**Scilab – Delo z matrikami**

**Scilab- Working with matrices**

**Uroš Ribič**

[**uros.ribic@student.fmf.uni-lj.si**](mailto:uros.ribic@student.fmf.uni-lj.si)

**Povzetek**

V programu Scilab si bomo pobližje ogledali delo z matrikami. Začeli bomo z najbolj osnovnimi stvarmi t.j. kako v programu ustvarimo matriko, kako dostopamo do posameznega elementa in kako lahko posamezni element spremenimo. V nadaljevanju si bomo pogledali še: kako lahko spreminjamo velikost matrike, kako s pomočjo matrik ustvarjamo vektorje in katere operacije lahko uporabljamo na matrikah. V sklopu naloge bomo rešili linearni sistem Ax = b s pomočjo LU razcepa. Za konec bomo pogledali še združljivost z MatLabom in katere so najbolj pogoste težave pri uvažanju skriptnih datotek.

**Ključne besede**

Scilab, matrike,vektorji,Matlab

**Abstract**

In Scilab, we will take a closer look at the work with matrices. We'll start with the most basic things e.g. how to create a matrix, how to access the individual elements and how each element can be changed. Further on we'll look at how can you change the size of the matrices, how to use matrices to create vectors and some operations which can be used on matrices. In the context of the tasks we solve the linear system Ax = b by using the LU decomposition. At the end we will look compatibility with Matlab and what are the most common problems when importing script files.

**Key words**

Scilab, matrices, vectors, Matlab

**Uvod**

Namen prispevka je, da si naučimo uporabo matrik v programu Scilab. Bralce želim seznaniti s prednostnimi in slabosti programa. Dotaknili se bomo tudi sorodnih programov (Matlab) in v katerih pogledih so si podobni in v katerih razlikujejo. Pogledali si bomo tudi kako je z družljivostjo s programom Matlab.

**Predstavitev orodja**

**1. Osnovne značilnosti orodja**

Scilab je odprtokoden (napisan v programskih jezikih: C, C++, Java, Fortran), tako da je koda programa na voljo vsakemu uporabniku. Je zelo dobra alternativa programoma Matlab in GNU Octave. Ukazi v Scilabu so podobni ukazom v Matlabu. Za razliko od programa Octave pa daje Scilab manj poudarka na dvosmerno skladanje združljivosti z Matlabom.

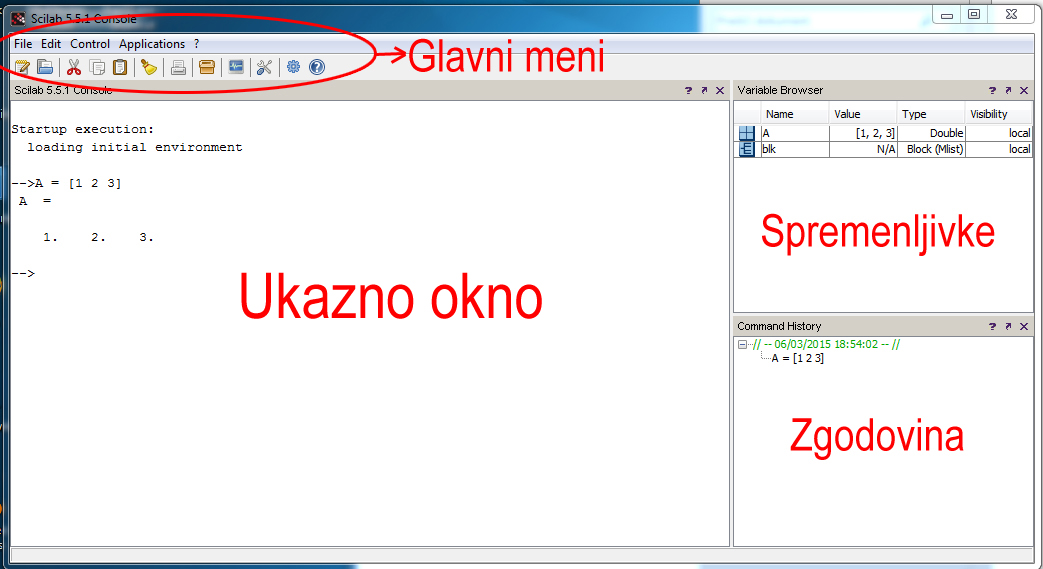
Scilab je brezplačen numerično usmerjen programski jezik. Na voljo je uporabnikom operacijskih sistemov: Linux, Mac OS X in Windows. Uporablja se ga lahko za delo z matrikami, ali pa za kakšne bolj kompliciranje stvari, kot so npr.: statistična analiza, simulacija dinamike tekočin, numerična optimizacija in modeliranje, simulacija eksplicitnih in implicitnih dinamičnih sistemov, simbolno manipuliranje itd.

