Predavanje 6 - Simbolička izračunavanja

Tatjana Tomović

Institut za matematiku i informatiku
Prirodno-matematički fakultet
Univerzitet u Kragujevcu

Kragujevac, 2014.

Simbolička izračunavanja

- Simbolička algebra podrazumeva je razvoj algoritama koji su u stanju da rešavaju probleme simboličkog izračunavanja, tj. koji su u stanju da manipulišu matematičkim izrazima tretirajući promenljive uopšteno, ne podrazumevajući nikakve posebne vrednosti, niti posebne tipove podataka.

Simbolička izračunavanja

- Simbolička algebra podrazumeva je razvoj algoritama koji su u stanju da rešavaju probleme simboličkog izračunavanja, tj. koji su u stanju da manipulišu matematičkim izrazima tretirajući promenljive uopšteno, ne podrazumevajući nikakve posebne vrednosti, niti posebne tipove podataka.
- Koristeći MuPAD, Matlab je u mogućnosti da vrši diferenciranje, integraciju, simplifikaciju, transformacije i rešavanje jednačina u simboličkom obliku.

Simbolička izračunavanja

- Simbolička algebra podrazumeva je razvoj algoritama koji su u stanju da rešavaju probleme simboličkog izračunavanja, tj. koji su u stanju da manipulišu matematičkim izrazima tretirajući promenljive uopšteno, ne podrazumevajući nikakve posebne vrednosti, niti posebne tipove podataka.
- Koristeći MuPAD, Matlab je u mogućnosti da vrši diferenciranje, integraciju, simplifikaciju, transformacije i rešavanje jednačina u simboličkom obliku.
- Izvršenje naredbe *mupadwelcome* pokreće notebook interface MuPADa, koji omogućava direktnu komunikaciju korisnika i MuPADa bez posredovanja Matlaba.

Simbolička izračunavanja

- Simbolička algebra podrazumeva je razvoj algoritama koji su u stanju da rešavaju probleme simboličkog izračunavanja, tj. koji su u stanju da manipulišu matematičkim izrazima tretirajući promenljive uopšteno, ne podrazumevajući nikakve posebne vrednosti, niti posebne tipove podataka.
- Koristeći MuPAD, Matlab je u mogućnosti da vrši diferenciranje, integraciju, simplifikaciju, transformacije i rešavanje jednačina u simboličkom obliku.
- Izvršenje naredbe *mupadwelcome* pokreće notebook interface MuPADa, koji omogućava direktnu komunikaciju korisnika i MuPADa bez posredovanja Matlaba.
- Naredbom *mupad*('*ime_fajla*') se pokreće notebook interface sa otvaranjem novog ili već postojećeg fajla.

Simboličke promenljive i izrazi

- Koristeći funkciju *sym* moguće je kreirati simboličke promenljive i izraze. Funkcija *syms* kreira više simboličkih promenljivih istovremeno.

Simboličke promenljive i izrazi

- Koristeći funkciju *sym* moguće je kreirati simboličke promenljive i izraze. Funkcija *syms* kreira više simboličkih promenljivih istovremeno.

```
>> x = sym('x');                    >> whos x
>> y = x*x/x + x + 1/2              >> whos y
>> x = 1;                           >> whos x
>> syms x t
```

Simboličke promenljive i izrazi

- Koristeći funkciju *sym* moguće je kreirati simboličke promenljive i izraze. Funkcija *syms* kreira više simboličkih promenljivih istovremeno.

```
>> x = sym('x');          >> whos x
>> y = x*x/x+x+1/2        >> whos y
>> x = 1;                 >> whos x
>> syms x t
```

Dodela numeričke vrednosti simboličkoj promenljivoj:

Simboličke promenljive i izrazi

- Koristeći funkciju *sym* moguće je kreirati simboličke promenljive i izraze. Funkcija *syms* kreira više simboličkih promenljivih istovremeno.

```
>> x = sym('x');          >> whos x
>> y = x*x/x+x+1/2        >> whos y
>> x = 1;                 >> whos x
>> syms x t
```

Dodela numeričke vrednosti simboličkoj promenljivoj:

```
>> gr = sym('(1+sqrt(5))/2')   >> whos gr
```

Simboličke promenljive i izrazi

- Koristeći funkciju *sym* moguće je kreirati simboličke promenljive i izraze. Funkcija *syms* kreira više simboličkih promenljivih istovremeno.

```
>> x = sym('x');          >> whos x
>> y = x*x/x+x+1/2        >> whos y
>> x = 1;                 >> whos x
>> syms x t
```

Dodela numeričke vrednosti simboličkoj promenljivoj:

```
>> gr = sym('(1+sqrt(5))/2')   >> whos gr
```

Od izuzetnog je značaja da argument funkcije sym bude tipa string.

Promenljive sa specifičnim vrednostima

- Moguće je kreirati simboličke promenljive čije vrednosti pripadaju samo određenim skupovima brojeva.

Promenljive sa specifičnim vrednostima

- Moguće je kreirati simboličke promenljive čije vrednosti pripadaju samo određenim skupovima brojeva.
- Naredba *syms x tip* je skraćeni zapis za *x=sym('x','tip')*.

