Fundusze Europejskie Wiedza Edukacja Rozwój
Politechnika Wrocławska
Unia Europejska Europejski Fundusz Społeczny

„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Na prawach rękopisu

PAKIETY OBLICZENIOWE

Materiały dydaktyczne do kursu dla studentów I roku
Wydziału Mechaniczno-Energetycznego

Opracowanie:

Adam Ruziewicz
Józef Rak

Wrocław, 2019

Spis treści

Wstęp

Niniejszy skrypt stanowi materiały pomocnicze do kursu Pakiety Obliczeniowe, przewidzianego dla studentów I roku Wydziału Mechaniczno-Energetycznego Politechniki Wrocławskiej, prowadzonego w formie ćwiczeń laboratoryjnych na komputerach.

Celem ćwiczeń jest zapoznanie studentów z możliwościami programów komputerowych wykorzystywanych do prostych i bardziej złożonych obliczeń inżynierskich oraz prezentacji wyników. Umiejętności zdobyte podczas kursu będą wykorzystywane przez studentów w toku studiów, zarówno jako wsparcie w realizacji projektów i innych zajęć laboratoryjnych, jak i podczas przygotowania prac inżynierskich.

W ramach zajęć przewidziana jest nauka programów *MS Excel* oraz *Mathcad* i ich możliwości w zakresie obliczeń inżynierskich, obróbki danych i prezentacji wyników obliczeń. Na podstawie różnego typu zadań obliczeniowych prowadzonych w *MS Excel* i *Mathcad* studenci będą mogli poznać możliwości obydwu programów, a także porównać ich zastosowania i funkcjonalności.

Niniejszy skrypt zawiera opisy 7 ćwiczeń laboratoryjnych w tym 2 z *MS Excel* i 5 z *Mathcad*. Na 8. ćwiczeniach przewidziane jest kolokwium zaliczeniowe, które zakres jest podany w ostatnim rozdziale skryptu.

Prezentowane materiały mają następujący układ:

- Opisy ćwiczeń znajdują się w osobnych rozdziałach, np.

Ćwiczenia 4 - Mathcad

- W podtytule rozdziału (ćwiczeń) podano ich zakres tematyczny, np.

Obliczenia symboliczne. Całkowanie i różniczkowanie. Wektory i macierze.

- Każde z omawianych zagadnień poprzedzone jest nagłówkiem, np.

Obliczenia symboliczne

- Zagadnienia dotyczące programów są omówione bezpośrednio na przykładzie zadań (np. **Zad. 1**), tzn. pod treścią każdego zadania przedstawiono sposób jego rozwiązania oraz zaprezentowano funkcje programów.
- Dodatkowo na końcu każdego rozdziału (każdych ćwiczeń) zebrana jest sama lista zadań, dla osób które preferują od razu rozwiązywać zadania samodzielnie. Listy zadań poszerzone są o zadania z gwiazdką * (np. **Zad. 9***), których rozwiązań nie przedstawiono w skrypcie.

Ćwiczenia 1 – MS Excel

Wprowadzenie. Formatowanie komórek. Adresy względne, bezwzględne i mieszane. Wykorzystanie funkcji matematycznych. Importowanie danych.

W programie MS Excel obliczenia na danych prowadzi się w pliku zwanym *Skoroszytem,* który może składać się z jednego lub większej liczby *Arkuszy*. Każdy *Arkusz* ma formę tabeli, w której komórkach wpisuje się dane oraz formuły (działania na danych). Arkusze wyświetlają się formie zakładek na dolnym pasku programu, a nowy arkusz można dodać po kliknięciu znaku **+** . Każde kolejne zadanie z ćwiczeń wykonaj w nowym arkuszu, zmieniając jego nazwę na kolejno *Zad. 1*, *Zad.2* …

Formatowanie komórek, adresy względne, bezwzględne i mieszane

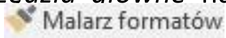
Zad. 1

Przygotuj arkusz kalkulacyjny do obliczenia rocznych kosztów energii elektrycznej różnych użytkowników (klientów), w zależności od wybranego dostawcy (sprzedawcy) energii. Taryfy dostawców różnią się, a w skład opłat wchodzą: opłata za energię, zależna od zużycia (*zł/kWh*) oraz opłata handlowa pobierana raz w miesiącu (*zł/m-c*). Należy również uwzględnić opłaty dystrybucyjne, które są niezależne od sprzedawcy energii. Należą do nich: składnik zmienny stawki sieciowej (*zł/kWh*), składnik stały stawki sieciowej (*zł/m-c*) oraz stawka opłaty abonamentowej (*zł/m-c*). Docelowo arkusz powinien zawierać tabele jak poniżej.

		Taryfy sprzedawców			
		Taulon	**Forkum**	**Elea**	**Egzerga**
Cena energii	*zł/kWh*	**0,3007**	**0,2383**	**0,2962**	**0,3400**
Opłata hadlowa	*zł/m-c*	**0,00**	**15,99**	**20,71**	**17,99**
Nr klienta	*kWh*	**Roczny koszt energii elektrycznej**			
1	**801**	487,32 zł	629,22 zł	732,24 zł	734,68 zł
2	**569**	366,13 zł	522,50 zł	612,09 zł	604,37 zł
3	**962**	571,43 zł	703,28 zł	815,62 zł	825,12 zł
4	**1114**	650,83 zł	773,20 zł	894,34 zł	910,49 zł
5	**1602**	905,76 zł	997,68 zł	1 147,08 zł	1 184,60 zł

Dobrze by obliczenia kosztów dla danej taryfy sprzedawcy znajdowały się pod cenami tej taryfy. Tabelę z opłatami dystrybucyjnymi można stworzyć w osobnym miejscu arkusza:

Opłaty dystrybucyjne			
Stawka sieciowa	składnik zmieny	**0,2217**	*zł/kWh*
	składnik stały	**4,80**	*zł/m-c*
Opłata abonamentowa		**0,94**	*zł/m-c*

Zacznij od przygotowania i sformatowania tabel. Przepisz jedynie dane liczbowe napisane pogrubioną czcionką. *Roczny koszt energii elektrycznej* ma być wynikiem obliczeń. Do graficznego sformatowania tabel można użyć funkcji z zakładki *Narzedzia główne* na *Pasku zadań*. Przydatnym narzędziem może się okazać *Malarz formatów* 

, który kopiuje format zaznaczonych komórek i przypisuje go do dowolnie dużego zaznaczonego obszaru. Dane w komórkach mogą mieć różne formaty, co jest ważne przy obliczeniach. Mogą to być formaty: *Liczbowe*, *Księgowe*, *Procentowe*, *Data*, *Czas*, *Naukowe*, *Tekstowe*, itp. Można je zmieniać na *Pasku zadań* lub po kliknięciu prawym przyciskiem myszy danej komórki (lub obszaru komórek) i wybraniu *Formatuj komórki*. Zadbaj, by taryfy i opłaty miały format liczbowy o odpowiedniej liczbie miejsc po przecinku, a obliczony koszt miał format księgowy.

Obliczenie rocznego kosztu energii elektrycznej wykonaj za pomocą wpisania jednej formuły w pierwsza komórkę i dwukrotnego przeciągnięcia formuł: w poziomie i w pionie. W tym celu należy odpowiednio odwołać się do komórek danych używając adresów względnych, bezwzględnych i mieszanych. Adres bezwzględny to taki którym odwołujemy się zawsze do jednej komórki mimo przeciągania formuł. Zapisuje się go dodając znak **$**. Przykładowo *$C$6*. Adresem mieszanym jest zablokowanie tylko wiersza *C$6* lub kolumny danej komórki *$C6*. Dla ułatwienia, adresy te można zmieniać przyciskiem **F4**. Adres względny zmienia komórkę, do której się odwołujemy wraz z przeciąganiem formuły.

Sprawdź, ile energii elektrycznej mógłby zużyć w ciągu roku klient 5, jeśli korzysta z taryfy firmy Taulon, a ma budżet 1000 zł. W tym celu skorzystaj z opcji *Szukaj wyniku* w zakładce *Dane* → *Analiza warunkowa*. Zaznacz wynik czerwoną czcionką.

Importowanie danych

Zad. 2

Butla o objętości $V=11\ \mathrm{l}$ zawierająca hel, na skutek nieszczelności zaworu traciła gaz. Przez miesiąc rejestrowano mierzone ciśnienie gazu w butli i zapisano w pliku tekstowym (*dane.txt*). Stwórz arkusz, w którym zostaną zaimportowane dane z pliku tekstowego, a następnie obliczony ubytek masy gazu znajdującego się w butli. Skorzystaj z równania stanu gazu doskonałego:

$$p\,V = m\frac{MR}{M}\,T$$

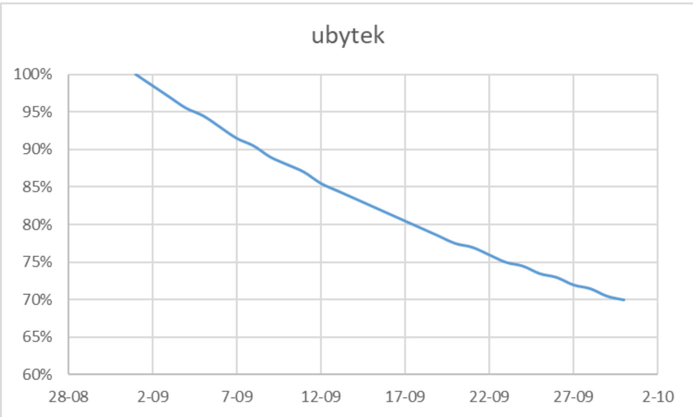
Należy pamiętać, że w powyższym równaniu trzeba podstawiać wartości w jednostkach SI, tj. Pa, $\mathrm{m^3}$, kg, K.

- Uniwersalna stała gazowa: $MR=8315\,\frac{\mathrm{J}}{\mathrm{kmol \cdot K}}$
- Masa molowa helu: $M=4\,\frac{\mathrm{kg}}{\mathrm{kmol}}$

- Temperatura: T=293 K

W celu zaimportowania danych przejdź do *Plik* → *Otwórz* wybierając plik *dane.txt*. Otworzy się kreator importu tekstu, w którym jest kilka opcji do wyboru. Należy przejść kroki kreatora wybierając tekst oddzielany znakami tabulacji. Po zakończeniu otworzy się nowy plik. Przekopiuj dane i wklej do arkusza *Zad.2* w skoroszycie *Cwiczenia1*. Dopilnuj, aby kolumna miała format czasu RRRR-MM-DD. Zmień format wyświetlania ciśnienia z naukowego na liczbowy. Następnie utwórz trzecią kolumnę w której obliczona zostanie masa gazu w butli każdego dnia (z dokładnością 3. miejsca po przecinku). W czwartej kolumnie oblicz dodatkowo procentowy ubytek masy (ile procent zostało w butli). Przy dwukrotnym kliknięciu na krzyżyk w prawym dolnym rogu komórki formuła zostanie automatycznie przeciągnięta do ostatniego wypełnionego po lewej stronie wiersza.

Stwórz wykres zawartości procentowej masy gazu w butli od czasu. W tym celu wstaw *wykres punktowy*, najlepiej *z wygładzonymi liniami* z zakładki *Wstawianie*. Gdy najpierw z przyciskiem **Ctrl** zaznaczysz odpowiednie kolumny wraz z etykietami, wykres powinien zaprezentować dane automatycznie. Przytrzymując Ctrl+Shift+↓ można łatwo zaznaczyć całą kolumnę danych. Sformatuj osie wykresu tak, aby zakres osi y zawierał się od 60% do 100% a data była wyświetlana w formacie DD-MM.



Wykorzystanie funkcji matematycznych

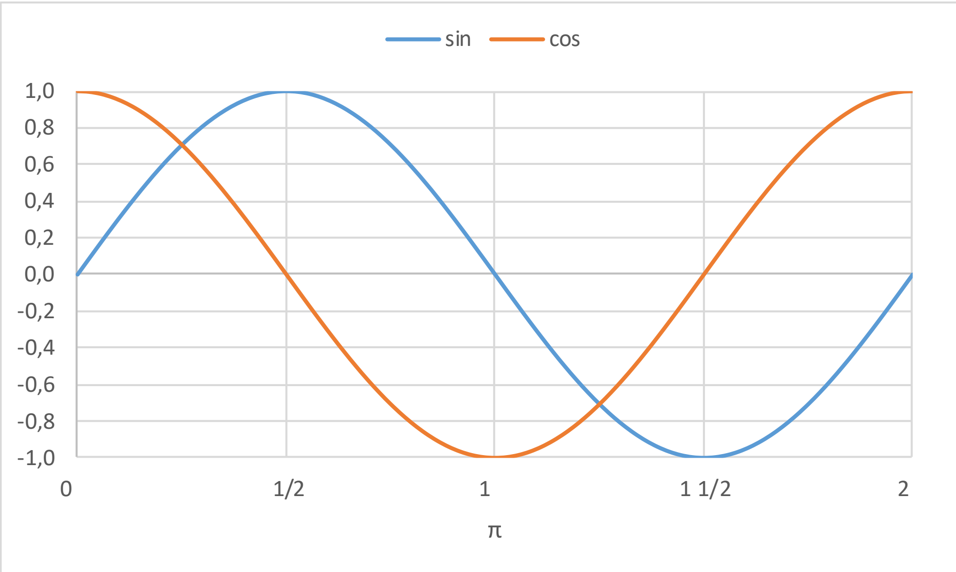
Zad. 3

Stwórz arkusz w którym obliczone zostaną funkcje trygonometryczne dla kątów w przedziale 0°-360° co 5°. W tym celu skorzystaj z wbudowanych funkcji matematycznych.

W przypadku braku znajomości składni istniejących funkcji, można znaleźć je na liście znajdującej się w zakładce *Formuły*. Są tam funkcje *Finansowe*, *Logiczne*, *Tekstowe*, *Matematyczne* i inne. Tym razem skorzystamy z funkcji matematycznych. Funkcje trygonometryczne obliczają wartości dla kąta w radianach, dlatego warto przeliczyć kąt w stopniach używając funkcji *RADIANY*. Wpisując funkcje w komórce po znaku **=** i otwarciu nawiasu, program sam podpowiada jakie dane wejściowe są potrzebne. W arkuszu oblicz wartości funkcji *sinus*, *cosinus*, *tangens* i *cotangens*, a także wykorzystując funkcję *PI()* oblicz kąt jako wielokrotność liczby π w ułamku zwykłym.

°	rad	π	sin	cos	tg	ctg
0	0,000	0	0,000	1,000	0	#DZIEL/0!
5	0,087	1/36	0,087	0,996	0,087489	11,43005
10	0,175	1/18	0,174	0,985	0,176327	5,671282
15	0,262	1/12	0,259	0,966	0,267949	3,732051
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Następnie stwórz wykres funkcji sin i cos w zależności od kąta jako wielokrotności liczby π.



Lista zadań 1

Zad. 1

Przygotuj arkusz kalkulacyjny do obliczenia rocznych kosztów energii elektrycznej różnych użytkowników (klientów), w zależności wybranego dostawcy (sprzedawcy) energii. Taryfy dostawców różnią się, a w skład opłat wchodzą: opłata za energię, zależna od zużycia (*zł/kWh*) oraz opłata handlowa pobierana raz w miesiącu (*zł/m-c*). Należy również uwzględnić opłaty dystrybucyjne, które są niezależne od sprzedawcy energii. Należą do nich: składnik zmienny stawki sieciowej (*zł/kWh*), składnik stały stawki sieciowej (*zł/m-c*) oraz stawka opłaty abonamentowej (*zł/m-c*).

Zad. 2

Butla o objętości $\mathrm{V}=11\ \mathrm{l}$ zawierająca hel, na skutek nieszczelności zaworu traciła gaz. Przez miesiąc rejestrowano mierzone ciśnienie gazu w butli i zapisano w pliku tekstowym (*dane.txt*). Stwórz arkusz, w którym zostaną zaimportowane dane z pliku tekstowego, a następnie obliczony ubytek masy gazu znajdującego się w butli. Skorzystaj z równania stanu gazu doskonałego.

Zad. 3

Stwórz arkusz w którym obliczone zostaną funkcje trygonometryczne dla kątów w przedziale 0°-360° co 5°. W tym celu skorzystaj z wbudowanych funkcji matematycznych. W arkuszu oblicz wartości funkcji sinus, cosinus, tangens i cotangens, a także wykorzystując funkcję *PI()* oblicz kąt jako wielokrotność liczby π w ułamku zwykłym. Następnie stwórz wykres funkcji sin i cos w zależności od kąta jako wielokrotności liczby π.

Ćwiczenia 2 – MS Excel

Praca z danymi. Tworzenie wykresów. Budowanie prostych modeli fizycznych. Solver.

Celem ćwiczenia jest praca z danymi w arkuszu kalkulacyjnym *MS Excel* na przykładzie modeli zjawisk fizycznych.

Zad. 1

Przygotuj arkusz kalkulacyjny symulujący strzelanie pocisku z katapulty, wykorzystując równania rzutu ukośnego. Uwzględnij równania współrzędnych pocisku:

- bez oporu powietrza:

$$x(t) = v_{0x}\, t$$

$$y(t) = v_{0y}\, t - \frac{g\, t^2}{2}$$

- z oporem powietrza (równania uwzględniają siłę oporu powietrza proporcjonalną do prędkości $F_x = -k\, v_x$, $F_y = -k\, v_y$):

$$x(t) = \frac{v_{0x}}{B}\left(1 - e^{-Bt}\right)$$

$$y(t) = \left(\frac{v_{0y}}{B} + \frac{g}{B^2}\right)\left(1 - e^{-Bt}\right) - \frac{g\, t}{B}$$

gdzie:

$v_{0x} = v_o \cos(\alpha)$	-	składowa x prędkości początkowej
$v_{0y} = v_o \sin(\alpha)$	-	składowa y prędkości początkowej
$g = 9{,}81\,\frac{m}{s^2}$	-	przyspieszenie ziemskie
$B = \frac{k}{m}\,, s^{-1}$	-	jednostkowy opór powietrza

Dane:

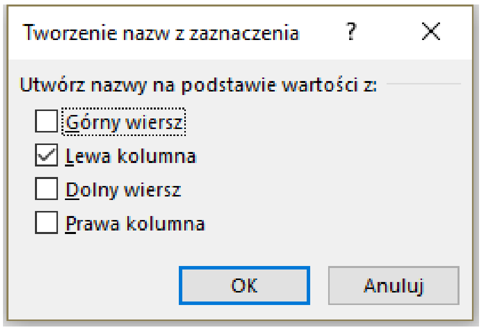
$m = 10\ kg$	-	masa pocisku
$k = 1\,\frac{kg}{s}$	-	opór powietrza
$\alpha = 40°$	-	kąt wystrzału pocisku
$v_o = 55\,\frac{m}{s}$	-	prędkość początkowa

Stwórz tabelkę ze stałymi przedstawionymi powyżej i utwórz nazwy tych stałych odnoszące się do komórek z ich wartościami. Wygeneruj i sformatuj wykres, który pokaże kierunek wystrzału pocisku oraz trajektorię lotu pocisku przy uwzględnieniu oporu powietrza i bez. Wyszukaj współrzędne pocisku po 2 sekundach lotu. Znajdź maksymalną wysokość pocisku i odpowiadającą mu współrzędną x, a także zasięg pocisku. Przy pomocy *Solvera* określ różne warianty wartości kąta i prędkości początkowej, dla których maksymalna wysokość wynosi $50\ m$.

Zacznij od stworzenia tabelki , w której zdefiniujesz stałe potrzebne do obliczenia współrzędnych pocisku. Zwróć uwagę, że część z nich jest obliczana formułami. Przykładowa tabelka może wyglądać następująco:

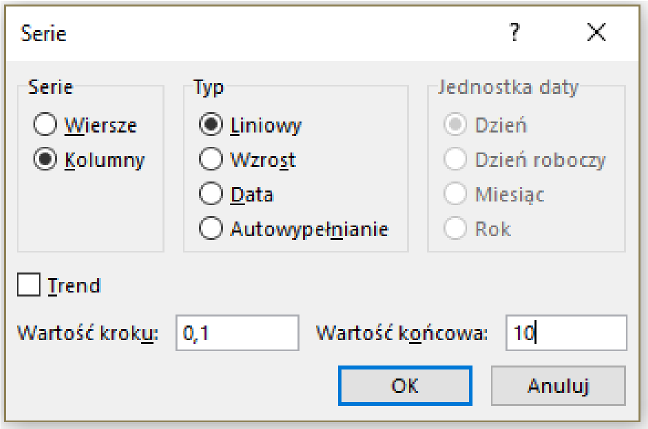
DANE	nazwa	wartość	jednostka
masa pocisku	**m**	10	kg
przyspieszenie ziemskie	**g**	9,81	m/s2
wpsółczynnik oporu	**k**	1	kg/s
opór relatywny	**B**	0,1	1/s
kąt wurzutu	**alfa**	40	deg
prędkość początkowa	**Vo**	55	m/s
prękośc początkowa x	**Vox**	42,13	m/s
prękośc początkowa y	**Voy**	35,35	m/s

Abu ułatwić sobie wpisywanie skomplikowanych formuł, nazwij zmienne tak, aby zamiast adresu komórek z wartościami można stosować ich nazwy. W tym celu przejdź do zakładki *Formuły* $\rightarrow$ *Nazwy zdefiniowane* i wybierz opcję *Utwórz z zaznaczenia*. Jeśli wcześniej zaznaczysz komórki nazw i wartości w układzie jak powyżej, w okienku, które się otworzy zaznacz aby program utworzył nazwy na podstawie lewej kolumny.



