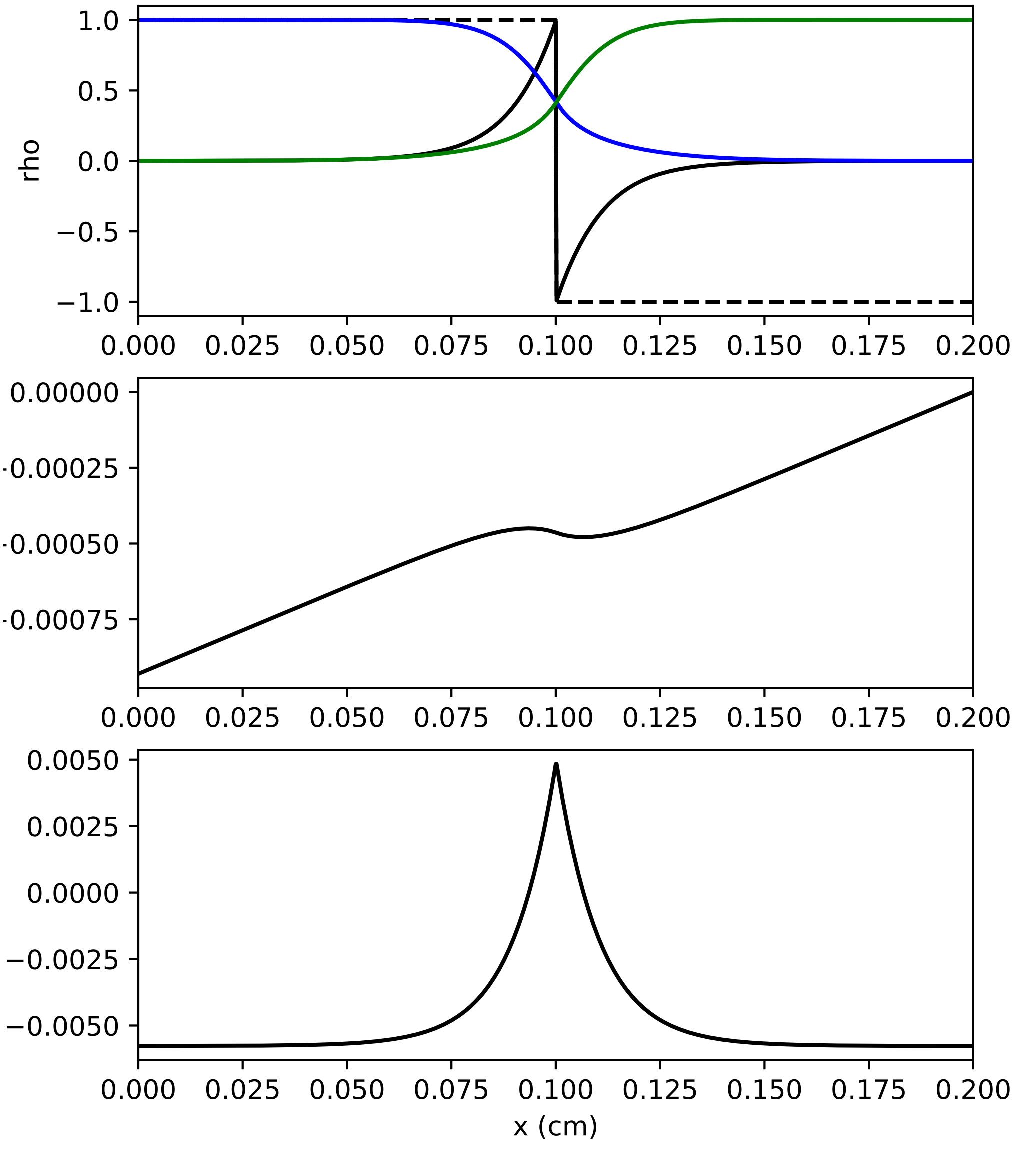

rho
1.0
0.5
0.0
−0.5
−1.0
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0.00015
0.00010
0.00005
0.00000
0.0100
0.0075
0.0050
0.0025
0.0000
x (cm)

Obr. 1:

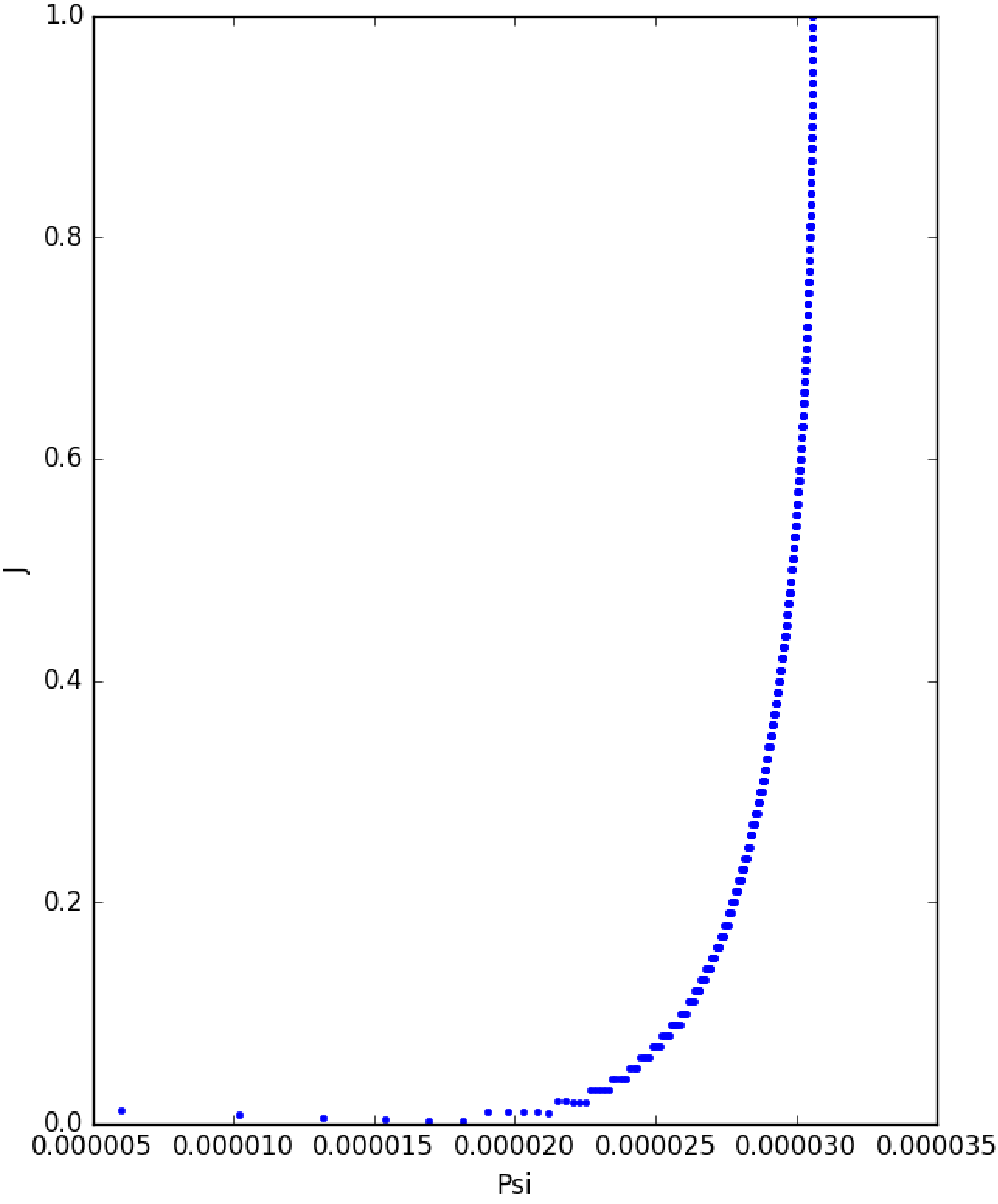


rho
1.0
0.5
0.0
−0.5
−1.0
Psi
0.003
0.002
0.001
0.000
grad(Psi)
0.05
0.04
0.03
0.02
0.01
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
x (cm)


Obr. 2:


1.0
0.5
0.0
−0.5
−1.0
rho
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175 0.200
0.00000
0.00025
0.00050
0.00075
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175 0.200
0.0050
0.0025
0.0000
−0.0025
−0.0050
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175 0.200
x (cm)

Obr. 3:


1.0
0.8
0.6
0.4
0.2
0.0
J
0.000005
0.000010
0.000015
0.000020
0.000025
0.000030
0.000035
Psi

Obr. 4: