

Računalniška orodja v fiziki - Iterativni postopki

Primož Cigler (28090039)

18. maj 2010

Povzetek

V zadnji letošnji nalogi pri računalniških orodjih si bom ogledal, kako lahko celične avtomate uporabimo v fiziki za določitev končnega stanja, ki se vzpostavi po celo dolgem času. V teoriji to pomeni, da ko pošljemo čas (oz. število korakov celičnega avtomata) proti neskončno, vendar nimamo toliko časa, tako da si bomo ogledali primere, ko celični avtomat naredi toliko ponovitev, da smo s približkom končnega stanja zadovoljni.

1 Naloga

Določi temperaturni profil v stacionarnem stanju (= po dolgem času) za dobro prevodno kovinsko palico, ki je izolirana po plašču, da poteka prevajanje toplote le v vzdolžni smeri. Konca palice sta hlajena na temperaturo $T = 0$, desno polovico palice pa grejemo z visokofrekvenčnim električnim poljem. Gretje opišemo s konstantnim členom, kjer je P/V gostota moči in λ toplotna prevodnost palice, ki ga dodamo VsotiSosedov. Primerno število celic N v linearni verigi je 20, konstantni člen pa naj bo enak 1.

1.1 Rešitev

Prvo nalogo sem rešil tako, da sem samo malenkost spremenil PHP program iz prejšnje naloge.

```
<?php

$palica = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);

$i = 0;

do
{
    $backup = $palica;
    $lim = 0;

    for ($j = 1; $j < 31; $j++)
    {
        $palica [$j] = round(($backup [$j-1] + $backup [$j+1])*0.5, 10);
        if ($j>15)
            $palica[$j] += 1;
    }
    $palica [0] = 0;
    $palica [31] = 0;

    for ($j = 0; $j < 32; $j++) {
        $lim += ($backup[$j]-$palica[$j])*($backup[$j]-$palica[$j]);
    }

    if ($i%30 == 0)
    {
        for ($k = 0; $k < 32; $k++) {
            $out [$k] .= str_pad($palica[$k], 15, "0", STR_PAD_LEFT) . "\t";
            $out2 [$k] = $palica[$k];
        }
    }

    $i++;
    $backup = NULL;
} while ($lim > 1e-10);

function sosede ($matrix, $x)
{
    $sum = 0;

    for ($i = $x-1; $i <= $x+1; $i++)
```

```

    {
        for ($j = $y-1; $j <= $y+1; $j++)
        {
            if ($i == $x && $j == $y) {
                $sum += 0;
            } else {
                $sum += $matrix[$i][$j];
            }
        }
    }
    $sum /= 8;

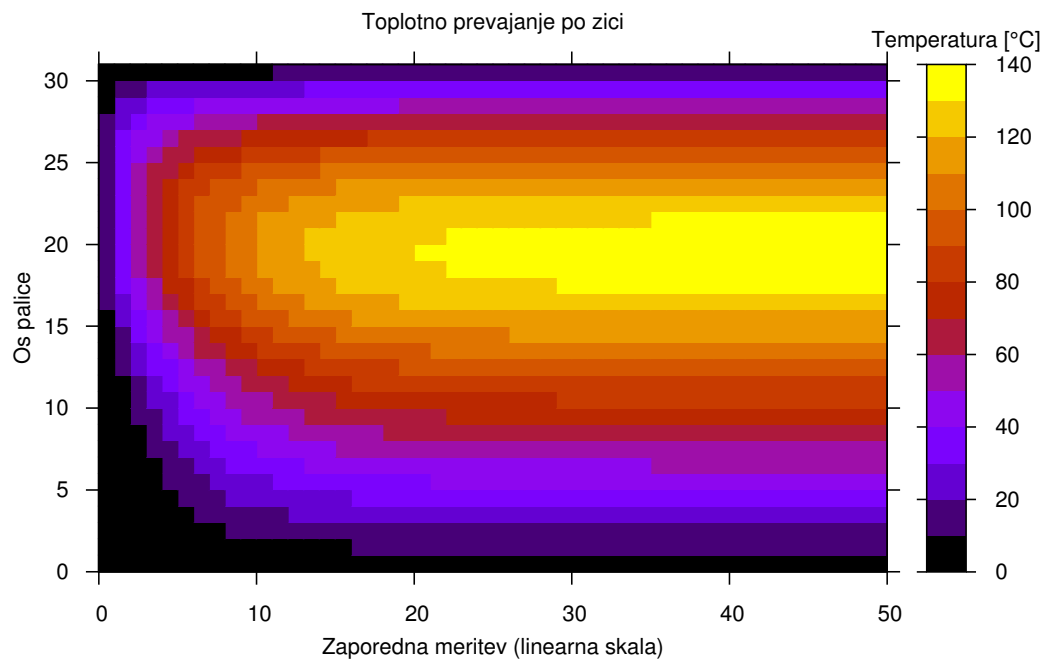
    return $sum;
}

$write = implode ("\n", $out);
echo $write . "\n\n";

$i = 0;
foreach ($out2 as $num) {
    echo $i . " " . $num . "\n";
    $i++;
}