Możesz sprawdzić, że po kliknięciu w jedną z komórek na pasku zamiast adresu pojawia się zdefiniowana nazwa.

W następnym kroku wypełnij danymi kolumnę wartości czasu t, dla którego będą obliczane współrzędne pocisku. W tym celu z zakładki *Narzędzia główne* $\rightarrow$ *Edytowanie* wybierz menu *Wypełnij*  i opcję *Seria danych…* . Niech to będzie czas od 0 do 10 sekund z krokiem 0,1 sekundy. Gdy chcemy serię danych w kolumnie musimy zaznaczyć *Kolumny*. Typ *Liniowy* dodaje wartość kroku do kolejnych komórek natomiast *Wzrost* mnoży komórki przez nią. Komórka, która jest zaznaczona powinna zawierać wartość początkową. Wartość kroku i wartość końcową zaznacza się w okienku.



Oblicz współrzędne $x(t)$ i $y(t)$ pocisku przy uwzględnieniu oporu oraz bez oporu powietrza. Wpisując formułę możesz posłużyć się zdefiniowanymi nazwami, co pozwoli uniknąć użycia wielu adresów bezwzględnych. Przykładowo formuła na $x(t)$ dla rzutu z oporem wygląda następująco: *=Vox/B*(1-EXP(-B*G8))*, gdzie $G8$ jest adresem wartości czasu t. Dodatkowo, korzystając z funkcji warunkowej *JEŻELI* spowoduj zatrzymanie pocisku na powierzchni. Funkcja ma następującą składnię:

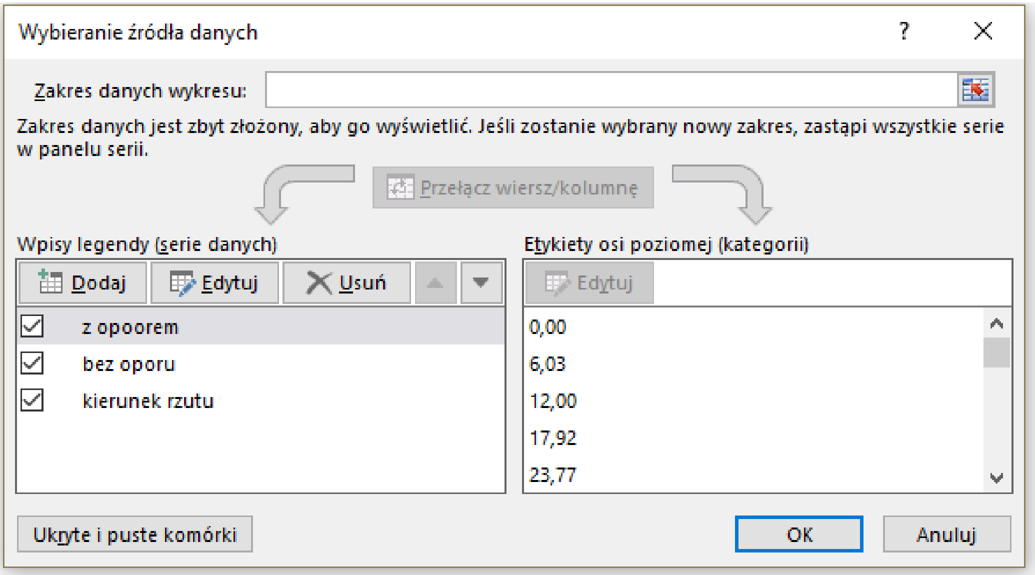
JEŻELI(**test_logiczny**; [wartość_jeżeli_prawda]; [wartość_jeżeli_fałsz])

Wystarczy w teście logicznym sprawdzić czy $y(t) > 0$ i w innym przypadku przypisać wartość 0. Tabelka obliczeń np. może wyglądać następująco:

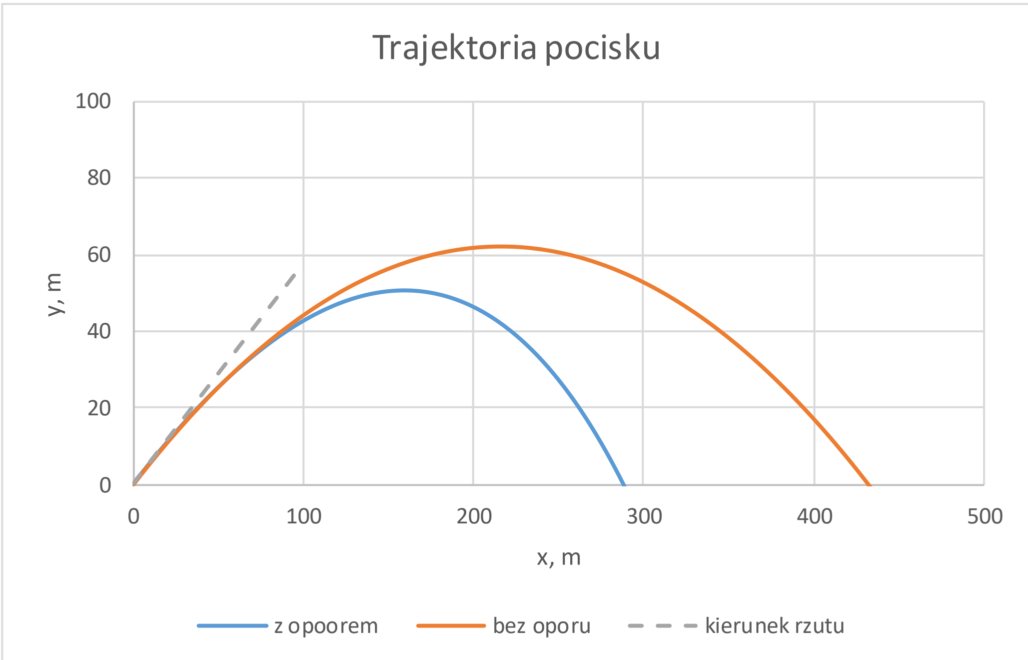
	Z oporem			**Bez oporu**		
czas, s	współrzędne, m			współrzędne, m		
t	**x(t)**	**y(t)**	**y+(t)**	**x(t)**	**y(t)**	**y+(t)**
0,0	0,00	0,00	0,00	0,00	0,00	0,00
0,1	4,19	3,47	3,47	4,21	3,49	3,49
…	…	…	…	…	…	…

Zaznaczywszy dane $x(t)$ i $y(t)$ wstaw wykres punktowy z wygładzonymi liniami. Po kliknięciu na wykres pojawią się dodatkowe zakładki *Projektowanie* i *Formatowanie*. W zakładce *Projektowanie* wybierz *Zaznacz dane*, aby dodać kolejne serie danych. Otworzy się okienko, w którym można dodawać serie danych, a także edytować obecne. Dodaj serię danych współrzędnych pocisku bez oporu.

Dodaj także dane potrzebne do wyświetlenia kierunku wyrzutu. W tym celu stwórz dodatkową tabelkę dla $x = (0,100)$ i y zależnych od kąta α. Edytuj etykiety danych, co przyda się do stworzenia legendy na wykresie.



Po kliknięciu na osie wykresu, sformatuj wartości osi oraz ustaw ich minima i maksima i jednostki siatki. W zakładce *Projektowanie* z menu *Dodaj element wykresu* można wybrać elementy, które będą wyświetlane na wykresie. Dodaj tytuły osi, tytuł wykresu i legendę. Wykres powinien wyglądać następująco:

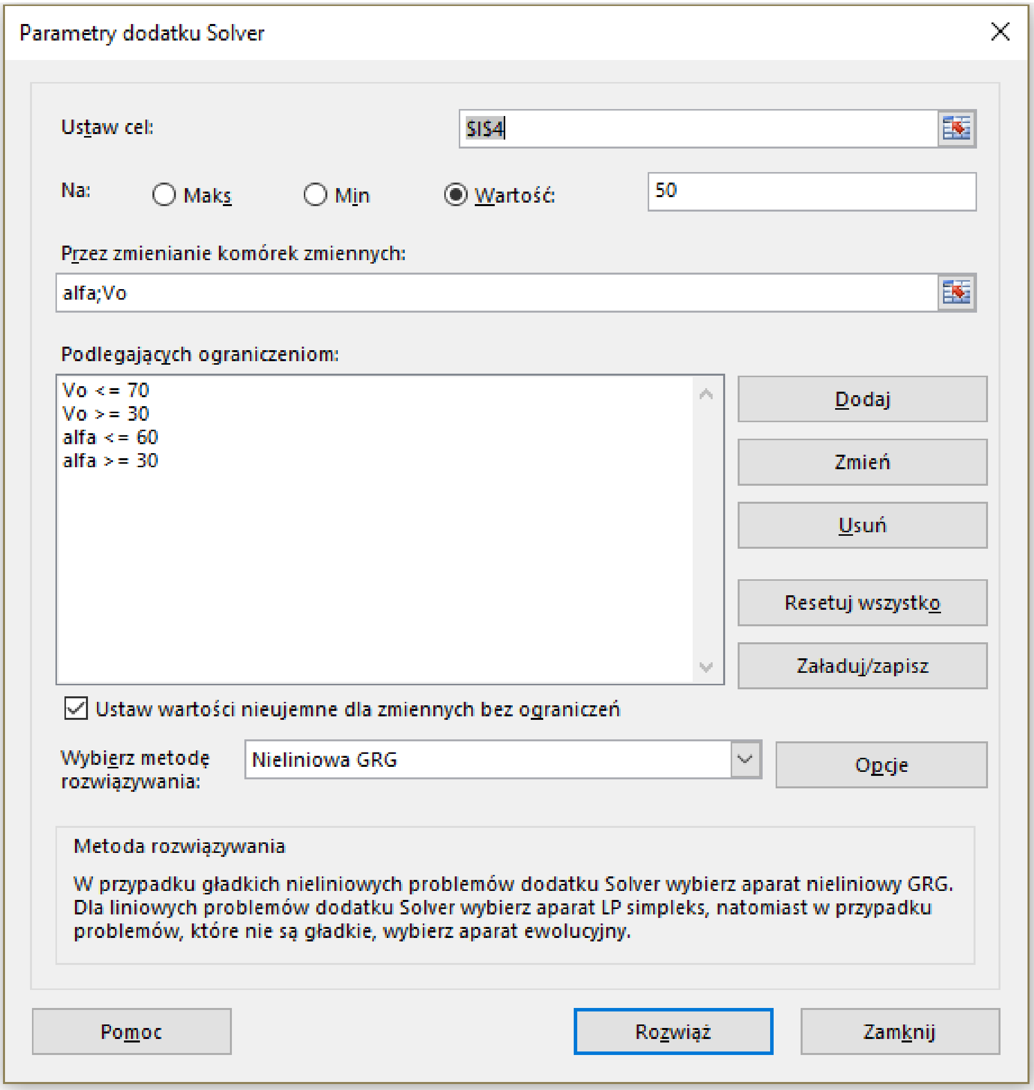


Aby wyświetlić współrzędne pocisku po 2 sekundach lotu skorzystaj z funkcji *WYSZUKAJ.PIONOWO*. Funkcja wyszukuje wartość w pierwszej kolumnie wybranej tabeli i zwraca wartość z tego samego wiersza i wskazanej kolumny (należy podać jej numer w tabeli).

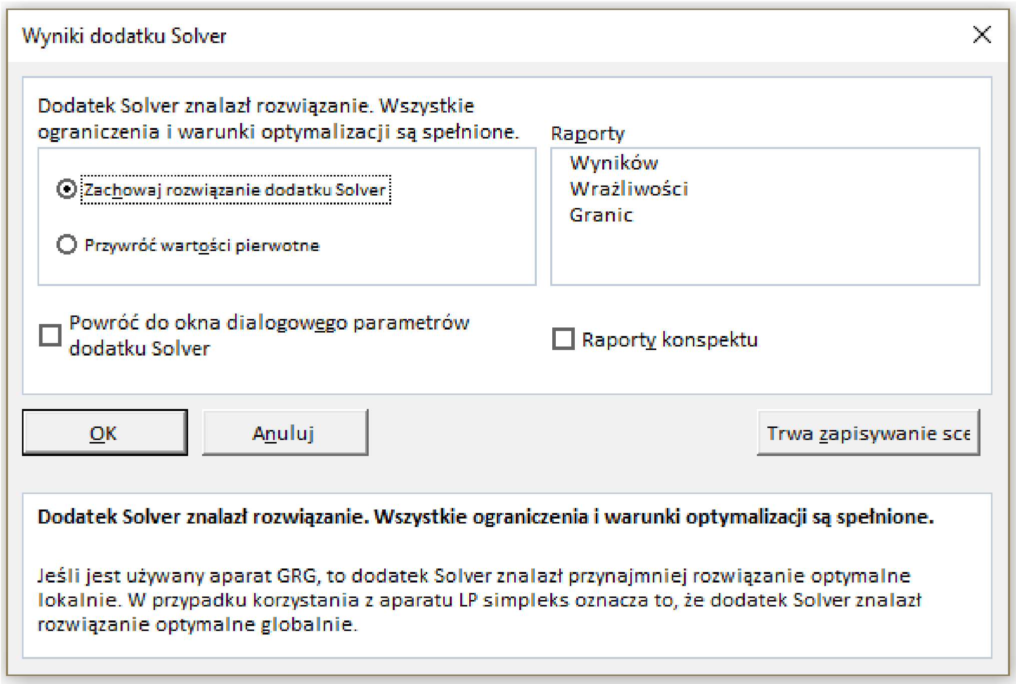
Maksymalną wysokość pocisku znajdź korzystając z funkcji *MAX*. Odpowiadającą jej odległość x można by znaleźć funkcją *WYSZUKAJ.PIONOWO* jeśli kolumna x byłaby po prawej stronie od y. W innym przypadku trzeba skorzystać z funkcji *INDEKS* oraz *PODAJ.POZYCJĘ*. Korzystając z Pomocy zaproponuj rozwiązanie.

Zasięg pocisku znajdziesz w podobny sposób, z tą różnicą, że szukanym wierszem będzie ten, w którym *y(t)=0*. Wykorzystaj dane z dodatnimi wartościami *y* (w której użyłeś funkcji *JEŻELI*).

Solver jest narzędziem które numerycznie znajduje rozwiązanie lub jedno z rozwiązań dla określonych warunków brzegowych. Aby rozwiązać zadanie, z zakładki *Dane* otwórz *Solver*.



W otwartym oknie ustaw cel. Celem jest aby wyliczane formułą maksymalne położenie było równe 50. Jako cel można też ustawić znalezienie minimum albo maksimum. Ustaw, że zmieniane mają być wartości kąta oraz prędkości początkowej. Dodatkowo dodaj pewne ograniczania, np. żeby kąt i prędkość były z zakresów $\alpha \in (30°, 60°)$, $v_o \in \left(30\frac{m}{s}, 70\frac{m}{s}\right)$. Po kliknięciu *Rozwiąż*, program obliczy wartości zmienianych komórek i wyświetli okno:



Można zachować rozwiązanie w arkuszu lub wrócić do pierwotnych wartości. Zapisz scenariusz klikając *Trwa zapisywanie Scenariusza* i nazwij go *a*. Zmieniając wartości początkowe kąta i prędkości wykonaj obliczenia jeszcze kilka razy i zapisz kolejne scenariusze: *a, b, c…* . Z zakładki *Dane* $\rightarrow$ *Analiza warunkowa* wyświetl *Podsumowanie scenariuszy* wybierając komórki wynikowe, które mają zostać pokazane.

Lista zadań 2

Zad. 1

Przygotuj arkusz kalkulacyjny symulujący strzelanie pocisku z katapulty, wykorzystując równania rzutu ukośnego. Uwzględnij równania współrzędnych pocisku:

- bez oporu powietrza:
- z oporem powietrza (uwzględniając siłę oporu powietrza proporcjonalną do prędkości $F_x = -k\, v_x$, $F_y = -k\, v_y$):

Stwórz tabelkę ze stałymi przedstawionymi powyżej i utwórz nazwy tych stałych odnoszące się do komórek z ich wartościami. Wygeneruj i sformatuj wykres, który pokaże kierunek wystrzału pocisku oraz trajektorię lotu pocisku przy uwzględnieniu oporu powietrza i bez. Wyszukaj współrzędne pocisku po 2 sekundach lotu. Znajdź maksymalną wysokość pocisku i odpowiadającą mu współrzędną x, a także zasięg pocisku. Przy pomocy *Solvera* określ różne warianty wartości kąta i prędkości początkowej, dla których maksymalna wysokość wynosi $50\ m$.

Zad. 2*

Dla danych z poprzedniego zadania i przypadku uwzględniającego opór powietrza oblicz i zaprezentuj na wykresie trajektorię lotu pocisku wystrzelonego pod różnymi kątami: α_1=30°, α_2=45°, α_3=50°. Wszystkie tory loty powinny znajdować się na wspólnym wykresie współrzędnych x, y.

Zad. 3*

Analogicznie do zad. 2 stwórz wykres trajektorii lotu pocisku dla różnych wartości oporu powietrza: $k_1=0{,}1\,\frac{kg}{s}$, $k_2=1\,\frac{kg}{s}$, $k_3=5\,\frac{kg}{s}$. Pozostałe dane przyjmij jak w zad. 1.

Ćwiczenia 3 - Mathcad

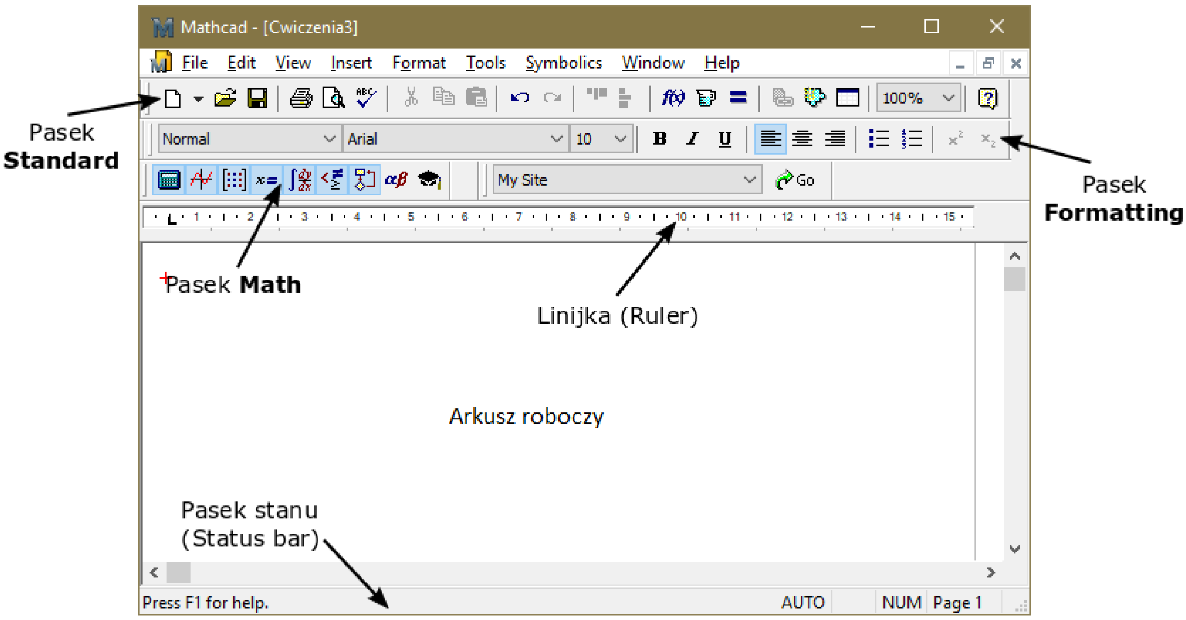
Wprowadzenie. Obliczenia, typy zmiennych, jednostki miar, funkcje, operatory logiczne.