Program je dostopen na spletni povezavi <http://www.scilab.org/download/5.5.1>

Na voljo sta tako 32-bitna različica kot tudi 64-bitna različica programa. Na zgoraj omejeni povezavi pa lahko dobite tudi celotno koda programa. Trenutno zadnja izdana verzija (preverjeno 6.3.2015) je Scilab 5.5.1.

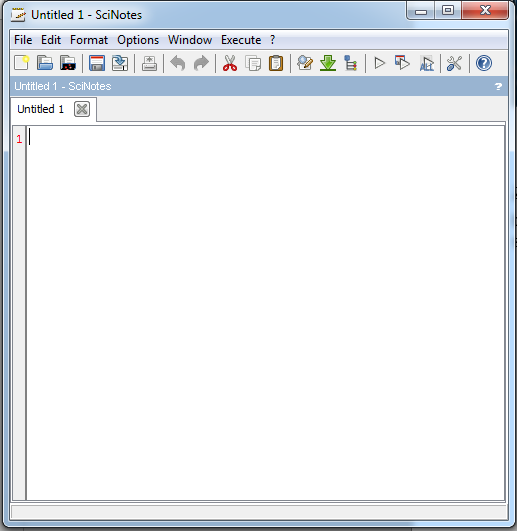
**2. Osnovni videz orodja**

Osnovni videz programa je sestavljen iz glavnega menija in treh oken (slika 1). Prvo okno je ukazno okno. To okno je najpomembnejše tu vpisujemo razne ukaze, računamo, definiramo razne spremenljivke itd.. Drugo okno je okno spremenljivk. V njem so spremenljivke, ki smo jih definirali v ukaznem oknu. Prikazuje ime spremenljivke, kakšna je njena vrednost, katerega tipa so itd. Zadnje tretje okno pa se imenuje zgodovina, tu se zapisuje zgodovina npr. kdaj smo ustvarili neko spremenljivko.



**Slika 1: Začetni videz**

Za pisanje bolj zahtevnih funkcij oz. skriptnih datotek uporabljamo t.i. Scilabovo beležko (Slika 2). Kliknemo na ikono v meniju . In odpre se nam novo okno v katerem zdaj lahko napišemo želeno funkcijo.

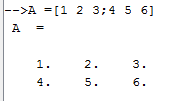


**Slika 2: Scilabova beležnica za pisanje skriptnih datotek**

**3. Ustvarjanje matrik**

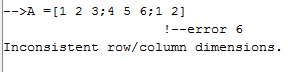
V tem prispevku se bomo osredotočili na delo v ukaznem oknu, tako da lahko za enkrat na Scilabovo beležnico pozabimo.

Matriko v Scilabu ustvarjamo v oglatih oklepajih (slika 3). Elemente med sabo ločimo s presledkom lahko pa tudi z vejico. Za decimalni zapis pa uporabimo piko (.). Elementi ločeni s presledkom (ali vejico) se bodo nahajali v isti vrstici. Za skok v novo vrstico pa uporabimo podpičje (;).



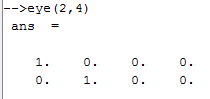
**Slika 3:Primer vnosa matrike**

Pri vnosu je pomembno, da je v vsaki vrstici enko elementov, sicer nam bo program vrnil napako (slika 4).



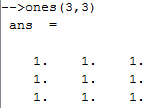
**Slika 4: Primer napake**

Identiteto ustvarjamo z ukazom eye (slika 5). Ukaz prejme dva argumenta. S prvim argumentom podamo število vrstic z drugim pa število stolpcev. Če želimo poiskati identiteto matrike A pa v argument napišemo njeno ime in program nam bo vrnil identiteto velikosti matrike A.

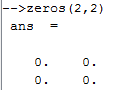


**Slika 5: Primer Identitete**

Podobno velja tudi za ukaza ones in zeros. Onse (slika 6) vrne matriko samih enk medtem, ko zeros vrne matriko samih ničel (slika 7).