Promenljive sa specifičnim vrednostima

- Moguće je kreirati simboličke promenljive čije vrednosti pripadaju samo određenim skupovima brojeva.
- Naredba *syms x tip* je skraćeni zapis za *x=sym('x','tip')*.
- Da bi se skup mogućih vrednosti koje može uzimati promenljiva proširio na skup kompleksnih brojeva koristi se naredba *syms x clear* ili *sym('x','clear')*.

```
>> syms x y real
>> syms x positive
>> y = sym('1 + sqrt(2)')
>> syms y clear
```

Kreiranje matrice simboličkih promenljivih

- Moguće je koristiti već postojeće simboličke promenljive i pomoću njih kreirati matrice.

Kreiranje matrice simboličkih promenljivih

- Moguće je koristiti već postojeće simboličke promenljive i pomoću njih kreirati matrice.
- U drugom slučaju elementi matrice ne postoje unutar adresnog prostora Matlaba. Ako nam je referenca unutar adresnog prostora Matlaba neophodna možemo je kreirati.

Kreiranje matrice simboličkih promenljivih

- Moguće je koristiti već postojeće simboličke promenljive i pomoću njih kreirati matrice.
- U drugom slučaju elementi matrice ne postoje unutar adresnog prostora Matlaba. Ako nam je referenca unutar adresnog prostora Matlaba neophodna možemo je kreirati.

```
>> syms a11 a12 a13 a21 a22 a23 a31 a32 a33
>> A = [a11 a12 a13; a21 a22 a23;  a31 a32 a33]
>> A = sym('A', [3, 2])
>> B11 = sym('B11');
```

Kreiranje matrice simboličkih promenljivih

- Sa matricama čiji su elementi simboličke promenljive moguće je vršiti aritmetičke operacije.

Kreiranje matrice simboličkih promenljivih

- Sa matricama čiji su elementi simboličke promenljive moguće je vršiti aritmetičke operacije.
- Koristeći funkciju *sym* moguće je kreirati matrice koristeći operator konstrukcije [], na potpuno isti način kao u Matlabu.

Kreiranje matrice simboličkih promenljivih

- Sa matricama čiji su elementi simboličke promenljive moguće je vršiti aritmetičke operacije.
- Koristeći funkciju *sym* moguće je kreirati matrice koristeći operator konstrukcije [], na potpuno isti način kao u Matlabu.

```
>> A = sym('A', [2, 2]);
>> B = sym('B', [2, 2]);
>> A + B
>> A = sym('[1 2 3; 4 5 6; a b c]')
```

Funkcija sym sa numeričkim argumentom

- Funkcija *sym*, primenjena na numerički argument, u najopštijem slučaju ima sledeći format $sym(num, flag)$, gde je *num* podatak tipa *numeric*, a *flag* predstavlja opciju koja određuje način konverzije numeričke u odgovarajuću racionalnu vrednost.

Funkcija sym sa numeričkim argumentom

- Funkcija *sym*, primenjena na numerički argument, u najopštijem slučaju ima sledeći format $sym(num, flag)$, gde je *num* podatak tipa *numeric*, a *flag* predstavlja opciju koja određuje način konverzije numeričke u odgovarajuću racionalnu vrednost.
- Podrazumevana vrednost za drugi argument je 'r', a sve moguće vrednosti obuhvataju 'f', 'r','e', 'd'.

Funkcija sym sa numeričkim argumentom

- Funkcija *sym*, primenjena na numerički argument, u najopštijem slučaju ima sledeći format $sym(num, flag)$, gde je *num* podatak tipa *numeric*, a *flag* predstavlja opciju koja određuje način konverzije numeričke u odgovarajuću racionalnu vrednost.
- Podrazumevana vrednost za drugi argument je 'r', a sve moguće vrednosti obuhvataju 'f', 'r','e', 'd'.
- Vrednost 'f' znači da se prvi argument konvertuje u racionalan broj oblika $N * 2^e$, gde su N i e celobrojne vrednosti, koje su tako odabrane da se prvi argument funkcije *sym* i rezultat funkcije *sym* ne razlikuju u double formatu.

Funkcija sym sa numeričkim argumentom

- Funkcija *sym*, primenjena na numerički argument, u najopštijem slučaju ima sledeći format $sym(num, flag)$, gde je *num* podatak tipa *numeric*, a *flag* predstavlja opciju koja određuje način konverzije numeričke u odgovarajuću racionalnu vrednost.
- Podrazumevana vrednost za drugi argument je 'r', a sve moguće vrednosti obuhvataju 'f', 'r','e', 'd'.
- Vrednost 'f' znači da se prvi argument konvertuje u racionalan broj oblika $N * 2^e$, gde su N i e celobrojne vrednosti, koje su tako odabrane da se prvi argument funkcije *sym* i rezultat funkcije *sym* ne razlikuju u double formatu.
- Vrednost drugog argumenta postavljena na 'r', omogućava egzaktno konvertovanje numeričkih vrednosti izraza tipa p/q, $p\pi/q$, $\sqrt{p}$, 2^p i 10^p iz double formata u odgovarajuću egzaktnu vrednost.

Funkcija sym sa numeričkim argumentom

- Ako drugi argument u pozivu funkcije *sym* postavimo na vrednost e, dobijamo rezultat koji je isti kao i u slučaju sa drugim argumentom postavljenim na f, izuzev što dobijamo i informaciju o apsolutnoj grešci izraženoj korišćenjem *eps*.