Wprowadzenie - Interfejs

Mathcad 15.0 jest programem służącym do obliczeń inżynierskich. Interfejs programu składa się z menu, pasków narzędzi i arkusza roboczego przypominającego białą kartkę, w którym zarówno wpisuje się tekst i prowadzi obliczenia. Po otwarciu programu zapisz arkusz jako *Cwiczenia3.xmcd*. Z *Menu* → *View* można włączyć różne paski narzędzi *Toolbars* (cześć z nich jest włączona domyślnie). Do najważniejszych należą:

- Pasek *Standard* - standardowe narzędzia
- Pasek *Formatting* - formatowanie tekstu
- Pasek *Math* - palety symboli i funkcji matematycznych

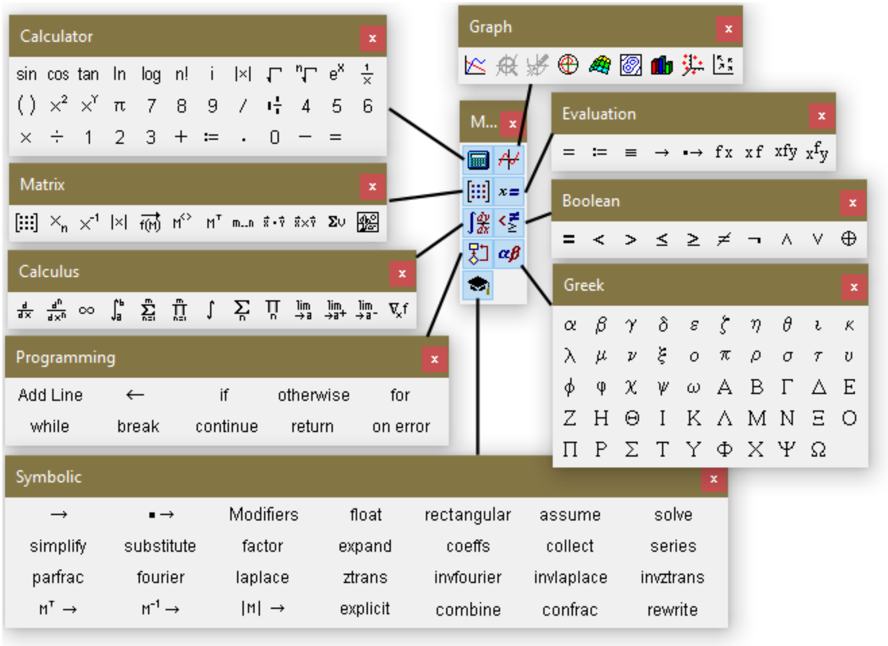
Możemy również włączyć Pasek stanu *Status bar* oraz dla ułatwienia edycji dokumentu - Linijkę *Ruler*.



Palety symboli z paska *Math* są przydatne podczas obliczeń i rozwiązywania różnych zagadnień. Zakres symboli jest szeroki i zawiera palety:

- *Calculator* – podstawowe operacje matematyczne
- *Evaluation* – przypisywanie wartości zmiennym, wyświetlanie wyniku
- *Graph* – wykresy 2D i 3D
- *Matrix* – operatory wektorów i macierzy
- *Boolean* – operatory logiczne
- *Calculus* – operatory rachunkowe (całki, pochodne, granice)

- *Greek* – greckie litery
- *Symbolic* – operatory do obliczeń symbolicznych
- *Programming* – operatory służące do programowania



Z *Menu* → *View* można także edytować nagłówek i stopkę dokumentu *Header and Footer*. Nie będą one widoczne w arkuszu roboczym, a dopiero na wydruku dokumentu.

Zad. 0

Ustaw nagłówek, w którym po lewej będzie Twoje imię i nazwisko, po środku tytuł pliku, a po prawej data (dwa ostatnie wstawiane automatycznie). W prawej części stopki powinien się wyświetlać numer strony oraz liczba stron dokumentu w formacie *numer strony/liczba stron*. Dodaj ramkę wokół strony dokumentu (arkusza roboczego). Sprawdź wygląd konspektu w podglądzie wydruku *Print preview*.

Napisz tytuł ćwiczenia „**Ćwiczenia 3 – Wprowadzenie do programu Mathcad**" wyśrodkowany, czcionką *Calibri 16*.

Mathcad domyślnie po rozpoczęciu pisania otwiera *region matematyczny*. *Region tekstowy* uruchamia się przyciskiem cudzysłowu **„** na klawiaturze lub po pierwszym kliknięciu spacji podczas wpisywania tekstu.

Każde zadanie zaczynaj od nagłówka **Zad. …** w domyślnym stylu *Heading 1*.

Typy zmiennych

Zad. 1

Przepisz do arkusza następujące wyrażenia matematyczne:

$$x_1^2+x_2+5$$

$$\frac{1}{2}+\frac{1}{3}=\frac{5}{6}$$

$$\sqrt[3]{27}=3$$

$$\ln(e^7)=7$$

$$\frac{1}{\frac{1}{10^2}+\frac{3}{9^{1.4}}+\frac{1}{17+3^{0.4}}}\cdot 10^{-5}=$$

$$|5-7|=2$$

Zwróć uwagę, że po użyciu klawisza = program zwraca wynik. Po jego dwukrotnym kliknięciu można ustawić jego format. Do wyboru są: ogólny *General*, zmiennoprzecinkowy *Decimal*, naukowy *Scientific*, inżynierski *Engineering* oraz ułamkowy *Fraction*. Można również ustawić liczbę miejsc wyświetlanych po przecinku.

Indeksy dolne przy symbolach dodaje się przy pomocy **kropki**. W obszarze matematycznym przy pomocy **spacji** można przeskakiwać między poziomami danego wyrażenia, np. ułamka. Jest to spore ułatwienie przy wpisywaniu skomplikowanych równań. Należy pamiętać też, że *Mathcad* używa notacji anglosaskiej, w której część ułamkową liczby pisze się po **kropce**. Wybrane operatory matematyczne zebrano w *Tab. 1 Skróty klawiszowe przydatne w pisaniu operacji matematycznych* pod koniec tych ćwiczeń.

Zad. 2

Pojazd porusza się ruchem prostoliniowym jednostajnie przyspieszonym z przyspieszeniem 2 m/s^2. Po jednej sekundzie prędkość pojazdu wynosiła 6 m/s. Jaka była prędkość początkowa pojazdu?

Zadanie należy rozwiązać posługując się zmiennymi, czyli obiektami określonymi danym symbolem i przechowującymi różne wartości. W *Mathcad* istnieją cztery typy zmiennych, które w programie wpisuje się następująco:

- skalarną - $a := 5$
- zakresową - $b := 1,2..100$
- znakową - $c := \text{"}tekst\text{"}$
- tablicową - $d := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

Zmienne, jak w powyższych przykładach definiuje się znakiem **:=** , który uzyskuje się z palety symboli *Evaluation* lub po kliknięciu na klawiaturze dwukropka **:** . Jest to *zmienna lokalna*. Ważne jest, że można się do niej odwołać jedynie poniżej lub po prawej stronie od jej definicji. Dla odróżnienia *zmienna globalna* jest widoczna w całym arkuszu, niezależnie od miejsca jej definicji. Definiuje się ją przy użyciu znaku **≡** się z palety symboli *Evaluation* lub po kliknięciu na klawiaturze tyldy **~**. Skróty klawiaturowe pomocne przy definicji zmiennych zebrano w *Tab. 1 Skróty klawiszowe przydatne w pisaniu operacji matematycznych* pod koniec tych ćwiczeń.

W zadaniu użyj zmiennych lokalnych v, v_0, t, a. Zadanie rozwiąż jedynie przy użyciu liczb. Zobacz jak zmienia się wynik, gdy zmienisz wartość zmiennych.

Jednostki miar

Zad. 3

Rozwiąż zad. 2 przy użyciu jednostek wbudowanych w *Matchcad*. Zwróć uwagę, że te same zmienne zdefiniowane ponownie w arkuszu program podkreśla na zielono. W obszarze poniżej nowej definicji zmiennej, po odniesieniu do niej program zwróci nową wartość.

Matchcad zna wiele jednostek miar. Warto wykorzystywać je w obliczeniach. Program, korzystając z własnej bazy może też służyć jako konwerter jednostek. Jednostki miar dodaje się z *Menu* → *Insert* → *Unit* lub po kliknięciu przycisku skrótu  na pasku *Standard*. Otwiera się okno z bazą przeróżnych jednostek pogrupowanych wg ich zastosowania. Warto zauważyć, że są to jednostki podstawowe. Np. brak jest w bazie jednostki przyspieszenia m/s^2 , gdyż *Mathcad* zna zarówno metr m jak i sekundę s. Jeśli znasz składnię jednostek, możesz je wpisywać bezpośrednio z klawiatury zaraz po wartości zmiennej. Uwaga: zdefiniowanie nowej zmiennej o tej samej nazwie co istniejąca jednostka, zmieni również jednostkę. Przykładowo: najlepiej nie używać oznaczeń m jako masy, czy s jako drogi, gdyż są to symbole przypisane do jednostek metra i sekundy

Mathcad zna też niektóre przedrostki mnożników, jak *kilo*, *centy*, *mili*. Możesz to sprawdzić wyświetlając w innym miejscu wartość zmiennej i wpisując inną jednostkę np. km/s zamiast m/s.

Gdy brak jest danej jednostki zawsze można ją zdefiniować jako zmienną globalną. Zdefiniuj jednostkę decymetra dm w odniesiu do metra m, i sprawdź wartości zmiennych w dm/hr, km/hr, hr. hr w *Mathcad* oznacza jednostkę godziny.

Funkcje

Zad. 4

Zdefiniuj zmienną zakresową kąta α od 0° do 360° co 10°. Przedstaw wartości kąta α w stopniach °, radianach *rad*, wielokrotności liczby π. Oblicz funkcje trygonometryczne *sinus*, *cosinus*, *tangens* i *cotangens* kąta α.

Litery greckie można dodawać z palety symboli *Greek* lub po wpisaniu polskiego znaku i wciśnięciu klawiszy **Ctrl + g**. Np. α można wpisać jako **a** po czym użyć skrótu **Ctrl + g**. Liczba π w *Mathcad* jest traktowana jako stała.

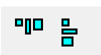
Zmienna zakresowa jest po prostu ciągiem arytmetycznym, który definiuje się za pomocą zakresu wartości oraz kroku. W *Mathcad* wygląda to następująco: najpierw podaje się pierwsza wartość zakresu, następnie drugą (w ten sposób program odczytuje krok), a na końcu ostatnią wartość. Składnia jest następująca:

$$a := 1,3..11$$

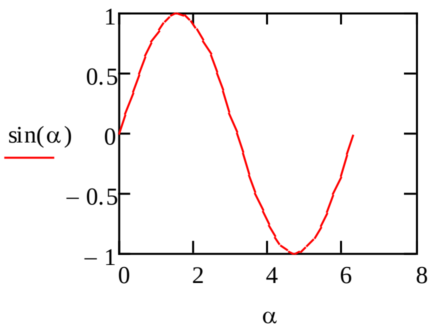
W ten sposób zdefiniowany został ciąg liczb : 1,3,5,7,9,11. Pierwszy i drugi wyraz ciągu oddziela się przecinkiem **,** , a drugi i ostatni średnikiem **;** (program wyświetla to jako dwie kropki). Po wyświetleniu wartości zmiennej program wyświetla tabelę wartości. Gdy jest ona długa, pojawia się pasek przewijania.

Na bazie istniejącej zmiennej zakresowej nie można utworzyć kolejnej zmiennej poprzez działanie arytmetyczne na pierwszej. Można jednak wyświetlić wynik tego działania. Spróbuj wyświetlić wartość kąta 2α. Spróbuj następnie zdefiniować kąt $\beta = 2\alpha$.

Mathcad ma wiele wbudowanych funkcji, m.in. są to funkcje trygonometryczne. Można je znaleźć na palecie symboli Calculator. Dodatkowo pełną listę funkcji można otworzyć w nowym oknie przez *Menu* → *Insert* → *Function* lub po kliknięciu przycisku skrótu  na pasku *Standard*. Najbardziej popularne funkcje wbudowane i stałe zebrano w *Tab. 2 Stałe i funkcje wbudowane* pod koniec tych ćwiczeń.

Tabelki wyświetlanych wyników ułóż równo korzystając z przycisków wyrównania w pinie i poziomie , które uaktywniają się po zaznaczeniu więcej niż jednego obiektu.

Na końcu korzystając z palety symboli Graph → *X-Y Plot*, możesz dodać wykres *sinusa* (jak poniżej). Wystarczy w pustych miejscach podpisać osie. Zwróć uwagę, że wykres łączy punkty. Spróbuj zmienić krok zmiennej zakresowej na 45° i zobacz różnicę.



Zad. 5

Napisz funkcję obliczającą drogę pojazdu z zad. 3. w zależności od czasu. Oblicz wartość tej drogi po 20 sekundach. Następnie, wykorzystując zmienną zakresową, oblicz wartość przebytej drogi po kolejnych sekundach jazdy od 1 sek. do 10 sek. Pamiętaj o wykorzystaniu jednostek. Przepisz jeszcze raz funkcję i przekształć ją na 3-argumentową. Zmieniaj wartości przyspieszenia, prędkości początkowej i czasu.

Mathcad pozwala na definiowanie dowolnej funkcji użytkownika. Każde wyrażenie można zdefiniować jako funkcję jedno- lub wieloargumentową. Wystarczy napisać nazwę funkcji, po której w nawiasach okrągłych znajdą się jej argumenty, po czym zdefiniować samą funkcję używając operatora definicji $:=$. Należy pamiętać, że zmienne znajdujące się w wyrażeniu funkcji muszą występować jako argumenty, bądź być zdefiniowane wcześniej. Funkcja w zadaniu będzie wyglądać następująco:

$$S(t) = \frac{at^2}{2} + v_0 t$$

Aby się do niej odwołać wystarczy zadeklarować zmienną, np. t_1 i odwołać się do tej zmiennej: $S(t_1)$ lub bezpośrednio wpisać $S(20s)$. Zmienna do której odwołuje się funkcja może być zmienną zakresową. Funkcja 3-argumentowa, gdzie wszystkie parametry będą argumentami, będzie wyglądać następująco:

$$S_2(t, a, v_0) = \frac{at^2}{2} + v_0 t$$

Operatory logiczne

Operatory logiczne dodaje się z klawiatury lub z palety symboli *Boolean*. Twardy znak równości **=** wprowadza się za pomocą skrótu **Ctrl + =** . Warto to zapamiętać, gdyż jest to często używany znak (nie tylko w operacjach logicznych), który ma inną funkcje niż znak wyniku $=$ oraz równości definicyjnej $:=$.

Twardym znakiem równości sprawdza się warunek PRAWDA czy FAŁSZ. W *Mathcad* gdy warunek jest prawdziwy program zwraca wartość 1, a gdy fałszywy – wartość 0. Aby wyświetlić zwracaną wartość należy wpisać znak wyniku $=$.

Zad. 6

Zdefiniuj 4 zmienne a, b, x, y takie, aby jedynie x i y były sobie równe. Sprawdź warunki logiczne:

$$a = b, a \neq b, a < b, a > b$$

$$x = y, x \neq y, x \leq y, x \geq y, x > y$$

A następnie:

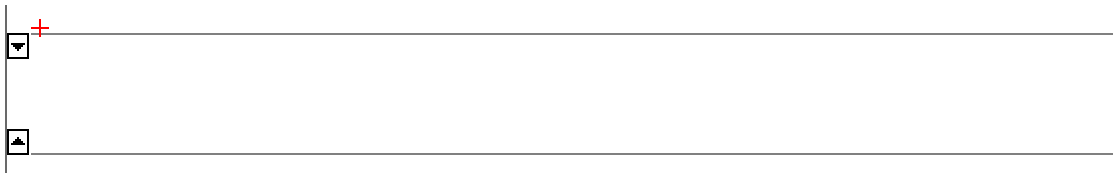
$$a = b \wedge x = y$$

$$a = b \ \vee x = y$$

$$\neg\,(a = b) \wedge x = y$$

Inne

Przydatną funkcją może być dodawanie obszarów, których zawartość można łatwo zwijać. Po kliknięciu *Menu* → *Insert* → *Area* w miejscu kursora pojawią się dwie poziome linie, które można dowolnie ustawić w arkuszu:



Po kliknięciu jednego z trójkątów po lewej stronie, obszar zwija się do górnej linii, ukrywając w ten sposób swoją treść.

Spróbuj dodać obszary do wszystkich zadań w pliku *Cwiczenia3*, tak aby można było ukryć rozwiązania zadań, a widoczne były tylko nagłówki.

Co pewien czas warto odświeżyć okno programu, np. gdy zostaje obraz przesuwanych obiektów. Do tego celu używa się skrótu **Ctrl + R**, który uruchamia opcję *Menu* → *View* → *Refresh*.

Arkusz programu przewija się dość niewygodnie. Szybsze przewijanie po jednej stronie można osiągnąć scrollem myszy z przytrzymanym przyciskiem **Ctrl** lub przyciskami **PageUp** i **PageDown**.

Tab. 1 Skróty klawiszowe przydatne w pisaniu operacji matematycznych

OPERATOR	KLAWISZ	SYMBOL
Definiowanie zmiennych		
Definicja zmiennej	:	$:=$
Wyświetlenie wyniku	=	$=$
Globalna definicja zmiennej	~	$\equiv$
Twardy znak równości	Ctrl + =	**=**
Oddzielenie elementów ciągu	,	,
Zakres ciągu	;	..
Indeks dolny zmiennej (nazwa)	.	
Indeks dolny elementów ciągu	[	
Operacje matematyczne		
Dodawanie	+	$a+b$
Odejmowanie	-	$a-b$
Mnożenie	*	$a \cdot b$
Dzielenie	/	$\frac{a}{b}$
Pierwiastek kwadratowy	\	$\sqrt{a}$
Potęgowanie	^	a^b
Wartość bezwzględna	|	$\|a\|$
Silnia	!	$a!$
Obydwa nawiasy	‘	(a)
Iloczyn ciągu	#	$\prod_n a$
Suma ciągu	$	$\sum_n a$
Pochodna pierwszego rzędu	?	$\frac{df}{dt}$
Pochodna dowolnego rzędu	Ctrl + ?	$\frac{d^n}{dt^n} f$
Całka oznaczona	&	$\int_a^b f\,dt$
Całka nieoznaczona	Ctrl + i	$\int f\,dt$
Granica	Crtl + l	$\lim_{t \to n} f$

Tab. 2 Stałe i funkcje wbudowane

SPOSÓB WPISYWANIA	STAŁA/FUNKCJA
Stałe	
Ctrl + z	∞
e	e
Ctrl + Shift + p (p Crtrl+g)	π
%	%
Funkcje	
sin(x)	sinus
cos(x)	cosinus
tan(x)	tangens
cot(x)	cotangens
sec(x)	secans
csc(x)	cosecans
asin(x)	arcus sinus
acos(x)	arcus cosinus
atan(x)	arcus tangens
exp(x)	e^x
log(x)	logarytm dziesiętny
ln(x)	logarytm naturalny
floor(x)	zaokrąglenie w dół
ceil(x)	zaokrąglenie w górę
round(x,n)	zaokrąglenie do n-tego miejsca po przecinku
trunc(x)	część całkowita liczby rzeczywistej
polyroots(v)	wyznaczenie pierwiastków wielomianu, gdzie *v* to wektor jego współczynników
find(x,y,…)	znalezienie rozwiązania układu równań i nierówności
mean(A)	średnia arytmetyczna wyrazów tablicy A
median(A)	mediana wyrazów tablicy A
stdev(A)	odchylenie standardowe wyrazów tablicy A

Lista zadań 3

Zad. 0

Ustaw nagłówek, w którym po lewej będzie Twoje imię i nazwisko, po środku tytuł pliku, a po prawej data (dwa ostatnie wstawiane automatycznie). W prawej części stopki powinien się wyświetlać numer strony oraz liczba stron dokumentu w formacie *numer strony/liczba stron*. Dodaj ramkę wokół strony dokumentu (arkusza roboczego). Sprawdź wygląd konspektu w podglądzie wydruku *Print preview*.

Napisz tytuł ćwiczenia „**Ćwiczenia 3 – Wprowadzenie do programu Mathcad**" wyśrodkowany, czcionką *Calibri 16*.

Każde zadanie zaczynaj od nagłówka **Zad. …** w domyślnym stylu *Heading 1*.

Zad. 1

Przepisz do arkusza następujące wyrażenia matematyczne:

$$x_1^2+x_2+5$$

$$\frac{1}{2}+\frac{1}{3}=\frac{5}{6}$$

$$\sqrt[2]{27}=3$$

$$\ln(e^7)=7$$

$$\frac{1}{\frac{1}{10^2}+\frac{3}{9^{1.4}}+\frac{1}{17+3^{0.4}}}\cdot 10^{-5}=$$

$$|5\text{-}7|=2$$

Zad. 2

Pojazd porusza się ruchem prostoliniowym jednostajnie przyspieszonym z przyspieszeniem 2 m/s^2. Po jednej sekundzie prędkość pojazdu wynosiła 6 m/s. Jaka była prędkość początkowa pojazdu? Zadanie należy rozwiązać posługując się zmiennymi.

Zad. 3

Rozwiąż zad. 2 przy użyciu jednostek wbudowanych w *Matchcad*. Zdefiniuj jednostkę decymetra dm w odniesiu do metra m, a także godziny h w odniesiu do sekundy s i sprawdź wartości zmiennych w dm/h, km/h, h.

Zad. 4

Zdefiniuj zmienną zakresową kąta α od 0° do 360° co 10°. Przedstaw wartości kąta α w stopniach °, radianach *rad*, wielokrotności liczby π. Oblicz funkcje trygonometryczne sinus, cosinus, tangens i cotangens kąta α. Na końcu korzystając z palety symboli Graph → *X-Y Plot*, dodaj wykres sinusa.

Zad. 5

Napisz funkcję obliczającą drogę pojazdu z zadania 3. w zależności od czasu. Oblicz wartość tej drogi po 20 sekundach. Następnie wykorzystując zmienną zakresową oblicz wartość przebytej drogi po kolejnych sekundach jazdy od 1 sek. do 10 sek. Pamiętaj o wykorzystaniu jednostek. Przepisz jeszcze raz funkcję zmieniając na 3-argumentową i zmieniaj wartości przyspieszenia, prędkości początkowej i czasu.