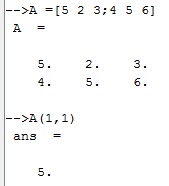
**Slika 6: Matrika samih enk**



**Slika 7: Matrika samih ničel**

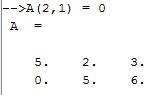
**4. Dostopanje do elementa**

Indeksiranje se začne z 1. Do elementa dostopamo z okroglimi oklepaji (slika 8). Znotraj oklepajev podamo dva argumenta. Prvi določa vrstico, drugi pa stolpec. Če stolpec oz. vrstica v dani matriki ne obstajata bo Scilab vrnil napako.



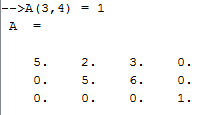
**Slika 8: Dostop do elementa**

Če želimo posamezni element spremenit poleg oklepajev dodamo še znak je enako (=) in zapišemo vrednost v katero želimo želeni element spremeniti (slika 9).



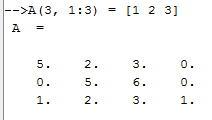
**Slika 9: Spreminjanje elementa**

V tem primeru če vrstica ali stolpec v trenutni matriki še ne obstajata bo program sam ustvari novo vrstico ali stolpec (slika 10). Na želeno mesto bo postavil vrednost elementa, preostala mesta pa bo zapolnil z ničlami.



**Slika 10: Ustvarjanje stolpca**

Na enkrat lahko dodajamo tudi več elementov (slika 11). Pri tem moramo biti pozorni, da se bo število mest ujemalo s številom novih elementov.



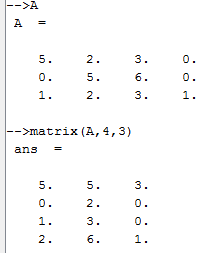
**Slika 11:Dodajanje več elementov na enkrat**

**5. Spreminjanje oblike matrik**

Pri spreminjanju oblike je pomembno, da je produkt število vrstic in število stolpcev pred spreminjanjem enak produktu po spreminjanju.

Ukaz za preoblikovanje matrike je matrix (slika 12). Ukaz prejme tri argumente pri čemer je prvi matrika, ki jo želimo spremeniti, drugi število vrstic, tretji število   
stolpcev.

Če je matrika večja si lahko pomagamo z ukazom size, ki izračuna velikosti matrike. Kot rezultat vrne število vrstic in število stolpcev.

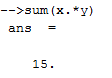


**Slika 12: Spreminjanje oblike**

**6. Vektorji**

Vektorje podamo kot stolpce. To storimo tako, da elemente ločimo s podpičji ali pa jih vnesemo kot vrstično matriko in jo kasneje transponiramo.

Dve najbolj pomembni operaciji, ki ju lahko uporabljamo na vektorjih sta skalarni produkt in vektorski produkt. Za slednega imamo že vgrajeno funkcijo cross medtem, ko skalarni produkt ne obstaja vgrajena funkcija. Skalarni produkt najlažje izračunamo tako, da vektorja zmnožimo po komponentah (to storimo, tako da pred operacijo postavimo piko) in nato te komponente seštejemo (to storimo z ukazom sum) (slika 13).



**Slika 13: Primer skalarnega produkta**

**7. Operacije na matrikah**

|  |  |
| --- | --- |
| ukaz | Opis |
| max (A) | največji element |
| min | najmanjši element |
| A\*B | navadno matrično množenje |
| tril(A) | spodnje trikotna matrika |
| triu(A) | zgornje trikotna matrika |
| lu(A) | LU razcep |
| inv(A) | inverz matrike |
| A' | transponiranje matrike |
| rank(A) | rang matrike |
| spec(A) | lastne vrednosti |
| det(A) | determinanta |
| trace(A) | sled matrike A |
| poly(A,'x') | karakteristični polinom matrike A, x je spremenljivka in mora biti napisana kot niz |
| roots(p) | ničle polinoma p |
| diag(A) | diagonalna matrika |
| A\b | reševanje linearnega sistema Ax = b |

**8. Primer uporabe**

Ogledali si bomo dva primera uporabe Scilaba pri reševanju naloge.

**1. Primer:**

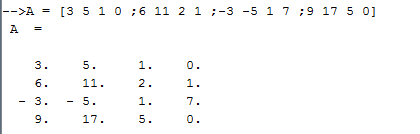
Podana je matrika A in vektor b.

A = ,b =

Izračunaj LU razcep matrike A in rešite sistem Ax = b.

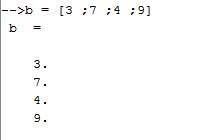
**Reševanje naloge:**

Najprej podamo matriko A (Slika 14).



