

Univerza v Ljubljani  
Fakulteta za elektrotehniko

**Nelma-cap**  
numerično računanje kapacitivnosti v tiskanem  
vezju

Tomaž Šolc

Seminarska naloga pri predmetu elektromagnetika

Ljubljana, september 2006

# Kazalo

<b>1</b>	<b>Uvod</b>	<b>2</b>
1.1	Motivacija . . . . .	2
1.2	Izbira metode računanja kapacitivnosti . . . . .	2
<b>2</b>	<b>Metode</b>	<b>4</b>
2.1	Električno polje in kapacitivnosti . . . . .	4
2.2	Metoda končnih diferenc . . . . .	5
2.3	SOR iteracijska metoda . . . . .	8
<b>3</b>	<b>Implementacija</b>	<b>9</b>
3.1	Uporaba programa . . . . .	9
3.2	Opis geometrije vezja . . . . .	12
<b>4</b>	<b>Rezultati</b>	<b>17</b>
4.1	Primerjava z analitičnimi izračuni . . . . .	17
4.2	Preizkus simetrije . . . . .	21
<b>5</b>	<b>Zaključek</b>	<b>21</b>
<b>6</b>	<b>Literatura</b>	<b>25</b>

# 1 Uvod

## 1.1 Motivacija

Pri načrtovanju elektronskih vezij za signale nizkih in srednjih frekvenc običajno privzamemo, da so mehanske dimenzije vezja majhne v primerjavi z valovnimi dolžinami elektromagnetnega valovanja, ki ga povzročajo signali v vezju. Ta predpostavka nam omogoča, da elektronske komponente obravnavamo kot koncentrirane (točkaste) elemente, vpliv povezav med njimi pa zanemarimo.

Ko se frekvence signalov v vezju večajo pa določenih vplivov povezav ne moremo več zanemariti. Zaradi geometrije običajnih vezij (to je vezij na tiskanini), se kot prvi nezaželen vpliv običajno pokažejo stresane kapacitivnosti med povezavami v vezju. Te povzročajo stik signalnih linij z ozemljitvijo in povečujejo presluh med posameznimi signalnimi linijami. Skupaj z uporabnostmi povezav in elementov povzročajo stresane kapacitivnosti tudi zakasnitve pri prenosu signalov. Pri še višjih frekvencah signalov pridejo do izraza še stresane induktivnosti in tedaj moramo povezave v vezju obravnavati kot elektromagnetne linije s čimer je konec poenostavitev za koncentrirana vezja.

Ker je obravnava vezij z linijami precej bolj zahtevna od obravnave koncentriranih vezij je očitno, da se želimo čim dlje držati omenjenih poenostavitev. Stresane kapacitivnosti lahko relativno enostavno vključimo v model koncentriranih vezij (v približku jih lahko v vezje dodamo kot koncentrirane elemente), zato je to najenostavnejši način, da se poenostavitev držimo tudi še rahlo čez mejo, ko sicer ne bi več veljale. Seveda pa to zahteva, da stresane kapacitivnosti poznamo, saj jih le tako lahko upoštevamo pri načrtovanju vezja.

## 1.2 Izbira metode računanja kapacitivnosti

Tiskano vezje je v osnovi tridimenzionalna struktura, sestavljena iz prevodnikov in dielektrikov. Edina lastnost geometrije, ki nam lahko pomaga pri izračunu, je da ima tiskano vezje plastno strukturo: po eni geometrijski osi ga lahko razdelimo na plasti, v katerih je po tej osi material homogen. Velikokrat pa naletimo tudi na plasti, ki so homogene tudi po ostalih dveh oseh (na primer substrat tiskanega vezja, ki nima povezav med prevodnimi plastmi).

Povezave na tiskanem vezju so v veliki večini primerov nepravilnih oblik, zato kakršnakoli analitična obravnava kapacitivnosti odpade. Tako nam osta-

nejo le še numerične metode. V literaturi je opisanih več numeričnih pristopov:

Za popolnoma splošno računanje kapacitivnosti med prevodnimi telesi, ki jih obdajajo dielektriki, lahko uporabimo ali metode, ki se osredotočajo na računanje električnega polja (metoda končnih diferenc in metoda končnih elementov) ali pa metode, ki se osredotočajo na računanje porazdelitve naboja na prevodnih telesih. Za slednje je značilno, da računski kompleksnost narašča z kompleksnostjo geometrije prevodnih in dielektričnih teles, ki jih obravnavamo. Po drugi strani pa računski kompleksnost prvih metod narašča z velikostjo prostora, ki ga obravnavamo.

Obstajajo pa tudi metode, ki se osredotočajo na prej omenjeno posebnost geometrije tiskanega vezja, in ki omogočajo poenostavljeno računanje. Najbolj zanimiva od teh je DRT (Dimension Reduction Technique) [1], ki omogoča računanje polja za vsako plast vezja posebej. Po drugi strani pa takšne metode večinoma zahtevajo, da je mogoče vezje razdeliti na neodvisne enote tudi po drugih dveh koordinatnih oseh [2].

Ker tiskano vezje za svojo velikost vsebuje precej kompleksno geometrijo, imajo po mojem mnenju prednost metode, ki se osredotočajo na računanje polja. Te pri običajnih tiskanih vezjih omogočajo boljši izkoristek računske moči. Po drugi strani pa te metode še vedno zahtevajo precejšnje količine računalniškega pomnilnika. V tem pogledu ima metoda končnih diferenc bistveno prednost pred metodo končnih elementov. Enakomerno porazdeljeno mrežo točk lahko v pomnilniku shranimo bistveno bolj učinkovito kot neenakomerno porazdeljene mejne točke v metodi končnih elementov.

Napredne metode, kot je DRT, sicer obljubljaajo še boljši izkoristek računske moči, vendar pa po mojih izkušnjah za računanje kapacitivnosti v tiskanih vezjih niso najbolj primerne. Običajnih tiskanih vezij namreč večinoma ni mogoče razdeliti v neodvisne enote, tako kot je to mogoče pri integriranih vezij (za katera je DRT prvenstveno namenjena).

