

wxMaxima, czyli nakładka graficzna

Maxima w skrócie to sprytny program do obliczeń matematycznych i rysowania wykresów. W przeciwieństwie do kalkulatora nie przetwarza tylko liczby, ale może pracować na symbolach, co przydaje się m.in. przy wielomianach, liczeniu pochodnych i całek, macierzach i wielu, wielu innych. Od znanych komercyjnych programów typu Mathematica różni się licencją – przy podobnych możliwościach Maxima należy do wolnego oprogramowania (GLP). Nic nie kosztuje, a przy tym jest ciągle rozwijana.



Rys. 1
logo Maximy

Maxima to program konsolowy (typu „linia komend”), a więc średnio przyjazny. Istnieją różne nakładki graficzne np. XMaxima (która jest „obudowanym programem konsolowym”) i wxMaxima, która dla laika ładnie wygląda (ma menu, ikony), jest wygodna i po polsku. Ponieważ lubię wygodę i szybką pracę, stąd poniżej przybliżę właśnie wxMaximę.

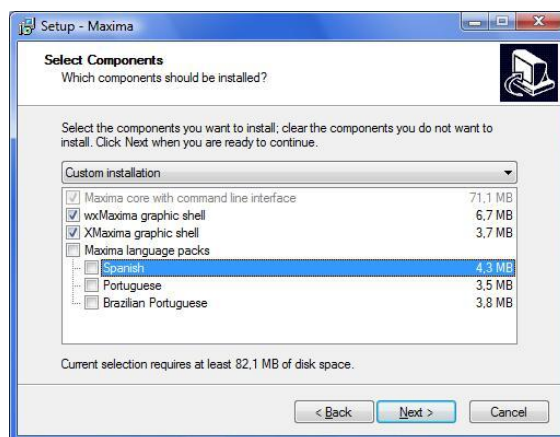
Instalacja

Cały pakiet Maximy i jej nakładek graficznych dla Windows, Linuxa i Maca pobrać można z oficjalnego downloadu Maximy, czyli

<http://sourceforge.net/projects/maxima/files/>

W chwili pisania tego artykułu (luty 2010 r.) ostatnia wersja Maximy miała oznaczenie 5.20.1, a wxMaximy - 0.8.4. Warto pobrać najnowszą wersję m.in. ze względu na spolszczenie programu.

Jeśli posiadamy Windowsy (tą wersję będę poniżej opisywał) wybieramy pierwszy plik z listy: maxima-5.20.1.exe (ok. 26 MB), zapisujemy na dysku i uruchamiamy. Instalator jest standardowy. W oknie „Select Components” polecam wybrać opcję „Full instalation”, przy czym wyłączyć checkboxy („ptaszki”) przy wersjach językowych „Spanish”, „Portuguese” i „Brazilian Portuguese” (patrz rysunek 2). Po zainstalowaniu program zajmie ok. 82 MB.

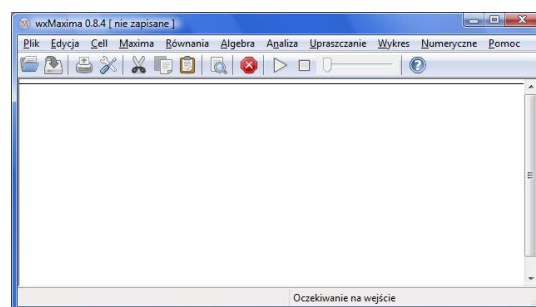


Rysunek 2

Uruchomienie

Jeśli nie zmienialiśmy domyślnych ustawień instalacji, to w menu Start > Programy znajdziemy folder o nazwie „Maxima-5.20.1”, a w nim trzy skróty do programów:

- „Command line Maxima”, czyli czarna mamba w całej okazałości
- „XMaxima” obudowana linia komend
- „wxMaxima” – ten program uruchamiamy i naszym oczom winien objawić się obraz jak obok (rysunek 3)



Rysunek 3

Rozgrzewka, czyli o co chooooooosi?

No to mamy duże białe pole, menu po polsku, a jak zaczynamy pisać to pojawia się jakaś czerwona szubieniczka...

