University *of Ljubljana*
Faculty *of Mathematics and Physics*

Modelska Analiza 1

2. naloga - Linearno programiranje

Avtor: Matic Lubej
Asistent: dr. Simon Čopar
Predavatelj: prof. dr. Alojz Kodre

Ljubljana, 13.10.2013

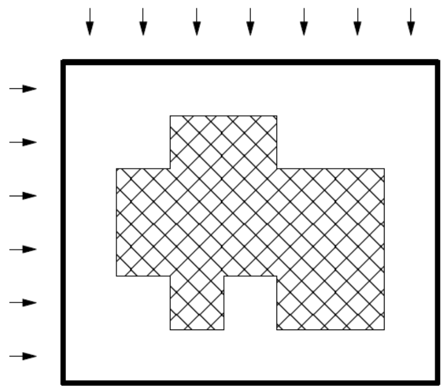
Naloga:

V nalogi smo predpostavili preprost 2D model rakastega tkiva, ki smo ga zdravili s sevanjem. V prvem delu naloge sem z linearnim programiranjem prišel do optimale krajevne odvisnosti jakosti žarkov med sevanjem. V drugem delu naloge sem linearno programiranje uporabil na problemu iz prejšnje naloge, kjer smo morali najti optimalno vožnjo avta do semaforja. Tu so enačbe linearne, zato smo problem nekoliko redefinirali. V tretjem delu naloge sem numerično rešil problem optimalne vožnje na način, ki mi je omogočal tudi nelinearnosti. Računal in grafično prikazal sem v programskem orodju Mathematica, kjer sem večinoma uporabljal funkcijo LinearProgramming ali pa NMinimize.

Del I

Obsevanje tkiva

V radioterapiji je poglavitni problem načrtovanje doze obsevanja. Izvire sevanja je treba razporediti tako, da v predelih tumorja zagotovijo dozo, ki je večja od kritične doze za uničenje tkiva, v zdravem okoliškem tkivu pa naj bo prejeta doza čim manjša. Postavili bomo najpreprostejši 2D model: obsevanje z vzporednim snopom sevanja ki ga enakomerno pomikamo prek tarče v dveh pravokotnih smereh.



Tumorsko območje smo označili senčeno, ostalo pa je zdravo tkivo. Izbral sem si smer žarkov od spodaj navzgor in nato od leve proti desni, torej v smeri urinega kazalca z začetkom spodaj levo, tako sem si izbral svoj vektor spremenljivk $\mathbf{x} = \{y_1, ..., y_6, x_1, ..., x_7\}$. Vsaka celica prejme dozo, ki je vsota $x_i + y_j$. Za celice tumorja sem zahteval, da je doza večja od kritične, za zdrave pa da je skupna prejeta doza minimalna. Naredil sem tudi inverzni račun, kjer zahtevamo, da je za zdrave celice doza manjša od kritične, za celice tumorja pa da je skupna doza maksimalna. Problema sem v vsakem primeru predstavil skupaj. V vseh primerih sem uporabil kritično dozo $D = 1$. Potrebno je opomniti, da v začetnih primerih absorpcija ni bila upoštevana, zato rezultati niso tako zanimivi kot v tistih primerih, kjer absorpcija je upoštevana.

Reševal sem v Mathematici s funkcijo LinearProgramming$[c, m, b]$, kjer parameter c predstavlja minimizirano količino $c \cdot x$, parameter m predstavlja količine, ki nastopajo v pogojih $m \cdot x = b$, parameter $b = \{\{b_1, s_1\}, \{b_2, s_2\}, ...\}$pa predstavlja pogoje $\leq b_i$ za $s_i = -1$, $\equiv b_i$ za $s_i = 0$ in $\geq b_i$ za $s_i = 1$. Parametri c, m in b so v primeru minimizacije prejete doze pri zdravem tkivu bili določeni na naslendji način:

$$c = \sum_{\{i,j\} \notin \text{Tumor}} (x_i + y_j)\,, \tag{1}$$

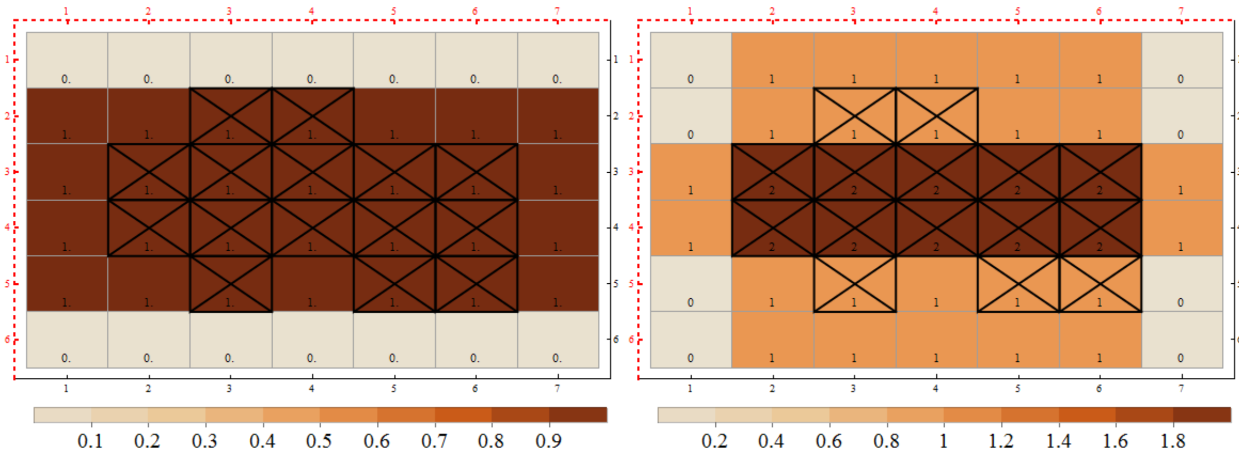
$$m = \begin{bmatrix} & & & & & \vdots & & & & & \\ 0 & \cdots & 0 & 1_j & 0 & \cdots & 0 & 1_i & 0 & \cdots & 0 \\ & & & & & \vdots & & & & & \end{bmatrix}, \tag{2}$$

$$b = \{\{D, 1\}, \{D, 1\}, \ldots\}\,, \tag{3}$$

če pa smo imeli inverzno formulacijo, so se c, m in b ustrezno spremenili, kjer smo naredili še $c \to -c$, zato da smo maksimizirali obsevanost rakastih celic, in ne minimizirali.

1 Primer

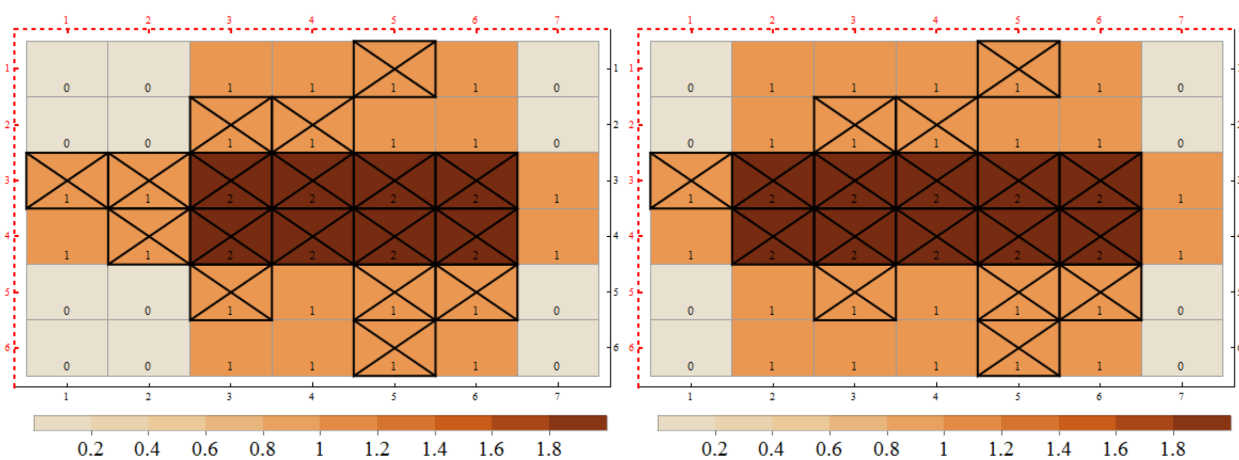
V tem razdelku si oglejmo osnovni problem, kot je nakazan v nalogi. Oglejmo si rešitve:



Vidimo, da so rešitve precej suhoparne, saj imajo žarki v vsaki poziciji le eno od dveh ali 3 možnih vrednosti. Leva slika prikazuje standarden problem, kjer vidimo, da so aktivirani le sredinski vzporedni žarki. Na prvi pogled se zdi, da najvišji žarek ni najbolj optimalen, saj ima jakost 1 v vrstici, kjer sta le 2 celici od sedmih rakasti, vendar če bi tega izključili in vključili navpična dva, bi s tem enako poškodovali enako število zdravih celic. Desna slika prikazuje inverzen problem, kjer je aktivacija žarkov malo bolj zanimiva. Tudi tu so vsi pogoji izpolnjeni, saj nobena zdrava celica ni bila preveč obsevana. Razlika pri inverznem problemu je torej večinska večja obsevanost rakastih celic, vendar pa je več celic bilo poškodovanih, a še vedno z izpolnjenim pogojem.