```

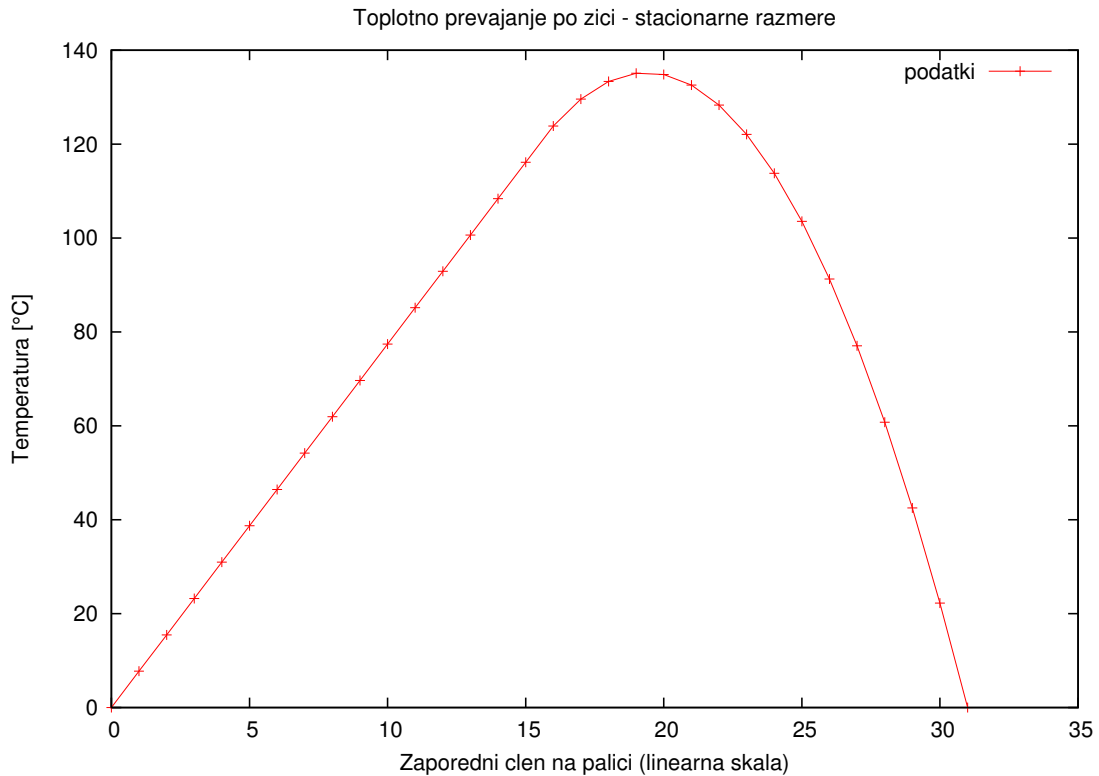
Program mi je vsakih 30 ponovitev izpisal stanje temperatur na členih palice. Ko sem imel podatke, sem z znanjem z 8. naloge narisal spodnji graf:



Graf 1: Graf temperature na palici v odvinosti od časa.

