# bPodstawy:

save ’plik’ zapisuje zmienne w pliku *plik* load ’plik’ wczytuje zmienne z pliku *plik* diary on zaczyna zapisywać sesję do pliku diary off kończy zapisywanie sesji

expm(A) funkcja wykładnicza dla macierzy *eA* sqrtm(A) macierz *B*, taka że *B*2 = *A* size(A) rozmiar macierzy *A*

length(x) długość wektora *x*

sum(x) suma współrzędnych wektora *x*

diary(’plik’) ustala nazwę pliku, do którego będzie zapisywana sesja

whos wypisuje listę wszystkich zmiennych

clear niszczy wszystkie zmienne

help polecenie wypisuje informacje o poleceniu *polecenie*

doc polecenie dokładniejsza informacja o poleceniu

A’ transpozycja macierzy *A*

dot(x,y) iloczyn skalarny wektorów *x* i *y*

# Tworzenie nowych macierzy:

rand(5,4) macierz 5x4 o losowych współrzędnych (rozkład jednostajny na [0*,* 1))

randn(5,4) macierz 5x4 o losowych współrzędnych (rozkład normalny N(0,1))

# Definiowanie i zmiana zmiennych:

zeros(4,2) macierz 4x2 wypełniona zerami

ones(1,4) wektor kolumnowy wypełniony jedynkami

x = 3 zmienna *x* staje się równa 3

x = [1 2 3] *x* staje się wektorem poziomym [1*,* 2*,* 3]

x = [1; 2; 3] *x* staje się wektorem pionowym [1*,* 2*,* 3]*T*

A = [1 2; 3 4; 5 6] *A* staje się macierzą 2x3

x(2) = 5 zmiana wartości drugiej współrzędnej wektora *x*

A(2,3) = 8 zmiana wartości macierzy w drugim wierszu i trzeciej kolumnie

# Działania na liczbach:

2+3, 2-3, 2\*3, 2/3 dodawanie, odejmowanie, itd.

2^2, 2^(1+i) potęgowanie

sqrt(-5) pierwiastek kwadratowy z -5

exp(4) *e*4

log(10), log10(10) oblicza ln 10 i log10 110 abs(4+3\*i) oblicza wartość bezwzględną liczby 4 + 3*i* sin(pi/2) oblicza sin( *π* )

2

# Działania na wektorach i macierzach:

3\*x każdą współrzędną wektora mnożymy przez 3 x+3 dodaje 3 do każdej współrzędnej wektora *x* x+y suma wektorów *x* i *y*

A\*x iloczyn macierzy *A* przez wektor *x*

A\*B iloczyn macierzy *A* i *B*

x.\*y iloczyn wektorów *x* i *y* po współrzędnych

x./y dzielenie po współrzędnych

x.^y potęgowanie po współrzędnych

A^3 trzecia potęga macierzy kwadratowej *A*

cos(x) cosinus każdej współrzędnej *x*

abs(A) wartość bezwzględna każdego elementu

exp(A) *e* do potęgi każdej współrzędnej *A*

sqrt(A) pierwiastek kwadratowy każdej współrzędnej *A*

eye(5) macierz jednostkowa 5x5

linspace(0,10,11) wektor poziomy o 11 liczbach równo wypełniających przedział [0*,* 10]

0:10 wektor poziomy [0*,* 1*,* 2*, . . . ,* 9*,* 10]

1:0.3:3 wektor poziomy [1*,* 1*.*3*,* 1*.*6*, . . . ,* 2*.*4*,* 2*.*8]

diag(x) macierz diagonalna o diagonali równej wektorowi *X*

# Fragmenty wektorów i macierzy:

x(2:5) współrzędne wektora *x* od drugiej do piątej

x(2:end) współrzędne od drugiej do końca

x(1:2:end) co druga współrzędna *x* począwszy od pierwszej

A(3,:) trzeci wiersz macierzy *A*

A(3,2:5) trzeci wiersz macierzy *A* o kolumnach od drugiej do piątej

A(:,2) druga kolumna macierzy *A*

diag(A) główna przekątna macierzy *A*

# Równania liniowe:

A b rozwiązanie równanie *Ax* = *b*

*\*

b/A rozwiązuje równanie *xA* = *b*

eig(A) zwraca wartości własne macierzy *A*

[V,D] = eig(A) *V* macierz wektorów własnych (w kolumnach),

*D* — macierz o diagonali z wartości własnych

det(A) wyznacznik macierzy *A*

# Rysowanie:

plot(y) rysuje wykres *y* względem osi *x* = 1*,* 2*, . . .*

plot(x,y) rysuje wykres *y* względem *x*

plot(x,A) rysuje wykres wierszy *A* względem *x* (liczba kolumn musi się zgadzać)