2 Primer

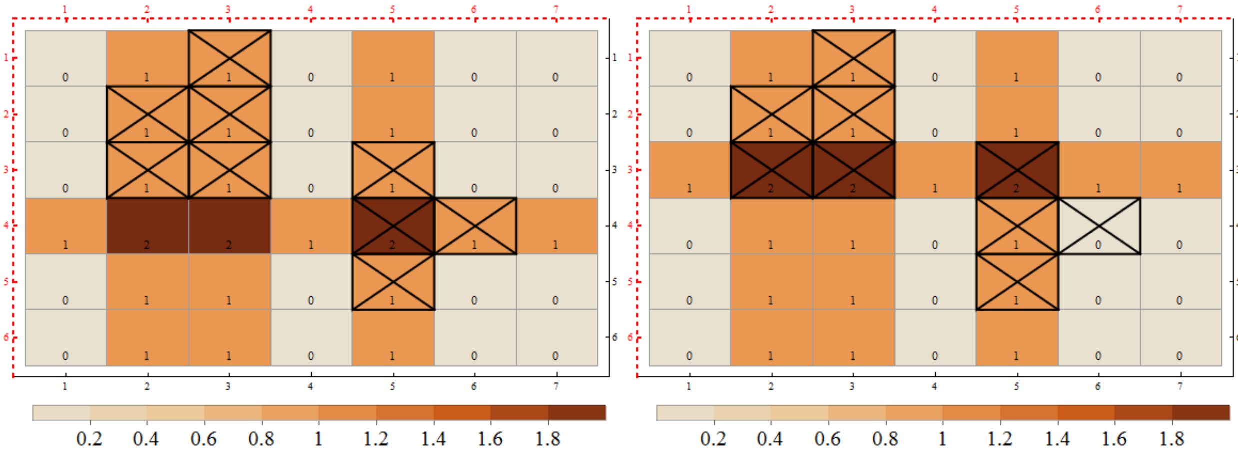
V tem primeru sem rakastemu tkivo dodal še kakšno celico zraven, tako da je malo bolj komplicirana oblika, kar se tiče izpolnjevanja pogojev:



Dodal sem 3 nove rakaste celice in rešil problem. Ponovno so pogoji v obeh primerih izpolnjeni. V tem primeru se dobro vidi razlika med obema načinoma, torej da je poudarek pri levi sliki minimiziranje sevanja pri zdravih celicah, in na desni sliki maksimiziranje sevanja pri rakastih.

3 Primer

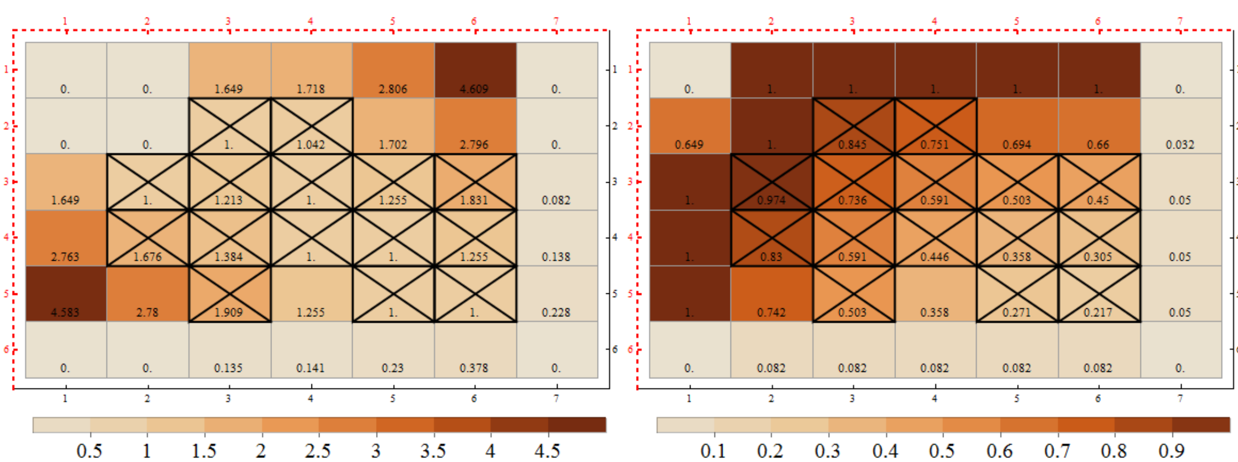
Izmislil sem si še svojo obliko tumorja, vendar rezultati niso kaj prida drugačni in še vedno predstavljajo dolgočasne rešitve:



V tem primeru imamo 2 tumorja, oziroma je tumor razdeljen na 2 dela. Tu se zdi, da prvotni način ni najbolj optimalen, saj so v nekaterih primerih zdrave celice obsevane z večjo dozo kot pa rakaste, ampak to ni narobe, saj na zdravih celicah ni bilo pogoja, le minimizacija doze obsevanja. Inverzna formulacija je bolj optimalna, saj so nekatere rakaste celice obsevanje z 2-kratno dozo, s kritično dozo je obsevana le ena celica več, z več kot kritično dozo pa ni obsevana nobena zdrava celica.

4 Primer - Absorpcija

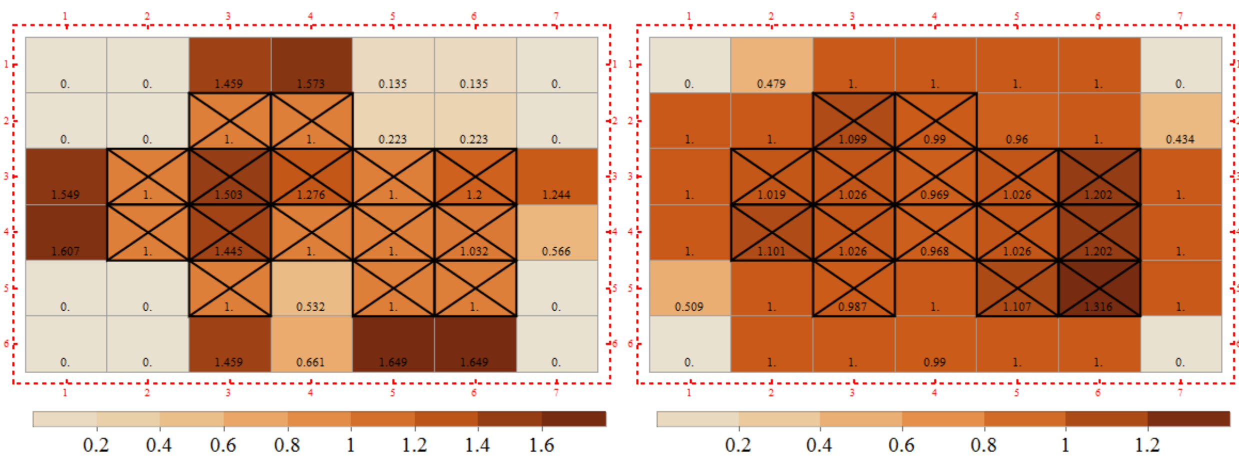
Malo bolj zanimivi rezultati se pokažejo, če vključimo absorpcijo sevanja v tkivu. Predpostavil sem, da rob 2D tkiva predstavlja koža, ostalo pa je notranjost. Z enačbo $d(r) = d_0 e^{-\lambda r}$, kjer je d_0 doza na površini kože, r globina tkiva in λparameter absorpcije, sem upošteval absorpcijo v tkivu. Po nekaj testih se je izkazala najbolj optimalna izbira $\lambda = 0.5$. Če je λ prevelik, potem so robovi dosti bolj obsevani kot notranjost, če pa je premajhen, potem pa je situacija preveč podobna prejšnji, kjer absorpcije ni bilo. Poglejmo si primer, kjer smo imeli prvotno obliko tumorja, žarka le iz dveh smeri in upoštevao absorpcijo:



Tokrat pa rešitve niso več celoštevilske jakosti žarkov, ampak imamo ne-cele vrednosti. Na levi sliki opazimo, da jakost žarkov narašča z večjo oddaljenostjo od zgornjega levega kota, kar je smiselno, saj bi bila drugače sešteta doza v kotu spodaj desno vedno manjša, tako pa so rešitve omogočile vsaj kritično vrednost sevanja po diagonali vmes. Vidimo, da površinske celice prejmejo kar precejšnje doze, kar pa spet ne krši pogojev, saj imamo le minimizacijo. Pri inverznem problemu smo omejili obsevanost zdravih celic na kritično. Vidimo, da imajo to vrednost vse tiste celice, ki imajo „za sabo“ tumor. To je edini način, da so rakaste celice maksimalno obsevane, kar pa ni najbolj optimalno, saj je obsevanost roba tumorja spodaj desno dosti pod kritično vrednostjo in to je maksimalna obsevanost.