Zad. 6

Zdefiniuj 4 zmienne a, b, x, y takie, aby jedynie x i y były sobie równe. Sprawdź warunki logiczne:

$$a = b, a \neq b, a < b, a > b$$

$$x = y, x \neq y, x \leq y, x \geq y, x > y$$

A następnie:

$$a = b \wedge x = y$$

$$a = b \ \vee x = y$$

$$\neg\, (a = b) \wedge x = y$$

Ćwiczenia 4 - Mathcad

Obliczenia symboliczne. Całkowanie i różniczkowanie. Wektory i macierze.

Obliczenia symboliczne

W programie *Mathcad* można prowadzić obliczenia w dwóch postaciach:

- numerycznej
- symbolicznej

Obliczenia numeryczne były wykorzystywane na poprzednich zajęciach. Wykonuje się je wpisując znak równości (wyniku) $=$, który można wpisać z klawiatury bądź wybrać z palety *Calculator* lub *Evaluation*. Otrzymany wynik niecałkowity podawany jest w formie ułamka dziesiętnego.

Obliczeń symbolicznych używa się, gdy potrzeba przeanalizować wpływ wybranych parametrów na całe wyrażenie, uprościć lub rozwinąć wyrażenie, wyznaczyć symbolicznie granice, pochodne i całki nieoznaczone funkcji, a także wyznaczyć rozwiązanie liczbowe w formie ułamka prostego. Paleta funkcji symbolicznych w Mathcad jest bardzo szeroka. Niektóre z nich są opisane poniżej.

Obliczenia symboliczne się wykonuje się korzystając z funkcji dostępnych w palecie *Symbolic* lub wybierając odpowiednią funkcję z *Menu* $\rightarrow$ *Symbolics*. Przede wszystkim zamiast klasycznego znaku równości $=$ używa się opertaora strzałki $\rightarrow$, który można wybrać z palety *Symbolic* lub *Evaluation*, jak również używając skrótu **Ctrl + .** . Rozwiązanie symboliczne można też uzyskać z *Menu* $\rightarrow$ *Symbolics* $\rightarrow$ *Evaluate* $\rightarrow$ *Symbolically* lub przy użyciu skrótu **Shift+F9**. Ten sposób wyrzuca wynik obok lub pod spodem wyrażenia, czasami z komentarzem dotyczącym przeprowadzonego działania. Styl wyświetlania można zmienić w *Menu* $\rightarrow$ *Symbolics* $\rightarrow$ *Evaluation Style*. Z *Menu* $\rightarrow$ *Symbolics* można wywołać także podstawowe operacje symboliczne, jak np. uproszczenie wyrażenia *Simplify* i rozwinięcie wyrażenia *Expand*.

Zad. 1

Wykonaj następujące operacje symboliczne:

- rozwiąż równanie:

$$\frac{1}{2}+\frac{2}{3}$$

- rozwiń wyrażenie:

$$x(2x+1)$$

- uprość wyrażenie:

$$a^2+2ab+b^2$$

Wykorzystaj funkcje z palety *Symbolic*, a następnie *Menu* $\rightarrow$ *Symbolics*. Porównaj obydwa sposoby. Ustaw wyświetlanie rozwiązania wraz z komentarzem obok wyrażenia.

Aby zastosować funkcje symboliczne, można wybrać je z palety *Symbolic* lub, znając składnię danej funkcji, wybrać operator *Keyword evaluation* ■ ⟶ i wpisać ręcznie nazwę funkcji. Do operatora przypisany jest skrót **Ctrl+Shift+.** . Opis funkcji symbolicznych znajduje się w *Tab.* 1 Operacje symboliczne pod koniec tego ćwiczenia. Większość z nich dotyczy całego wyrażenia, jednak przy niektórych trzeba określić zmienną, której dotyczy operacja. Symbolicznie można również wykonywać operacje na macierzach.

Zad. 2

Korzystając z palety *Symbolic* wykonaj następujące operacje symboliczne:

- Uprość wyrażenia:

$$\sin^3 x + \sin x \ \cos^2 x$$

$$\sqrt{x^2}$$

Używając funkcji *assume* określ zakres zmiennej na rzeczywistą liczbę ujemną (przy użyciu dodatkowej palety *Modifiers*). Co zwróci program gdy zabraknie tego założenia?

- Rozłóż na czynniki wyrażenie:

$$\frac{24a+12b}{3ac+9a^2}$$

- Wydziel czynnik x przed nawias w wyrażeniach:

$$x^2 \cdot 2y-(x+y)^3+(x-y)^2$$

$$(x^2+5xy+10x+7y+18)(x^2+1)$$

- Podstaw wartości $x = 5$, $y = 3$ do wyrażenia:

$$2x^2+5y+1$$

- Zaokrąglij następujące liczby:

  π , do 10 miejsc po przecinku

  e , do 5 miejsc po przecinku

  $\sqrt{2}$, do 6 miejsc po przecinku

- Wyznacz współczynniki wielomianów:

$$(x+3)(x+5)$$

$$(x^2+1)(x^2+2x+5)$$

  Porównaj rezultat z rozwinięciem wyrażeń funkcją *expand*.

- Rozwiąż równania (ze względu na x):

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

$$2x+3y+2$$

- Rozłóż na sumę ułamków prostych wyrażenie:

$$\frac{x+2}{(x-1)^2(x^2-2x+2)}$$

Różniczkowanie

W programie *Mathcad* z powodzeniem można obliczać pochodne i całki. Zapis pochodnych w *Mathcad* jest tradycyjny, jednak wpisanie z klawiatury wyrażenia $\frac{d}{dx}$ przed funkcją nie zadziała. Operator pochodnej można wybrać z palety *Calculus*. Do dyspozycji jest operator pochodnej pierwszego rzędu $\frac{d}{dx}$ (skrót klawiaturowy **?**) oraz pochodnej wyższych rzędów $\frac{d}{dx}$ (skrót klawiaturowy **Ctrl +?**). Używając operatów pochodnych należy zdefiniować zmienną oraz funkcję tej zmiennej. Może to być bezpośrednio wyrażenie matematyczne lub nazwa wcześniej zadeklarowanej funkcji np.: $\frac{d}{dx}f(x)$. Chcąc poznać postać funkcyjną pochodnej, trzeba użyć operatora strzałki → do obliczeń symbolicznych. Zadeklarowanie zmiennej pozwoli obliczyć wartość pochodnej w punkcie. W tym wypadku możliwe jest użycie operatora wyniku $=$.

Zad. 3

Oblicz symbolicznie pochodne (po dx) następujących wyrażeń i zadeklarowanych funkcji:

$$f(x) = 4x^2 + 3x + 5$$

$$\sin x$$

$$x^n$$

$$x^2 + 3y$$

$$h(x,y) = 2xy^2 + 2x + 7y$$

Oblicz pochodną po dy ostatniej funkcji.

Oblicz pochodną funkcji $f(x)$ w punkcie $x = 1.5$.

UWAGA! Zadeklarowanie zmiennej x będzie skutkowało wykorzystaniem tej zmiennej we wszystkich obliczeniach poniżej. Aby wyczyścić wartość zmiennej do obliczeń symbolicznych, należy ponownie ją zadeklarować jako:

$$x := x$$

Zad. 4

Pojazd porusza się ruchem jednostajnie przyspieszonym. Zdefiniuj funkcję drogi pojazdu od czasu:

$$S(t) = \frac{at^2}{2} + v_0 t$$

Korzystając z faktu, że prędkość jest pierwszą pochodną drogi po czasie a przyspieszenie drugą pochodną, oblicz symbolicznie obydwa wyrażenia i sprawdź ich poprawność. Następnie oblicz prędkość chwilową pojazdu w 20 sekundzie, jeśli jego przyspieszenie wynosi $2\ \frac{m}{s^2}$, a prędkość początkowa $5\,\frac{m}{s}$.

Całkowanie

Mathcad umożliwia obliczanie całek nieoznaczonych i oznaczonych. Operatory całkowania można wybrać z palety *Calculus* klikając symbol $\int$ dla całki nieoznaczonej i $\int_a^b$ dla całki oznaczonej. Można zastosować też skróty klawiszowe, kolejno **Ctrl +** i oraz **&**. Obliczając całkę należy, podobnie jak przy pochodnych, zdefiniować zmienną oraz funkcję, a dla całki oznaczonej - także zakres. Całki oblicza się symbolicznie przy pomocy operatora strzałki. W przypadku całki oznaczonej wynik wyświetla się w postaci symbolicznej, np. w ułamku lub przy użyciu funkcji elementarnych. Dopiero po wpisaniu znaku wyniku $=$, zostanie wyświetlona wartość liczbowa w formacie zmiennoprzecinkowym.

Zad. 5

Oblicz całki nieoznaczone następujących funkcji i wyrażeń po dx:

$$f(x) = 4x^2 + 3x + 5$$

$$\sin^2(x)$$

$$x^n$$

$$x^2 \ln^2 x$$

Uprość wyrażenia, jeśli to możliwe.

Zad. 6

Oblicz całki oznaczone następujących funkcji na przedziałach:

$$\frac{1}{81 + x^2}, \langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$$

$$g(x) = \frac{1}{\sqrt{1 + x^3}}, \langle 0, 1 \rangle$$

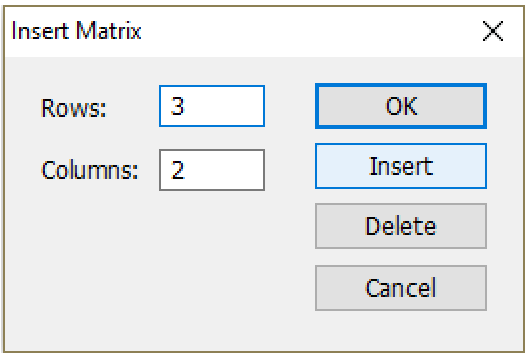
$$f(x) = x^{10}, \langle -1, 1 \rangle$$

$$\sin x\ , \langle -\pi, \pi \rangle$$

Wektory i macierze

Mathcad z powodzeniem nadaje się do obliczeń przy użyciu wektorów i macierzy. Służą do tego odpowiednie funkcje znajdujące się w palecie symboli *Matrix*. Program nie rozróżnia szczególnie wektorów i macierzy, traktując wektor jako macierz jednokolumnową.

Zarówno wektor, jak i macierz można stworzyć klikając ikonę lub skrót **Ctrl + M**. Otworzy się okno wstawiania macierzy *Insert Matrix*, w którym trzeba zdefiniować liczbę rzędów (*rows*) i kolumn (*columns*):



Z palety symboli *Matrix* można wybrać podstawowe funkcje do obliczania macierzy takie jak: odwrócenie macierzy X^{-1}, wyznacznik $|x|$, transpozycja M^T oraz do obliczania wektorów: iloczyn skalarny $\vec{x} \cdot \vec{y}$, iloczyn wektorowy $\vec{x} \times \vec{y}$, oraz suma wektora Σv.

Wektory i macierze mogą także (przede wszystkim) służyć do przechowywania większej ilości danych w jednej zmiennej. Można je traktować jako tablicę wartości.

Zad. 7

Stwórz wektor czasu zawierający wartości od 1 do 50 sekund. Dla danych z zad. 4 oblicz wartości drogi i prędkości po kolejnych sekundach ruchu. Odwołując się do elementów wektorów sprawdź wartość przebytej drogi i chwilowej prędkości po 21 i po 37 sekundach. Zakładając, że pojazd waży 500 kg, oblicz jego energię kinetyczną w kolejnych sekundach ruchu.

Oprócz tworzenia wektora przez wstawienie go funkcją *Insert Matrix* i ręcznym wpisaniem wartości istnieją inne dwa proste sposoby. Najłatwiej jest zadeklarować zmienną zakresową, po czym wstawić znak wyniku =. Pod spodem pojawi się tabela wartości, a zmienna traktowana będzie jako wektor.

$$t_1 := 1s, 2s .. 50s$$

1	s
2	
3	
4	
5	
6	
...	

Uwaga, brak kliknięcia = spowoduje, że zmienna zostanie zakresowa i nie będzie na jej podstawie można tworzyć innych zmiennych.

Drugim sposobem jest stworzenie zmiennej zakresowej i tworzącej indeksy wektora po czym odwołanie się do niej za pomocą funkcji *Subsript* x_n. Można wykonać jakiekolwiek działanie na i lub dodać jednostkę:

$$i := 1..50 \qquad Czas_i := i \cdot s \qquad Czas =$$

	1
1	1
2	2
3	3
4	4
5	5
6	6
7	...

s

Domyślnie Mathcad numeruje elementy wektora od 0. Aby to zmienić należy zmienić wartość zmiennej $ORIGIN$ na wybrana wartość, np. $ORIGIN := 1$. Długość wektora można sprawdzić funkcją $length(x)$.

Mając wektor czasu można zdefiniować kolejno funkcje drogi od czasu, prędkości od czasu i energii kinetycznej od prędkości i przypisać im zmienne. Jeśli funkcje odwołują się do wektora, utworzone zmienne też będą wektorami.

Aby wyznaczyć drogę i prędkość po 21 i 37 sekundach wystarczy wyciągnąć dane z 21. i 37. elementu wektora (z 20. i 36. jeśli wektor jest numerowany od 0). Trzeba do tego użyć funkcji *Subscript*, która można wywołać skrótem klawiszowym **[** . Należy zwrócić uwagę, że indeks wektora i macierzy, mimo że w zapisie taki sam, różni się od zwykłego indeksu zmiennej wpisywanego po kropce.

Zad. 8

Zdefiniuj wektory $x = 1,2 \ldots 10$ oraz $y = 2,4 \ldots 8$, a następnie stwórz macierz wyników funkcji $x^2 + y^2$ w zakresie x i y. Wyciągnij z tablicy wynik dla $x = 9$ i $y = 6$ i porównaj z wynikiem zdefiniowanej funkcji $f(x) = x^2 + y^2$. Korzystając z istniejącej macierzy stwórz macierz wyników funkcji $(x^2 + y^2)^2$.

Załóżmy, że wektor x będzie odpowiadał wierszom macierzy a wektor y kolumnom. W macierzy potrzeba przedstawić funkcję każdej pary x i y, dlatego macierz będzie miała wymiary 10x4. Pierwsza kolumna będzie miała postać funkcji $x^2 + y_1^2$ gdzie y_1 jest odwołaniem do pierwszego elementu wektora y (pod warunkiem, że $ORIGIN := 1$). Korzystając z funkcji *Matrix Column* $M^{\langle\rangle}$, możemy w ten sposób zdefiniować każdą z 4 kolumn macierzy., np.:

$$M^{\langle 1\rangle} := x^2 + (y_1)^2 \qquad M^{\langle 2\rangle} := x^2 + (y_2)^2$$

Ab znaleźć wynik dla $x = 9$ i $y = 6$ trzeba odwołać się do odpowiedniego elementu macierzy. Ponieważ $y = 6$ odpowiada kolumnie 3, będzie to element $M_{9,3}$. W przypadku zdefiniowanej funkcji wystarczy podstawić wartość x i y : $f(9{,}6)$.

Do znalezienia wyników funkcji $(x^2 + y^2)^2$ może posłużyć macierz M. Przy podniesieniu macierzy M do kwadratu wystąpi błąd i komunikat, że macierz musi być kwadratowa. Operacje mnożenia i potęgowania macierzy są wykonywanie wg standardowego mnożenia Cauchy'ego. Chcąc wymnożyć tylko wartości odpowiadających sobie elementów trzeba skorzystać z funkcji *Vectorize* $\overrightarrow{f(M)}$. Jej postać będzie wyglądała następująco: $\overrightarrow{M^2}$

Tab. 1 Operacje symboliczne

FUNKCJA Z MENU	OPERACJA	ZAKRES
float	zaokrąglenie do danej liczby miejsc	wyrażenie
rectangular	oddzielnie liczby zespolonej na część rzeczywistą i urojoną	wyrażenie
assume	założenie zakresu zmiennej	zmienna
solve	symboliczne rozwiązanie równania	zmienna
simplify	algebraiczne uproszczenie	wyrażenie
substitute	podstawienie wartości lub wyrażeń pod zmienne	zmienna
factor	rozkład na czynniki	wyrażenie
expand	algebraiczne rozwinięcie	wyrażenie
coeffs	znalezienie współczynników wielomianu	wyrażenie
collect	wydzielenie czynników jako potęg zmiennej	wyrażenie
series	rozwinięcie w szereg Taylora	wyrażenie
parfrac	rozkład na sumę ułamków prostych	wyrażenie
fourier	transformata Fouriera	wyrażenie
laplace	transformata Laplace'a	wyrażenie
ztrans	transformata Z	wyrażenie
invfouriem	odwrotna transformata Fouriera	wyrażenie
invlaplace	odwrotna transformata Laplace'a	wyrażenie
invztrans	odwrotna transformata Z	wyrażenie
$\boldsymbol{M^T} \rightarrow$	transpozycja	macierz
$\boldsymbol{M^{-1}} \rightarrow$	odwrotność	macierz

$\|\boldsymbol{M}\| \rightarrow$	wyznacznik	macierz
explicit	podstawienie wartości zmiennych bez ich upraszczania	wyrażenie
combine	łączenie wyrażeń przy wykorzystaniu elementarnych funkcji	wyrażenie
rewrite	przekształcenie wrażenia przy wykorzystaniu elementarnych funkcji	wyrażenie

Lista zadań 4

Zad. 1

Wykonaj następujące operacje symboliczne:

- rozwiąż równanie: $\frac{1}{2}+\frac{2}{3}$
- rozwiń wyrażenie: $x(2x+1)$
- uprość wyrażenie: $a^2+2ab+b^2$

Zad. 2

Korzystając z palety *Symbolic* wykonaj następujące operacje symboliczne:

- Uprość wyrażenia: $\sin^3 x+\sin x\ \cos^2 x$

$$\sqrt{x^2}$$

- Rozłóż na czynniki wyrażenie: $\frac{24a+12b}{3ac+9a^2}$
- Wydziel czynnik x przed nawias w wyrażeniach:

$$x^2\cdot 2y-(x+y)^3+(x-y)^2$$
$$(x^2+5xy+10x+7y+18)(x^2+1)$$

- Podstaw wartości $x=5$, $y=3$ do wyrażenia:

$$2x^2+5y+1$$

- Zaokrąglij następujące liczby:

  π , do 10 miejsc po przecinku

  e , do 5 miejsc po przecinku

  $\sqrt{2}$, do 6 miejsc po przecinku

- Wyznacz współczynniki wielomianów:

$$(x+3)(x+5)$$
$$(x^2+1)(x^2+2x+5)$$

  Porównaj rezultat z rozwinięciem wyrażeń funkcją *expand*.

- Rozwiąż równania (ze względu na x):

$$x^2+2x-3$$
$$2x+3y+2$$

- Rozłóż na sumę ułamków prostych wyrażenie:

$$\frac{x+2}{(x-1)^2(x^2-2x+2)}$$

Zad. 3

Oblicz symbolicznie pochodne (po dx) następujących wyrażeń i zadeklarowanych funkcji:

$$f(x) = 4x^2 + 3x + 5$$

$$\sin x$$

$$x^n$$

$$x^2 + 3y$$

$$h(x, y) = 2xy^2 + 2x + 7y$$

Oblicz pochodną po dy ostatniej funkcji.

Oblicz pochodną funkcji $f(x)$ w punkcie $x = 1.5$.

Zad. 4

Pojazd porusza się ruchem jednostajnie przyspieszonym. Zdefiniuj funkcję drogi pojazdu od czasu:

$$S(t) = \frac{at^2}{2} + v_0 t$$

Korzystając z faktu, że prędkość jest pierwszą pochodną drogi po czasie a przyspieszenie drugą pochodną, oblicz symbolicznie obydwa wyrażenia i sprawdź ich poprawność. Następnie oblicz prędkość chwilową pojazdu w 20 sekundzie, jeśli jego przyspieszenie wynosi 2 $\frac{m}{s^2}$, a prędkość początkowa 5 $\frac{m}{s}$.

Zad. 5

Oblicz całki nieoznaczone następujących funkcji i wyrażeń po dx:

$$f(x) = 4x^2 + 3x + 5$$

$$\sin^2(x)$$

$$x^n$$

$$x^2 \ln^2 x$$

Uprość wyrażenia, jeśli to możliwe.

Zad. 6

Oblicz całki oznaczone następujących funkcji na przedziałach:

$$\frac{1}{81 + x^2}, \langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$$

$$g(x) = \frac{1}{\sqrt{1 + x^3}}, \langle 0, 1 \rangle$$

$$f(x) = x^{10}, \langle -1, 1 \rangle$$

$$\sin x \ ,\langle -\pi, \pi \rangle$$

Zad. 7

Stwórz wektor czasu zawierający wartości od 1 do 50 sekund. Dla danych z zad. 4 oblicz wartości drogi i prędkości po kolejnych sekundach ruchu. Odwołując się do elementów wektorów sprawdź wartość przebytej drogi i chwilowej prędkości po 21 i po 37 sekundach. Zakładając, że pojazd waży $500\ kg$, oblicz jego energię kinetyczną w kolejnych sekundach ruchu.

Zad. 8

Zdefiniuj wektory $x = 1{,}2 \dots 10$ oraz $y = 2{,}4 \dots 8$, a następnie stwórz macierz wyników funkcji $x^2 + y^2$ w zakresie x i y. Następnie wyciągnij z tablicy wynik dla $x = 9$ i $y = 6$ i porównaj z wynikiem zdefiniowanej funkcji $f(x) = x^2 + y^2$. Korzystając z istniejącej macierzy stwórz macierz wyników funkcji $(x^2 + y^2)^2$.