Zacznijmy od tego, że w wxMaximie wszystko co pojawia się w białym polu to albo wejście (nasze komendy) albo wyjście (to co program nam wyrzuci). Program po lewej stronie oznacza linie wejścia (input) jako „(%i_kolejne_wejście)” np. „(%i24)”, a linie wyjścia (output) rozpoznamy po „(%o_kolejne_wyjście)”, np. „(%o24)”. Dzięki tym oznaczeniom możemy odwoływać się do poprzednich wejść/wyjść. WxMaxima symbole te sama wstawia (łącząc je jedno pod drugim w parę), nie musimy je wpisywać.

Komendy lub liczby i symbole wpisujemy ręcznie lub wybieramy poprzez menu programu, a na końcu każdej linii wejścia umieszczamy „;” (średnik) i naciskamy jednocześnie klawisze Shift+Enter. Przewagą interfejsu wxMaximy jest to, że gdy zapomnimy o średniku, program go sam uzupełni. Wciskając na końcu samo Enter przejdziemy do następnej linii tego samego wejścia, bez wykonania komendy. Pamiętajmy o tym, że program rozpoznaje wielkość liter.

Komendy do wpisania będę podawał w następującej konotacji: (zielona czcionka Courier) `invert(s);` Po każdej komendzie pamiętajmy o naciśnięciu jednocześnie Shift i Enter.

Na początek użyjmy programu jako kalkulatora, wpismy

`2+2*4-1;`

W efekcie program odpowie 9. Sprawdźmy potęgowanie i silnię

`3^2*4!;`

Wynik to oczywiście 216.

Program ostatnią wartość wyjścia zapamiętuje w zmiennej „%” (procent).

Jest to domyślna zmienna używana w poleceniach wywoływanych z menu.

Przykładowo po wprowadzeniu macierzy kwadratowej wybierzmy z menu

`Algebra > Wyznacznik`, co odpowiada wejściu

`determinant(%);`

```
(%i1) 2+2*4-1;  
(%o1) 9  
(%i2) 3^2*4!;  
(%o2) 216  
(%i3) %-16;  
(%o3) 200  
(%i4) %o2-16;  
(%o4) 200
```

Rysunek 4

My wykorzystamy zmienną „%” inaczej – odwołamy się do niej jak do liczby

`%-16;`

otrzymamy 200. Co ciekawe możemy odwołać się do dowolnego poprzedniego wejścia albo wyjścia wpisując jego numer, np.

`%o2-16;`

Macierze

W wxMaximie możemy wygodnie dokonywać działań na macierzach typu dodawanie, odejmowanie, mnożenie przez siebie i liczbę, wyznaczać macierz odwrotną, wyznacznik i rząd, itd.

Zacznijmy od wprowadzenia macierzy – tu sprawę ułatwi nam menu programu. Wybieramy `Algebra`

```
(%i5) Q: matrix(  
      [1,2,0],  
      [-1,4,-5],  
      [0,-5,4]  
);  
(%o5) 
$$\begin{bmatrix} 1 & 2 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$

```

Rysunek 5

`> Wprowadź macierz`. Wpisujemy rozmiar macierzy, typ najczęściej zostawiamy bez zmian („zwykła”) i nadajemy opcjonalnie nazwę. Warto nazwać macierz, gdyż dzięki temu po pewnym czasie będziemy mogli odwołać się do tej macierzy i wykonać na niej jakieś operacje.

Przykładowo wybierzmy zwykłą macierz 3x3 o nazwie „Q” (w wxMaximie możemy używać także małych liter w nazwach) `> OK`. Wypełniamy ją liczbami, przechodząc do kolejnych komórek klawiszem Tab(ulatora), a liczby ujemne wpisując jako np. „-2”, a nie „2-”. Kończymy wybierając OK. Na ekranie zobaczymy zrzut jak na rysunku 5.

Program sam wpisał wejście, co może nam się przydać, gdy zrobimy „literówkę” w macierzy. Nie musimy jeszcze raz wybierać z menu formularza macierzy i go uzupełniać. Wystarczy skopiować całe poprzednie wejście, wkleić, poprawić błąd, np.

```
INNANAZWA: matrix(
[7,7,0],
[-1,4,-5],
[0,-5,4]
);
```

i nacisnąć Shift+Enter. Wpisując ręcznie komendę tworzenia macierzy nie musimy rozdzielać kolejne wiersze klawiszem Enter – całość możemy wpisać w jednej linii, ważne tylko by nie pominąć żadnego przecinka czy nawiasu. Powyższy zapis możemy zastąpić komendą:

```
INNANAZWA: matrix([7,7,0],[-1,4,-5],[0,-5,4]);
```

Dzięki temu skorygowaliśmy wartości i utworzyliśmy nową macierz pod nową nazwą.