Vzel sem malenkost daljšo palico, kot piše v navodilih (32 členov), kar ne spremeni končne slike, ker lahko vedno model reskaliramo.

Narisal sem še 2-dimenzionalen graf samo za končno sliko in sicer kako je temperatura razporejena na koncu:



Graf 2: Graf temperaturnega profila na palici po dolgem času.

Vidimo, da na nesegrevanem delu palice temperatura linearno pada proti koncu, na segrevanem delu pa ima na prvi pogled obliko parabole. Predvsem je očitno, da graf ni simetričen, kar tudi pričakujemo glede na potem segrevanja in prevajanja.

2 Naloga

Prožno opno napnemo na okvir s kvadratnim tlorisom 20 cm x 20 cm, v katerem sta dve nasprotni stranici parabolično zapognjeni, tako da sežeta za 2 cm nad ravnino, ki jo določata preostali (ravni) stranici. Kakšna je njena oblika? Za natančnost računa zadošča centimetrska mreža. Vsaka celica ima štiri sosede, tiste, s katerimi se stika prek stranice. Z našim povprečevalnim avtomatom dosežemo, da se vsaka celica najglajše vleže med sosede, tako da je celotna površina opne najmanjša.

2.1 Rešitev

V PHP-ju sem napisal naslednji program:

```
<?php
for ($x = -10; $x < 11; $x++)
{
    for ($y = -10; $y < 11; $y++)
    {
        $matrix[$x][$y] = f($x);
    }
}
```

```

display($matrix);

function f($x)
{
    $val = 6 -(3/50)*$x*$x;
    return $val;
}

do {
    $backup = NULL;
    $backup = $matrix;
    $lim = 0;

    for ($x = -9; $x < 10; $x++)
    {
        for ($y = -9; $y < 10; $y++)
        {
            $matrix[$x][$y] = sosede ($backup, $x, $y);
        }
    }

    for ($x = -10; $x < 11; $x++)
    {
        for ($y = -10; $y < 11; $y++)
        {
            $lim += ($backup[$x][$y]-$matrix[$x][$y])*($backup[$x][$y]-$matrix[$x][$y]);
        }
    }
}
while ($lim > 1e-10);

echo "\n\n";
display ($matrix);

function display ($matrix)
{
    for ($x = -10; $x < 11; $x++)
    {
        for ($y = -10; $y < 11; $y++)
        {
            $out .= str_pad($matrix[$x][$y], 10, "0", STR_PAD_LEFT) . "\t";
        }
        $out .= "\n";
    }

    echo $out;
}

function sosede ($matrix, $x, $y)
{
    $sum = 0;
    $sum += $matrix[$x][$y-1];
    $sum += $matrix[$x][$y+1];
}