5 Primer - Absorpcija

Nekoliko bolj realen problem imamo, če omogočimo obsevanje žarkov iz vseh smeri, torej po celem tkivu:



Sedaj smo prišli do najbolj realnih in optimalnih rezultatov. Z obsevanostjo iz vseh smeri omogočimo večjo obsevanost pri celicah na nasprotnih straneh, kar zniža jakosti na robovih, kot je bilo videti v prejšnjem primeru na levi sliki. tem primeru na levi sliki vidimo, da imajo nekatere zdrave celice nekoliko večjo obsevanost od kritične, vendar pa večina zdravih celic sploh ni prejela nobene doze, oziroma je ta bila zanemarljiva, med tem ko so vse rakaste celice prejele vsaj kritčno dozo. Na desni sliki je kar nekaj zdravih celic, ki so prejele kritično dozo, vendar nič več. Tudi ta primer je zelo optimalen, saj vse rakaste celice prejmejo vsaj kritično dozo, pri pogoju da zdrave celice prejmejo največ kritično.

Del II

Problem optimalne vožnje - Linearno programiranje

S to metodo se da rešiti tudi problem optimalne vožnje, ki nas je spremljal pri prvem poročilu. Glavna razlika pri tem načinu je seveda omejitev zaradi linearnosti, saj nam ta onemogoča minimizacijo kumulativnega kvadrata pospeška, ker vsebuje kvadratne člene hitrosti. Glavna prednost pri tem načinu je, da nam linearno programiranje omogoča, da imamo med rešitvami le pozitivne hitrosti, hkrati pa lahko hitrosti in pospeške tudi strogo omejimo. Ker torej ne moremo minimizirati kumulativnega kvadrata pospeška, zato maksimiziramo končno hitrost. Nalogo sem reševal tako, da sem območje in hitrost diskretiziral na $\mathbf{x} = \{x_1, x_2, ..., x_N\}$, in $\mathbf{y} = \{y_1, y_2, ..., y_N\}$. Za najbolj optimalno vrednost se je izkazala $N = 30$, tako da sem jo uporabil v vseh primerih. Kot pri prvem poročilu pa moramo spet upoštevati, da je prevožena pot enaka $l = 1$. To upoštevamo s pogojem:

$$\sum_{i=1}^{N} y_i \mathrm{d}t = l, \tag{4}$$

kjer je $\mathrm{d}t = \frac{l}{N y_0} = \frac{1}{N}$. Ponovno sem računal s funkcijo LinearProgramming$[c, m, b]$, kjer so ti parametri pri $N = 5$ bili takšni:

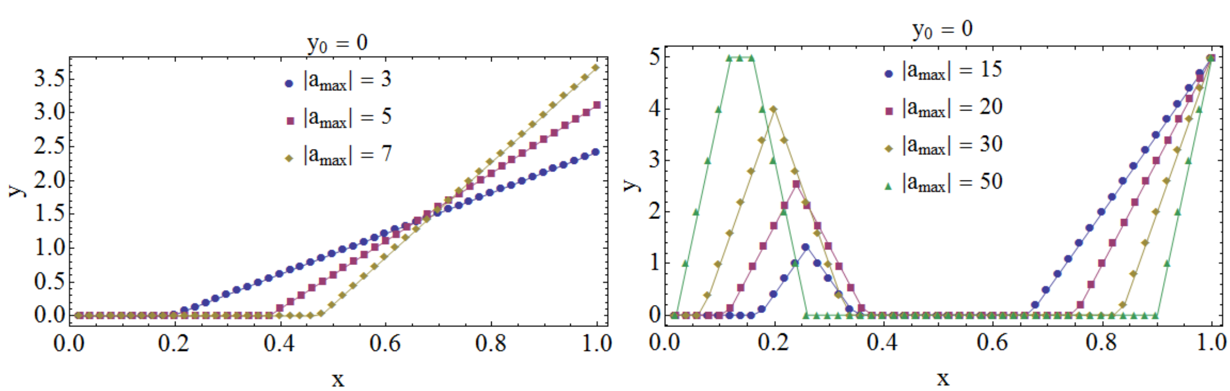
$$c = \{0, 0, 0, 0, -1\}, \qquad m = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & -1 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \tag{5}$$

$$b = \left\{ \{y_0, 0\}, \left\{ \frac{l}{\mathrm{d}t}, 0 \right\}, \{y_{max}, -1\}, ..., \{a_{max}, -1\}, ..., \{-a_{max}, 1\} \right\}, \tag{6}$$

c pa mora biti negativen, ker minimizacija $-v_N$ pomeni maksimizacijo v_N. Poglejmo si rezultate za različne začetne hitrosti y_0 pri različnih omejitvah pospeška in pri omejitvi hitrosti $v_{max} = 5$.