Zad. 9*

Korzystając z obliczeń symbolicznych oblicz ekstrema, punkty przegięcia oraz określ przedziały wklęsłości i wypukłości funkcji: $f(x) = x^3 + 2x^2 - 7x - 3$

Ćwiczenia 5 - Mathcad

Równania i układy równań. Wykresy. Importowanie danych, obiektów, współpraca z MS Excel.

Równania i układy równań

Mathcad umożliwia rozwiązywanie równań i układów równań na różne sposoby. Często wybrane rozwiązanie zależy od formy skomplikowania równania, a także oczekiwanej formy rozwiązania. Równania można rozwiązywać zarówno przy użyciu obliczeń symbolicznych, jak i wykorzystując wbudowane w programie funkcje. Funkcje wraz z ich opisem i składnią można wybrać z *Menu* → *Insert* → *Function*. Funkcje służące do rozwiązywania równań znajdują się w kategorii *Solving*.

Zad. 1

Rozwiąż symbolicznie następujące równania:

$$-x^2 + x + 6 = 0$$

$$ax^2 + bx + c = 0$$

Do rozwiązania użyj funkcji *solve* z palety *Symbolic*. Pamiętaj, że definiując równanie czy nierówność należy użyć znaków z palety *Boolean*, w tym wypadku twardego znaku równości **=** . Jeśli w równaniu jest więcej niż jedna zmienna, należy po przecinku po słowie *solve* zadeklarować zmienną, dla której program ma zwrócić rozwiązanie. Aby uprościć symboliczne rozwiązanie, można po kolejnym przecinku dodać funkcję *simplify*. Rozwiązanie można przypisać do zmiennej. Spróbuj przypisać rozwiązanie pierwszego równania do zmiennej *wynik*, która będzie 2-elementowym wektorem.

Zad. 2

Znajdź miejsca zerowe następujących wielomianów:

$$x^3 + 2x^2 - 7x - 3$$

$$\frac{(x+4)(x+1)(x-1)(x-3)}{14} + 0{,}5$$

W przypadku wielomianu wyższego rzędu niż 2, funkcja symboliczna *solve* może mieć problem ze znalezieniem dokładnego rozwiązania, albo rozwiązanie będzie miało bardzo skomplikowaną postać. Dużo lepiej radzi sobie z tym funkcja *polyroots*. Argumentem funkcji musi być wektor współczynników wielomianu. Aby otrzymać taki wektor, wystarczy użyć symbolicznej funkcji *coeffs*. Dla ułatwienia można przepisać wielomian jako funkcję i użyć jej postaci do obliczenia symbolicznego. Funkcja *polyroots* zwróci wektor wszystkich miejsc zerowych wielomianu. W przypadku pierwszego wielomianu rozwiązanie będzie wyglądać następująco:

$$b := x^3 + 2x^2 - 7x - 3 \text{ coeffs } \rightarrow \begin{pmatrix} -3 \\ -7 \\ 2 \\ 1 \end{pmatrix} \qquad \text{polyroots}(b) = \begin{pmatrix} -3.68 \\ -0.393 \\ 2.074 \end{pmatrix}$$

Zad. 3

Znajdź miejsca zerowe funkcji:

$$f(x) = 2x + 10$$

$$g(x) = -x^2 + x + 6$$

Do obliczania miejsc zerowych służy funkcja *root*. Aby jej użyć, najpierw trzeba zdefiniować funkcję $f(x)$. Jako argumenty funkcji podaje się nazwę funkcji, zmienną, oraz ewentualnie zakres w którym powinno się znaleźć rozwiązanie. Jeśli nie jest znany zakres, w którym znajduje się miejsce zerowe, musimy zdefiniować wartość początkową x, od której program rozpocznie iteracyjne rozwiązywanie funkcji.

Drugą opcją jest użycie procedury *Given-Find*. Działanie procedury jest podobne do funkcji *root*, różni się jednak zapisem. Tak samo najpierw trzeba zdefiniować wartość zmiennej, od której program zacznie obliczać równanie. Następnie należy napisać wyrażenie *Given*, pod którym dopiero trzeba napisać równanie (oczywiście używając twardego znaku równości **=**). Na końcu funkcja *Find*, której argumentem jest zmienna, zwraca rozwiązanie. Oczywiście można przypisać rozwiązanie do dowolnej zmiennej. Dla pierwszego równania zapis będzie wyglądał następująco:

$$x := 1$$

$$\text{Given}$$

$$2x + 10 \mathbf{=} 0$$

$$\text{Find}(x) = -5$$

W przypadku równania nieliniowego obydwoma metodami można znaleźć jedynie jedno rozwiązanie. Program wyświetla pierwsze otrzymane rozwiązanie, które zależy od wartości początkowej zmiennej. Znając rozwiązanie drugiego równania z zadania 1, spróbuj tak zmienić początkową wartość x, aby otrzymać najpierw pierwsze a potem drugie rozwiązanie. Jedno z nich może wyglądać następująco:

$x := -5$

Given

$-x^2 + x + 6 = 0$

$wynik_0 := \mathrm{Find}(x)$

$wynik = (-2)$

Zad. 4

Rozwiąż następujące układy równań:

$$\begin{cases} 2x + y = 1 \\ x - y = 4 \end{cases}$$

$$\begin{cases} x^2 - y^2 = 5 \\ x + y = 1 \end{cases}$$

$$\begin{cases} \sin x + \cos x = 2 \\ x + y = \dfrac{\pi}{2} \end{cases}$$

Układy równań najlepiej rozwiązuje się, podobnie jak równania, procedurą *Given-Find*. Różnica polega na tym, że na początku należy zdefiniować wartości wszystkich zmiennych pojawiających się w równaniach. Wywołując funkcję *Find* należy podać jako argumenty nazwy wszystkich zmiennych. Zwróci ona wektor z rozwiązaniami o takiej długości ile jest zmiennych.

$x := 1$
$y := 1$

Given

$2x + y = 1$

$x - y = 4$

$wynik := \mathrm{Find}(x, y)$

$wynik = \begin{pmatrix} 1.667 \\ -2.333 \end{pmatrix}$

Wykresy 2D

Mathcad posiada szerokie możliwości prezentowania danych na wykresach dwuwymiarowych. Tworzenie, edycja i formatowanie wykresów jest bardzo proste. Wszystkie rodzaje wykresów oraz związane z nimi narzędzia znajdują się na palecie symboli *Graph*. Jest kilka różnych opcji informacji potrzebnych do stworzenia wykresów:

- 2 wektory tej samej długości z danymi liczbowymi
- funkcja jednej zmiennej

- funkcja jednej zmiennej oraz wektor wartości tej zmiennej
- 2 funkcje tej samej zmiennej - dotyczy wykresów we współrzędnych parametrycznych

Wykresy można dodatkowo można tworzyć w dwóch układach: kartezjańskim i biegunowym.

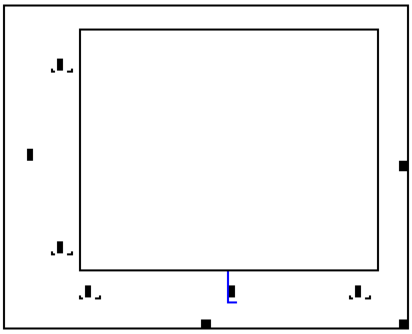
Zad. 5

Przedstaw w układzie kartezjańskim i sformatuj odpowiednio wykres dwóch funkcji:

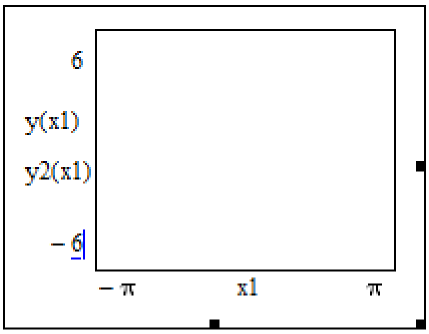
$$y(x) = 4\sin x$$

$$y2(x) = \sin x - x$$

Na początku zdefiniuj obydwie funkcje. Następnie, z palety *Graph* lub za pomocą skrótu klawiszowego **@** otwórz nowy wykres.



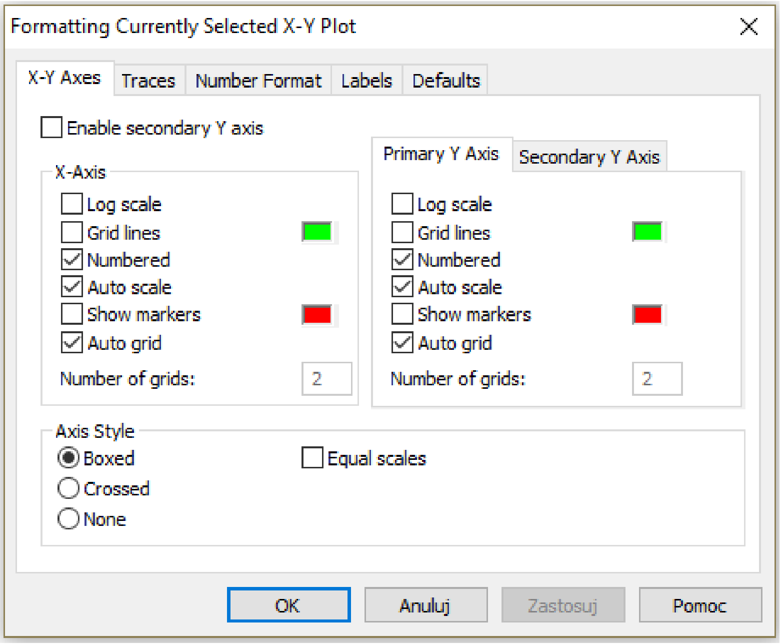
Wykres tworzy się w bardzo prosty sposób: w puste miejsca wpisuje się funkcję (oś odciętych) i zmienną (oś rzędnych). Dodatkowo na końcach osi można określić ich zakresy. W przeciwnym razie *Mathcad* ustala domyślny zakres. Ustaw zakres osi x na $(-\pi, \pi)$, a osi y na $(-6,6)$. Zmienna może nazywać się inaczej niż w funkcji, pod warunkiem, że funkcja zostanie wywołana argumentem o nowej nazwie. Chcąc wyświetlić więcej niż jedną funkcję trzeba wpisać kolejne nazwy po przecinku. Mogą być to funkcje dla różnych zmiennych - wtedy kolejne zmienne także muszą być oddzielone przecinkami.



Mathcad domyślnie dobiera zakres zmiennej tak aby wykres był w miarę gładki. Można zmienić je zakres wyświetlania poprzez definicję zmiennej zakresowej. Zmień zakres tak, aby wyświetlał dokładnie całą funkcję sinus:

$$x1 := -\pi, -0.99\pi..\pi$$

Aby sformatować wykres kliknij w niego 2-krotnie. Otworzy się okno *Formatting Currently Selected X-Y Plot*.



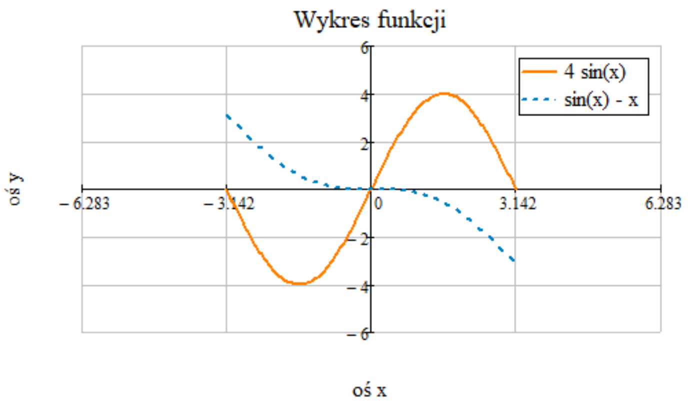
W oknie wyświetla się 5 zakładek, w których można formatować:

- osie liczbowe - *X-Y Axes*
- linie na wykresie - *Traces*
- formaty liczb - *Number Format*
- etykiety osi i tytuł - *Labels*
- ustawienia domyślne - *Defaults*

Sformatuj wykres tak aby:

- osie krzyżowały się w zerze (*Axis Style Crossed*)
- wyświetlane były szare linie siatki (*Grid lines*), liczba linii ustawiana ręcznie
- linie były kolorowe, jedna przerywana, grubość (*Line Weight*) 2.
- ukryte byłe argumenty (Hide arguments), a wyświetlana była legenda z poprawnymi nazwami
- wyświetlał się tytuł wykresu i podpisy osi

Wykres powinien wyglądać następująco:



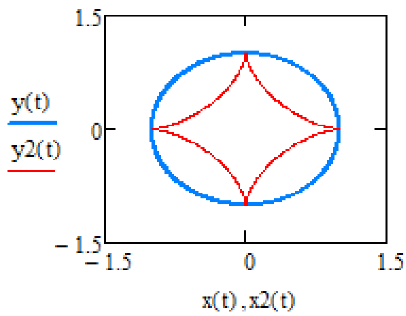
Zad. 6

Narysuj kształt asteroidy (hipocykloidy) na wykresie w układzie kartezjańskim. Skorzystaj z par funkcji:

$$x(t) = \cos t \ , \qquad y(t) = \sin t$$

$$x2(t) = \cos^3 t \ , \qquad y2(t) = \sin^3 t$$

Wykres, na którym do osi przypisane są dwie funkcje tej samej zmiennej, nazywany jest wykresem we współrzędnych parametrycznych, bądź po prostu wykresem parametrycznym. Wykres dla wyżej wymienionych par funkcji będzie wyglądał następująco:



Zad. 7

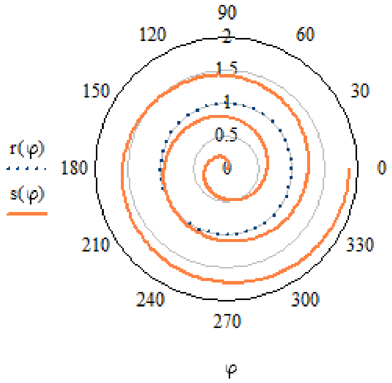
W układzie biegunowym narysuj okrąg $r(\varphi) = 1$ i spiralę Archimedesa $s(\varphi) = 0{,}1\varphi$.

Na wykresie o współrzędnych biegunowych położenie punktu definiuje się za pomocą kąta i promienia. W *Mathcad* wykres *Polar plot* można dodać z palety *Graph* lub przy pomocy skrótu klawiszowego **Ctrl + 7**.

Zdefiniuj funkcje i dodaj je jako argumentu wykresu. Zwróć uwagę, ze Mathcad domyślnie określa zakres zmiennej jako $0..2\pi$. Aby zwiększyć obszar rysowania spirali należy zdefiniować przed wykresem zmienną, pamiętając o małym kroku, tak aby wykres był gładki, np.:

$$\varphi := 0, 01\pi .. 6\pi$$

Wykres ma wyglądać następująco:



Wykresy 3D

Gdy trzeba zaprezentować więcej serii danych na jednym wykresie, warto rozważyć przedstawienie ich w formie jednego wykresu trójwymiarowego. Tak samo, gdy prezentowanym wynikiem ma być przebieg funkcji dwóch zmiennych, lub macierz danych (np. punktów położonych w przestrzeni trójwymiarowej (x, y, z). *Mathcad* proponuje kilka możliwości tego typu danych. Wszystkie rodzaje wykresów trójwymiarowych znajdują się na palecie symboli *Graph*. Do dyspozycji są:

- wykres powierzchniowy - *Surface Plot*
- wykres punktowy - *3D Scatter Plot*
- wykres słupkowy - *3D Bar Plot*
- wykres konturowy (poziomicowy) - *Contour Plot*
- wykres wektorowy - *Vector Field Plot*

Dodatkowo, każdy z nich można przedstawić w układzie kartezjańskim, sferycznym lub walcowym. Wykres 3D można stworzyć na kilka sposobów, tzn. podając:

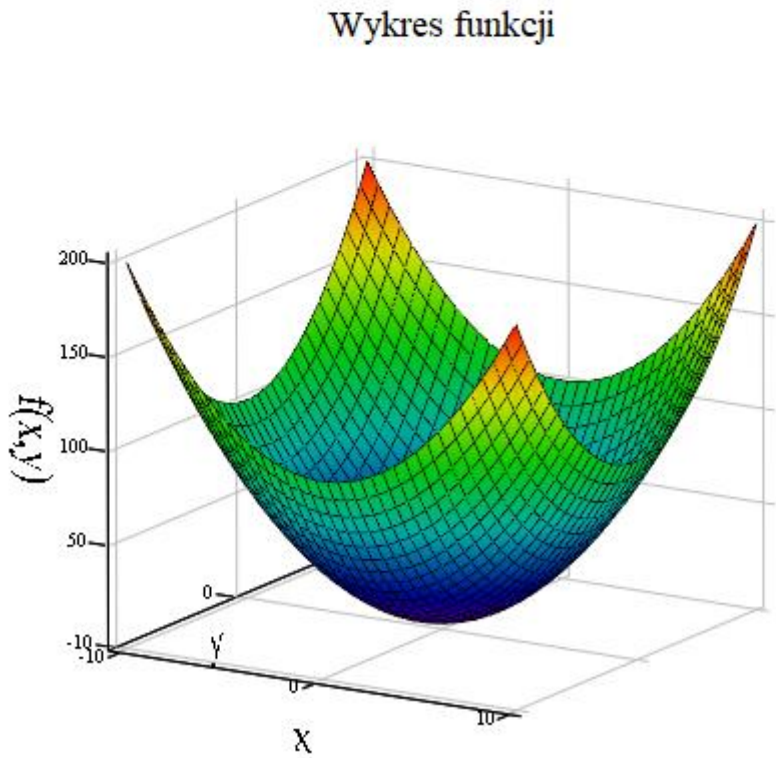
- macierz liczbową – w tym przypadku indeksy wierszy są wartościami x a indeksy kolumn wartościami y
- funkcję dwóch zmiennych
- wektor trzech funkcji dwóch zmiennych
- wektor trzech funkcje jednej zmiennej

Zad. 8

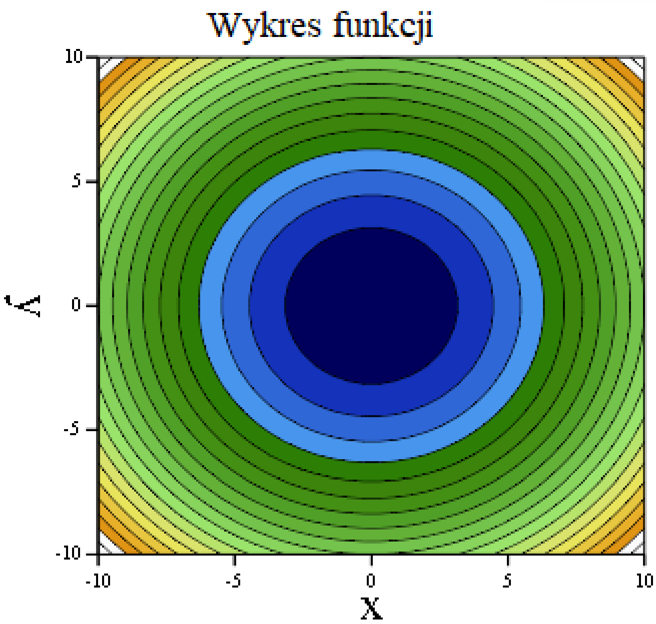
Przedstaw funkcję $f(x) = x^2 + y^2$ na wykresie powierzchniowym oraz poziomicowym.

Aby stworzyć wykres 3D wystarczy zdefiniować funkcję, wybrać *Surface Plot* z palety *Graph* i w wolne pole wpisać nazwę funkcji. Uwaga: w przypadku wykresów 3D podaje się samą nazwę funkcji bez argumentów. Pojawi się czarno-biała siatka punktów. Aby sformatować wykres, kliknij w niego dwukrotnie.

W zakładce *General* można zmienić rodzaj wykresu 3D na dowolny. W ostatniej zakładce *QiuckPlotData* można zmienić układ wykresu a także ustalić obszar obliczania gęstość siatki. Zmień wartości tak, aby funkcja obliczana była w zakresach $(0{,}10)$ i 30 liniami siatki. Następnie w zakładce *Axes* zmień domyślne ustawienia i nazwij etykiety osi. W zakładce *Apperance* edytuje się wypełnienie oraz linie wykresów. Ustaw mapę kolorów i zmień grubość ryzowanych linii siatki na $0{,}5$. Wykres można przybliżać i oddalać za pomocą scrolla oraz dowolnie obracać przytrzymując klawisz myszy. Ostatecznie wykres powinien wyglądać następująco:



Aby przedstawić wykres w formie konturowej, najlepiej przekopiuj ten istniejący i zmień jego ustawienia. W zakładce *Apperance* ustaw wypełnienie konturów, a w zakładce *Advanced* zmień mapę kolorów na *topograf*. Po edycji linii siatki wykres wygląda następująco:



Zad. 9

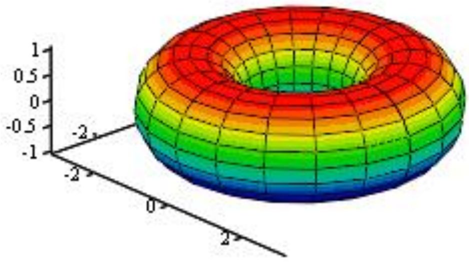
Stwórz wykres parametryczny na bazie 3 funkcji:

$$f(x,y) = (2 + \cos x)\cos y$$

$$g(x,y) = (2 + \cos x)\sin y$$

$$h(x) = \sin(x)$$

Najpierw zdefiniuj funkcje oraz trzyelementowy wektor $G(x,y)$ zawierający te funkcje. Wybierz *Surface Plot* i wpisz jak argument G. Zmień zakres obliczeń do wykresu (*QuickPlotData*) na $(0,2\pi)$. Niestety w oknie formatowania trzeba wpisać przybliżoną wartość liczbową π. Zaznaczenie *Equal Scales* w zakładce *General* spowoduje równe skalowanie wykresu i jego prawidłowy wygląd:



Importowanie danych i obiektów

Do arkusza Mathcad można wstawiać elementy z plików różnych formatów. Służą do funkcje z *Menu* → *Insert*. Można dodawać komponent, który otwiera inny program z pozycji arkusza (*Component*), dane z innych plików (*Data*), kontrolki (*Controls*), obiekty innych formatów (*Object*) i odwołania do innych plików *Mathcad* (*References*).