```

```

$sum += $matrix[$x-1][$y];
$sum += $matrix[$x+1][$y];

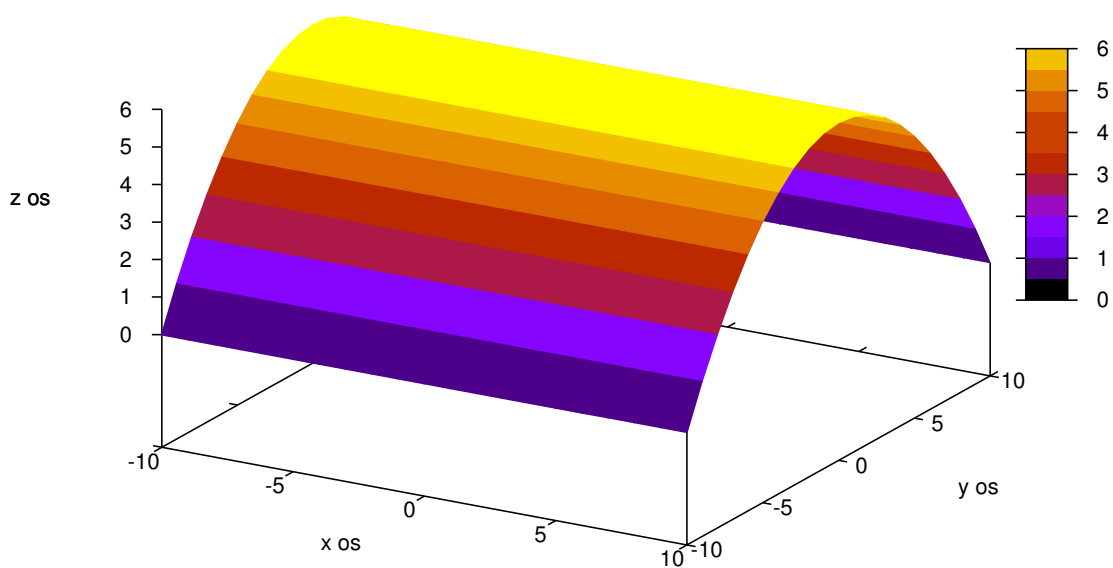
$sum /= 4;

return $sum;
}

```

Najprej sem na roko izračunal parametre za funkcijo $f(x) = ax^2 + bx + c$ in določil parametre a, b in c, če naj bosta ničli te parabole na ± 10 , višina pa 6. Dobil sem $a = -\frac{3}{50}$, $b = 0$ in $c = 6$. Nato sem to vnesel v program, da mi je izračunal začetno sliko:

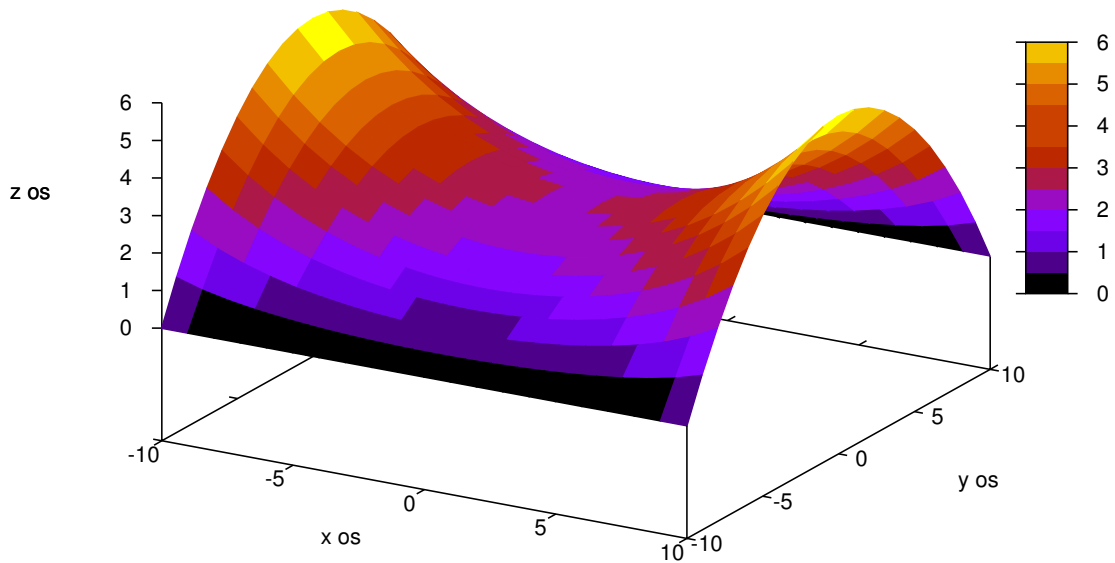
Opna



Graf 3: Začetna slika za opno.