Funkcija sym sa numeričkim argumentom

- Ako drugi argument u pozivu funkcije *sym* postavimo na vrednost e, dobijamo rezultat koji je isti kao i u slučaju sa drugim argumentom postavljenim na f, izuzev što dobijamo i informaciju o apsolutnoj grešci izraženoj korišćenjem *eps*.
- Ako je drugi argument u pozivu funkcije *sym* postavljen na vrednost d, onda se vrši konverzija u broj koji ima onoliko dekadnih cifara kolika je vrednost promenljive *digits*.

Funkcija sym sa numeričkim argumentom

- Ako drugi argument u pozivu funkcije *sym* postavimo na vrednost e, dobijamo rezultat koji je isti kao i u slučaju sa drugim argumentom postavljenim na f, izuzev što dobijamo i informaciju o apsolutnoj grešci izraženoj korišćenjem *eps*.
- Ako je drugi argument u pozivu funkcije *sym* postavljen na vrednost d, onda se vrši konverzija u broj koji ima onoliko dekadnih cifara kolika je vrednost promenljive *digits*.

$>> b = sym(1/3,'f')$
$>> 1/3 - 6004799503160661/18014398509481984$
$>> b = sym(1/3,'r'), \quad >> b = sym(1/3,'e')$

Pojednostavljivanje izraza

- MuPAD ne pokušava automatski da primeni sve identitete kojima raspolaže da uprosti izraze, već se mora koristiti funkcija *simplify* ili funkcija *simple*.

Pojednostavljivanje izraza

- MuPAD ne pokušava automatski da primeni sve identitete kojima raspolaže da uprosti izraze, već se mora koristiti funkcija *simplify* ili funkcija *simple*.
- Funkcija *simplify* može biti pozvana i sa dva argumenta. Drugi argument je broj koji ograničava broj primena identiteta prilikom pronalaženja najjednostavnijeg izraza. Podrazumevana vrednost je 100 primena.

Pojednostavljivanje izraza

- MuPAD ne pokušava automatski da primeni sve identitete kojima raspolaže da uprosti izraze, već se mora koristiti funkcija *simplify* ili funkcija *simple*.
- Funkcija *simplify* može biti pozvana i sa dva argumenta. Drugi argument je broj koji ograničava broj primena identiteta prilikom pronalaženja najjednostavnijeg izraza. Podrazumevana vrednost je 100 primena.

```
>> x = sym('(1 + sqrt(5))/2');
>> simplify(x^2 - x - 1)
>> simple(sin(2 * asin(x)))
```

Racionalni izrazi

- Pisanje polinoma u razvijenom obliku postiže se funkcijom *expand*.

Racionalni izrazi

- Pisanje polinoma u razvijenom obliku postiže se funkcijom *expand*.
- Faktorizacija polinoma se postiže upotrebom funkcije *factor*.

Racionalni izrazi

- Pisanje polinoma u razvijenom obliku postiže se funkcijom *expand*.
- Faktorizacija polinoma se postiže upotrebom funkcije *factor*.
- Funkcija *collect* vraća polinom zapisan u sređenom obliku po nekoj promenljivoj.

Racionalni izrazi

- Pisanje polinoma u razvijenom obliku postiže se funkcijom *expand*.
- Faktorizacija polinoma se postiže upotrebom funkcije *factor*.
- Funkcija *collect* vraća polinom zapisan u sređenom obliku po nekoj promenljivoj.
- Funkcija *coeffs* vraća vektor koeficijenata polinoma po zadatoj promenljivoj.

Racionalni izrazi

- Pisanje polinoma u razvijenom obliku postiže se funkcijom *expand*.
- Faktorizacija polinoma se postiže upotrebom funkcije *factor*.
- Funkcija *collect* vraća polinom zapisan u sređenom obliku po nekoj promenljivoj.
- Funkcija *coeffs* vraća vektor koeficijenata polinoma po zadatoj promenljivoj.
- Funkcija *numden* vraća brojilac i imenilac razlomka nekog simboličkog izraza.

Racionalni izrazi

- Pisanje polinoma u razvijenom obliku postiže se funkcijom *expand*.
- Faktorizacija polinoma se postiže upotrebom funkcije *factor*.
- Funkcija *collect* vraća polinom zapisan u sređenom obliku po nekoj promenljivoj.
- Funkcija *coeffs* vraća vektor koeficijenata polinoma po zadatoj promenljivoj.
- Funkcija *numden* vraća brojilac i imenilac razlomka nekog simboličkog izraza.

$>>$ *syms x* $\quad >> f = (x^2 + x + 1) * (x^3 + x + 2) + 3 * x + 4$
$>>$ *expand*(f) $\quad >>$ *factor*$(poly)$ $\quad >>$ *syms x a b clear*
$>> p = (a * x + 1) * (1 + b * x^2 - a * x) + x; \quad >>$ *collect*(p, x)

Zamene

- Ukoliko želimo moguće je zameniti neku promenljivu u simboličkom izrazu nekom konkretnom vrednošću.

Zamene

- Ukoliko želimo moguće je zameniti neku promenljivu u simboličkom izrazu nekom konkretnom vrednošću.
- Moguće je menjati umesto simbolički promenljivih i cele izraze koji sadrže druge simboličke promenljive.

Zamene

- Ukoliko želimo moguće je zameniti neku promenljivu u simboličkom izrazu nekom konkretnom vrednošću.
- Moguće je menjati umesto simbolički promenljivih i cele izraze koji sadrže druge simboličke promenljive.

```
>> syms x y z
>> f = x^2 + y^3 + z^4;
>> subs(f, x, 1/3)
>> f
>> syms x u v clear
>> f = x^2 + x + 1;
>> subs(f, x, u + v)
```

Zamene

- Ako izraz sadrži bar jednu simboličku promenljivu u tom slučaju Matlab i ne pokušava da interpretira izraz.