1 Primer

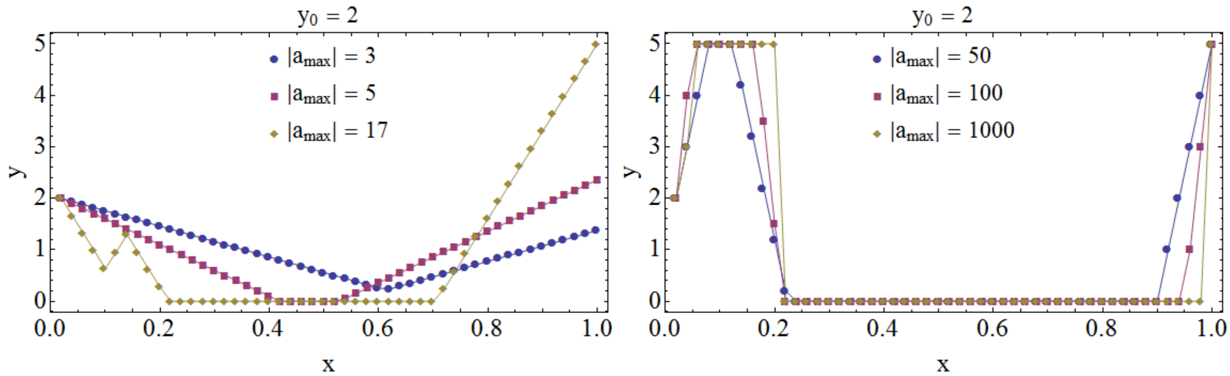
V tem razdelku si poglejmo primer, kjer imamo $y_0 = 0$ in omejitve pospeškov $a_{max} = \{3, 5, 7, 15, 20, 30, 50\}$:



Vidimo, da se pri majhnih omejitvah pospeška zgodi, da avto nekaj časa miruje, potem pa spelje in enakomerno pospešuje do konca, tako da z maksimalno hitrostjo prevozi zeleno luč na semaforju. Z dopuščanjem večjih pospeškov avto vedno dlje časa miruje, potem pa hitreje pospeši, kjer vedno ohranja enako količino prepotovane poti. Pri večjih pospeških je situacija nekoliko obratna, saj avto prej spelje, če dopuščamo večje pospeške, nato nekaj časa pospešuje, nato pa začne zavirati takoj, ali pa po nekaj trenutkih enakomerne vožnje, če bi hitrost sicer prekoračila omejitev. Tako zavira do hitrosti $y = 0$, nekaj časa miruje, potem pa spet enakomerno pospešuje to semaforja s čimvečjim možnim pospeškom, da je končna hitrost maksimalna. Vidimo, da rešitev vožnje, kot jo predlaga linearno programiranje, ni kaj prida gladka in tudi ne praktična.

2 Primer

Poglejmo si še primer kjer imamo začetno hitrost $y_0 = 2$ pri istih omejitvah pospeškov $a_{max} = \{3, 5, 7, 50, 100, 1000\}$:



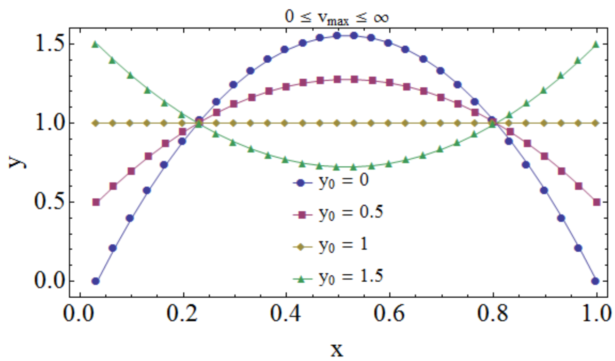
V tem primeru rešitve niso nič kaj veliko bogatejše kot v prejšnjem primeru. Tu ima avto neko začetno hitrost, zahtevamo pa iste pogoje kot prej, zato tudi dobimo podobne rešitve kot prej. Pri majhnih pospeških se končne hitrosti znižajo, saj na začetku opravi večji del poti. Pri omejitvah pospeška $|a_{max}| = 3$ in 5 avto zavira, se v enem primeru ustavi in nato začne pospeševati do semaforja, pri omejitvi $|a_{max}| = 17$ pa se zgodi čudna reč, saj opazimo „nagubanost“ hitrosti na začetku. Nagubanost nastopi zato, ker so pospeški že tako veliki, da bi omejitev na koncu odrezala končno hitrost in verjetno tako kak pogoj ne bi bil izpolnjen. Desna slika je praktično ista kot v prejšnjem primeru, le da sem tu povečal pospeške tako, da je omejitev hitrosti bolj očitna.