Dopisal sem funkcijo “sosede”, ki mi je za vsako točko izračunala aritmetično sredino sosednjih štirih in pognal algoritem, dokler razlika popravkov med dvema zaporednima obhodom ni bila dovolj majhna ($1e-10$). Dobil sem naslednjo končno sliko:

Opna



Graf 4: Končna slika za opno.

Končna slika je takšna, kot sem pričakoval, da bo. Ta oblika površine, če se robovi ne smejo spreminjati, ima najmanjšo površino za zvezno ploskev.

3 Naloga

Neobvezna naloga: poskusi sam izračunati temperaturni profil dimnika, ki smo ga risali v nalogi 8.2, le da vzemi luknjo gladke pravokotne oblike istih dimenzij.

3.1 Rešitev

Lotil sem se tudi neobvezne naloge. Spet sem samo malo spremenil skripto iz prejšnje naloge:

```
<?php
```

```
for ($x = 0; $x < 50; $x++)  
{  
    for ($y = 0; $y < 50; $y++)  
    {  
        if ($x>20 && $x<30 && $y > 20 && $y<30) {  
            $matrix[$x][$y] = 200;  
        } else {  
            $matrix[$x][$y] = 0;  
        }  
    }  
}
```



```

display($matrix);

do {
    $backup = NULL;
    $backup = $matrix;
    $lim = 0;

    for ($x = 1; $x < 49; $x++)
    {
        for ($y = 1; $y < 49; $y++)
        {
            if ($x>20 && $x<30 && $y > 20 && $y<30)
            {
                $matrix[$x][$y] = 200;
            } else {
                $matrix[$x][$y] = sosede ($backup, $x, $y);
            }
        }
    }

    for ($x = 0; $x < 50; $x++)
    {
        for ($y = 0; $y < 50; $y++)
        {
            $lim += ($backup[$x][$y]-$matrix[$x][$y])*($backup[$x][$y]-$matrix[$x][$y]);
        }
    }
}
while ($lim > 10);

echo "\n\n";
display ($matrix);

function display ($matrix)
{
    for ($x = 0; $x < 50; $x++)
    {
        for ($y = 0; $y < 50; $y++)
        {
            $out .= str_pad($matrix[$x][$y], 10, "0", STR_PAD_LEFT) . "\t";
        }
        $out .= "\n";
    }

    echo $out;
}

function sosede ($matrix, $x, $y)
{
    $sum = 0;
    $sum += $matrix[$x][$y-1];
    $sum += $matrix[$x][$y+1];
    $sum += $matrix[$x-1][$y];
    $sum += $matrix[$x+1][$y];
}

