**ORODJE YACAS**

**Mathematical tool Yacas**

**Dimitar Stoilkov**

**dimitar.stoilkov@student.fmf.uni-lj.si**

**Povzetek**

Pri predmetu Računalniška orodja v matematiki si je vsak študent izbral različno orodje, ki nam bi znalo pomagati pri matematičnih nalogah, in ga predstavil na študentski konferenci ROM. Izbral sem si orodje YACAS. Program je brezplačen in zelo enostavno se ga namesti na katerikoli operacijski sistem. V svoji predstavitvi delovanja orodja sem se osredotočil na primere in naloge iz področja linearne algebre. Poleg algebre, program ima funkcije s katerimi lahko računamo tudi naloge iz ostalih področij matematike. Program ni priporočljiv za ljudi, ki so vizualnega tipa, je pa pisan na kožo programerjem, saj ne potrebujejo miške, ampak samo pišejo ukaze, kar je tudi posebnost tega programa. Na prvi pogled skromen program skriva v sebi veliko več, kot je mogoče opaziti.

**Ključne besede**

Algebra, vektor, matrika

**Abstract**

As a part of the first semesters' subject called Computer tools in mathematics, each student selected a random program, which could prove itself useful in solving mathematical problems, and presented it on the conference for students ROM. I selected a tool called Yacas. It is free and extremely easy to install on every computer operating system. In my presentation I focused on examples taken from linear algebra. In addition to solving algebra problems, the tool has many other options which help us in other fields of mathematics. People of the visual type, might not find this tool interesting, but programmers will love it, because the mouse is not needed. On the surface a program that looks very simple and dull, hides inside much more than one can see.

**Key words**

Algebra, vector, matrix

**Uvod**

Glavni namen prispevka je predstavitev orodja in opis njegovega delovanja. Pri opisu njegovega delovanja bom rešil 3 naloge iz področja linearne algebre, ki so bile predstavljene tudi na konferenci, ampak tokrat bolj obširno in podrobno. Naloge vsebujejo računanje lastne vrednosti in vektorje, inverz matrike in baze prostora. Na koncu bom orodje primerjal z že znanimi programi.

1. **Opis orodja**

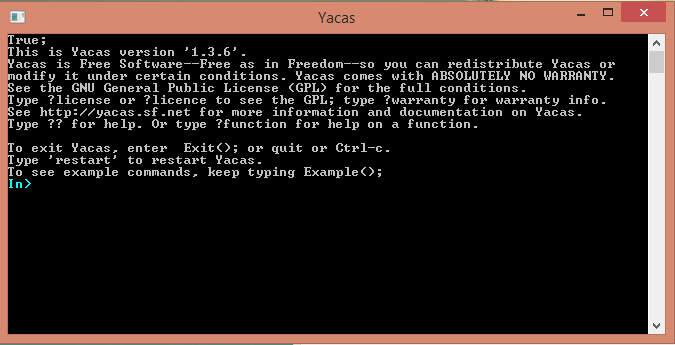
Yacas je računalniško orodje za simbolično manipulacijo matematičnih izrazov. YACAS je akronim za ''Yet Another Computer Algebra System'', kar tudi je, saj je programov, ki delujejo na področju matematike ogromno. Program je izdelalo podjetje Ayal Pinkus et al. leta 1999, kar pomeni da je program že nekoliko star, ampak se uspešno kosa z novejšimi programi. Program omogoča tudi programiranje, saj uporablja jezik C++, kar omogoča uporabo programerske sintakse, vse od zank *while* do pogojnih stavkov *if* in seznamov *list*. Ko se program zažene, se nam odpre okno v katerem vidimo terminal. Za razliko od programov, ki so bili predstavljeni na konferenci, program nima ne orodne vrstice, ne svetlikajočih se gumbkov na katere lahko klikamo, ampak dobimo samo okno s črnim ozadjem in dolgočasno sivo barvo pisavo. Ukaze oziroma funkcije najdemo na spletni strani, kjer je program dosegljiv, kjer odpremo zavihek *My Yacas*(v nadaljevanju opisan), ki nam pomaga pri delu s programom.