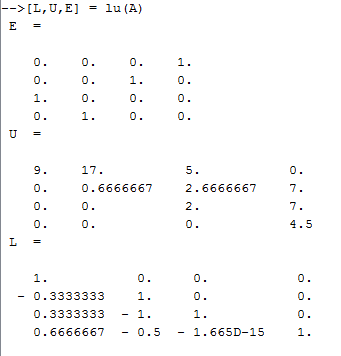
**Slika 14: Vnos matrike A**

Sedaj pa vnesemo še vektor b (slika 15).



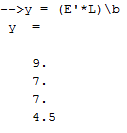
**Slika 15: Vnos vektorja b**

Ko definiramo matriko A in vektor b se lotimo LU razcepa (slika 16). Pri LU razcepu dobimo tri matrike. Prva matrika L je zgornje trikotna matrika. Druga matrika U je spodnje trikotna. Tretja matrika E pa je permutacijska matrika.



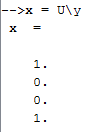
**Slika 16: LU razcep**

V naslednjem koraku bomo poiskali novo spremenljivko y (Slika 17). To storimo tako, da matriko E transponiramo in jo pomnožimo z matriko L. Vse skupaj pa delimo (računamo linearni sistem) z b.



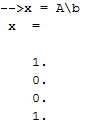
**Slika 17: Izračunamo novo spremenljivko y**

Zdaj, ko imamo izračunan y, lahko poiščemo neznani vektor x. Matriko U delimo z vektorjem y in dobimo rešitev za x.



**Slika 18: Izračun vektorja X**

Lahko preverimo še če je naša rešitev (slika 18) pravilna , tako da linearni sistem rešimo še brez LU razcepa. Torej A delimo z b (slika 19).



**Slika 19**

Prednost LU razcepa je, da je pri reševanju sistema Ax = b veliko manj operacij. Posledično računalnik veliko hitreje izračuna rešitev, če je matrika velika je razlika opazna.

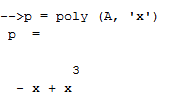
**2. Primer**

Podana je matrika A = , izračunaj karakteristični polinom in njene lastne vrednosti.

**Reševanje naloge:**

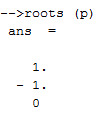
Začeli bomo z vnosom matrike A (Slika 14).

V naslednjem koraku lahko izračunamo karakteristični polinom to naredimo z ukazom poly (slika 20). Ukaz poly sprejme dva argumenta. Prvi je matrika v kateri želimo izračunati karakteristični polinom. Drugi argument pa je spremenljivka karakterističnega polinoma. Pri drugem argumentu je pomembno, da ga podamo kot niz.



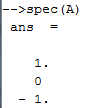
Slika 20

Lastne vrednosti lahko zdaj izračunamo na dva načina. Prvi način je da izračunamo ničle polinoma, to naredimo z ukazom roots, ali pa uporabimo ukaz spec. Mi bomo nalogo rešili na oba načina. Najprej izračunamo lastne vrednosti na prvi način (slika 21).



Slika 21

Da se prepričam, da je rešitev zares pravila storimo to še na drugi način (slika 22). Ukaz spec prejme za argument matriko v kateri želimo poiskati lastne vrednosti.



Slika

Vidimo, da dobim enake rezultate le da je vrstni red drugačen.

**9. Združljivost z Matlabom**

Scilabove skriptne datoteka lahko odpremo tudi v Matlabu in obratno. Najbolj pogosta težava pri pravilnem delovanju skriptnih datotekah pa lahko nastane že pri komentiranju. Matlab za komentiranje uporablja % med tem, ko Scilab uporablja za komentiranje //. Težavo lahko rešimo tako, da pred uvažanjem, iz enega v drug programa, pobrišemo vse komentarje ali pa znake za komentiranje spremenimo.

Ena pogostih težav je tudi, da ima Matlab več vgrajenih funkcij. Sicer to še ne pomeni, da nekega problema zdaj ne moremo rešiti v Scilabu. To pomeni le, da je v Scilabu potrebno poiskati kakšen drug pristop in narediti kak korak več (lahko še vedno obstaja tudi možnost, da se problema ne da rešiti). Zaradi tega je velika večja možnost, da bodo Scilabove skriptne datoteke pravilno delovale v Matlabu, kot pa Matlabove skriptne datoteka v Scilabu.