axis equal wymusza takie samo skalowania na obu osiach title(’Tytul’) dodaje *Tytul* nad rysunkiem xlabel(’ala’) podpisuje oś *OX* jako *ala*

legend(’f’, ’g’) podpisuje dwie krzywe jako *f* i *g*

grid dodaje siatkę na rysunku

pascal(x)

C = randi(10,3,2) (zakres, wiersz, kol)

„;” wiersz

„ „ kolumna

Magic(x)

Ones(x)

Format long

A=[min:krok:max]

Polecenie generuje wektor poczynając od elementu o wartości min, kończąc na elemencie owartości maxz krokiem krok. Jeżeli parametrkrokzostanie pominięty, przyjmuje się, iżkrok=1

mean(A) średnia kolumn

a.\*b mnożenie tablicowe

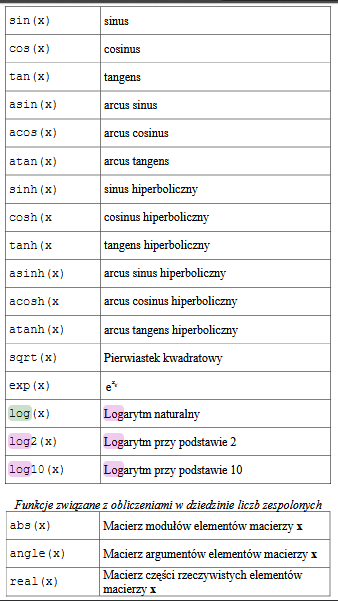
det(A)- obliczanie wyznacznika macierzyA•

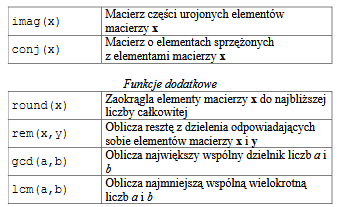
eig(A)- obliczanie wartości własnych macierzyA•

poly(A)-obliczaniewspółczynników wielomianu charakterystycznego macierzyA•

rank(A)-obliczanierzędu macierzyA•

diag(A)- wyznaczanie elementów leżących na głównej przekątnej macierzyA





K(:,1)=[ ]

%usunięcie pierwszego wiersza

K(1,:)=[ ]

%usunięcie likwiduje dany wiersz bądź kolumnę z macierzy

for c = 1:ncols

for r = 1:nrows

if r == c

A(r,c) = 2;

elseif abs(r-c) == 1

A(r,c) = -1;

else

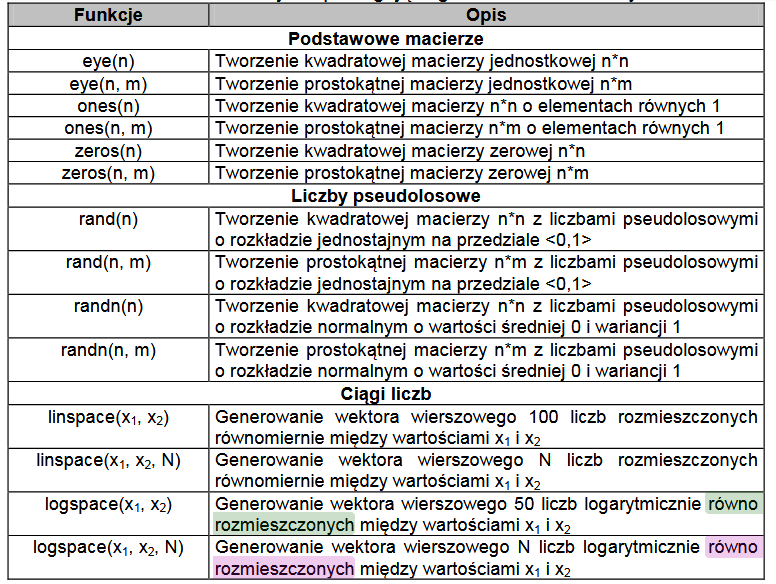
A(r,c) = 0;

end

end

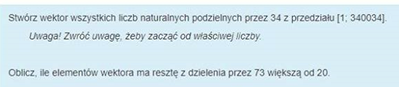
end

A



Wiersz x kolumna!!