Zaradi teh zgoraj naštetih vzrokov sem se odločil, da bom uporabil metodo končnih diferenc. Kasneje se bo izkazalo, da ima poleg naštetih prednosti še to prednost, da je mogoče računalniški algoritem precej optimizirati, če upoštevamo plastovitost tiskanega vezja. Hkrati pa predstavlja prednost tudi precejšnja preprostost metode, ki ugodno vpliva na število napak pri programiranju.

## 2 Metode

Sledi podrobnejši opis uporabljenih metod za računanje kapacitivnosti.

V nadaljevanju pomenijo deli prevodnika vozlišča vezja, med katerimi računamo kapacitivnosti.

### 2.1 Električno polje in kapacitivnosti

Osnovni princip računanja kapacitivnosti, ki sem ga uporabil je sledeč:

V prostor postavimo vse prevodne in neprevodne dele tiskanega vezja. Na  $i$ -ti prevodni del priključimo napetost  $1V$ , vse ostale dele pa ozemljimo. V prostor postavimo še ozemljeno prevodno škatlo ter se tako omejimo na končni volumen, v katerem bomo računali električno polje.

Velikost ozemljene škatle določimo tako, da zajame ves del prevodnika pod napetostjo in še določeno njegovo okolico. Od velikosti okolice je odvisna natančnost računanja in računska zahtevnost (večja okolica pomeni večji volumen, v katerem moramo izračunati polje). S velikostjo ozemljene škatle v bistvu povemo, kako oddaljena sta še lahko dva ozemljena predmeta, da lahko za kapacitivnost med njima privzamemo, da je enaka nič.

V prostoru znotraj škatle sedaj velja Laplaceova diferencialna enačba za potencial (v prostoru ni prostih nosilcev naboja):

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0 \quad (1)$$

Na podlagi te enačbe in robnih pogojev, ki smo jih določili z škatlo in prevodnimi telesi na znanih potencialih, lahko izračunamo potencial v vseh točkah prostora. Iz vrednosti potenciala lahko nato izračunamo polje po enačbi:

$$\vec{E} = -\nabla V \quad (2)$$

Ko poznamo električno polje v prostoru, lahko po Gaussovem izreku izračunamo naboj na  $j$ -tem prevodniku tako, da integriramo polje po ploskvi  $\mathcal{A}$ , ki objema izbrani prevodnik:

$$\oint_{\mathcal{A}} \varepsilon \vec{E} \cdot d\vec{a} = Q_j \quad (3)$$

Sedaj lahko izračunamo stresane kapacitivnosti med prevodnikom  $i$  in prevodnikom  $j$  po definiciji:

$$C_{i,j} = \frac{Q_j}{U_{i,j}} = \frac{Q_j}{1V} \quad i \neq j \quad (4)$$

Ko ponovimo zgornji postopek za vse  $i$  in  $j$  smo zračunali vse kapacitivnosti v vezju. Pravzaprav lahko brez posebnega večanja računske zahtevnosti, vsako kapacitivnost izračunamo dvakrat ( $C_{i,j} = C_{j,i}$ ), saj je računanje polja daleč najbolj zahteven korak, in dodatno računanje integralov ne vpliva bistveno na čas računanja. Po drugi strani pa s tem dodatnim računanjem dobimo način, kako oceniti napako računanja. Če se  $C_{i,j}$  dosti razlikuje od  $C_{j,i}$ , potem naš izračun ni bil dovolj natančen.

## 2.2 Metoda končnih diferenc

Metoda končnih diferenc nam omogoča, da reševanje zgoraj omenjene Laplaceve diferencialne enačbe z robnimi pogoji prevedemo v reševanje sistema linearnih enačb ([3], strani 80-88).

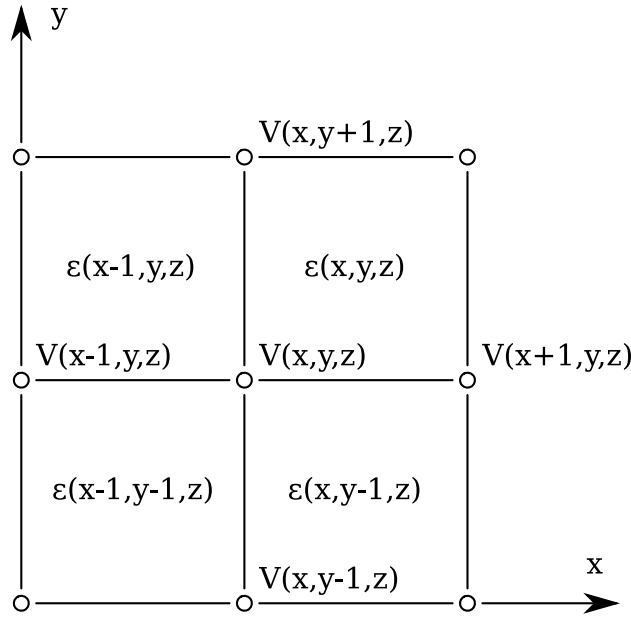
To storimo tako, da v računski prostor postavimo mrežo. Vrednosti potenciala nato računamo samo na vozliščih mreže. Material (prevodnik ali dielektrik) pa naj bo homogen znotraj enega razdelka mreže. V mojem primeru sem računanje poenostavil še tako, da je korak mreže za vsako dimenzijo posebej konstanten (ti koraki so v nadaljevanju označeni z  $h_x$ ,  $h_y$  in  $h_z$ ).