Zad. 10

Wstaw do arkusza obraz *katapulta.jpg* oraz *katapulta.pdf*. Porównaj rezultaty.

Wejdź w *Menu* → *Insert*→ *Object*, a następnie *Utwórz z pliku* i wybierz odpowiedni plik. Niestety Mathcad nie wyświetla obrazów o rozszerzeniu *.jpg* zostawiając jedynie skrót do pliku.

Zad. 11

Zapisz kopię pliku *MS Excel: Cwiczenia2.xlsx*. Pobierz dane współrzędnych x i y pocisku uwzględniających opór powietrza. W arkuszu Mathcad oblicz odległość pocisku od punktu wystrzału dla każdej pary współrzędnych.

Aby pobrać dane z pliku wejdź w *Menu* → *Insert*→ *Data File* → *Input*. po czym wybierz plik *Cwiczenia2-kopia.xlsx*. Następnie przejdź dalej i wybierz odpowiedni zakres komórek z arkusza MS Excel, który ma być zaimportowany. Pojawi się ikona dyskietki, do której można przypisać zmienną. Zmienna ta będzie macierzą. Z macierzy należy wydzielić wektory x i y a następnie obliczyć odległość punktu od początku układu tworząc nowy wektor element po elemencie.

Współpraca z MS Excel

Mathcad może wymieniać dane z plikami *MS Excel* dzięki wbudowanym funkcjom importu *READEXCEL* i eksportu *WRITEXCEL*.

Zad. 12

Wykonaj polecenie z zad. 11 za pomocą funkcji *READEXCEL* a następnie wpisz obliczone dane do wolnej kolumny w pliku, z którego zaimportowano dane za pomocą funkcji *WRITEEXCEL*. Sprawdź czy funkcje zadziałały poprawnie.

Korzystając z funkcji READEXCEL należy podać w cudzysłowach nazwę pliku w cudzysłowie oraz zakres komórek z arkusza MS Excel oraz przypisać zmienną. Tym razem rozwiązanie zadania zajmuje zaledwie parę linijek, dlatego można bezpośrednio zaimportować wektory x i y omijając tworzenie macierzy i wyciąganie danych. Przykładowe rozwiązanie wygląda następująco:

$x_1 := \mathrm{READEXCEL}(\text{"Cwiczenia2-kopia.xlsx"}, \text{"H8:H108"})$

$y_1 := \mathrm{READEXCEL}(\text{"Cwiczenia2-kopia.xlsx"}, \text{"I8:I108"})$

$odl_1 := \overrightarrow{\sqrt{x1^2 + y1^2}}$

$wpis := \mathrm{WRITEEXCEL}\left(odl_1, \text{"Cwiczenia2-kopia.xlsx"}, \text{"O8:O108"}\right)$

Lista zadań 5

Zad. 1

Rozwiąż symbolicznie następujące równania:

$$-x^2 + x + 6 = 0$$

$$ax^2 + bx + c = 0$$

Zad. 2

Znajdź miejsca zerowe następujących wielomianów:

$$x^3 + 2x^2 - 7x - 3$$

$$\frac{(x+4)(x+1)(x-1)(x-3)}{14} + 0{,}5$$

Zad. 3

Znajdź miejsca zerowe funkcji:

$$f(x) = 2x + 10$$

$$g(x) = -x^2 + x + 6$$

Zad. 4

Rozwiąż następujące układy równań:

$$\begin{cases} 2x + y = 1 \\ x - y = 4 \end{cases}$$

$$\begin{cases} x^2 - y^2 = 5 \\ x + y = 1 \end{cases}$$

$$\begin{cases} \sin x + \cos x = 2 \\ x + y = \dfrac{\pi}{2} \end{cases}$$

Zad. 5

Przedstaw w układzie kartezjańskim i sformatuj odpowiednio wykres dwóch funkcji:

$$y(x) = 4 \sin x$$

$$y2(x) = \sin x - x$$

Zad. 6

Narysuj kształt asteroidy (hipocykloidy) na wykresie w układzie kartezjańskim. Skorzystaj z par funkcji:

$$x(t) = \cos t \,, \qquad y(t) = \sin t$$

$$x2(t) = \cos^3 t \ , \qquad y2(t) = \sin^3 t$$

Zad. 7

W układzie biegunowym narysuj okrąg $r(\varphi) = 1$ i spiralę Archimedesa $s(\varphi) = 0{,}1\varphi$.

Zad. 8

Przedstaw funkcję $f(x) = x^2 + y^2$ na wykresie powierzchniowym oraz poziomicowym.

Zad. 9

Stwórz wykres parametryczny na bazie 3 funkcji:

$$f(x, y) = (2 + \cos x) \cos y$$

$$g(x, y) = (2 + \cos x) \sin y$$

$$h(x) = \sin(x)$$

Zad. 10

Wstaw do arkusza obraz *katapulta.jpg* oraz *katapulta.pdf*. Porównaj rezultaty.

Zad. 11

Zapisz kopię pliku *MS Excel: Cwiczenia2.xlsx*. Pobierz dane współrzędnych x i y pocisku uwzględniających opór powietrza. W arkuszu Mathcad oblicz odległość pocisku od punktu wystrzału dla każdej pary współrzędnych.

Zad. 12

Wykonaj polecenie z zad. 11 za pomocą funkcji *READEXCEL* a następnie wpisz obliczone dane do wolnej kolumny w pliku, z którego zaimportowano dane za pomocą funkcji *WRITEEXCEL*. Sprawdź czy funkcje zadziałały poprawnie.

Zad. 13*

Za pomocą sumy dwóch funkcji falowych:

$$y(t) = A \sin\left(2\pi \frac{t}{T}\right)$$

o różnych okresach T stwórz wykres zjawiska dudnienia.

Zad. 14*

Stwórz wykres krzywej w przestrzeni przy pomocy funkcji:

$$f(t) = \sin(t)$$

$$g(t) = \cos(t)$$

$$h(t) = t$$

Ćwiczenia 6 - Mathcad

Instrukcje warunkowe. Pętle. Programowanie funkcji własnych.

W praktyce inżynierskiej często zdarza się, że obliczenia wykonane muszą być wielokrotnie lub tok obliczeń wybierany jest w zależności od warunków. Do tego celu służą np. funkcje, które pozwalały na przyspieszenie obliczeń. *Mathcad* pozwala na tworzenie o wiele bardziej skomplikowanych funkcji niż te poznane na Ćwiczeniach 3. W tym celu należy wykorzystać konstrukcje programowe. Konstrukcje programowe są dostępne z paska narzędziowego *Programming Toolbar* oraz *Menu* → *View* → *Toolbars* → *Programming.*

Bloki

Zad. 1

Na ćwiczeniach 3 zdefiniowałeś funkcję obliczającą drogę pokonaną przez ciało w ruchu jednostajnie zmiennym.

$$s(t) = \frac{at^2}{2} + v_0 t$$

Zapisz ją wykorzystując bloki dostępne z paska narzędziowego *Programming* Add Line lub z wykorzystaniem skrótu klawiszowego **]**. Każdorazowe wywołanie zwiększa wielkość bloku i możliwych do zapisania instrukcji. Zwróć uwagę, że ostatnie pole dostępne w bloku jest polem zwracającym wartość zmiennej w nim zapisanym i zazwyczaj wykorzystywane jest do zwrócenia wyniku ostatecznego. Przypisanie wartości do zmiennej programowane jest przy pomocy strzałki skierowanej w lewo ← dostępnej z palety *Programming* lub z pomocą skrótu klawiszowego **{**.

$$s(t) := \left| \begin{array}{l} s_1 \leftarrow a * t^2 \\ s_2 \leftarrow v_0 * t \\ wynik \leftarrow s_1 + s_2 \end{array} \right.$$

Pamiętaj, że zmienne muszą być zdefiniowane wcześniej lub wpisane jako argumenty do funkcji, np.:

$$s(t, a, v_0) := \left| \begin{array}{l} s_1 \leftarrow a * t^2 \\ s_2 \leftarrow v_0 * t \\ wynik \leftarrow s_1 + s_2 \end{array} \right.$$

Zad. 2

Wyznacz pole trójkąta jeśli podane są długości boków trójkąta a, b, c. Skorzystaj ze wzorów:

$p = \frac{a+b+c}{2}$ – połowa obwodu trójkąta

$A = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}$ – pole powierzchni trójkąta

Instrukcje warunkowe

Zad. 3

Napisz instrukcję, która obliczy wartość działania $\frac{x-4}{4x}$, ale w przypadku dzielenia przez zero, zwróci informację o błędzie. Pierwszym etapem będzie stworzenie *bloku* instrukcji, który zaczyna się od instrukcji warunkowej if (pamiętaj o wybraniu instrukcji z palety, nie wpisuj jej ręcznie!). Zwróć uwagę, że przed instrukcją warunkową znajduje się wartość wybierana, w wypadku gdy warunek instrukcji jest spełniony, a następnie przypisywana do zmiennej a.

$$p_1(x) := \left| \begin{array}{l} a \leftarrow \dfrac{x-3}{4x} \ \ if\ x \neq 0 \\ a \end{array} \right.$$

Tę samą instrukcję można również zapisać w inny, bardziej rozbudowany sposób. Po ustawieniu kursora na wolnym polu przed instrukcją if, wybierz operator Add Line, lub użyj skrótu **}**.

$$p_2(x) := \left| \begin{array}{l} if\ x \neq 0 \\ \quad \left| \begin{array}{l} a \leftarrow x - 3 \\ b \leftarrow 4x \\ wynik \leftarrow \dfrac{a}{b} \end{array} \right. \\ wynik \end{array} \right.$$

Zwrócenie błędu można zaprogramować przez wykorzystanie drugiej instrukcji *if*, oraz przypisanie do zmiennej wynikowej tekstu błędu (<u>zauważ, że porównanie logiczne musi zostać wybrane z palety *Boolean*</u>).

$$p_3(x) := \left| \begin{array}{l} if\ x \neq 0 \\ \quad \left| \begin{array}{l} a \leftarrow x - 3 \\ b \leftarrow 4x \\ wynik \leftarrow \dfrac{a}{b} \end{array} \right. \\ \ \ wynik \leftarrow \text{"}Dzielenie\ przez\ zero\ jest\ niedozwolone\text{"}\ if\ x = 0 \\ wynik \end{array} \right.$$

Zad. 4

Bardzo przydatnym operatorem jest on error. Pozwala on na zdefiniowanie działania programu w przypadku napotkania jakiegokolwiek błędu. Dzięki temu zaprogramowanie przykładu z zad. 3 można skrócić do jednej linijki. Pamiętaj o wywoływaniu operatorów z palety.

$$p_4 := \text{"Dzielenie przez zero lub inny błąd"}\ on\ error\ \frac{x-3}{4x}$$

Zad. 5

Jeśli x nie jest różne od zera to niepotrzebne staje się wykorzystanie kolejnej instrukcji warunkowej. Dlatego w programowaniu przewidziana została instrukcja *else* (co oznacza: „w innym wypadku), która w oprogramowaniu *Mathcad* nazywana jest instrukcją otherwise. Jej zastosowanie jest podobne jak w przypadku poprzedniego zadania, jedyna różnica to zastąpienie ostatniej instrukcji warunkowej wraz z warunkiem instrukcją *otherwise* wybraną z palety lub za pomocą skrótu **Ctrl** + **}**. Spróbuj napisać zadanie 3 wykorzystując tę instrukcję.

Zad. 6

W zależności od obliczonej wartości liczby Reynoldsa określającej charakterystykę przepływu wyznacz moc cieplną traconą przez rurociąg. Jeśli liczba Reynoldsa jest mniejsza od 40, zwróć informację o zbyt małej prędkości. Będziesz potrzebować następujących informacji:

$$Re = \frac{u \cdot d}{\nu}$$

Dla $Re > 4000$

$$Q = A \cdot \Delta T \cdot \frac{\lambda}{d} \cdot 0{,}369\ Re^{0{,}618}$$

Dla $40 < Re < 4000$

$$Q = A \cdot \Delta T \cdot \frac{\lambda}{d} \cdot 1{,}31\ Re^{0{,}466}$$

Gdzie:

u – prędkość wyrażona w m/s

$d = 0{,}02\ m$ – średnica charakterystyczna

$\nu = 0{,}9 \cdot 10^{-6}\ m^2/s$ – lepkość kinematyczna

$\lambda = 0{,}6\ W/(m \cdot K)$ – przewodność cieplna

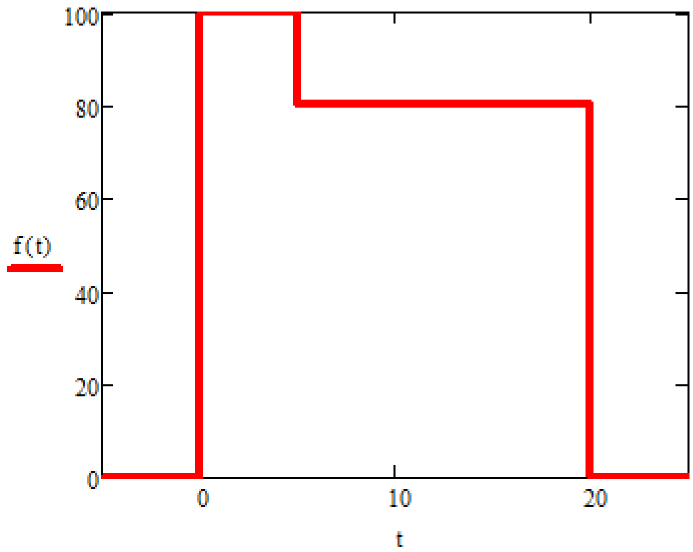
$A = 1\ m^2$ – powierzchnia wymiany ciepła

$\Delta T = 10\ K$ – różnica temperatur

Zad. 7

Zdarza się, że wartość jakiegoś parametru nie jest możliwa do przedstawienia jedną funkcją lecz większą ilością funkcji lub wartości, w takim przypadku bardzo pomocne okazuje się zastosowanie instrukcji warunkowych.

Napisz funkcję, która dla $t < 0$ będzie przyjmować wartość 0, dla $t \in < 0{,}5)$ będzie przyjmować wartość 100, dla $t \in < 5{,}20)$ wartość 80, a dla pozostałego zakresu 0.



Pętle

Bardzo często w praktyce inżynierskiej należy przeprowadzić pewne obliczenia wielokrotnie. O ile trzykrotne wykonanie pewnych obliczeń jest co najwyżej uciążliwe to przeprowadzenie nawet prostych obliczeń 100 lub więcej razy staje się czasochłonne lub nawet niemożliwe. Z tego powodu stosuje się pętle, które pozwalają na przeprowadzenie tych obliczeń dla zmiennych przybierających różne wartości.

Zad. 8

Pierwszym programem wykorzystującym pętlę `for` będzie funkcja zwracająca wartości funkcji sinus. Po przepisaniu poniższego kodu przedstaw go na wykresie.

$$sinus := \left| \begin{array}{l} for\ i \in 1..360 \\ \quad \left| \begin{array}{l} a_i \leftarrow \sin(i \cdot {}^\circ) \\ a \end{array} \right. \\ a \end{array} \right.$$

Zad. 9

Kolejnym programem wykorzystującym pętle, będzie funkcja licząca średnią z podanych wartości. W pierwszym kroku należy uznać pierwszy wyraz wektora jako największą liczbę, następnie porównywać kolejne elementy wektora z największą liczbą, a jeżeli nowa liczba jest większa od poprzednie to przypisywać jej wartość do aktualnej najwyższej.

$$maksymalna(M) := \left| \begin{array}{l} wynik \leftarrow M_0 \\ for\ licz \in 1..length(M)-1 \\ \quad wynik \leftarrow M_{licz}\ \ if\ \ wynik < M\_licz \\ wynik \end{array} \right.$$

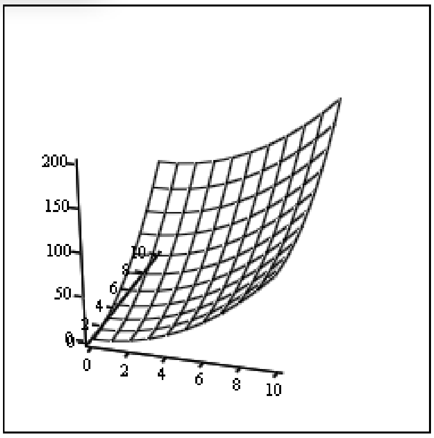
Pamiętaj, że wszystkie operatory musisz wywołać z palety narzędziowej lub skrótem klawiszowym. Zwróć również uwagę, że wykorzystana została wbudowana funkcja *length(x)* zwracająca długość wektora.

Zad. 10

Wyznacz macierz w której wartości będą wynikiem równania $x^2 + y^2$, gdzie x jest pierwszym argumentem macierzy (numerem wiersza), a y jest drugim argumentem (numerem kolumny). Zwróć uwagę, że dolny indeks w zapisie jest wywoływany funkcją *Subscript* lub skrótem klawiszowym **[**.

$$z := \left| \begin{array}{l} for\ \ i \in 0..10 \\ \quad for\ \ i \in 0..10 \\ \quad\quad wynik_{i,j} \leftarrow x_i^2 + y_j^2 \\ wynik \end{array} \right.$$

Wyznaczona macierz z przedstawiona w formie graficznej powinna wyglądać jak na wykresie poniżej:



z

Zad. 11

Napisz funkcję, która na podstawie danych wejściowych liczby i wektora zwróci pierwszy element wektora większy od podanej wartości liczbowej oraz informację, który jest to element. Zwróć uwagę, że wynik jest wektorem, można to osiągnąć tworząc wektor z palety *Matrix*.

$$thresh(t,M) := \left|\begin{array}{l} i \leftarrow 0 \\ while\ M_i < t \\ \qquad i \leftarrow i + 1 \\ \begin{pmatrix} i \\ M_i \end{pmatrix} \end{array}\right.$$

Zad. 12

Jedną z instrukcji wykorzystywanych w programowaniu jest instrukcja *break*, pozwala na przerwanie wykonywania instrukcji w której jest użyta. Dla przykładu poniżej pokazany został pseudokod programu przerywającego instrukcję for po osiągnięciu wartości 5 przez zmienną j. Wewnętrzna pętla *for* jest przerywana, natomiast zewnętrzna jest powtarzana dopóki i nie osiągnie wartości 3. Oznacza to, że wewnętrzna pętla będzie wykonywana dokładnie 3 razy.

$for\ i \in 1{,}3$

$\qquad for\ j \in 1{,}10$

$\qquad\qquad suma = suma + j$

$\qquad\qquad break\ if\ i = 3$

Napisz program sumujący pierwsze trzy elementy macierzy większe od 5, wykorzystując instrukcję break.

$$sumowanie(M) := \left|\begin{array}{l} i \leftarrow 0 \\ for\ j \in length(M) - 1 \\ \quad \left|\begin{array}{l} if\ M_j > 5 \\ \quad \left|\begin{array}{l} suma \leftarrow +M_j \\ i \leftarrow i + 1 \end{array}\right. \\ break\ if\ i \geq 3 \end{array}\right. \\ suma \end{array}\right.$$

Zad. 13

Napisz program sprawdzający jak dużo liczb parzystych znajduje się w zakresie od 1 do n, gdzie n jest argumentem funkcji. Pamiętaj o wywoływaniu operatorów z palet narzędziowych. Wykorzystaj wbudowaną funkcję *mod(x,a)* – modulo. Napisany program pomija instrukcję zwiększania wartości zmiennej wynik jeśli liczba jest nieparzysta.

$$ileparzystych(n) := \left|\begin{array}{l} wynik \leftarrow 0 \\ for\ i \in 1..n \\ \qquad \left|\begin{array}{l} continue\quad if\ mod(i,2) = 1 \\ wynik \leftarrow wynik + 1 \end{array}\right. \\ wynik \end{array}\right.$$

Zad. 14

Napisz program zwracający piąty element listy 1 do n, z wykorzystaniem operatora *return*.

$$retur(n) := \left| \begin{array}{l} wynik \leftarrow 0 \\ for\ \ i \leftarrow 1..n \\ \quad \left| \begin{array}{l} wynik \leftarrow wynik + 1 \\ return\ \ wynik\ \ \ if\ \ i = 5 \end{array} \right. \\ wynik \end{array} \right.$$

Funkcje własne

Zad. 15

Napisz program, który znajdzie wszystkie liczby pierwsze z zakresu 1 do n. Pamiętaj, że za pomocą operatora *Subscript* możesz dodawać kolejne elementy wektora bez potrzeby deklarowania jego wielkości.