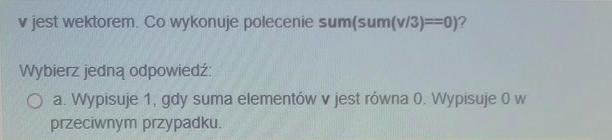


>> d=[34:34:340034]

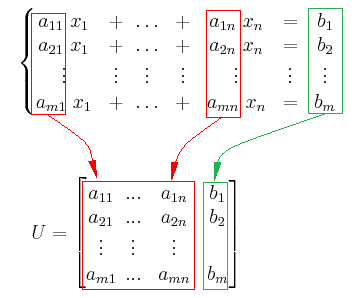
d(mod(d,73)<=20)=[]

length(d)

ans = 7124



Crammer



posiada przynajmniej jedno rozwiązanie wtedy i tylko wtedy, gdy

rankA =rankU

n – m = licz parametrów

Tzn. rząd macierzy współczynników i rząd macierzy uzupełnionej układu są sobie równe.

Ponadto, jeżeli wprowadzimy następujące oznaczenia:

r = rankA=rankU

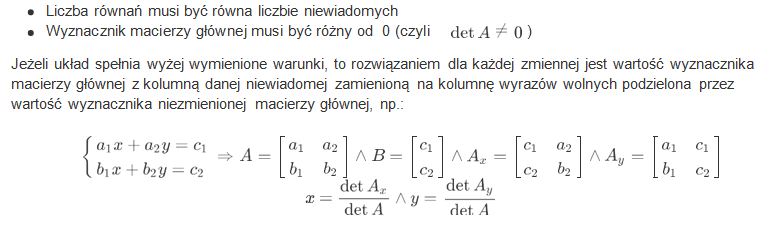
n niewiadome; m liczba równań

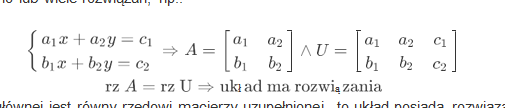
to gdy

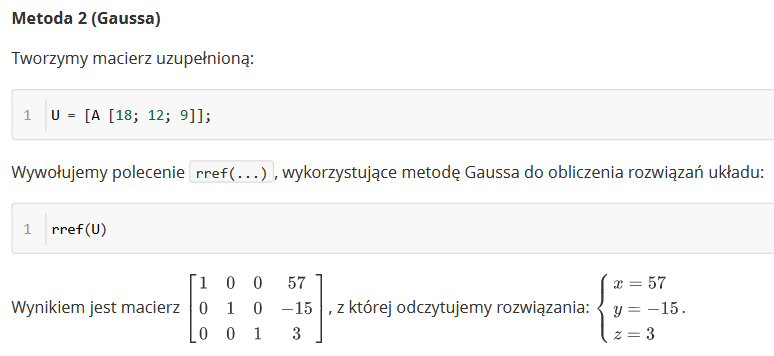
r = n - układ ma dokładnie jedno rozwiązanie,

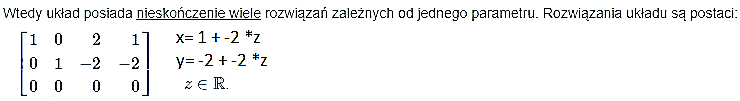
r < n - układ ma nieskończenie wiele rozwiązań, które zależą od parametrów.

Gry rankA != rank U układ sprzeczny

rank() rząd macierzy







Rysowanie Plot(x,y, col-pat)

Clf czyszczenie

Hold on, zatrzymanie

Hold off

x=linspace(-2 \* pi, 3 \* pi, 100)

plot(x,sin(x), 'g!')

x = linspace(-pi,0,250)

w = linspace(0,pi,250)

plot(x,-sin(x),’r’,w,sin(w),’r’)

x2 = [-pi,0]

y2 = [0,-2]

plot(x2,y2,’r’)

x3 = [pi,0]

y3 =[0,-2]

plot(x3,y3,’r’)