Zamene

- Ako izraz sadrži bar jednu simboličku promenljivu u tom slučaju Matlab i ne pokušava da interpretira izraz.
- Ako umesto simboličke promenljive zamenimo matricu, MuPAD će kreirati matricu čiji su elementi izrazi u kojima su zamenjeni pojedini elementi matrice.

Zamene

- Ako izraz sadrži bar jednu simboličku promenljivu u tom slučaju Matlab i ne pokušava da interpretira izraz.
- Ako umesto simboličke promenljive zamenimo matricu, MuPAD će kreirati matricu čiji su elementi izrazi u kojima su zamenjeni pojedini elementi matrice.

```
>> syms x y clear   >> f = x*y/3;
>> a = subs(f,[x y],[2 4])
>> syms x y clear   >> f = x^3+x*y+3;
>> subs(f,x,[1 2 3;4 5 6])
```

Zamene

- Slično se dešava i ako zamenimo matricom čiji su elementi automatski generisane simboličke promenljive.

Zamene

- Slično se dešava i ako zamenimo matricom čiji su elementi automatski generisane simboličke promenljive.
- Funkcija *symvar* pronalazi sve simboličke promenljive unutar simboličke promenljive/izraza.

Zamene

- Slično se dešava i ako zamenimo matricom čiji su elementi automatski generisane simboličke promenljive.
- Funkcija *symvar* pronalazi sve simboličke promenljive unutar simboličke promenljive/izraza.

```
>> syms x y A clear
>> A = sym('A', [2 3])
>> f = x^3 + x * y + 3;
>> g = subs(f, x, A)
>> symvar(g)
```

Zamene

- Ako želimo da zamenimo matricu u izraz koji predstavlja polinom, ali tako da matricu tretiramo kao jedinstveni algebarski objekat, tj. ako želimo da izračunamo vrednost polinoma $p(x) = x^2 + 3x + 1$ u matrici A, treba izračunati izraz $p(A) = A^2 + 3A + I$.

Zamene

- Ako želimo da zamenimo matricu u izraz koji predstavlja polinom, ali tako da matricu tretiramo kao jedinstveni algebarski objekat, tj. ako želimo da izračunamo vrednost polinoma $p(x) = x^2 + 3x + 1$ u matrici A, treba izračunati izraz $p(A) = A^2 + 3A + I$.
- Funkcija *sym2poly* keira vektor vrstu koja reprezentuje polinom unutar Matlaba, a onda funkcija *polyvalm*, koja nije deo paketa za simbolička izračunavanja, računa vrednost polinoma sa matričnim argumentom.

Zamene

- Ako želimo da zamenimo matricu u izraz koji predstavlja polinom, ali tako da matricu tretiramo kao jedinstveni algebarski objekat, tj. ako želimo da izračunamo vrednost polinoma $p(x) = x^2 + 3x + 1$ u matrici A, treba izračunati izraz $p(A) = A^2 + 3A + I$.
- Funkcija *sym2poly* keira vektor vrstu koja reprezentuje polinom unutar Matlaba, a onda funkcija *polyvalm*, koja nije deo paketa za simbolička izračunavanja, računa vrednost polinoma sa matričnim argumentom.

```
>> syms x clear
>> f = x^2 + 3*x + 1;
>> polyvalm(sym2poly(f), [1 2 3; 4 5 6; 7 8 9])
```

Zamene

- Funkcija *subexpr* pronalazi zajednički izraz u nizu izraza koji je prvi argument i ponovo ispisuje niz izraza, gde je zajednički izraz zamenjen jednom promenljivom.

Zamene

- Funkcija *subexpr* pronalazi zajednički izraz u nizu izraza koji je prvi argument i ponovo ispisuje niz izraza, gde je zajednički izraz zamenjen jednom promenljivom.
- Drugi argument u pozivu funkcije, ulazni i izlazni je ime zajedničkog izraza. Drugi izlazni argument dobija vrednost zajedničkog izraza.

Zamene

- Funkcija *subexpr* pronalazi zajednički izraz u nizu izraza koji je prvi argument i ponovo ispisuje niz izraza, gde je zajednički izraz zamenjen jednom promenljivom.
- Drugi argument u pozivu funkcije, ulazni i izlazni je ime zajedničkog izraza. Drugi izlazni argument dobija vrednost zajedničkog izraza.

```
>> syms b c x clear
>> sol = [(-b + sqrt(b^2 - 4 * c))/2 (-b - sqrt(b^2 - 4 * c))/2];
>> [r, s] = subexpr(sol,' s')
```

Diferenciranje

- Funkcija *diff* omogućava diferenciranje simboličkih izraza.

Diferenciranje

- Funkcija *diff* omogućava diferenciranje simboličkih izraza.
- Najopštiji poziv je $diff(y, x, n)$, gde je y izraz koji treba diferencirati, argument x je promenljiva po kojoj treba diferencirati, i argument $n$je red izvoda koji se traži.

Diferenciranje

- Funkcija *diff* omogućava diferenciranje simboličkih izraza.
- Najopštiji poziv je $diff(y, x, n)$, gde je y izraz koji treba diferencirati, argument x je promenljiva po kojoj treba diferencirati, i argument $n$je red izvoda koji se traži.
- Kad nedostaje ime promenljive po kojoj se izvod traži MuPAD pretražuje izraz i pronalazi promenljivu čije ime, u smislu engleske abecede, je najbliže slovu x, i potoj promenljivoj traži izvod zadatog reda.