Jeśli zapomnieliśmy nazwać wpisaną przed chwilą macierz – najprościej jak użyjemy operatora przypisania „:” (dwukropka) i zmienną „%”, np. nazwijmy ją „ALGEBRA”

```
ALGEBRA:%;
```

Możemy sprawdzić co program zapamiętał pod określoną zmienną (nazwą) poprzez podanie zmiennej. Pamiętajmy o tym, że ważna jest wielkość liter:

```
ALGEBRA;
```

Całość będzie wyglądać jak na zrzucie obok (rysunek 6).

```
(%i6) INNANAZWA: matrix(
[7,7,0],
[-1,4,-5],
[0,-5,4]
);
```

```
(%o6) 
$$\begin{bmatrix} 7 & 7 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$

```

```
(%i7) ALGEBRA:%;
```

```
(%o7) 
$$\begin{bmatrix} 7 & 7 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$

```

```
(%i8) ALGEBRA;
```

```
(%o8) 
$$\begin{bmatrix} 7 & 7 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$

```

Rysunek 6

Macierz odwrotna

Użyjmy wcześniej utworzoną macierz „Q”. Jeśli po jej wprowadzeniu używaliśmy innych komend wywołajmy ją ponownie poleceniem

```
Q;
```

Jeśli w poprzednim wejściu wpisaliśmy macierz nie musimy jej wywoływać. Następnie wybieramy menu **Algebra** > **Macierz odwrotna** i tak oto jednym kliknięciem uzyskujemy Q^{-1} . Program sam uzupełnił wejście wpisując

```
invert(%);
```

Przypomnę, że zmienna „%” oznacza ostatnie (tuż powyżej) wyjście. Alternatywnie możemy ręcznie wpisać komendę invert(nazwa_macierzy). Obliczmy odwrotność macierzy Q:

```
invert(Q);
```

Powinniśmy uzyskać ten sam wynik co powyżej. Odkrywamy tutaj kolejną przewagę wxMaxima – program sam po wpisaniu pierwszego nawiasu dopisuje drugi, zamykający nawias (opcję tą zmienić możemy w menu **Edycja** > **Preferencje** > **Opcje** > **Match parenthesis in text controls** – automatycznie zamyka nawiasy w polach tekstowych).

Zapamiętajmy ten wynik w zmiennej „W”:

```
W:%;
```

Jeśli teraz wybierzemy z menu **Algebra** > **Macierz odwrotna** otrzymamy odwrotność odwrotności, czyli po prostu macierz „Q”. Całość widzimy na rysunku 7.

```
(%i9) Q;
```

```
(%o9) 
$$\begin{bmatrix} 1 & 2 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$

```

```
(%i10) invert(%);
```

```
(%o10) 
$$\begin{bmatrix} 9 & 8 & 10 \\ -4 & -4 & -5 \\ -5 & -5 & -6 \end{bmatrix}$$

```

```
(%i11) invert(Q);
```

```
(%o11) 
$$\begin{bmatrix} 9 & 8 & 10 \\ -4 & -4 & -5 \\ -5 & -5 & -6 \end{bmatrix}$$

```

```
(%i12) W:%;
```

```
(%o12) 
$$\begin{bmatrix} 9 & 8 & 10 \\ -4 & -4 & -5 \\ -5 & -5 & -6 \end{bmatrix}$$

```

```
(%i13) invert(%);
```

```
(%o13) 
$$\begin{bmatrix} 1 & 2 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$

```

Rysunek 7

Mnożenie, dodawanie i odejmowanie macierzy

Możemy sprawdzić, czy wxMaxima nie zrobiła błędu i wymnożymy macierz „Q” razy „W”. W celu mnożenia macierzy przez siebie używamy operatora „.” (kropka), użycie znaku „*” (gwiazdka) jest błędem

```
Q.W;
```

Mnożąc macierz Q razy jej odwrotność otrzymamy macierz jednostkową. Ten sam efekt uzyskamy po wpisaniu komendy

```
Q.invert(Q);
```

Możemy też przećwiczyć mnożenie różnych macierzy i sprawdzić, że $A.B \neq B.A$, czy też przekonać się, że nie można wymnożyć macierzy 2x2 przez 3x3 (błąd „attempt to multiply nonconformable matrices”).