Splošna enačba v sistemu linearnih enačb, v katerega se pretvori Laplaceova enačba, sedaj izgleda takole:

$$\begin{aligned} & -(k_{x1} + k_{x2} + k_{y1} + k_{y2} + k_{z1} + k_{z2})V_{x,y,z} \\ & + k_{x1}V_{x-1,y,z} + k_{x2}V_{x+1,y,z} \\ & + k_{y1}V_{x,y-1,z} + k_{y2}V_{x,y+1,z} \\ & + k_{z1}V_{x,y,z-1} + k_{z2}V_{x,y,z+1} = 0 \end{aligned}$$

Pri čemer velja

$$\begin{aligned}
k_{x1} &= (\varepsilon_{x-1,y-1,z-1} + \varepsilon_{x-1,y-1,z} + \varepsilon_{x-1,y,z-1} + \varepsilon_{x-1,y,z}) \cdot \frac{h_z h_y}{4h_x} \\
k_{x2} &= (\varepsilon_{x,y-1,z-1} + \varepsilon_{x,y-1,z} + \varepsilon_{x,y,z-1} + \varepsilon_{x,y,z}) \cdot \frac{h_z h_y}{4h_x} \\
k_{y1} &= (\varepsilon_{x-1,y-1,z-1} + \varepsilon_{x-1,y-1,z} + \varepsilon_{x,y-1,z-1} + \varepsilon_{x,y-1,z}) \cdot \frac{h_z h_x}{4h_y} \\
k_{y2} &= (\varepsilon_{x-1,y,z-1} + \varepsilon_{x-1,y,z} + \varepsilon_{x,y,z-1} + \varepsilon_{x,y,z}) \cdot \frac{h_z h_x}{4h_y} \\
k_{z1} &= (\varepsilon_{x-1,y-1,z-1} + \varepsilon_{x-1,y,z-1} + \varepsilon_{x,y-1,z-1} + \varepsilon_{x,y,z-1}) \cdot \frac{h_x h_y}{4h_z} \\
k_{z2} &= (\varepsilon_{x-1,y-1,z} + \varepsilon_{x-1,y,z} + \varepsilon_{x,y-1,z} + \varepsilon_{x,y,z}) \cdot \frac{h_x h_y}{4h_z}
\end{aligned}$$



Slika 1: Interpretacija indeksov pri  $V$  in  $\varepsilon$ .

Robni pogoji pa vnesejo v sistem še enačbe v obliki:

$$\begin{aligned}
V_{x',y',z'} &= 0 \\
V_{x'',y'',z''} &= 1
\end{aligned}$$

Zgoraj omenjeno splošno enačbo potrebujemo v resnici le na mejah med dvema plastema. Znotraj plasti je snov po z osi homogena, kar pomeni, da se enačba poenostavi na naslednji način:

$$\begin{aligned}
& -(l_{x1} + l_{x2} + l_{y1} + l_{y2} + 2l_z)V_{x,y,z} \\
& + l_{x1}V_{x-1,y,z} + l_{x2}V_{x+1,y,z} \\
& + l_{y1}V_{x,y-1,z} + l_{y2}V_{x,y+1,z} \\
& + l_zV_{x,y,z-1} + l_zV_{x,y,z+1} = 0
\end{aligned}$$

Pri čemer velja

$$\begin{aligned}
l_{x1} &= (\varepsilon_{x-1,y-1,z} + \varepsilon_{x-1,y,z}) \cdot \frac{h_z h_y}{2h_x} \\
l_{x2} &= (\varepsilon_{x,y-1,z} + \varepsilon_{x,y,z}) \cdot \frac{h_z h_y}{2h_x} \\
l_{y1} &= (\varepsilon_{x-1,y-1,z} + \varepsilon_{x,y-1,z}) \cdot \frac{h_z h_x}{2h_y} \\
l_{y2} &= (\varepsilon_{x-1,y,z} + \varepsilon_{x,y,z}) \cdot \frac{h_z h_x}{2h_y} \\
l_z &= (\varepsilon_{x-1,y-1,z} + \varepsilon_{x-1,y,z} + \varepsilon_{x,y-1,z} + \varepsilon_{x,y,z}) \cdot \frac{h_x h_y}{4h_z}
\end{aligned}$$

Za plasti, ki so homogene tudi po x in y oseh, pa uporabimo enačbo:

$$\begin{aligned}
& -(2m_x + 2m_y + 2m_z)V_{x,y,z} \\
& + m_xV_{x-1,y,z} + m_xV_{x+1,y,z} \\
& + m_yV_{x,y-1,z} + m_yV_{x,y+1,z} \\
& + m_zV_{x,y,z-1} + m_zV_{x,y,z+1} = 0
\end{aligned}$$

Pri čemer velja

$$\begin{aligned}
m_x &= \varepsilon_{x,y,z} \cdot \frac{h_z h_y}{h_x} \\
m_y &= \varepsilon_{x,y,z} \cdot \frac{h_z h_x}{h_y}
\end{aligned}$$



$$m_z = \varepsilon_{x,y,z} \cdot \frac{h_x h_y}{h_z}$$

## 2.3 SOR iteracijska metoda

Število neznank in s tem število enačb v sistemu, ki ga dobimo s pomočjo končnih diferenc, je zelo veliko - enako je številu vozlišč v mreži, ta pa narašča s tretjo potenco dimenzij vezja, ki ga obravnavamo. Že manjša vezja prinesejo sisteme z  $10^6$  neznank. Pri večjih vezjih oziroma bolj gosti računski mreži pa hitro dosežemo tudi  $10^9$  neznank, s čimer pa že dosežemo meje zmogljivosti delovnega pomnilnika osebnih računalnikov.

Očitno je, da neiteracijske metode odpadejo za reševanje tako velikih sistemov. Vse te metode namreč zahtevajo popoln zapis enačb, ki nastopajo v sistemu, to pa zahteva zmogljivost pomnilnika, ki je odvisna od kvadrata števila spremenljivk v sistemu.

Od razpoložljivih iteracijskih metod sem se odločil za uporabo SOR iteracije zaradi naslednjih razlogov:

- Je relativno preprosta, kar spet ugodno vpliva na število napak pri programiranju.
- Konvergenca je zagotovljena, saj je sistem enačb diagonalno dominanten.
- Ob primerni izbiri parametra  $\omega$  zagotavlja hitro konvergenco.
- Za razliko od nekaterih drugih metod ne zahteva, da v pomnilniku računalnika shranjujemo rezultate dveh zaporednih iteracij. To pomeni, da pri enako velikem pomnilniku omogoča reševanje dvakrat večjega problema.

