

Algebra liniowa - laboratorium nr 2, liczby zespolone

Wstęp

Zapoznaj się z podanymi poniżej funkcjami, działanie niektórych z nich możesz przetestować za pomocą podanych przykładowych komend. Ważna informacja: jednostkę urojoną i otrzymujemy w Maximie za pomocą

`%i`.

Następnie wykonaj zadania ze strony 3 i 4.

Przydatne funkcje

- `print(wyrażenie)` - drukuje wyrażenie

```
print("mój tekst")
a:3; print("a =", a)
a:4; print("kwadrat liczby", a, "wynosi", a^2)
```

- `realpart(z)` - część rzeczywista liczby zespolonej z
- `imagpart(z)` - część urojona liczby zespolonej z
- `rectform(z)` - postać algebraiczna liczby zespolonej

```
rectform((2+4*%i)/(-5+2*%i));
```

- `abs(a)` - wartość bezwzględna liczby a
- `cabs(z)` - zwraca moduł liczby zespolonej z

```
cabs(1);
cabs(1 + %i);
cabs(a+%i*b);
```

- `carg(z)` - zwraca argument liczby zespolonej

```
carg(1);
carg(1 + %i);
carg(1+sqrt(3)*%i);
```

- `sqrt(a)` - zwraca pierwiastek liczby a

```
sqrt(2);
```

- `expand(w)` - wymnaża nawiasy w wyrażeniu w , rozwija wyrażenie w

```
expand((x-1)^2*(y+3));
```

- `rhs(w)` - zwraca prawą stronę równania (analogicznie `lhs(w)` zwraca lewą stronę równania)

```
rhs(x^2+y^2=3*x-2*y);
```

- `float(a)` - zwraca postać dziesiętną liczby a (przybliżoną)

```
float(%pi);  
float(sqrt(2));
```

- `round(a)` - zwraca przybliżenie do jedności liczby a (liczby podzielne przez 0.5 są przybliżane do najbliższej liczby parzystej)

```
round(%pi);  
round(1/2);  
round(3/2);
```

- `subst(a,b,c)` - podstawia a za b w wyrażeniu c

```
row:x^4-5*x^2+6=0; subst(t,x^2,row);  
exptsubst: not exptsubst; - zmusza maxime do podstawiania niejawnych wyrażeń  
row:x^4-5*x^2+6=0; subst(t,x^2,row);
```

- `solve(równanie,x)` - rozwiązuje równanie względem x , ogólna postać `solve([lista równań],[lista zmiennych])`

```
solve(x^4-5*x^2+6=0,x);
```

- `realonly:true` - ogranicza funkcję `solve()` do zwrotu tylko rozwiązań rzeczywistych
- `length(L)` - zwraca liczbę elementów listy L
- `push(a,L)` - dodaje "z lewej" element a do listy L

Funkcje pakietu "draw"

Należy załadować pakiet `draw` poleceniem `load(draw)` Wszystkie poniższe funkcje należy wpisywać wewnątrz funkcji `draw2d()`, `wxdraw2d()`

- `xaxis=true` - rysuje oś X
- `yaxis=true` - rysuje oś Y
- `xrange=[a,b]` - ogranicza rysunek wykresu do $x \in [a, b]$
- `yrange=[a,b]` - ogranicza rysunek wykresu do $y \in [a, b]$
- `points([L])` - rysuje punkty z listy L na wykresie

```
points([[0,0],[0,1]])
```

- `point_type=filled_circle` - zmiana stylu punktów na koła
- `polygon([L])` - rysuje wielokąt o bokach w punktach listy L
- `explicit(F,x,a,b)` - rysuje funkcję F zmiennej x na przedziale $[a, b]$
- `implicit(F,x,a,b,y,c,d)` - rysuje wykres funkcji uwikłanej F zmiennych x, y dla $x \in [a, b]$ oraz $y \in [c, d]$
- `parametric(X,Y,t,a,b)` - rysuje krzywą określoną parametrycznie $x = x(t)$, $y = y(t)$ dla $t \in [a, b]$
- `proportional_axes = xy` - jednostki na osiach X i Y będą proporcjonalne

Zadania

1. Napisać funkcję `pos_tryg()` przyjmującą liczbę zespoloną z w postaci algebraicznej a zwracającą postać trygonometryczną liczby z , której argument podany jest w stopniach.

```
pos_tryg(1+sqrt(3)*%i) zwraca 2 [cos( 60 ) + % i sin( 60 )]
```

```
pos_tryg(7+%i*10) zwraca \sqrt{149} [cos( 55 ) + % i sin( 55 )]
```

2. Napisać funkcję `zwin_wiel()` przyjmującą dwa argumenty. Pierwszy to lista pierwiastków, druga to lista krotności pierwiastków. Funkcja ta zwraca rozwinięty wielomian, którego pierwiastkami są liczby z pierwszej listy o krotnościach z drugiej.

```
zwin_wiel([1,2],[1,1]) zwraca
```

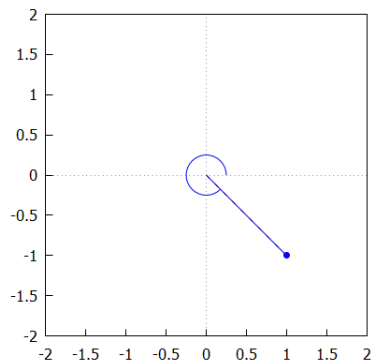
$$x^2 - 3x + 2.$$

```
zwin_wiel([2,3,1],[7,8,3]) zwraca
```

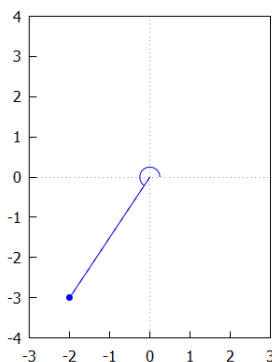
$$x^{18} - 41x^{17} + 789x^{16} - 9467x^{15} + 79348x^{14} - 493206x^{13} + 2355458x^{12} - 8834950x^{11} + 26375673x^{10} - 63118885x^9 + 121322321x^8 - 186786111x^7 + 228533634x^6 - 219094092x^5 + 160886088x^4 - 87273936x^3 + 32915808x^2 - 7698240x + 839808.$$

3. Napisać funkcję `zes_na_wyk()` przyjmującą liczbę zespoloną z , która rysuje jej interpretację geometryczną wraz z promieniem i argumentem.

```
zes_na_wyk(1-1*%i) zwraca
```



```
zes_na_wyk(-2-3*%i) zwraca
```



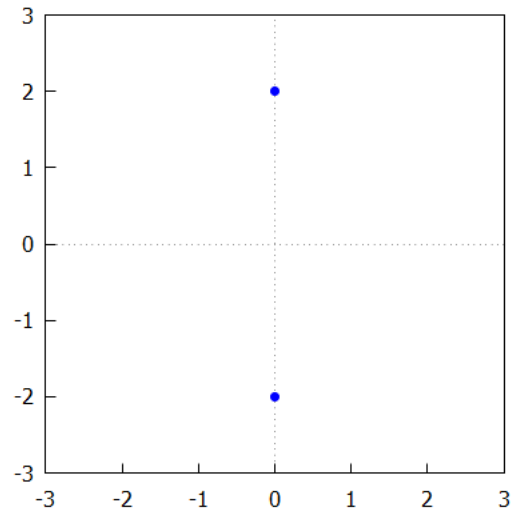
4. Napisać funkcję `rozw_wiel()` przyjmującą równanie zespolone i zwracającą liczby zespolone, które je spełniają.

```
rozw_wiel(z^2-z=8+z) zwraca [-2,4]
```

```
rozw_wiel(z^2-(6+%i)*z+11-7*%i=0) zwraca [1-2 % i, 3 % i+5]
```

5. Napisać funkcję `rys_pier()` przyjmującą dwa argumenty. Pierwszy to liczba zespolona, drugi to stopień pierwiastka. Funkcja ta zwraca rysunek z zaznaczonymi wszystkimi pierwiastkami.

```
rys_pier(-4,2) zwraca
```



```
rys_pier(-34+%i*50,8) zwraca
```