Diferenciranje

- Funkcija *diff* omogućava diferenciranje simboličkih izraza.
- Najopštiji poziv je $diff(y, x, n)$, gde je y izraz koji treba diferencirati, argument x je promenljiva po kojoj treba diferencirati, i argument $n$je red izvoda koji se traži.
- Kad nedostaje ime promenljive po kojoj se izvod traži MuPAD pretražuje izraz i pronalazi promenljivu čije ime, u smislu engleske abecede, je najbliže slovu x, i potoj promenljivoj traži izvod zadatog reda.
- Podrazumevan red izvoda je jedan.

Diferenciranje

- Funkcija *diff* omogućava diferenciranje simboličkih izraza.
- Najopštiji poziv je $diff(y, x, n)$, gde je y izraz koji treba diferencirati, argument x je promenljiva po kojoj treba diferencirati, i argument $n$je red izvoda koji se traži.
- Kad nedostaje ime promenljive po kojoj se izvod traži MuPAD pretražuje izraz i pronalazi promenljivu čije ime, u smislu engleske abecede, je najbliže slovu x, i potoj promenljivoj traži izvod zadatog reda.
- Podrazumevan red izvoda je jedan.

$>> syms\ x\ y\ clear \quad >> f = x^2 + y^3 * sin(x);$
$>> diff(f, x, 2) \quad >> diff(f, 2) \quad >> diff(f) \quad >> diff(f, y)$

Integracija

- Funkcija *int* omogućava integraciju simboličkih izraza.

Integracija

- Funkcija *int* omogućava integraciju simboličkih izraza.
- Najopštiji oblik funkcije *int* je $int(y, x, a, b)$, gde je y izraz koji se integrali (podintegralna funkcija), x je promenljiva po kojoj se vrši integracija, a je donja, a b gornja granica integracije.

Integracija

- Funkcija *int* omogućava integraciju simboličkih izraza.
- Najopštiji oblik funkcije *int* je $int(y, x, a, b)$, gde je y izraz koji se integrali (podintegralna funkcija), x je promenljiva po kojoj se vrši integracija, a je donja, a b gornja granica integracije.
- Ukoliko se ime promenljive po kojoj se integrali ne navede, promenljiva po kojoj se vrši integracija je podrazumevana promenljiva izraza y. Ako se ne navedu granice integracije, izračunava se neodređeni integral.

Integracija

- Funkcija *int* omogućava integraciju simboličkih izraza.
- Najopštiji oblik funkcije *int* je $int(y, x, a, b)$, gde je y izraz koji se integrali (podintegralna funkcija), x je promenljiva po kojoj se vrši integracija, a je donja, a b gornja granica integracije.
- Ukoliko se ime promenljive po kojoj se integrali ne navede, promenljiva po kojoj se vrši integracija je podrazumevana promenljiva izraza y. Ako se ne navedu granice integracije, izračunava se neodređeni integral.

```
>> syms x clear    >> f = x^2 * sin(x);
>> int(f)    >> int(f,x)
>> int(f,x,0,pi)
```

Integracija

- Moguća je primena ograničenja skupa mogućih vrednosti promenljive.

Integracija

- Moguća je primena ograničenja skupa mogućih vrednosti promenljive.

```
>> syms x a clear
>> int(exp(-a*x^2),x,-inf,inf)
>> syms a x clear
>> syms a positive
>> int(exp(-a*x^2,x,-inf,inf)
```

Nalaženje graničnih vrednosti

- Funkcija *limit* služi za određivanje graničnih vrednosti. Najopštiji oblik funkcije je $limit(y, x, a, strana)$, gde je y izraz od koga se traži granična vrednost, dok je x promenljiva po kojoj se traži granična vrednost, a je tačka u kojoj se granična vrednost izračunava, i strana je opcioni argument, sa vrednostima 'left' i 'right' da označi levu ili desnu graničnu vrednost.

Nalaženje graničnih vrednosti

- Funkcija *limit* služi za određivanje graničnih vrednosti. Najopštiji oblik funkcije je $limit(y, x, a, strana)$, gde je y izraz od koga se traži granična vrednost, dok je x promenljiva po kojoj se traži granična vrednost, a je tačka u kojoj se granična vrednost izračunava, i strana je opcioni argument, sa vrednostima 'left' i 'right' da označi levu ili desnu graničnu vrednost.

$>> syms\ x\ clear$
$>> y = sin(x);\quad >> limit(y, x, 0)$
$>> syms\ x\ a\ clear$
$>> f = (1 + a/x)^x;$
$>> limit(f, x, inf)\quad >> limit(x^x, x, 0, 'right')$

Sumiranje

- Funkcija *symsum* se korisiti za simboličko sumiranje redova. Najopštiji oblik za sumiranje redova izgleda ovako $symsum(y, x, a, b)$, gde je y opšti član reda, x je promenljiva po kojoj se izvodi sumiranje, a i b su donja i gornja granica po kojoj se menja promenljiva po kojoj se izvodi sumiranje.