Prednosti programa:

* Brezplačen in enostaven za uporabo
* Omogoča programiranje
* Obsežna zbirka matematičnih funkcij
* Majhna poraba pomnilnika
* Spletna aplikacija za pomoč uporabnikom

Slabosti:

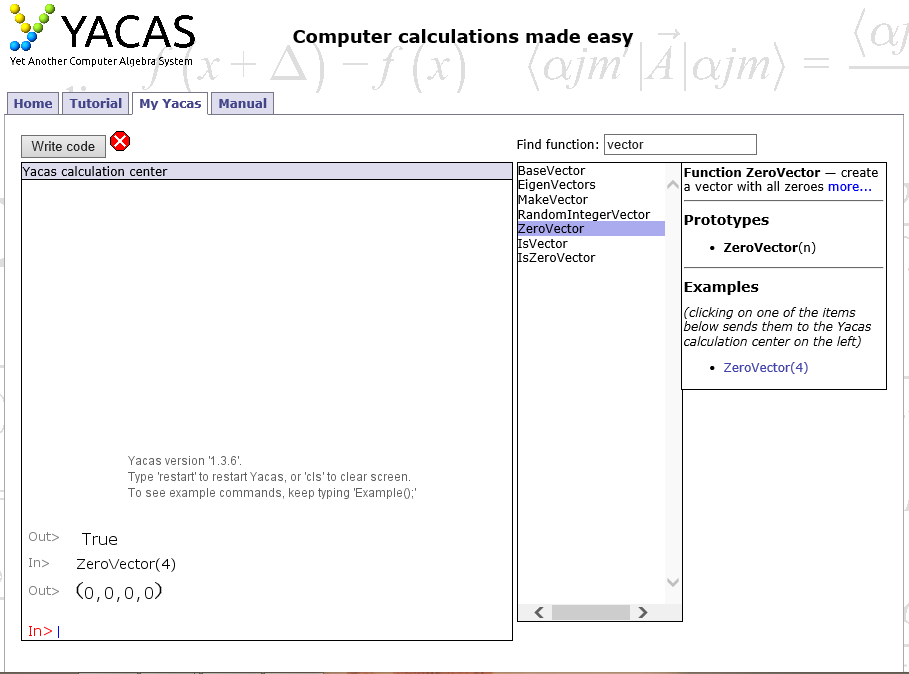
* Vizualno zelo skromen
* Velikokrat se zgodi, da se program sesuje
* Veliko malo se ga uporablja, zato ni razširjen po internetu



Slika : Okno, ki nas pričaka, ko se program odpre

1. **Yacas-ova aplikacija za pomoč uporabnikom**

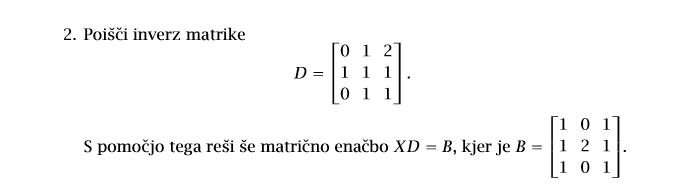
Pri YACAS-u so zelo potrudili, da bi delo z njihovim programom uporabnikom olajšali. Ko odpremo njihovo spletno stran <http://yacas.sourceforge.net/homepage.html>, opazimo, da imamo 4 zavihke: *Home, Tutorial, My Yacas* in *Manual.* Prvi zavihek je domača stran programa, kjer program lahko prenesemo na naš računalnik, ostale tri so namenjene izključno pomaganju uporabniku. Zavihek *Tutorial* nam pojasni kaj vse program lahko naredi, od predstavitve sintakse programa, definiranje funkcij pa vse do zelo kompleksnih izrazov. Drugi zavihek, že v prejšnjem odstavku omenjen *My Yacas*, je najbolj pomemben med vsemi zavihki na spletni strani. Poleg tega, da nam omogoča neposredno delo v terminalu, imamo polje *Find function:*, kjer vpišemo funkcijo ali samo košček besede, ki ga funkcija vsebuje, in nam izpiše vse funkcije povezane z našim vnosom. Ko se z miško približamo funkciji, nam na desni strani izpiše definicijo funkcije, kako se funkcija vnese v terminal in primer uporabe funkcije. Če hočemo takoj ugotoviti, kako funkcija deluje, lahko kliknemo na dan primer in nam ga takoj vnese v terminal. Če pa se hočemo igrati z funkcijo, lahko brez problemov vnašamo funkcijo kolikokrat želimo v terminal. Na koncu je še zavihek *Manual*, ki vsebuje datoteke, ki pa so namenjen tistim, ki bi hoteli še bolj raziskati program in njegovo delovanje. Tukaj najdemo dve poglavji, ki se ukvarjata s programiranjem v Yacasu, od osnovnega pa vse do naprednega.