Uporabil sem naslednji algoritem [4]:

$$x_i^{(k)} = \omega \bar{x}_i^{(k)} + (1 - \omega)x_i^{(k-1)} \quad (5)$$

$$x_i = V_{x,y,z} \quad (6)$$

$$i = x + S_x \cdot y + S_x S_y \cdot z \quad (7)$$

$$x \in [0, S_x - 1] \quad y \in [0, S_y - 1] \quad z \in [0, S_z - 1] \quad (8)$$

Pri čemer je  $x_i^{(k)}$  vrednost  $i$ -te spremenljivke (v mojem primeru vrednost potenciala v neki točki na mreži) po  $k$ -ti iteraciji.  $\omega \in (0, 2)$  je ekstrapolacijski parameter, od katerega je odvisna hitrost konvergence.  $\bar{x}_i$  pa je vrednost  $i$ -te spremenljivke, izračunana s pomočjo Gauss-Seidel iteracije.

Izračun vrednosti s Gauss-Seidel iteracijo ([5], stran 44) je v mojem primeru (če uporabimo splošno enačbo za potencial na enem vozlišču v računski mreži, ki je napisana zgoraj):

$$\begin{aligned} V_{x,y,z}^{(k)} = & (k_{x1} + k_{x2} + k_{y1} + k_{y2} + k_{z1} + k_{z2})^{-1} \\ & \cdot (k_{x1} V_{x-1,y,z}^{(k-1)} + k_{x2} V_{x+1,y,z}^{(k-1)} \\ & + k_{y1} V_{x,y-1,z}^{(k-1)} + k_{y2} V_{x,y+1,z}^{(k-1)} \\ & + k_{z1} V_{x,y,z-1}^{(k-1)} + k_{z2} V_{x,y,z+1}^{(k-1)}) \end{aligned}$$

### 3 Implementacija

Program je napisan v programskem jeziku C in obsega približno 4800 vrstic kode. Uporablja se z ukazne vrstice v UNIXu podobnih operacijskih sistemih (preizkušeno na operacijskih sistemih Debian GNU/Linux in Darwin/Mac OS X).

Izvorna koda je dosegljiva na naslovu <http://www.tablix.org/~avian/nelma>.

#### 3.1 Uporaba programa

Sintaksa parametrov programa:

---

```
nelma-cap [ -s STANDOFF ]
          [ -n ITERATIONS ]
          [ -w SOR_OMEGA ]
          [ -e MAX_ERROR ]
          [ -d ]
```

[ -v VERBOSITY ]  
config\_file.em

-s STANDOFF Določitev najmanjše razdalje med ozemljeno škatlo in prevodnikom. STANDOFF je pozitivno celo število.  
-n ITERATIONS Določitev minimalnega števila iteracij. ITERATIONS je pozitivno celo število.  
-w SOR\_OMEGA Vrednost ekstrapolacijskega parametra za SOR iteracijo. SOR\_OMEGA je realno število med 0 in 2.  
-e MAX\_ERROR Določitev konvergenčnega kriterija (največja relativna napaka pri izračunu kapacitivnosti). MAX\_ERROR je realno število med 0 in 1.  
-v VERBOSITY Nivo zgovornosti programa. VERBOSITY je pozitivno celo število. Večje kot je, več sporočil program izpiše na zaslon.  
-d Zastavica, ki vključi izpisovanje izračunanega električnega polja v datoteko na trdem disku.  
config\_file.em Pot do datoteke, ki vsebuje opis geometrije tiskanega vezja.

---

Program po končanem izračunu izpiše na standardni izhod (običajno zaslon terminala) vrednosti kapacitivnosti v obliki, ki je kompatibilna s programi za simulacijo vezij SPICE.

Primer:

---

```
* Created by NELMA capacitance calculator, version 2.0
* Wed Sep 27 18:08:31 2006
*
* Command line: nelma-cap -s 50 -n 100 -w 1.8 -e 0.02 -d layout.em

C0001 gnd line2 3.289585e-12
* numerical error: +/- 1.18e-13 ( 4 %)
C0002 gnd line1 3.688931e-12
* numerical error: +/- 1.29e-13 ( 3 %)
C0102 line2 line1 1.026117e-14
* numerical error: +/- 2.47e-17 ( 0 %)
```

---

Pri čemer na primer vrstici

---

C0001 gnd line2 3.289585e-12  
\* numerical error: +/- 1.18e-13 ( 4 %)

---

Pomenita, da je bila izračunana kapacitivnost med vozliščema `gnd` in `line2` 3.29 pF in da je bila ocenjena relativna napaka izračuna 4% (to je napaka, ki izhaja iz postavitve ozemljene škatle, in ne iz reševanja linearnih enačb).

V kolikor je bila programu podana tudi zastavica `-d`, bo program izpisal na disk tudi večje število datotek z izračunanimi vrednostimi električnega potenciala in polja.

Imena izpisanih datotek:

<code>imevoztisca-x</code>	Izračunano polje v preseku $x = \frac{S_x}{2}$ , ko je bilo na napetost priključeno vozlišče z imenom <code>imevoztisca</code> .
<code>imevoztisca-y</code>	Izračunano polje v preseku $y = \frac{S_y}{2}$ , ko je bilo na napetost priključeno vozlišče z imenom <code>imevoztisca</code> .
<code>imevoztisca-imeplasti</code>	Izračunano polje v preseku $z = z_{imeplasti}$ , ko je bilo na napetost priključeno vozlišče z imenom <code>imevoztisca</code> . $z_{imeplasti}$ je z koordinata sredine plasti z imenom <code>imeplasti</code> .

Oblika vrstic v izpisanih datotekah:

---

`i j field_ij potential_ij con_ij`

<code>i</code>	Prva koordinata točke (odvisno od preseka).
<code>j</code>	Druga koordinata točke (odvisno od preseka).
<code>field_ij</code>	Velikost električnega polja v tej točki.
<code>potential_ij</code>	Vrednost električnega potenciala v tej točki.
<code>con_ij</code>	Vrednost, ki pove ali je ta točka del robnih pogojev (1) ali ne (0).

---

### 3.2 Opis geometrije vezja

Program prebere geometrijo vezja iz tekstovne datoteke in ene ali več slikovnih datotek v formatu PNG.

Tekstovna datoteka je sestavljena iz več delov, ki opisujejo materiale, ki so uporabljeni v tiskanem vezju, geometrijo povezav in podobno.