Sumiranje

- Funkcija *symsum* se korisiti za simboličko sumiranje redova. Najopštiji oblik za sumiranje redova izgleda ovako $symsum(y, x, a, b)$, gde je y opšti član reda, x je promenljiva po kojoj se izvodi sumiranje, a i b su donja i gornja granica po kojoj se menja promenljiva po kojoj se izvodi sumiranje.
- Ako ime promenljive po kojoj se vrši sumiranje nije navedeno, sumiranje se izvodi po podrazumevanoj promenljivoj izraza y.

Sumiranje

- Funkcija *symsum* se korisiti za simboličko sumiranje redova. Najopštiji oblik za sumiranje redova izgleda ovako $symsum(y, x, a, b)$, gde je y opšti član reda, x je promenljiva po kojoj se izvodi sumiranje, a i b su donja i gornja granica po kojoj se menja promenljiva po kojoj se izvodi sumiranje.
- Ako ime promenljive po kojoj se vrši sumiranje nije navedeno, sumiranje se izvodi po podrazumevanoj promenljivoj izraza y.

```
>> syms k clear
>> symsum(1/k^2, 1, inf)
>> symsum(1/k^4, k, 1, Inf)
```

Taylorov polinom

- Funkcija *taylor* razvija funkciju u Taylorov polinom u okolini zadate tačke sa zadatim stepenom.

Taylorov polinom

- Funkcija *taylor* razvija funkciju u Taylorov polinom u okolini zadate tačke sa zadatim stepenom.
- Opšti oblik funkcije je $taylor(y, x, n, a)$, gde je y izraz čiji se Taylorov polinom traži, x je promenljiva po kojoj se traži Taylorov polinom, $n-1$ je stepen traženog Taylorovog polinoma, i a je tačka u kojoj se traži razvoj u Taylorov polinom.

Taylorov polinom

- Funkcija *taylor* razvija funkciju u Taylorov polinom u okolini zadate tačke sa zadatim stepenom.
- Opšti oblik funkcije je $taylor(y, x, n, a)$, gde je y izraz čiji se Taylorov polinom traži, x je promenljiva po kojoj se traži Taylorov polinom, $n-1$ je stepen traženog Taylorovog polinoma, i a je tačka u kojoj se traži razvoj u Taylorov polinom.

```
>> syms x clear
>> f = log(1 - x^2);
>> taylor(f, x, 8, 0)
```

Algebarske jednačine

- Koristeći funkciju *solve* moguće je rešavati jednu ili sistem algebarskih jednačina. Najopštiji oblik funkcije *solve* izgleda ovako $solve(eqn1, ..., eqnN, x1, ..., xN)$, gde su $eqn1$ do $eqnN$ jednačine koje treba rešiti, a $x1$ do xN su promenljive po kojima treba rešiti jednačine.

Algebarske jednačine

- Koristeći funkciju *solve* moguće je rešavati jednu ili sistem algebarskih jednačina. Najopštiji oblik funkcije *solve* izgleda ovako $solve(eqn1, ..., eqnN, x1, ..., xN)$, gde su $eqn1$ do $eqnN$ jednačine koje treba rešiti, a $x1$ do xN su promenljive po kojima treba rešiti jednačine.
- Ako ograničimo skup mogućih vrednosti promneljive u nekim slučajevima nećemo dobiti rešenja jednačina koje imaju rešenja.

Algebarske jednačine

- Koristeći funkciju *solve* moguće je rešavati jednu ili sistem algebarskih jednačina. Najopštiji oblik funkcije *solve* izgleda ovako $solve(eqn1, ..., eqnN, x1, ..., xN)$, gde su $eqn1$ do $eqnN$ jednačine koje treba rešiti, a $x1$ do xN su promenljive po kojima treba rešiti jednačine.
- Ako ograničimo skup mogućih vrednosti promneljive u nekim slučajevima nećemo dobiti rešenja jednačina koje imaju rešenja.

$>>$ *syms x y a*
$>>$ $[u, v] = solve(x^2 + y^2 - a, x + y, x, y)$
$>>$ *syms x clear* $>>$ *syms x real*
$>>$ $u = solve(x^2 + 1, x)$

Diferencijalne jednačine

- Funkcija *dsolve* može se koristiti za rešavanje diferencijalnih jednačina. Najopštiji oblik korišćenja funkcije *dsolve* je sledeći *dsolve*(*jednacina*, *pocetniuslovi*), gde je argument *jednacina* string koji predstavlja jednačinu, argument *pocetniuslovi* je string koji predstavlja početne uslove.

Diferencijalne jednačine

- Funkcija *dsolve* može se koristiti za rešavanje diferencijalnih jednačina. Najopštiji oblik korišćenja funkcije *dsolve* je sledeći *dsolve*(*jednacina*, *pocetniuslovi*), gde je argument *jednacina* string koji predstavlja jednačinu, argument *pocetniuslovi* je string koji predstavlja početne uslove.
- Ako hoćemo da unesemo vrednosti izvoda u početne uslove koristimo slovo D.

Diferencijalne jednačine

- Funkcija *dsolve* može se koristiti za rešavanje diferencijalnih jednačina. Najopštiji oblik korišćenja funkcije *dsolve* je sledeći *dsolve*(*jednacina*, *pocetniuslovi*), gde je argument *jednacina* string koji predstavlja jednačinu, argument *pocetniuslovi* je string koji predstavlja početne uslove.
- Ako hoćemo da unesemo vrednosti izvoda u početne uslove koristimo slovo D.

$$>> simplify(dsolve('D^2y + Dy + y = 0','y(0) = 1','y(1) = 1'))$$
$$>> simplify(dsolve('D^2y + Dy + y = 0','Dy(0) = 1','y(0) = 1'))$$

Manipulacija matricama

- Funkcija *size* određuje tip simboličke matrice.