Zad. 16

Napisz program obliczający pierwiastki rzeczywiste równania kwadratowego. Wywołuj funkcję z podanymi współczynnikami a, b, c.

Operatory – programowanie

Tab. 2 Operatory, programowanie – lista skrótów

OPERACJA	SKRÓT	IKONA
Dodanie bloku programistycznego	**]**	Add Line
Przypisanie wartości do zmiennej	**{**	←
Instrukcja warunkowa *if*	**}**	if
Instrukcja warunkowa *else*, nazywana *otherwise* w programie Mathcad	**Ctrl** + **}**	otherwise
Pętla *for*	**Ctrl** + **"**	for
Pętla *while*	**Ctrl** + **]**	while
Przerwanie wykonywania instrukcji (wyjście z instrukcji)	**Ctrl** + **{**	break
Kontynuowanie wykonywania instrukcji (pominięcie pojedynczej iteracji)	**Ctrl** + **[**	continue
Zwracanie wyniku programu	**Ctrl** + **|**	return
Obsługa błędu	**Ctrl** + **'**	on error

Lista zadań 6

Zad. 1

W ćwiczeniu 3 zdefiniowałeś funkcję obliczającą drogę pokonaną przez ciało w ruchu jednostajnie zmiennym.

$$s(t) = \frac{at^2}{2} + v_0 t$$

Zapisz ją wykorzystując bloki dostępne z paska narzędziowego *Programming* 

lub z wykorzystaniem skrótu klawiszowego **]**.

Zad. 2

Wyznacz pole trójkąta jeśli podane są długości boków trójkąta a, b, c. Skorzystaj ze wzorów:

$p = \frac{a+b+c}{2}$ – połowa obwodu trójkąta

$A = \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$ – pole powierzchni trójkąta

Zad. 3-5

Napisz instrukcję, która obliczy wartość działania $\frac{x-4}{4x}$, ale w przypadku dzielenia przez zero, zwróci informację o błędzie.

Zad. 3

Wykorzystaj tylko instrukcje warunkowe *if*.

Zad. 4

Wykorzystaj obsługę błędu *on error.*

Zad. 5

Wykorzystaj instrukcję *otherwise*.

Zad. 6

W zależności od obliczonej wartości liczby Reynoldsa określającej charakterystykę przepływu wyznacz moc cieplną traconą przez rurociąg. Jeśli liczba Reynoldsa jest mniejsza od 40, zwróć informację o zbyt małej prędkości. Będziesz potrzebować następujących informacji:

$$Re = \frac{u \cdot d}{\nu}$$

Dla $Re > 4000$

$$Q = A \cdot \Delta T \cdot \frac{\lambda}{d} \cdot 0{,}369\ Re^{0{,}618}$$

Dla $40 < Re < 4000$

$$Q = A \cdot \Delta T \cdot \frac{\lambda}{d} \cdot 1{,}31\ Re^{0{,}466}$$

Gdzie:

u – prędkość wyrażona w m/s

$d = 0{,}02\ m$ – średnica charakterystyczna

$\nu = 0{,}9 \cdot 10^{-6}\ m^2/s$ – lepkość kinematyczna

$\lambda = 0{,}6\ W/(m \cdot K)$ – przewodność cieplna

$A = 1\ m^2$ – powierzchnia wymiany ciepła

$\Delta T = 10\ K$ – różnica temperatur

Zad. 7

Napisz funkcję, która dla $t < 0$ będzie przyjmować wartość 0, dla $t \in < 0{,}5)$ będzie przyjmować wartość 100, dla $t \in < 5{,}20)$ wartość 80, a dla pozostałego zakresu 0. Przedstaw wynik na wykresie.

Zad. 8

Napisz program przypisujący do kolejnych współrzędnych wektora wartości funkcji sinus (możesz skorzystać z wbudowanej funkcji *sin()*). Po napisaniu programu przedstaw wynik na wykresie.

Zad. 9

Napisz program liczący średnią wartość z podanego wektora. Może Ci się przydać wbudowana funkcja *length(M).*

Zad. 10

Wyznacz macierz w której wartości będą wynikiem równania $x^2 + y^2$, gdzie x jest pierwszym argumentem macierzy (numerem wiersza), a y jest drugim argumentem (numerem kolumny). Przedstaw wynik na wykresie 3D.

Zad. 11

Napisz funkcję, która na podstawie danych wejściowych liczby i macierzy zwróci pierwszy element macierzy większy od podanej wartości liczbowej oraz informację, który jest to element. Jako wynik zwróć wektor dwuelementowy gdzie pierwszym elementem jest numer, a drugim wartość wyniku.

Zad. 12

Napisz program sumujący pierwsze trzy elementy macierzy większe od 5. Możesz wykorzystać instrukcję *break*.

Zad. 13

Napisz program sprawdzający jak dużo liczb parzystych znajduje się w zakresie od 1 do n, gdzie n jest argumentem funkcji. Możesz wykorzystać wbudowaną funkcję modulo.

Zad. 14

Napisz program zwracający piąty element listy 1 do n, z wykorzystaniem operatora *return*.

Zad. 15

Napisz program, który znajdzie wszystkie liczby pierwsze z zakresu 1 do n. Pamiętaj, że za pomocą operatora *Subscript* możesz dodawać kolejne elementy wektora bez potrzeby deklarowania jego wielkości.

Zad. 16

Napisz program obliczający pierwiastki rzeczywiste równania kwadratowego. Wywołuj funkcję z podanymi współczynnikami a, b, c.

Zad. 17*

Napisz funkcję liczącą medianę z podanego przez użytkownika zbioru liczb. Napisz dwa osobne podprogramy dla parzystej liczby danych oraz nieparzystej, a następnie napisz program zarządzający, który wywoła odpowiedni podprogram.

Ćwiczenia 7 - Mathcad

Mathcad: budowanie analitycznych modeli fizycznych.

Rzut ukośny

Celem ćwiczenia jest wykorzystanie nabytych umiejętności z programu Mathcad do budowy modeli zjawisk fizycznych. Dla porównania z możliwościami MS Excel jest to samo zagadnienie co w Ćwiczeniach 2.

Zad. 1

Przygotuj arkusz symulujący strzelanie pocisku z katapulty, wykorzystując równania rzutu ukośnego. Uwzględnij równania współrzędnych pocisku:

- bez oporu powietrza:

$$x(t) = v_{0x}\, t$$

$$y(t) = v_{0y}\, t - \frac{g\, t^2}{2}$$

- z oporem powietrza (równania uwzględniają siłę oporu powietrza proporcjonalną do prędkości $F_x = -k\, v_x$, $F_y = -k\, v_y$):

$$x(t) = \frac{v_{0x}}{B}\left(1 - e^{-Bt}\right)$$

$$y(t) = \left(\frac{v_{0y}}{B} + \frac{g}{B^2}\right)\left(1 - e^{-Bt}\right) - \frac{g\, t}{B}$$

gdzie:

$v_{0x} = v_o \cos(\alpha)$	-	składowa x prędkości początkowej
$v_{0y} = v_o \sin(\alpha)$	-	składowa y prędkości początkowej
$g = 9{,}81\,\frac{m}{s^2}$	-	przyspieszenie ziemskie
$B = \frac{k}{m}\, , s^{-1}$	-	jednostkowy opór powietrza

Dane:

$m = 10\ kg$	-	masa pocisku
$k = 1\,\frac{kg}{s}$	-	opór powietrza
$\alpha = 40°$	-	kąt wystrzału pocisku
$v_o = 55\,\frac{m}{s}$	-	opór powietrza

Wygeneruj i sformatuj wykres, który pokaże kierunek wystrzału pocisku oraz trajektorię lotu pocisku przy uwzględnieniu oporu powietrza i bez oraz kierunek wyrzutu. Znajdź maksymalną wysokość pocisku i odpowiadającą mu współrzędną x, a także zasięg pocisku. Zadanie rozwiąż operując jednostkami fizycznymi.

Na początku należy zdefiniować funkcje osobno dla ruchu bez oporu i z oporem powietrza. Maksimum można znaleźć na kilka sposobów. Jednym z nich jest funkcja *Maximize*, która zwraca wartość zmiennej, dla której znajdzie maksimum. Jako argument funkcji należy podać nazwę funkcji (bez jej argumentu) oraz nazwę zmiennej. Konieczne jest zdefiniowanie wartości początkowej zmiennej analogicznie jak dla procedury *Given-Find*. Aby wyświetlić wartość maksymalną wystarczy następnie wywołać funkcję $y(t_{maks})$.

$$t := 1s$$

$$t_{maks} := \text{Maximize}(y, t) = 3.605\ s$$

$$y(t_{maks}) = 63.725\ m$$

Drugim sposobem może być zdefiniowanie wektora czasu z małym krokiem czasowym, np.: $t_1 := (0, 0.1s..\ 10s)$ i znalezienie maksimum funkcją *max* :

$$\max(y(t_1)) = 63.725\ m$$

To rozwiązanie może być mniej dokładne, jeśli zdefiniuje się wektor czasu z dużym krokiem czasowym, gdyż funkcja *max* przeszukuje jedynie wartości znajdujące się w wektorze.

Aby wyznaczyć zasięg pocisku, wystarczy rozwiązać w dowolny sposób równanie $y(t)=0$. Będą 2 rozwiązania tego równania: pierwsze w punkcie początkowym i drugie w chwili, gdy pocisk uderza o ziemię. Wystarczy do drugiego wyniku przypisać zmienną np. t_{Xmax}, po czym wywołać funkcję $x(t_{Xmax})$.

Rozwiązując równocześnie zadanie dla rzutu bez oporu i z oporem powietrza, pamiętaj, aby funkcje nazywać inaczej, tak żeby można je wywołać jednoczenie na wykresie. Wektor czasu natomiast może być ten sam dla obu przypadków.

Zad. 2

Dla danych z poprzedniego zadania i przypadku uwzględniającego opór powietrza oblicz i zaprezentuj na wykresie trajektorię lotu pocisku wystrzelonego pod różnymi kątami: $\alpha_1=30°$, $\alpha_2=45°$, $\alpha_3=50°$. Wszystkie tory lotu powinny znajdować się na wspólnym wykresie współrzędnych x, y.

Aby rozwiązać zadanie, zdefiniuj współrzędne położenia pocisku jako funkcje dwóch zmiennych (czasu i kąta), natomiast składowe prędkości jako funkcje jednej zmiennej (kąta).

Zdefiniuj kąt α jako wektor 3 elementowy, a następnie poprzez odpowiednie wywołanie stwórz osobne wektory współrzędnych x i y dla każdego kąta i wstaw do wykresu.

Lista zadań 7

Zad. 1

Przygotuj arkusz symulujący strzelanie pocisku z katapulty, wykorzystując równania rzutu ukośnego. Uwzględnij równania współrzędnych pocisku:

- bez oporu powietrza:

$$x(t) = v_{0x}\, t$$

$$y(t) = v_{0y}\, t - \frac{g\, t^2}{2}$$

- z oporem powietrza (równania uwzględniają siłę oporu powietrza proporcjonalną do prędkości $F_x = -k\, v_x$, $F_y = -k\, v_y$):

$$x(t) = \frac{v_{0x}}{B}(1 - e^{-Bt})$$

$$y(t) = \left(\frac{v_{0y}}{B} + \frac{g}{B^2}\right)(1 - e^{-Bt}) - \frac{g\, t}{B}$$

gdzie:

$v_{0x} = v_o \cos(\alpha)$	-	składowa x prędkości początkowej
$v_{0y} = v_o \sin(\alpha)$	-	składowa y prędkości początkowej
$g = 9{,}81\, \frac{m}{s^2}$	-	przyspieszenie ziemskie
$B = \frac{k}{m}\, , s^{-1}$	-	jednostkowy opór powietrza

Dane:

$m = 10\, kg$	-	masa pocisku
$k = 1\, \frac{kg}{s}$	-	opór powietrza
$\alpha = 40°$	-	kąt wystrzału pocisku
$v_o = 55\, \frac{m}{s}$	-	opór powietrza

Wygeneruj i sformatuj wykres, który pokaże kierunek wystrzału pocisku oraz trajektorię lotu pocisku przy uwzględnieniu oporu powietrza i bez oraz kierunek wyrzutu. Znajdź maksymalną wysokość pocisku i odpowiadającą mu współrzędną x, a także zasięg pocisku. Zadanie rozwiąż operując jednostkami fizycznymi.

Zad. 2

Dla danych z poprzedniego zadania i przypadku uwzględniającego opór powietrza oblicz i zaprezentuj na wykresie trajektorię lotu pocisku wystrzelonego pod różnymi kątami: α_1=30°, α_2=45°, α_3=50°. Wszystkie tory loty powinny znajdować się na wspólnym wykresie współrzędnych x, y.

Zad. 3*

Analogicznie do zad. 2 stwórz wykres trajektorii lotu pocisku dla różnych wartości oporu powietrza: k_1=0,1 $\frac{kg}{s}$, k_2=1 $\frac{kg}{s}$, k_3=5 $\frac{kg}{s}$. Pozostałe dane przyjmij jak w zad. 1.

Zad. 4*

Bazując na modelu rzutu ukośnego z oporem powietrza (z zad. 1) wyznacz maksymalną wysokość osiąganą przez pocisk dla wszystkich kombinacji wartości kąta i oporu powietrza z zakresu: α=30°, 32°.. 60° , k=0.5, 1.. 2.5 $\frac{kg}{s}$. Stwórz trójwymiarowy wykres konturowy $y_{max}(k,\alpha)$. Wyniki obliczeń wpisuj do macierzy przy pomocy pętli.

Ćwiczenia 8 - Mathcad

Kolokwium zaliczeniowe z programu Mathcad.

Cały czas zajęć tj. 1,5 godziny jest przewidziany na kolokwium zaliczeniowe z umiejętności obsługi programu Mathcad i wykonywania obliczeń. Kolokwium polega na samodzielnym rozwiązaniu przedstawionych zadań w środowisku Mathcad i zapisaniu ich w formie pliku.

Zakres kolokwium:

- posługiwanie się różnymi typami zmiennych
- operacje na danych przy wykorzystaniu wbudowanych i zdefiniowanych przez użytkownika jednostek miar
- zastosowanie wbudowanych funkcji programu
- wykorzystanie obliczeń symbolicznych w równaniach
- znajdowanie optimów lokalnych i globalnych funkcji
- rozwiązywanie równań i układów równań
- tworzenie i edycja wykresów dwuwymiarowych i trójwymiarowych
- importowanie danych, współpraca z plikami MS Excel
- podstawy programowania, wykorzystanie instrukcji warunkowych i pętli do obliczeń
- budowanie prostych analitycznych modeli zjawisk i zagadnień z fizyki

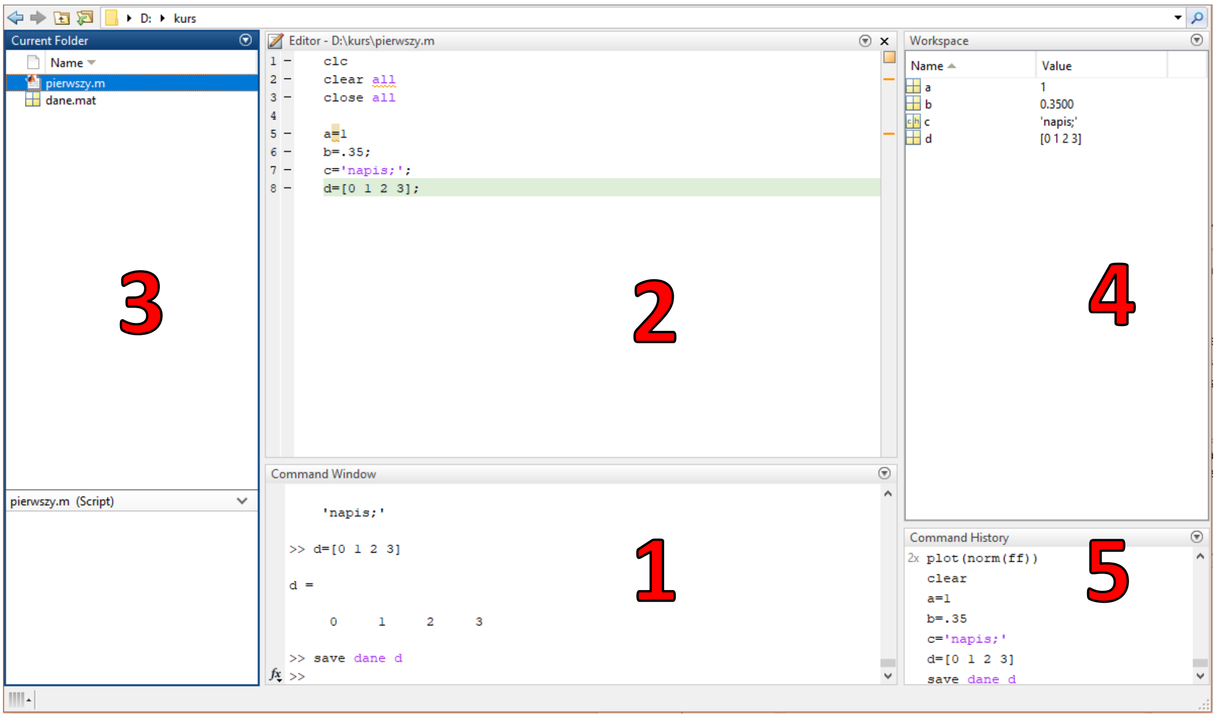
Ćwiczenie 9 – Matlab

Wprowadzenie, interfejs użytkownika, typy zmiennych, podstawowe operacje na danych, operatory logiczne.

Wprowadzenie – Interfejs

MatLab - MATrix LABoratory to program komputerowy do wykonywania obliczeń naukowych i inżynierskich, przetwarzania danych i ich wizualizacji, a także do tworzenia modeli i symulacji komputerowych.

Interfejs programu składa się z następujących elementów:



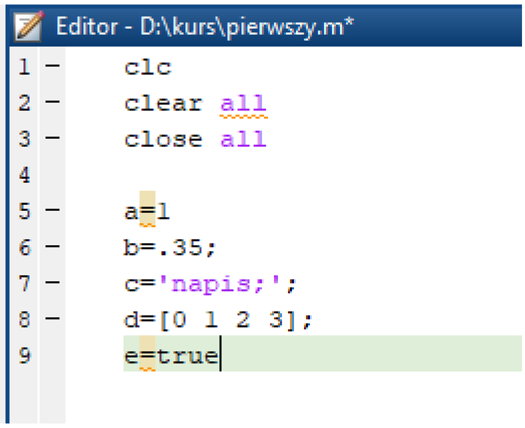
1) Linia komend – służy do obsługi programu za pomocą komend w języku *Matlab*.
2) Edytor – do pisania skryptów (m-pliki) i funkcji(również m-pliki, ale uruchamiane z parametrem).
3) Folder roboczy – podgląd zawartości aktualnego folderu.
4) Lista zmiennych – podgląd na zmienne przechowywane w pamięci, do których można się odwołać. Znajdują się tutaj informacje o rodzaju zmiennej i możliwość podglądu jej zawartości.
5) Historia komend.

Zad. 0

1. Ustaw swoje środowisko tak, aby zawierało przynajmniej edytor, linię komend i listę zmiennych i folder roboczy.
2. Utwórz swój folder w katalogu Student.

Głównym obszarem pracy jest edytor, w którym zapisujemy ciąg komend, które chcemy wykonać (polecenia wykonywane są sekwencyjnie, linijka po linijce).

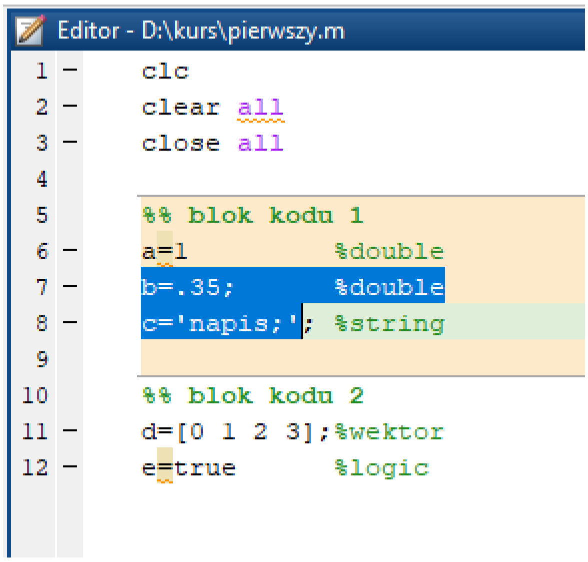
3. Przepisz do edytora następujący kod:



4. W celu odpalenia skryptu wciśnij klawisz F5 na klawiaturze. Zapisz plik w swoim folderze.

Efektem działania skryptu będzie wyświetlenie wartości zmiennych a i e w konsoli (linijki niezakończone średnikiem) i dodanie zestawu zmiennych do pamięci komputera.