Slika :Prikaz zavihka My Yacas

1. **Prikaz delovanja programa**

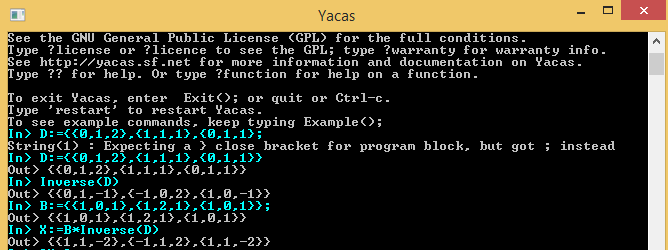
Da bi pokazal kako program deluje in kako poteka postopek reševanje nalog, sem si izbral 3 naloge, vsaka iz različnega področja linearne algebre. Pri vsaki nalogi bom napisal nekaj malega teorije in potem bom pokazal, kako se naloga rešuje z mojim programom. Naloge so pridobljene iz različnih kolokvijev pri predmetu Linearna algebra na Fakulteti za računalništvo in informatiko, Ljubljana.

* 1. Prva naloga

Slika 3:Prva naloga

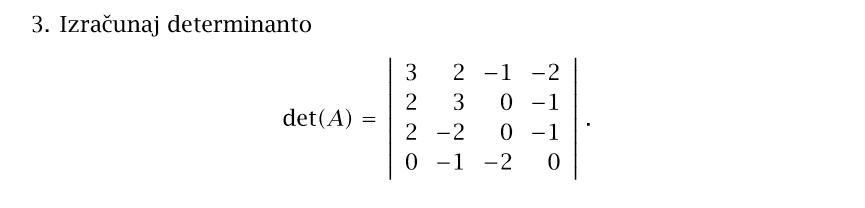
Kot lahko vidimo naloga od nas zahteva, da poiščemo inverzno matriko podane matrike D. To pa naredimo v ta namen, da lahko lažje rešimo enačbo, ki jo imamo spodaj. Enačbo množimo iz desne strani z inverzom matrike D in dobimo enačbo naše matrike X:

Da bi to nalogo lahko rešili v mojem programu, moramo najprej definirati matriko D. To naredimo tako, da vpišemo *D:={{},{},{}}*, kjer so v oklepajih številke posamezne vrstice matrike. Pokličemo funkcijo *Inverse(D)*, ki nam vrne inverz matrike D. Ko to enkrat storimo in definiramo, po istem postopku kot zgoraj, matriko B, lahko izračunamo naš X. Vpišemo v program *X:=B\*Inverse(D)* in dobimo rešitev za matriko X. Nalogo bi lahko tudi samo z enim klicem funkcije *Inverse* izračunali, ampak sem hotel bolj nazorno pokazati kako deluje funkcija.