Operator „*” (gwiazdka) należy użyć gdy chcemy pomnożyć liczbę razy macierz

```
3*Q;
```

Każdy element macierzy zostanie pomnożony x3. Dodajmy dwie macierze

```
Q+W;
```

i odejmijmy od macierzy W nią samą, co oczywiście powinno dać macierz zerową

```
W-W;
```

Wyznacznik, rząd i transpozycja

Po wprowadzeniu macierzy kwadratowej nazwijmy ją „Q” i wybierzmy z menu **Algebra** > **Wyznacznik**. Program wpisał do wejścia odpowiadającą temu komendę

```
determinant(%);
```

Oczywiście taki sam efekt da wywołanie

```
determinant(Q);
```

Rząd macierzy nie znajdziemy w menu, na szczęście komenda jest bardzo prosta. Rząd macierzy Q obliczymy wpisując

```
rank(Q);
```

Transpozycję macierzy wyznaczymy wybierając z menu **Algebra** > **Transponuj macierz**, co odpowiada komendzie

```
transpose(Q);
```

Myki i tricki

Na zakończenie zgłębiania wiedzy tajemnej proponuję trzy ułatwienia, które możemy uruchomić w opcjach wxMaximy.

W menu **Maxima** > **Panele** > **Historia** uruchamiamy po prawej stronie okna programu historię naszych komend. Najnowsze są na górze, starsze poniżej. Po dwukrotnym kliknięciu którejś z pozycji historii wywołujemy ją na ekranie i jeśli chcemy uruchamiamy klawiszami Shift+Enter.

Jeśli zapomnieliśmy użyte zmienne (nazwy macierzy) wywołamy je w menu **Maxima** > **Pokaż**

zmienną lub za pomocą komendy

```
values;
```

W wxMaximie możemy zmienić wielkość wyświetlanej czcionki w menu **Edycja** > **Ustaw przybliżenie** lub **Przybliź** lub **Oddal**.

```
(%i14) Q.W;  
      
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
  
(%o14)  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$   
(%i15) Q.invert(Q);  
      
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
  
(%o15)  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$   
(%i16) 3*Q;  
      
$$\begin{bmatrix} 3 & 6 & 0 \\ -3 & 12 & -15 \\ 0 & -15 & 12 \end{bmatrix}$$
  
(%o16)  $\begin{bmatrix} 3 & 6 & 0 \\ -3 & 12 & -15 \\ 0 & -15 & 12 \end{bmatrix}$   
(%i17) Q+W;  
      
$$\begin{bmatrix} 10 & 10 & 10 \\ -5 & 0 & -10 \\ -5 & -10 & -2 \end{bmatrix}$$
  
(%o17)  $\begin{bmatrix} 10 & 10 & 10 \\ -5 & 0 & -10 \\ -5 & -10 & -2 \end{bmatrix}$   
(%i18) W-W;  
      
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
  
(%o18)  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   
(%i19) Q;  
      
$$\begin{bmatrix} 1 & 2 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$
  
(%o19)  $\begin{bmatrix} 1 & 2 & 0 \\ -1 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$   
(%i20) determinant(%);  
(%o20) -1  
(%i21) determinant(Q);  
(%o21) -1  
(%i22) rank(Q);  
(%o22) 3  
(%i23) transpose(Q);  
      
$$\begin{bmatrix} 1 & -1 & 0 \\ 2 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$$
  
(%o23)  $\begin{bmatrix} 1 & -1 & 0 \\ 2 & 4 & -5 \\ 0 & -5 & 4 \end{bmatrix}$ 
```

Rysunek 8

Strony internetowe programów

- Maxima <http://maxima.sourceforge.net>
- wxMaxima <http://wxmaxima.sourceforge.net>

Najnowszą wersję tego artykułu znajdziecie pod adresem

<http://murowana.pl/szkola/wsb/wxmaxima.zip>

*Dominik
IiE WSB Poznań
wsb@murowana.pl*