Sledi podrobnejši opis delov datoteke na enem izmed primerov, ki so priloženi programu:

---

```
net line1 {
    objects = {
        "line1-front",
        "line1-back"
    }
}
net line2 {
    objects = {
        "line2-front",
        "line2-back"
    }
}
net gnd {
    objects = {
        "gnd-front",
        "gnd-back"
    }
}
```

---

Prvi del datoteke opisuje vozlišča (*nets*) vezja. Vozlišče je lahko sestavljeno iz enega ali več objektov (*objects*). V tem primeru so definirana tri vozlišča (`line1`, `line2` in `gnd`), vsako pa je sestavljeno iz dveh objektov (en objekt za vsako stran tiskanega vezja).

---

```
object gnd-front {
    position = { 0, 0 }

    material = "copper"

    type = "image"
    role = "net"

    file = "front.png"
    file-pos = { 30, 30 }
}

object gnd-back {
    position = { 0, 0 }

    material = "copper"

    type = "image"
    role = "net"

    file = "back.png"
    file-pos = { 30, 30 }
}

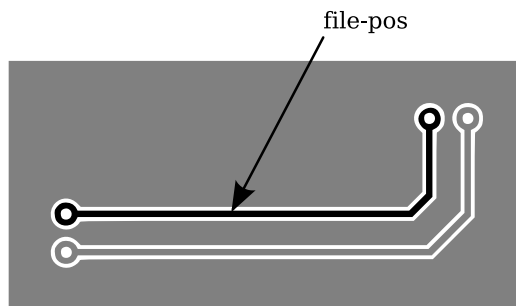
...
```

---

Naslednji del datoteke opisuje objekte. Objekt je definiran kot del prostora, ki je zapolnjen z določenim materialom. V tem primeru so objekti iz bakra (definicija `material`), njihov spodnji levi rob pa je postavljen v koordinatno izhodišče.

Objekti so tukaj definirani kot dvodimenzionalne strukture. Tretjo dimenzijo (višino) dobijo od plasti, v kateri so.

Sama geometrija objekta se naloži iz slikovne datoteke, ki vsebuje masko za jedkanje posameznega sloja tiskanega vezja (v tem primeru datoteki `front.png` in `back.png`). Definiciji `file-pos` vsebujeta koordinati objektov na sliki.



Slika 2: Določanje geometrije objekta iz maske. `file-pos` kaže na eno točko na objektu, program pa nato s pomočjo flood-fill algoritma določi geometrijo objekta iz slike.

Program za potrebe verifikacije podpira tudi podajanje geometrije objektov brez slikovne datoteke. V tem primeru je parameter `type` enak `rectangle` ali `circle`.

---

```
material composite {
    type = "dielectric"

    permittivity = 3.54e-11
    conductivity = 1e-10
    permeability = 0.0
}

material air {
    type = "dielectric"

    permittivity = 8.85e-12
    conductivity = 1e-10
    permeability = 0.0
}

material copper {
```

```
    type = "metal"

    permittivity = 0.0
    conductivity = 5.96e7
    permeability = 0.0
}
```

---

V nadaljevanju datoteke so definirani materiali, ki se uporabljajo v tiskalnem vezju. Za vsak material je podana dielektričnost (As/Vm), prevodnost (S/m) in permeabilnost (Vs/Am). Trenutno se uporablja le podatek za dielektričnost.

---

```
layer air-top {
    height = 50
    z-order = 10

    material = "air"
}

layer component {
    height = 3
    z-order = 6

    material = "air"

    objects = {
        "line1-back",
        "line2-back",
        "gnd-back"
    }
}

layer substrate {
    height = 30
    z-order = 5

    material = "composite"
}
```



```

layer solder {
    height = 3
    z-order = 4

    material = "air"

    objects = {
        "line1-front",
        "line2-front",
        "gnd-front"
    }
}

```

```

layer air-bottom {
    height = 50
    z-order = 0

    material = "air"
}

```

---

Sledi definicija plasti. Vsaka plast ima definirano višino (`height`), vrstni red (`z-order`) in osnovni material plasti (`material`). V te plasti so nato dodani prej definirani objekti.

---

```

space test {
    step = { 4.2333e-5, 4.2333e-5, 1e-4 }

    layers = {
        "air-top",
        "component",
        "substrate",
        "solder",
        "air-bottom"
    }
}

```

Zadnji del datoteke definira korak računske mreže (v metrih) in v prostor doda prej definirane plasti.

## 4 Rezultati

Pravilno delovanje programa sem preveril na več načinov. Najprej sem s programom izračunal kapacitivnosti preprostih struktur, za katere je mogoče natančno ali približno izračunati kapacitivnost tudi po analitični poti. Numerične rezultate sem nato primerjal z analitičnimi.

Kot drugi način verifikacije sem uporabil strukture, ki so simetrične po eni ali več oseh. V tem primeru sem primerjal numerično izračunane kapacitivnosti, ki bi zaradi simetrije morale biti enake.

### 4.1 Primerjava z analitičnimi izračuni

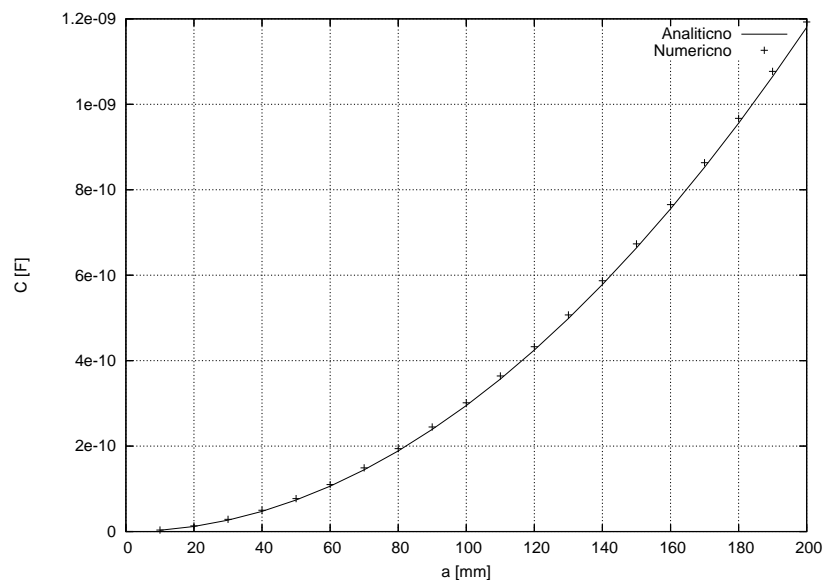
Na sliki 3 je prikazan graf kapacitivnosti ploščatega kondenzatorja s ploščami v obliki kvadrata v odvisnosti od stranice kvadrata. Plošče so bile v tem primeru pravokotne na os  $z$  (vzporedne s plastmi tiskanega vezja). Za analitični izračun kapacitivnosti je bila uporabljena naslednja enačba, ki ne upošteva robnih efektov:

$$C = \frac{\varepsilon \cdot a \cdot a}{d} \quad (9)$$

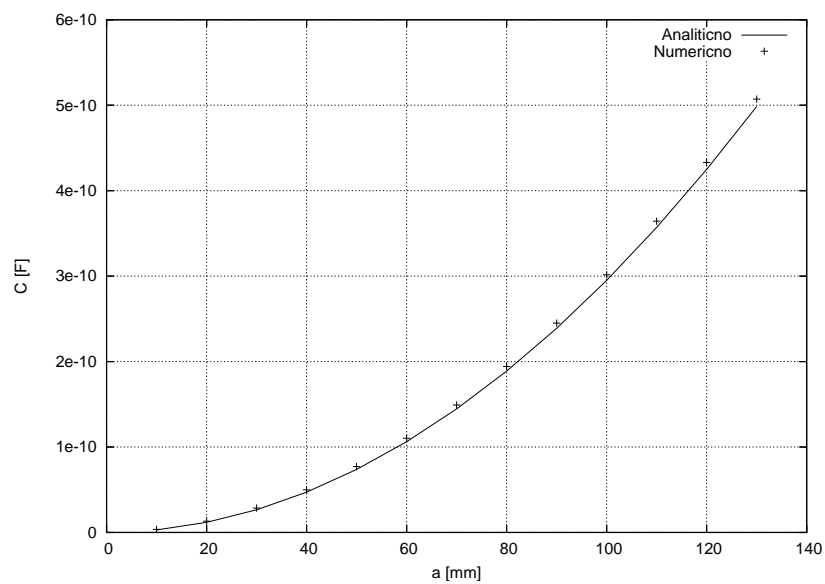
Kjer je  $a$  stranica plošče,  $d$  razmik med ploščama in  $\varepsilon$  dielektričnost materiala med ploščama. Zaradi robnih efektov pričakujemo, da bo numerično izračunana kapacitivnost nekoliko večja od analitično izračunane.

Na sliki 4 je znova prikazan graf kapacitivnosti ploščatega kondenzatorja, s to razliko, da so tokrat plošče ležale pravokotno na os  $x$  (torej so sekale plasti tiskanega vezja). Pričakujemo, da bodo ti rezultati enaki, kot pri prejšnjem izračunu.

Na sliki 5 je prikazan graf kapacitivnosti ploščatega kondenzatorja s okroglimi ploščami. Plošče so bile pravokotne na os  $z$ . Za analitični izračun kapacitivnosti je bila uporabljena Kirchhoffova enačba, ki upošteva tudi robne efekte:



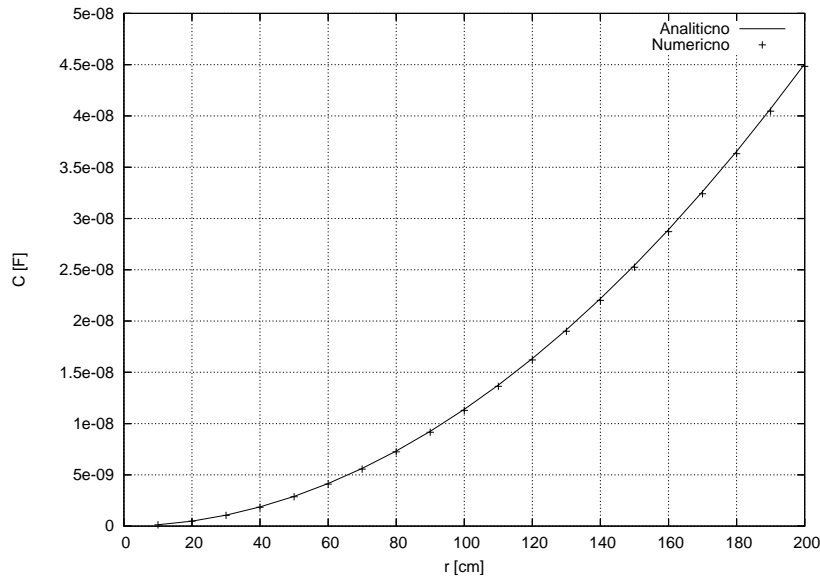
Slika 3: Odvisnost kapacitivnosti  $C$  planarnega ploščatega kondenzatorja v odvisnosti od stranice plošč  $a$ .



Slika 4: Odvisnost kapacitivnosti  $C$  lateralnega ploščatega kondenzatorja v odvisnosti od stranice plošč  $a$ .

$$C = \varepsilon \cdot \left( \frac{\pi r^2}{d} + r \log \left( \frac{16\pi r}{d} - 1 \right) \right) \quad (10)$$

Kjer je  $r$  polmer plošč,  $d$  razmik med ploščama in  $\varepsilon$  dielektričnost materiala, ki obdaja plošče. Ker uporabljena enačba upošteva tudi robne efekte, pričakujemo, da bodo numerični rezultati enaki analitičnim do numerične napake natančno.