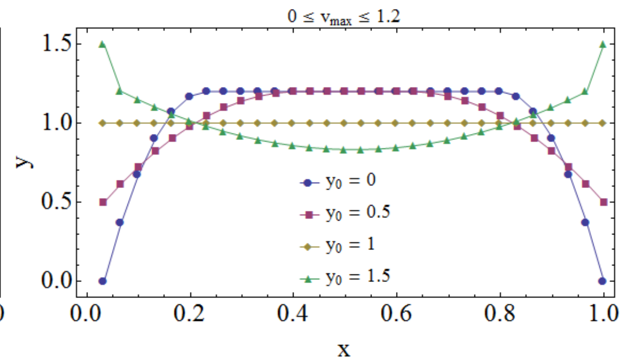
Del III
Problem optimalne vožnje - Numerično reševanje

V tem delu sem se lotil problema optimalne vožnje na malce drugačen način. Uporabil sem funkcijo NMinimize, ki numerično minimizira neko funkcijo pri danih pogojih. Glavna prednost te metode je, da je ne omejuje linearnost, kot je bil problem pri linearnem programiranju. To pomeni, da se lahko sedaj vrnemo k prvotni minimizaciji kumulativnega kvadrata pospeška, hkrati pa lahko še omejimo hitrosti, česar prej nismo mogli. Odločil sem se, da tu pospeškov ne bom omejeval, ker so objekt minimizacije. Ponovno sem razdelil območje in hitrost na diskretne vrednosti in dobil rešitve na podoben način kot je že bilo omenjeno v poročilu. V nobenem primeru nisem omejil začetne in končne hitrosti y_0 in y_N. Risal sem vedno po dve sliki, kjer so na levi sliki reproducirani podatki za prvo poročilo, na desni sliki pa so zraven omejitve hitrosti, kjer je omejitev hitrosti v vseh primerih enaka $y_{max} = 1.2$.

1 Periodični robni pogoj

Za periodičnost sem moral izpolniti robni pogoj $y_N = y_0$, kar sem narisal za več različnih začetnih hitrosti:

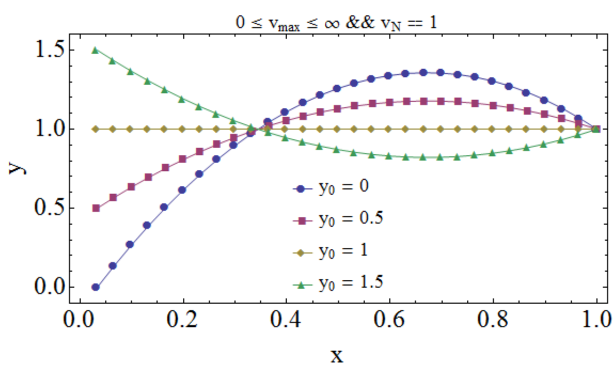


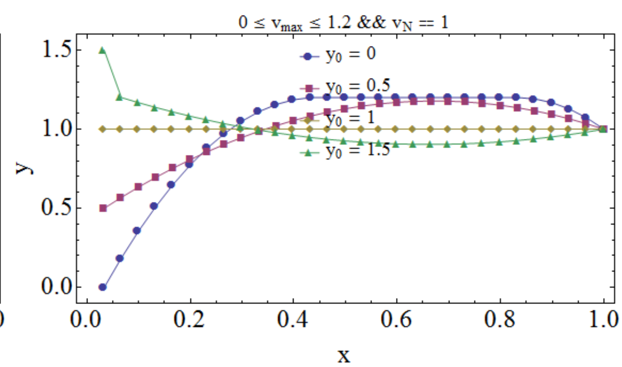


Če se bralec spomni, smo res reproducirali rezultate iz prve naloge. Na desni sliki vidimo kako bi potekala vožnja, če bi hitrost omejili na $y_{max} = 1.2$.Vidimo, da na nekatere funkcije omejitev vpliva tako, da jih splošči, kot je vidno za primer $y_0 = 0$, medtem ko se pri primeru z $y_0 = 1.5$ zgodi to, da hitrost hitro skoči nižje na omejitev, na koncu pa spet nazaj na isto vrednost kot na začetku. Takšno gibanje ni ravno optimalno, je pa vseeno nekoliko aplikativno v realnosti, kjer voznik na začetku naredi hiter negativen trzaj in na koncu pozitivnega.