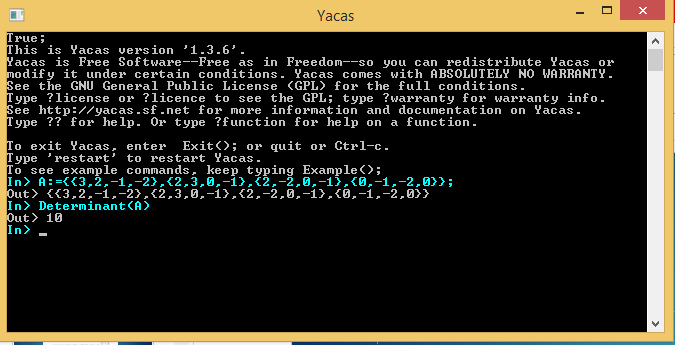
Slika 4:Rešitev naloge 1

* 1. Druga naloga



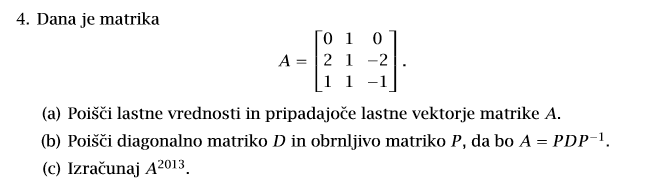
Slika : Druga naloga

Druga naloga od nas zahteva, da izračunamo determinanto matrike A. Načinov računanja determinantne matrik je veliko, kateri način si izberemo je odvisno od posameznika in od njegovega predznanja. V našem primeru je najlažje razviti matriko po kofaktorjih, tako da si izberemo vrstico ali stolpec, ki ima največ ničel in razvijamo po njem. Če pa smo preveč leni lahko odpremo Yacas, ki nam bo brez problemov in v zelo kratkem času izračunal determinanto. Tako kot v prejšnji nalogi definiramo matriko A. Ko je to končano, pokličemo funkcijo *Determinant(A)* in dobimo vrednost determinante. Pri tej matriki bi tudi, če bi se je lotili ročno reševati, porabili malo časa. Če pa dobimo matrike, ki niso tako lepe kot naša, ali pa so večje od naše, se z veseljem obrnemo na računalnik, ki nam bo takoj povedal odgovor.

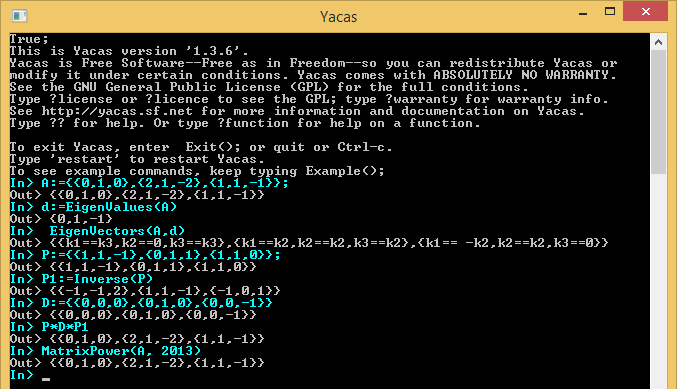


Slika :Rešitev druge naloge

* 1. Tretja naloga

Zadnja naloga je nekoliko zahtevnejša in je sestavljena iz treh delov. Časovna je naloga precej požrešna, saj nam lahko vzame tja od 30min pa do 1h, odvisno od predznanja, je pa velika verjetnost da bomo kje naredili napako, zaradi velikega števila računov, ki jih moramo narediti. Prvo, kar moramo narediti pri nalogi, je poiskati lastne vrednosti in lastne vektorje. Lastna vrednost je skalar , za katerega velja , kjer je lastni vektor matrike. Lastne vrednosti dobimo tako, da po diagonali odštejemo in izračunamo determinanto matrike. Ko dobimo vrednosti , jo vstavimo v prejšnjo matriko in dobimo lastne vektorje. Diagonalno matriko D tvorimo tako, da po diagonali postavimo vrednosti , na ostalih mestih so ničle. Obrnljiva matrika P pa je sestavljena iz lastnih vektorjev. Da bi vso to suhoparno teorijo pretvorili v prakso, definiramo matriko A. Lastne vrednosti bomo shranili v novi spremenljivki, zato zapišemo *d:=EigenValues(A)*. V drugem koraku pokličemo funkcijo *EigenVectors(A,d)*, ki nam bo vrnila lastne vektorje matrike A z lastno vrednostjo . Drugi del naloge je veliko lažji, lastne vrednosti postavimo v svojo matriko D, lastne vektorje pa v matriko P. Izračunamo inverz matrik P s funkcijo *Inverse(P)* in ko pomnožimo vse tri matrike skupaj dobimo nazaj A. Na konec še pokličemo funkcijo *MatrixPower(A, 2013)*, ki nam vrne nazaj matriko A. Z uporabo programa, smo čas, ki bi ga potrebovali, da ročno izračunamo to nalogo, občutno zmanjšali.