Manipulacija matricama

- Funkcija *size* određuje tip simboličke matrice.
- Funkcija *diag* kreira ili ekstrahuje dijagonalu simboličke matrice.

Manipulacija matricama

- Funkcija *size* određuje tip simboličke matrice.
- Funkcija *diag* kreira ili ekstrahuje dijagonalu simboličke matrice.
- Funkcije *tril* i *triu* vraćaju donju i gornju trougaonu matricu zadate matrice, redom.

```
>> syms a b c clear
>> diag([a b c])    >> diag([a, b, c], 1)
>> diag([a, 0, 0; 0 b 0; 0, 0, c])
>> A = sym('A', [3, 4]);
>> tril(A)
```

Determinanta i inverzna matrica

- Funkcija *det* se koristi za izračunavanje determinante matrice čiji su elementi simboličke promenljive ili izrazi.

Determinanta i inverzna matrica

- Funkcija *det* se koristi za izračunavanje determinante matrice čiji su elementi simboličke promenljive ili izrazi.
- Funkcija *inv* izračunava inverznu matricu simboličke matrice i vraća simboličku matricu.

Determinanta i inverzna matrica

- Funkcija *det* se koristi za izračunavanje determinante matrice čiji su elementi simboličke promenljive ili izrazi.
- Funkcija *inv* izračunava inverznu matricu simboličke matrice i vraća simboličku matricu.

```
>> syms a b c d clear
>> A = [a b; c d];
>> det(A)
>> inv(A)
```

Redukcija i rang

- Funkcija *rref* izračunava redukovanu trapeznu formu matrice, tj.

$$\begin{bmatrix} I & B \\ 0 & 0 \end{bmatrix}$$

Redukcija i rang

- Funkcija *rref* izračunava redukovanu trapeznu formu matrice, tj.

$$\begin{bmatrix} I & B \\ 0 & 0 \end{bmatrix}$$

- Funkcija *rank* računa rang simboličke matrice.

Redukcija i rang

- Funkcija *rref* izračunava redukovanu trapeznu formu matrice, tj.

$$\begin{bmatrix} I & B \\ 0 & 0 \end{bmatrix}$$

- Funkcija *rank* računa rang simboličke matrice.

```
>> A = sym('[1 2 3; 2 4 6; 1 3 7]');
>> rref(A)
>> syms a b c clear
>> A = [a b c; 2*a 2*b 2*c; 1 2 3];
>> rref(A)    >> rank(A)
```

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Funkcija *poly* vraća karakteristični polinom simboličke matrice.

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Funkcija *poly* vraća karakteristični polinom simboličke matrice.
- Ako je funkcija pozvana bez drugog argumenta karakteristični polinom se računa po promenljivoj x.

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Ako promenljiva x ne postoji na ovaj način ona neće biti kreirana. Promenljiva x postoji u MuPADu ali ne i Matlabu. Ako hoćemo da dobijemo mogućnost da promenimo vrednost promenljive x unutar karakterističnog polinoma matrice A, morali bismo prvo da dobijemo promenljivu u Matlabu koja pokazuje na odgovarajuću prmenljivu u MuPADu, pa tek da sa njom radimo.

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Ako promenljiva x ne postoji na ovaj način ona neće biti kreirana. Promenljiva x postoji u MuPADu ali ne i Matlabu. Ako hoćemo da dobijemo mogućnost da promenimo vrednost promenljive x unutar karakterističnog polinoma matrice A, morali bismo prvo da dobijemo promenljivu u Matlabu koja pokazuje na odgovarajuću prmenljivu u MuPADu, pa tek da sa njom radimo.

```
>> syms a b c y clear     >> poly(A,y)
>> A=[a b; a+b c];     >> h=poly(A);
>> x=sym('x');     >> subs(h,x,2)
```

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Sopstvene vrednosti i sopstveni vektori simboličkih matrica mogu se dobiti upotrebom funkcije *eig*.

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Sopstvene vrednosti i sopstveni vektori simboličkih matrica mogu se dobiti upotrebom funkcije *eig*.
- Ako imamo samo jedan izlazni argument, dobijamo vektor sopstvenih vrednosti. Ako imamo dva izlazna argumenta dobijamo matricu sopstvenih vektora, i matricu sopstvenih vrednosti, pri čemu je matrica sopstvenih vrednosti dijagonalna.

Karakteristični polinom, sopstvene vrednosti i sopstveni vektori

- Sopstvene vrednosti i sopstveni vektori simboličkih matrica mogu se dobiti upotrebom funkcije *eig*.
- Ako imamo samo jedan izlazni argument, dobijamo vektor sopstvenih vrednosti. Ako imamo dva izlazna argumenta dobijamo matricu sopstvenih vektora, i matricu sopstvenih vrednosti, pri čemu je matrica sopstvenih vrednosti dijagonalna.

```
>> syms a b clear
>> A = [a b; b a];
>> d = eig(A)    >> [d V] = eig(A)
```

Konverzija u stringove

- Konverzija u stringove se izvodi funkcijom *char*.

Konverzija u stringove

- Konverzija u stringove se izvodi funkcijom *char*.
- Funkcija *char* primenjena nad simboličkim izrazom proizvodi njegovu reprezentaciju u obliku stringa.