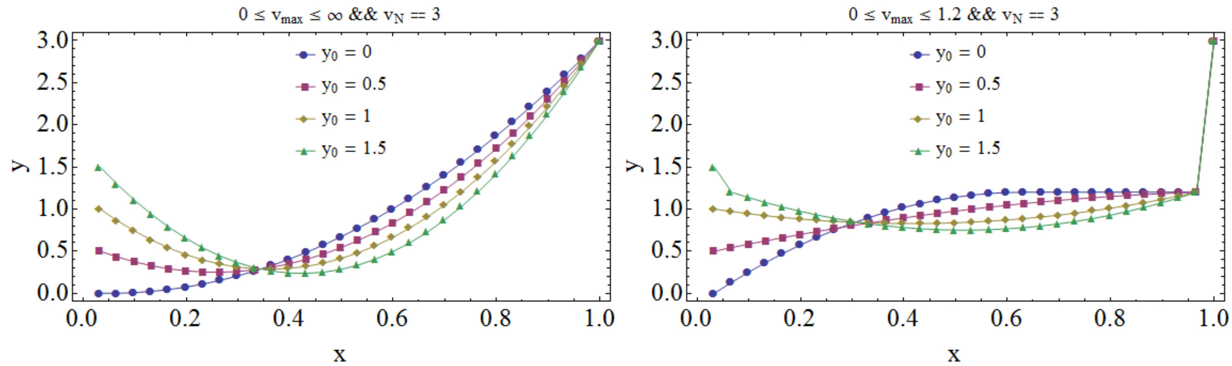
2 Fiksni robni pogoj

Za fiksen robni pogoj sem naredil primera z y_N =1 in $y_N = 3$, ponovno pa sem narisal rešitve za več različnih začetnih hitrosti. Rešitve za $y_N = 1$ izgledajo tako:





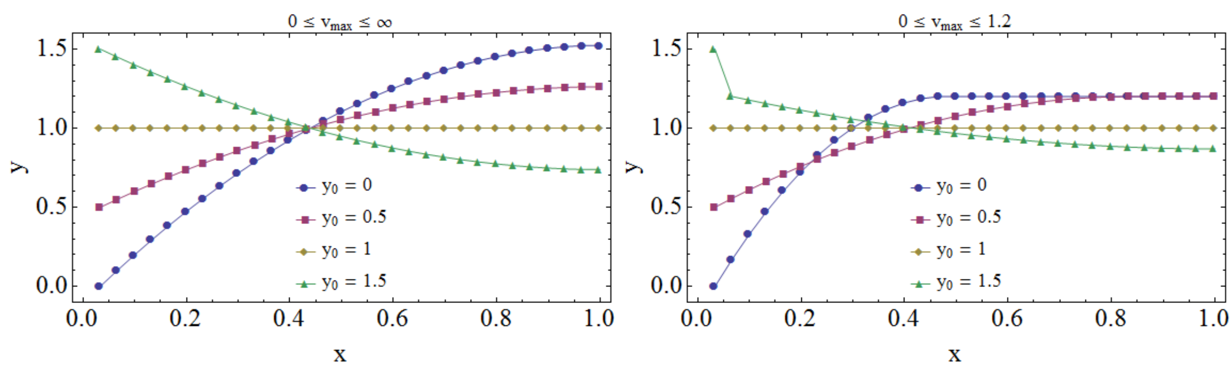
Leva slika je spet ista kot pri prvem poročilu, medtem ko je na desni sliki rešitev nekoliko sploščena, tako da ustreza omejitvam. Primer z najvišjo začetno hitrostjo spet vsebuje negativen trzaj na začetku vožnje, vse vožnje pa se nato nekako umirijo in končajo s končno hitrostjo. Poglejmo si še primer z $y_N = 3$:



Te rešitve so verjetno še najmajn optimalni in najmajn praktične, saj je situacija na začetku zelo podobna primeru z začetno hitrostjo $y_N = 1$, razlikuje se le na koncu, kjer imamo ogromen pospešek na končno hitrost, tako da ustreza pogoju dolžine poti in robnemu pogoju.

3 Dinamični robni pogoj

Pri dinamičnem robnem pogoju sem upošteval da je končni pospešek $a_N = 0$, kjer voznik prevozi zeleno luč brez pospeševanja:



Čeprav imamo tudi tu trzaj na začetku v primeru z najvišjo začetno hitrostjo, se zdi da so te rešitve najbolj optimalne in vožnja najbolj „gladka“, saj vse rešitve konvergirajo h konstanti hitrosti. Na tej točki opazimo, da če omejimo hitrost so si rešitve optimalne vožnje med seboj po vsebini precej podobne, razlikujejo se le na robovih zaradi drugačnih dobnih pogojev.