Slika 7:Tretja naloga



Slika :Rešitev tretje naloge

1. **Primerjava z drugimi orodji**

Od vseh orodij, ki so bili predstavljeni na konferenci, je Yacas na prvi pogled najbolj dolgočasno in neuporabno orodje. Ampak, ko posvetimo temu orodju malo več našega časa in ko ga dodobra spoznamo, ugotovimo da to sploh ni tako dolgočasno orodje kot smo pričakovali da bo. Ne moremo ga primerjati sicer z Mathematico in MatLabom, ki so plačljiva in se uporabljajo na univerzah in na srednjih šolah po celem svetu, lahko ga pa primerjamo z ostalimi orodji. Ena izmed prednosti, ki jih program ima, je ta da lahko shranjujemo spremenljivke in jih potem kličemo, kar je lažje kot da bi jih morali vsakič znova napisati. Druga prednost, v primerjavi z drugimi programi, je to, da ga je mogoče namestiti na vseh možnih operacijskih sistemih. Problem pa ja, ko spremenljivko z istim imenom definiramo z različnimi številkami, nas ne obvesti, da je ta spremenljivka že zasedena, ampak njene prejšnje vrednosti nadomesti z novimi. Tretja prednost oziroma slabost, odvisno od uporabnika, je preprostost tega programa. Za ljudi, ki so vizualnega tipa, ki jim je všeč da je graf funkcije lepo narisan, ali pa da je vektor napisan tako kot je treba(z oglatimi oklepaji), ta program ni prava izbira. Če pa se najde kdo, ki mu je všeč dejstvo, da je program skromen in brez nepotrebnih gumbov in funkcij, ki jih nikoli ne bo uporabil, je program ''zadetek v polno''. Kar je slabost programa, v primerjavi z drugimi, je to, da potrebujemo nekaj časa, da se naučimo kako se posamezna funkcija napiše in s katerimi argumenti jo moramo poklicati. Še ena slabost je to, da potrebujemo internetno povezavo. Če recimo potrebujemo ime neke funkcije, jo moramo poiskati na internetu in samo na internetu, kar je velik minus v primerjavi z drugimi programi, ki imajo navodila in opcijo *Help* že vgrajene v sam program in je vedno na razpolagi uporabniku. . V tem primeru si lahko pomagamo z znanjem, ki smo ga pridobili, če smo delali kdaj z MatLabom ali pa z Mathematico, saj so nekateri ukazi zelo podobni. Tako kot pri mnogih programih, lahko tudi tukaj spreminjamo velikost pisave in pa tudi barvo ne samo pisave, ampak tudi ozadja.

1. **Zaključek**

Na koncu bi strnil svoje misli in svoje ugotovitve, ki sem jih dobil pri delu s programom Yacas. Osebno mi je program zelo všeč, uporabljam ga pri reševanju domačih nalog in pri pripravah na kolokvije pri predmetu linearna algebra in sem zadovoljen z njegovim delovanjem. Vem pa da mnogim mojim sovrstnikom ali mlajšim generacijam, program ne bi bil všeč, kajti ni tako vizualno bogat kot nekateri programi, ki so prosto dostopni na internetu. Zato bi ga odsvetoval pri uporabi v srednjih in osnovnih šolah. Program je primeren za nekoga, ki bi rad začel uporabljati nov program, ker se je mogoče naveličal prejšnjega in ki ima dovolj prostega časa, da bi se usedel in bi se poglobil v program. Enostaven je za uporabo, ampak samo takrat, ko ga dobro spoznamo in se naučimo večino ukazov na pamet, ali pa jih imamo kje zapisano. Zasnovan je tako, da za vsak plus, vedno lahko najdemo minus. Mogoče edina stvar, ki me moti pri programu je ta, da to, kar počnemo v programu ni mogoče nikjer shraniti in potem nadaljevati. Potrebno je skozi delati posnetke zaslona in potem vse te slike sortirati in urejati, kar vzame čas, ki bi ga lahko porabili za kaj koristnega. Razen spletne strani od programa, ki je obsežna z informacijami in je zelo dobro urejena, je težko najti na internetu strani ali objave po različnih blogih, ki bi ponujali več informacij glede dela s programom. Tiste strani, ki pa vsebujejo informacije glede programa, so večinoma informacije dostopne na matični spletni strani programa. Ne glede na vse težave, na katere lahko naletimo, in ne glede na vse minuse in slabe lastnosti, ki jih program ima, ga bom še naprej uporabljal, saj vem da program, ki bi bil popoln, ne obstaja.

**Viri**

* <http://yacas.sourceforge.net/homepage.html> (ogled 12.4.2015)
* <http://en.wikipedia.org/wiki/Yacas> (ogled 12.4.2015)