```

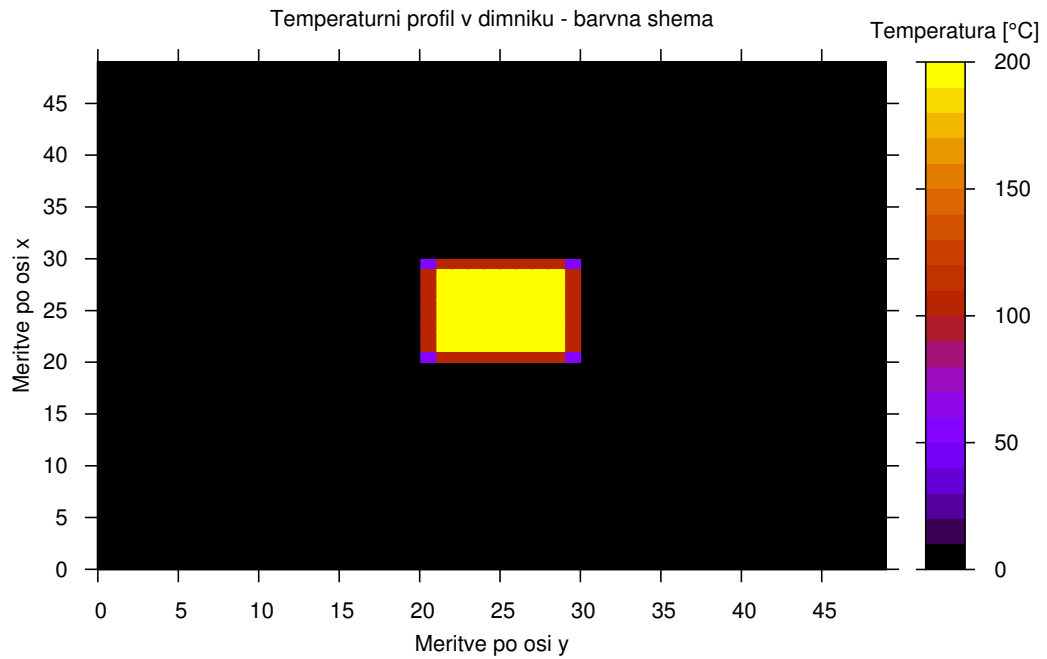
```

    $sum /= 4;

    return $sum;
}

```

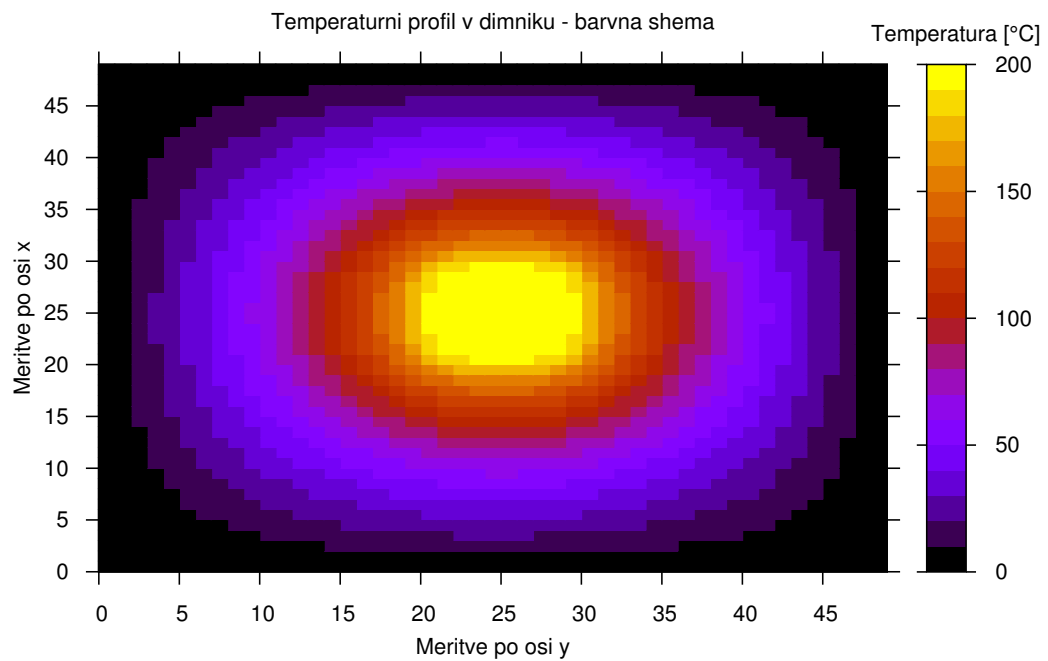
Ustvari sem polje 50 x 50 in v sredinskih 10 x 10 točk vstvail konstantnih 200 °C, vse ostalo pa pustil na 0. Začetna slika izgleda takole:



Graf 5: Začetna slika za temperature v dimniku.

Nato sem pognal program, dokler ni bila suma razlik med dvema ponovitvama manjša od 0.1 (računalnik je potreboval kar nekaj sekund in tukaj bi se verjetno dosti boljše v smislu hitrosti izkazal kakšen drug programski jezik, ki je bolj namenjen numerični obdelavi podatkov).

Dobil sem naslednjo končno sliko:



Graf 6: Končna slika za temperature v dimniku.

Dobim zelo podoben graf, kot pri 8. nalogi, ko smo podatke že dobili in smo morali samo narisati graf.