**Mnenja drugih uporabnikov**

Mnenje uporabnika Petra (2008):

»I'm a Scilab user: my opinion is that this program is very close to Matlab. There are minor differences in the name of some functions, and some basic commands, and of course scilab doesn't have all the functions from Matlab but I believe that after you learn Scilab you can easily adapt to Matlab. From my experience : after writing a long script in Matlab I've discover that the company to whom the script was addressed didn't had Matlab. They suggested me 2 alternatives : Excel or Scilab. So what I did was to run the Matlab script in Scilab (with the .sci extension added). For my surprise, only 3 minor error appeared.

1. the commentaries in scilab begin with // while in Matlab are with %
2. at the beginig of the script a data file data.m was loaded with the command "run data.m" in Matlab. In Scilab it became exec('data.sci')
3. some built in constants like PI in matlab are used like %PI in Scilab.

There are of course other differences, like the fact that if you write your own functions in Scilab you have to load them with the command getf("") while in Matlab you don't need to load the functions scripts (they must be in the matlab search path of course or the curent directory).

If you want something similar to Simulink you could use Scilab alternative - Scicos. Most of my colleagues consider Scilab as a open source alternative to Matlab (not the only one of course).

Finally, I recommend you to visit the home site of Scilab to view the available extra documentation. «

Z uporabnikom se strinjam. Program je zelo blizu Matlabu in če v njem pridobiš veliko znanja, se s tem znanjem lahko hitro navadiš tudi v Matlabu. Mogoče v obratno smer malce bolj dolgotrajno.

Uporabnik amca01 (2014)

»Scilab is at version 5.4.1, and is a mature product. I have come across a few comments in newgroups to the effect that Scilab is better than Octave for some advanced numerical work. However, I’m not sure if these are just opinions or are backed up with solid data and timings. I used to like Scilab a great deal, but now I don’t see that it offers anything over Octave other than Xcos (a dynamic systems modeller, similar to Simulink), and a few niche extras, such as solutions of boundary value problems.«

Sam programa Ocave in Scilab nisem še toliko spoznal, da bi lahko povedal kateri od njiju je bolši. Se mi pa zdi, da je Ocave bistveno bolj razširjen in popularen. Zato se več uporabnikov odloči zanj medtem, ko verjetno programa Scilab niso niti testirali.

Uporabnik in\_flu\_ence (2009)

»I personally prefer octave over scilab. I normally have a lot of matlab files from previous workers and converting them into scilab using the built-in convertor normally will cause minor errors. Whereas I can just run the .m files as it is in octave. Far more convenient for me in the sense and octave tends to work perfectly fine in all the previous matlab codes written. It maybe just me for using only a limited set of functions.

I manage to do loops, sort data and plot the sorted data in gnuplot + saving the files in csv. «

Pomanjkljivost programa Scilab je zagotovo združljivost z Matlabom. Tudi sam sem naletel na težave pri uvažanju in to že pri bolj preprostih funkcij. Med tem ko v Octave-u teh težav ni bilo.

**Zaključek**

Splošni vtis o programu je sicer pozitiven. Moti pa pred vsem dejstvo, da program nima toliko vgrajenih funkcij kakor Matlab. Včasih se zatakne že pri najpreprostejših funkcijah npr. skalarni produkt. Še ena od pomanjkljivosti programa je tudi, da ni deležen veliko novih posodobitev.

Zelo pozitivna stvar programa je predvsem to, da je na voljo brezplačno. Če ga primerjamo z Matlabom pa program zasede veliko manj prostora na računalniku in je bistveno hitrejši pri začetnem odpiranju. Videz programa je tudi dokaj podoben Matlabovimu, tako da večjih težav s prilagajanjem nanj ni.

Na domači spletni strani programa lahko najdemo tudi dokumentacijo program in veliko število rešenih primerov in vodičev. Lahko pa si, podobno kot v Matlabu ,pomagamo z ukazom help tako, da v ukazno vrstico zraven napišemo še ukaz, ki ga želimo uporabiti pa ne vemo kako. Program nam bo pomagal s knjižnico, kjer je ukaz opisan in kjer bomo našli tudi primere uporabe.

Poleg matrik, ga lahko uporabljamo tudi za računanje integralov, odvodov, fourierove vrste in še kaj bi se našlo. Zato je program primeren za fakultete.

**Viri:**

1. <http://www.scilab.org/> (ogled 6.3.2015)
2. <http://help.scilab.org/> (ogled 7.3.2015)
3. <http://en.wikipedia.org/wiki/Scilab> (ogled 7.3.2015)
4. <https://amca01.wordpress.com/2014/01/05/the-best-matlab-alternative-3/> (ogled 6.3.2015)
5. <http://ubuntuforums.org/showthread.php?t=594737> (ogled 6.3.2015)