Konverzija u stringove

- Konverzija u stringove se izvodi funkcijom *char*.
- Funkcija *char* primenjena nad simboličkim izrazom proizvodi njegovu reprezentaciju u obliku stringa.

```
>> syms x clear
>> a = char(x^3 + x);
>> whos a
```

Konverzija u numeričke tipove

- U numeričke tipove podataka možemo konvertovati izraze koji u sebi sadrže samo konstante, dakle, koji u sebi ne sadrže simboličke promenljive koje nemaju konstantne vrednosti.

Konverzija u numeričke tipove

- U numeričke tipove podataka možemo konvertovati izraze koji u sebi sadrže samo konstante, dakle, koji u sebi ne sadrže simboličke promenljive koje nemaju konstantne vrednosti.
- Funkcije koje se koriste za konverziju odgovaraju tipovima podataka u Matlabu. Za označene cele brojeve to su funkcije *int*8, *int*16, *int*32, *int*64, za neoznačene cele brojeve *uint*8, *uint*16, *uint*32, *uint*64, za tipove podataka u pokretnom zarezu *single* i *double*.

Konverzija u numeričke tipove

- U numeričke tipove podataka možemo konvertovati izraze koji u sebi sadrže samo konstante, dakle, koji u sebi ne sadrže simboličke promenljive koje nemaju konstantne vrednosti.
- Funkcije koje se koriste za konverziju odgovaraju tipovima podataka u Matlabu. Za označene cele brojeve to su funkcije $int8, int16, int32, int64$, za neoznačene cele brojeve $uint8, uint16, uint32, uint64$, za tipove podataka u pokretnom zarezu *single* i *double*.

```
>> A = sym('[1 1 + sqrt(2)]');
>> a = int8(A);    >> whos a
>> a = single(A)    >> whos a
```

Grafika

- Funkcija *ezplot* može se koristiti za crtanje grafika funkcije jedne promenljive, koja je data eksplicitno, implicitno ili parametarski.

Grafika

- Funkcija *ezplot* može se koristiti za crtanje grafika funkcije jedne promenljive, koja je data eksplicitno, implicitno ili parametarski.
- Za crtanje eksplicitno date funkcije koristi se format *ezplot*(*izraz*, [*a b*]), gde je izraz izraz koji zavisi od jedne promenljive, a *a* i *b* su granice intervala u kojem se crta grafik.

```
>> syms x clear
>> subplot(2,2,1), ezplot(x*sin(x),[-5,5]);
>> hold on
>> ezplot(x*cos(x));
>> hold off
```

Grafika

- U slučaju implicitno zadate funkcije format za pozivanje funkcije izgleda ovako *ezplot*(*izraz*, [*xmin xmax ymin ymax*]), gde je izraz implicitno data funkcija dve promenljive, zadata sa izraz=0.

Grafika

- U slučaju implicitno zadate funkcije format za pozivanje funkcije izgleda ovako *ezplot*(*izraz*, [*xmin xmax ymin ymax*]), gde je izraz implicitno data funkcija dve promenljive, zadata sa izraz=0.

$>> subplot(2,2,2), ezplot(x^2+y^2-9);$
$>> hold\ on$
$>> ezplot(x^2+y^2-25,[-5,5,-5,5]);$
$>> hold\ off$

Grafika

- Ako je funkcija parametarski data poziv funkcije ima sledeći format *ezplot*(*izrazx*, *izrazy*, [*tmin tmax*]), pri čemu je *izrazx* izraz koji opisuje zavisnost apscise od parametra, *izrazy* je izraz koji opisuje zavisnost ordinate od parametra, a *tmin* i *tmax* su granice u kojima se parametar menja.

Grafika

- Ako je funkcija parametarski data poziv funkcije ima sledeći format *ezplot*(*izrazx*, *izrazy*, [*tmin tmax*]), pri čemu je *izrazx* izraz koji opisuje zavisnost apscise od parametra, *izrazy* je izraz koji opisuje zavisnost ordinate od parametra, a *tmin* i *tmax* su granice u kojima se parametar menja.

```
>> subplot(2,2,3), ezplot(cos(t),sin(t));
>> subplot(2,2,4), ezplot(cosh(t),sinh(t),[-1,1]);
```

Grafika

- Funkcija *ezpolar* crta grafik funkcije u polarnim koordinatama. Format funkcije je sledeći *ezpolar*(*izraz*, [*a* *b*]).

Grafika

- Funkcija *ezpolar* crta grafik funkcije u polarnim koordinatama. Format funkcije je sledeći *ezpolar*(*izraz*, [*a b*]).
- Funkcija *ezplot*3 crta parametarski zadate trodimenzionalne krive. Osnovni format funkcije izgleda ovako *ezplot*(*izrazx*, *izrazy*, *izrazz*, [*tmin tmax*]).

Grafika

- Funkcija *ezpolar* crta grafik funkcije u polarnim koordinatama. Format funkcije je sledeći *ezpolar*(*izraz*, [*a b*]).
- Funkcija *ezplot*3 crta parametarski zadate trodimenzionalne krive. Osnovni format funkcije izgleda ovako *ezplot*(*izrazx*, *izrazy*, *izrazz*, [*tmin tmax*]).

```
>> syms t clear
>> ezpolar(t, [0 4 * pi]);
>> ezplot3(cos(t), sin(t), t, [0, 6 * pi],' animate');
```