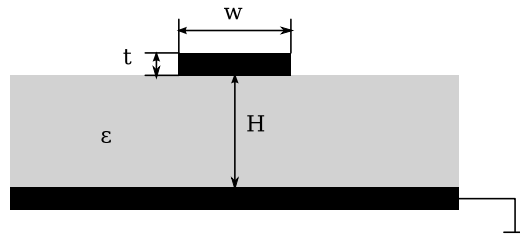
Slika 5: Odvisnost kapacitivnosti  $C$  ploščatega kondenzatorja v odvisnosti od polmera plošč  $r$ .

Na sliki 7 je prikazan graf kapacitivnosti mikrotrakaste linije proti masi v odvisnosti od njene širine. Za analitični izračun kapacitivnosti je bil uporabljen naslednji približek [6]:

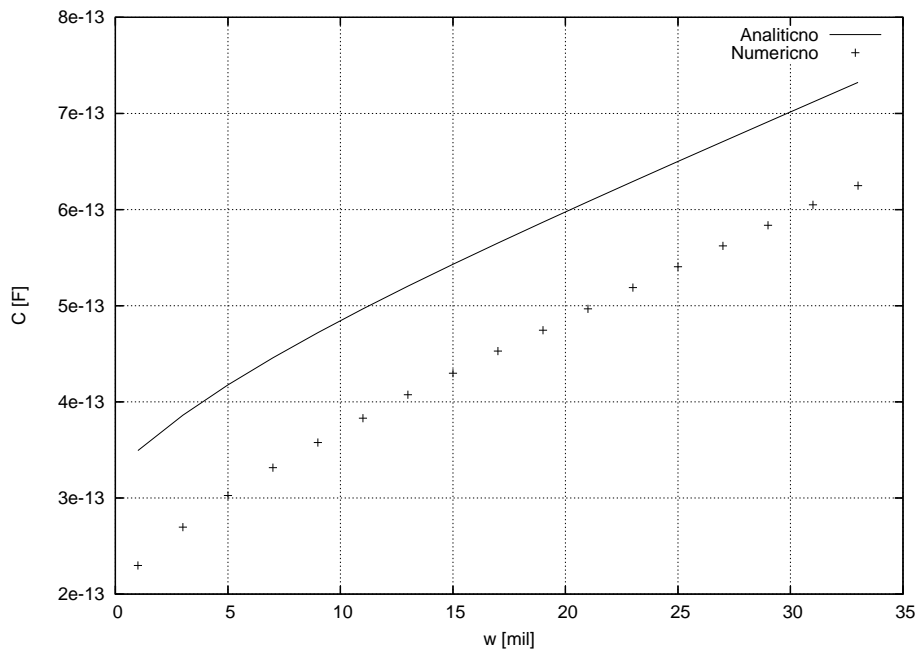
$$c [pF] = l [in] \cdot \frac{0.67 \cdot (\varepsilon_r + 1.41)}{\ln \frac{5.98 \cdot H}{0.8 \cdot w + t}} \quad (11)$$

Na sliki 9 je prikazan graf kapacitivnosti pokopane mikrotrakaste linije proti masi v odvisnosti od njene širine. Za analitični izračun kapacitivnosti je bil uporabljen naslednji približek [6]:

$$c [pF] = l [in] \cdot \frac{1.41 \cdot \varepsilon_r}{\ln \frac{3.81 \cdot H}{0.8 \cdot w + t}} \quad (12)$$



Slika 6: Mikrotrakasta linija.



Slika 7: Odvisnost kapacitivnosti  $C$  proti masi mikrotrakaste linije v odvisnosti od njene širine  $w$ .

## 4.2 Preizkus simetrije

Pri preizkusu simetrije sem v prostor simetrično postavil 8 kosov prevodnika v obliki kocke (slika 10) in računal kapacitivnosti med njimi. Zaradi simetrije bi morale biti vse kapacitivnosti po stranicah kocke (sem spada na primer kapacitivnost med `net1` in `net2`) enake. Prav tako bi morale biti med seboj enake tudi kapacitivnosti po diagonalah ploskev kocke (sem spada na primer kapacitivnost med `net1` in `net4`) in po prostorskih diagonalah (sem spada na primer kapacitivnost med `net1` in `net8`).

Rezultati izračuna so prikazani v tabelah 1, 2 in 3. Pri izračunu je bil kot konvergenčni kriterij za SOR iteracijo postavljena relativna napaka kapacitivnosti 1% ali manj.

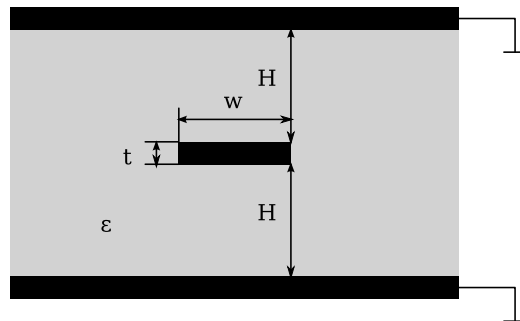
## 5 Zaključek

Preizkusi so pokazali, da se izbrane numerične metode dokaj dobro obnesejo pri računanju kapacitivnosti. Vsi preizkusi so dali pričakovane rezultate.

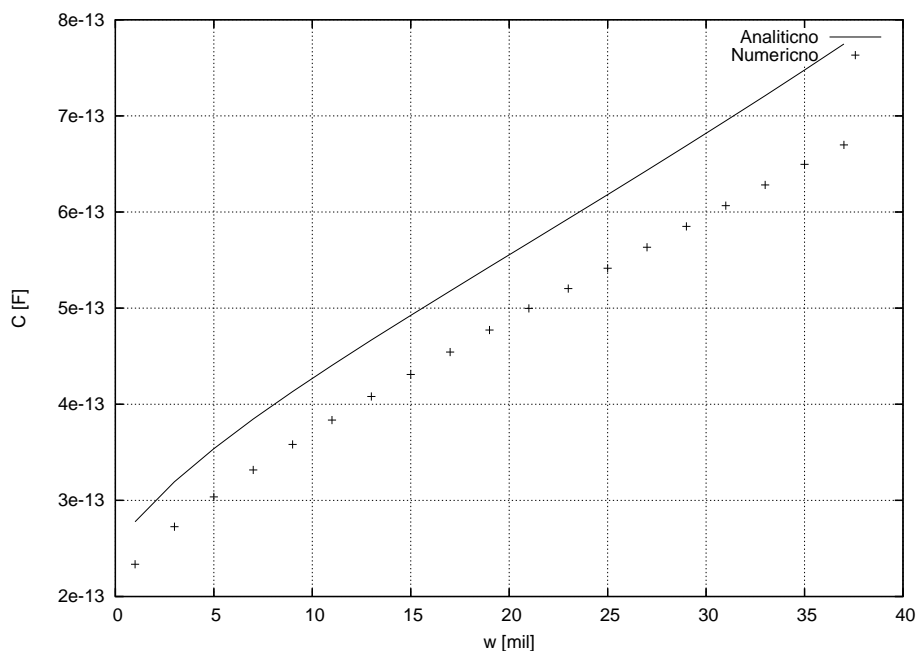
Najbolj izstopata izračuna kapacitivnosti za oba tipa mikrotrakaste linije. Vzroki za ta odstopanja bi lahko bili naslednji:

- Analitični izrazi so v teh dveh primerih le grobi približki, saj natančna analitična obravnava teh dveh struktur ni mogoča.
- Pri teh dveh izračunih tudi pride najbolj do izraza poenostavitev, ki jo naredimo s postavljanjem ozemljene škatle (ena prevodna struktura je dosti večja od druge).
- Lahko gre za napako v programu (verjetnost za to je majhna, saj so drugi preizkusi dali točne rezultate).

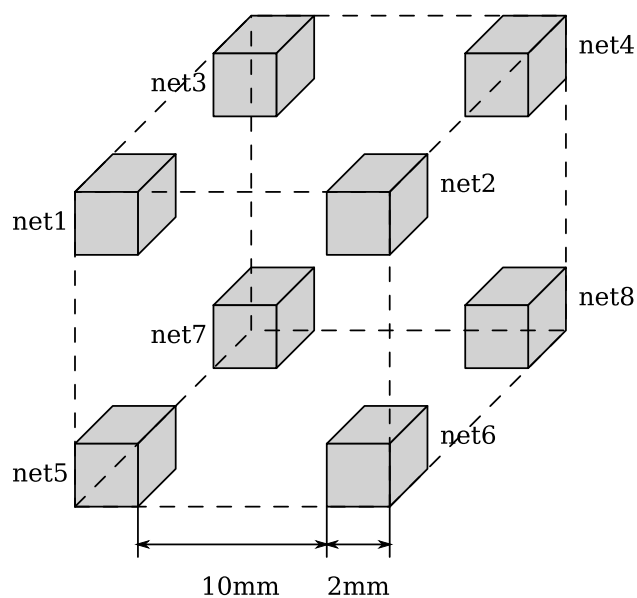
Vsekakor pa lahko velikost napake pri teh dveh primerih (ki sta sicer še najbližje pogojem v pravem tiskanem vezju) vzamemo kot tipično velikost napake, ki jo moramo upoštevati pri uporabi tega programa.



Slika 8: Pokopana mikrotrakasta linija



Slika 9: Odvisnost kapacitivnosti  $C$  proti masi vkopane mikrotrakaste linije v odvisnosti od njene širine  $w$ .



Slika 10: Razporeditev prevodnikov za preizkus simetrije.

```

C0007 net8 net1 3.331270e-15
C0106 net7 net2 3.331266e-15
C0205 net6 net3 3.331267e-15
C0304 net5 net4 3.324573e-15

```

Največje odstopanje: 6.697e-18 (0.2%)

Tabela 1: Kapacitivnosti po prostorskih diagonalah kocke pri preizkusu simetrije.



C0003 net8 net5 6.062178e-15  
C0005 net8 net3 6.162798e-15  
C0006 net8 net2 6.162798e-15  
C0102 net7 net6 6.069830e-15  
C0104 net7 net4 6.162416e-15  
C0107 net7 net1 6.162765e-15  
C0204 net6 net4 6.162416e-15  
C0207 net6 net1 6.162764e-15  
C0305 net5 net3 6.153117e-15  
C0306 net5 net2 6.153118e-15  
C0407 net4 net1 6.069553e-15  
C0506 net3 net2 6.069858e-15

Največje odstopanje: 1.0062e-16 (1.6%)

Tabela 2: Kapacitivnosti po diagonalah ploskev kocke pri preizkusu simetrije.

C0001 net8 net7 1.382707e-14  
C0002 net8 net6 1.382707e-14  
C0004 net8 net4 1.392616e-14  
C0103 net7 net5 1.381599e-14  
C0105 net7 net3 1.392612e-14  
C0203 net6 net5 1.381598e-14  
C0206 net6 net2 1.392612e-14  
C0307 net5 net1 1.391258e-14  
C0405 net4 net3 1.382710e-14  
C0406 net4 net2 1.382710e-14  
C0507 net3 net1 1.382706e-14  
C0607 net2 net1 1.382706e-14

Največje odstopanje: 9.906e-17 (0.7%)

Tabela 3: Kapacitivnosti po stranicah kocke pri preizkusu simetrije.

## 6 Literatura

### Literatura

- [1] Hong, W. et al.: *A Novel Dimension-Reduction Technique for the Capacitance Extraction of 3-D VLSI Interconnects*. MTT, Vol. 46, No. 8, strani 1037-1044.
- [2] Beeftink, F. et al.: *Accurate and Efficient Layout-to-Circuit Extraction for High-Speed MOS and Bipolar/BiCMOS Integrated Circuits*. ICCD Proceedings 1995, strani 360-365.
- [3] Sinigoj, A. R.: *ELMG polje*. Ljubljana, Založba Fakultete za elektrotehniko, 1996.
- [4] Noel, B.; Shirley M.: *Successive Overrelaxation Method*. 2002. (citirano 10. 1. 2006). Dostopno na naslovu <http://mathworld.wolfram.com/SuccessiveOverrelaxationMethod.html>
- [5] Orel, B.: *Osnove numerične matematike*. Ljubljana, Založba FE in FRI, 199.
- [6] IPC-D-317A: *Design Guidelines for Electronic Packaging Utilizing High-Speed Techniques*. Jan 1995, Section 5, strani 13-36.