Polecenie *clc* czyści zawartość konsoli, *clear all* powoduje wyczyszczenie wszystkich zmiennych z pamięci, natomiast *close all* zamyka wszelkie dodatkowe okna (np. wykresy). Takie rozpoczęcie skryptu zapewnia, że jego działanie odbędzie się bez ewentualnych pozostałości po poprzednich obliczeniach. Kod można komentować za pomocą znaku % (wypróbuj kombinacje klawiszy **ctrl+r** i **ctrl+t** do komentowania bloku kodu w edytorze). Z kolei zastosowanie podwójnego znaku %% powoduje podzielenie dokumentu na sekcje. Zaletą stosowania sekcji, oprócz zwiększenia czytelności, jest to, że można odpalić tylko kod z sekcji, w której znajduje się kursor za pomocą kombinacji **ctrl+enter**. Jeżeli chcesz wykonać tylko część kodu, zaznacz wybrany fragment i naciśnij klawisz **F9**.



Zad. 1

1) Wyzeruj pamięć wpisując w konsoli polecenie *clear all*. Następnie podziel kod w edytorze na bloki i dodaj komentarze, jak na rysunku.
2) Zaznacz fragment kodu i wykonaj go za pomocą klawisza **F9**. Co wyświetliła konsola?
3) Wykonaj kod z bloku 2. za pomocą kombinacji **ctrl+enter** .
4) Zbadaj opcje, jakie daje okno z podglądem zmiennych. Sprawdź opcje po kliknięciu prawym klawiszem myszy na nazwę zmiennej. Wypróbuj wszystkie narzędzia na zmiennych c i d.

Typy zmiennych

Tab. 3 Podstawowe typy zmiennych, używane są w środowisku Matlab

ZMIENNA	OPIS
double	zmienna podwójnej precyzji
single	pojedyncza precyzja
logical	zmienna logiczna przyjmująca wartości *true* lub *false*
uint8,uint16,uint32,uint64	8,16,32 i 64 bitowa liczba całkowita bez znaku
int8,int16,int32,int64	8,16,32 i 64 bitowa liczba całkowita ze znakiem
char, string	zmienna znakowa, ciąg znaków
cell	zmienna komórkowa
struct	struktura

Każda nowa zmienna jest domyślnie typu *double*, o ile nie ma innych przesłanek.

Sposoby wyświetlania

Wyświetlanie liczb możesz zmienić wpisując w linii komend polecenie format *nazwaFormatu.* Dostępne formaty to m. in.:

- *short* - 5 cyfr, stałoprzecinkowe
- *short e* - 5 cyfr, zmiennoprzecinkowa
- *short g* - 5 cyfr, stało- lub zmiennoprzecinkowa
- *long, long e, long g* - 15 cyfr, reprezentacja analogicznie do short
- *rat* - format ułamkowy
- *hex* - styl heksadecymalny
- *bank* - tryb walutowy

Zad. 2

1. Wykonaj ponownie skrypt z edytora w całości (klawisz **F5**).
2. W linii komend wpisz b i zatwierdź enterem.
3. Wpisz format rat i ponownie wyświetl zmienną b.
4. Zmień format na *short g*. Wyświetl wynik działania $b + int8(4)$ oraz $b + 4$.

Podstawowe operacje

W środowisku *Matlab* dokonywać będziemy zazwyczaj następujących, podstawowych operacji:

Operatory arytmetyczne

+- – dodawanie i odejmowanie

* – mnożenie

/ – dzielenie

\ – odwrotne dzielenie macierzowe

^ – potęgowanie

Operatory boolowskie

< mniejsze

> większe

<= mniejsze lub równe

>= większe lub równe

== równe

~= nierówne

Operatory logiczne

& AND

| OR

~ NOT

xor XOR

Zad. 3

1. Wykonaj działania.
   a) a+b
   b) a^2
   c) (a+b)/2
   d) a+b/2
2. Przećwicz dostęp do danych w wektorze.
   a) d(1)+d(2)+d(3)
   b) d(2:end)
   c) d(1:end-1)
   d) d(1:2:end)
3. Sprawdź podane zależności
   a) a==d
   b) d(a==d)
   c) d(a<=d)
4. Utwórz nową zmienną f i wyświetl elementy wektora o wartościach od 6 do 19.
   a) f = 1:2:25
   b) f<19 & f>6
   c) f(f<19 & f>6)
5. Innym sposobem na utworzenie wektora jest użycie funkcji linespace. Utwórz nową zmienną.
   a) g = linspace(45,90,16)

Zad. 4

1. Utwórz nowy skrypt o nazwie wektory1.m.
2. Zdefiniuj zmienne:

    a – wektor od 0 do 10, z krokiem 0,1

    b – wektor od 0 do 100 z krokiem 1

    s – s=rng

    c – c=rand(1,100)

    d – d=rand(1,100)
3. Zainicjalizuj generator liczb losowych do stanu sprzed utworzenia zmiennej c, za pomocą komendy:

    rng(s)
4. Utwórz zmienną e = rand(1,100).
5. Wykonaj obliczenia:
    a) a+b
    b) a.*b
    c) a*c i a*c'
    d) a.^b
6. Porównaj wektory c i e.
7. Wyświetl wynik poleceń
    a) [c d]
    b) [c; d]
8. Utwórz macierz A zawierającą wszystkie wektory i wyświetl ją w linii komend.
9. Podaj:
    a) sumę wszystkich elementów tablicy A
    b) sumę liczb w wierszu 3 oraz sumę liczb w przedostatniej kolumnie
    c) różnicę pomiędzy elementami wiersza pierwszego i ostatniego

Własne funkcje

Zaletą *Matlaba* jest możliwość prostego pisania własnych algorytmów w postaci funkcji. Funkcje mają postać m-pliku, tak jak skrypty. Mogą one być wywoływane wewnątrz programów, bądź z linii komend, identycznie jak używane są wbudowane funkcje, jak *sin*, czy *sum.* Przykładowa funkcja:

```
function [n,f] = mojaFunkcja(n,x1)

f = 10*x1.^2-16;

n=n+1;
```

Zwróci wynik w postaci:

```
>> mojaFunkcja(1,3)
```

ans = 2

lub

>> [x y] = mojaFunkcja(1,3)

x = 2

y = 74

Zmienne tworzone i używane wewnątrz wykonującej się funkcji mają oddzielne miejsce w pamięci komputera, są więc niezależne i ich nazwy mogą się dublować z tymi z głównego programu.

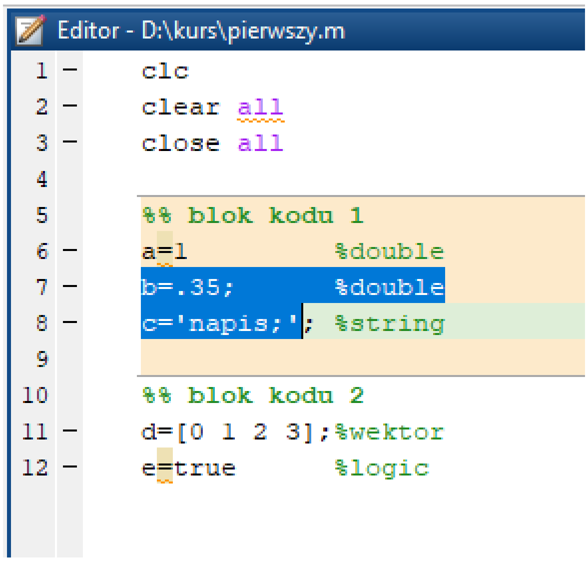
Zad. 5

1. Utwórz m-plik o nazwie mojaFunkcja.m.
2. Napisz funkcję obliczającą wartość funkcji wielomianowej dla zadanego argumentu x.
   a) $F(x) = 2x^3 + 4x - 2$
3. Wywołaj funkcję w linii komend. Zrób to dla trzech losowych argumentów. Upewnij się, że m-plik z funkcją jest w aktualnym folderze roboczym.

Lista zadań 9

Zad. 1

1. Wyzeruj pamięć wpisując w konsoli polecenie *clear all*. Następnie podziel kod w edytorze na bloki i dodaj komentarze, jak na rysunku.
2. Zaznacz fragment kodu i wykonaj go za pomocą klawisza **F9**. Co wyświetliła konsola?
3. Wykonaj kod z bloku 2. za pomocą kombinacji **ctrl+enter** .
4. Zbadaj opcje, jakie daje okno z podglądem zmiennych. Sprawdź opcje po kliknięciu prawym klawiszem myszy na nazwę zmiennej. Wypróbuj wszystkie narzędzia na zmiennych c i d.



Zad. 2

1. Wykonaj ponownie skrypt z edytora w całości (klawisz **F5**).
2. W linii komend wpisz b i zatwierdź enterem.
3. Wpisz format rat i ponownie wyświetl zmienną b.
4. Zmień format na *short g*. Wyświetl wynik działania $b + int8(4)$ oraz $b + 4$.

Zad. 3

1. Wykonaj działania.

   a) a+b
   b) a^2
   c) (a+b)/2
   d) a+b/2
2. Przećwicz dostęp do danych w wektorze.
   e) d(1)+d(2)+d(3)

f) d(2:end)
g) d(1:end-1)
h) d(1:2:end)

3 Sprawdź podane zależności

d) a==d
e) d(a==d)
f) d(a<=d)

4 Utwórz nową zmienną f i wyświetl elementy wektora o wartościach od 6 do 19.

d) f = 1:2:25
e) f<19 & f>6
f) f(f<19 & f>6)

5 Innym sposobem na utworzenie wektora jest użycie funkcji linespace. Utwórz nową zmienną.

g = linspace(45,90,16)

Zad. 4

1. Utwórz nowy skrypt o nazwie wektory1.m.
2. Zdefiniuj zmienne:
   a – wektor od 0 do 10, z krokiem 0,1
   b – wektor od 0 do 100 z krokiem 1
   s – s=rng
   c – c=rand(1,100)
   d – d=rand(1,100)
3. Zainicjalizuj generator liczb losowych do stanu sprzed utworzenia zmiennej c, za pomocą komendy:
   rng(s)
4. Utwórz zmienną e = rand(1,100).
5. Wykonaj obliczenia:
   a) a+b
   b) a.*b
   c) a*c i a*c’
   d) a.^b
6. Porównaj wektory c i e.
7. Wyświetl wynik poleceń
   a) [c d]
   b) [c; d]
8. Utwórz macierz A zawierającą wszystkie wektory i wyświetl ją w linii komend.
9. Podaj:
   a) sumę wszystkich elementów tablicy A

b) sumę liczb w wierszu 3 oraz sumę liczb w przedostatniej kolumnie
c) różnicę pomiędzy elementami wiersza pierwszego i ostatniego

Zad. 5

1. Utwórz m-plik o nazwie mojaFunkcja.m.
2. Napisz funkcję obliczającą wartość funkcji wielomianowej dla zadanego argumentu x.
   a) $F(x) = 2x^3 + 4x - 2$
3. Wywołaj funkcję w linii komend. Zrób to dla trzech losowych argumentów. Upewnij się, że m-plik z funkcją jest w aktualnym folderze roboczym.

Ćwiczenia 10 - Matlab

Prezentacja wyników: wykresy 2D i 3D

Funkcje jednej zmiennej

Jednym z podstawowych zastosowań programu *Matlab* jest wizualizacja danych i wyników obliczeń. Podstawową formą prezentowania jest dwuwymiarowy wykres XY.

Zad. 1

1. Utwórz nowy m-plik i zapisz go w swoim folderze roboczym.
2. Przygotuj dane do umieszczenia na wykresie, będzie to przebieg funkcji trygonometrycznych na przedziale -2π do 2π.

   ```
   x=linspace(-2*pi,2*pi,100);
   ys=sin(x);
   yc=2*cos(x);
   ```

3. Wykonaj wykres funkcji sinus. Przed każdym kolejnym podpunktem wykonaj polecenie *figure*, aby otworzyć nowe okno wykresu.

   ```
   plot(x,ys)
   ```

   Porównaj z plot(ys)

4. Wykonaj wykres funkcji cosinus.

   ```
   plot(x,yc)
   ```

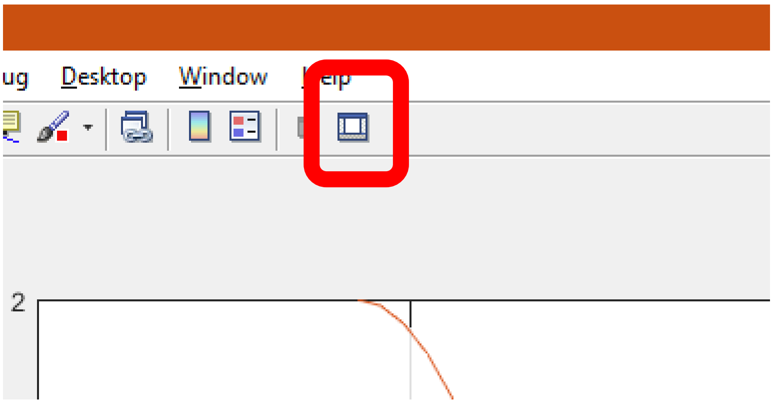
5. Wykonaj wykresy obu funkcji jednocześnie.

   ```
   plot(x,ys)
   hold on
   plot(x,yc)
   ```

6. Narysuj wykres, na którym koordynaty na osi poziomej to zmienna *ys,* a na osi pionowej *yc*.
   a) Sprawdź działanie poleceń:
      i) grid on
      ii) legend
      iii) xlim, ylim

iv) title
v) xlabel, ylabel

b) Dodaj kilka znaków specjalnych do opisów osi i tytułu. Skorzystaj z faktu, że *Matlab* posiada interpreter języka LateX i wstaw, np. \circC, \lambda, \frac{a}{b}.
c) Zapoznaj się ze sposobami upiększenia wykresu za pomocą narzędzi graficznych. Zmień styl i kolor linii, dodaj markery.



d) Zapisz wykres w postaci pliku *png* (File>Save As). Zamknij okno z wykresem.
e) Wykonaj ten sam wykres, tym razem użyj polecenia:

plot(x,ys,'r—', x,yc,'ks')

7. Utwórz nową zmienną, zawierającą wartości obu funkcji w dwóch wierszach.

y = [ys; yc];

xx = [x; x];

8. Utwórz wykres:

plot(xx, y)

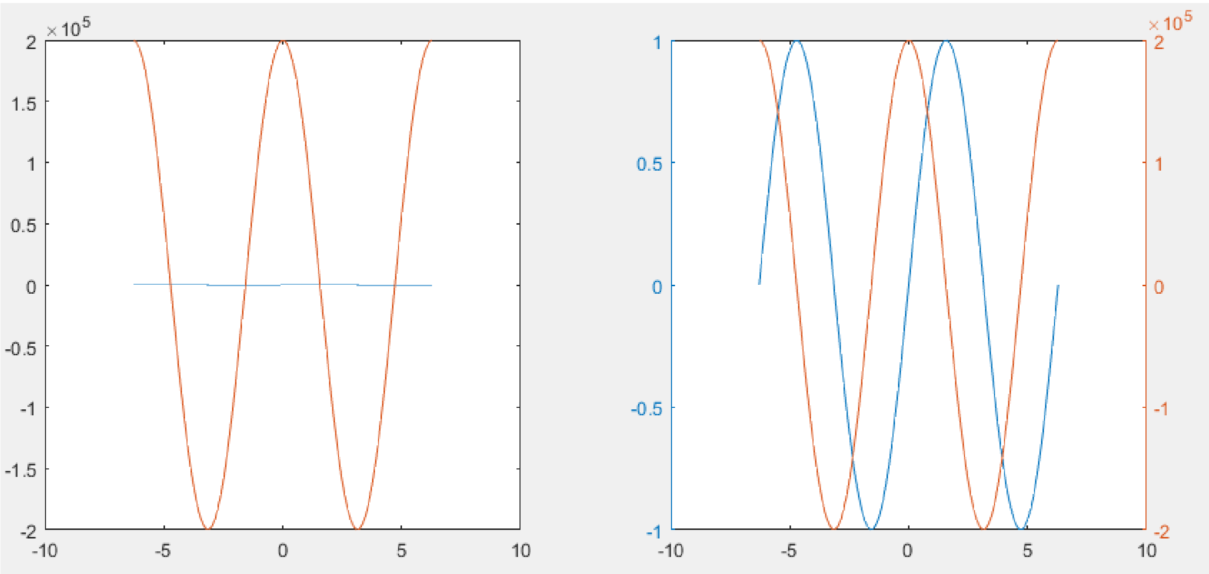
Wykres wygląda nietypowo, dlaczego? Spróbuj użyć zmiennych xx' i y'

Subplot

Za pomocą funkcji *subplot* możliwe jest umieszczanie wielu wykresów w pojedynczym oknie. Zadania 2.-3. wykonaj tak, aby utworzyć dwa wykresy obok siebie. Znajdź polecenie *subplot* w dokumentacji i zastosuj się do zawartych tam wskazówek.

Zad. 2

1) Utwórz nową zmienną y2=yc*1e5;
2) Utwórz wspólny wykres dla zmiennych ys i y2.
3) Utwórz wykres tych samych zmiennych, tym razem wykorzystując dwie niezależne osie pionowe:
   plotyy(x,ys,x,y2)



Zad. 3

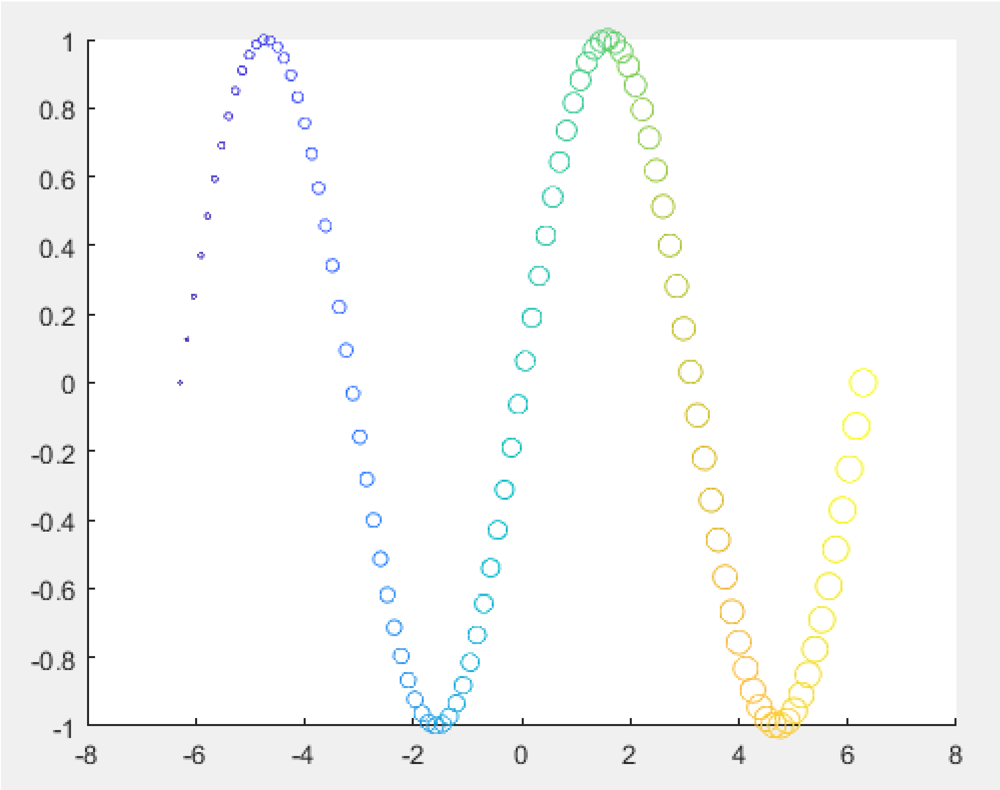
1) Stwórz wykres funkcji Lx=log(x)
2) Wykorzystaj funkcję semilogx(x,Lx). Skąd taki kształt wykresu?

Zad. 4

1) Wykonaj wizualizację danych typu *scatter*.

   scatter(x,ys,1:100,1:100)

Podobnie jak w funkcji plot, definiowane punkty wyświetlane na siatce kartezjańskiej. Funkcja *scatter* umożliwia dodatkowo zdefiniowanie wielkości i koloru każdego z markerów osobno.



Zad. 5

1. Napisz funkcję, która wyświetli nowe okno i narysuje w nim kwadrat. Długość boku kwadratu a oraz jego pozycja x,y będą argumentami funkcji.

   kwadrat(a,x,y)

2. Zadbaj o to, aby wykres rysowany był w obszarze trzech długości boku od środka kwadratu.

Funkcje dwóch zmiennych

Zad. 6

1) W celu wykonania wykresów 3D potrzeba zdefiniować trzy koordynaty plotowanych punktów. Utwórz zmienne:

   X = linspace(-10,10,100)

Y = X

Z = Y.^2 + X.^2

2) Wykonaj wykres

plot3(X,Y,Z)

3) W celu wykonania wykresu powierzchni należy utworzyć siatkę argumentów za pomocą funkcji meshgrid

[XX YY] = meshgrid(X,Y);

ZZ = XX.^2 + YY.^2;

plot3(XX,YY,ZZ)

4) Wypróbuj inne funkcje do wizualizacji danych trójwymiarowych:
   a) surf(XX,YY,ZZ)
   b) contour(XX,YY,ZZ)
   c) contourf(XX,YY,ZZ)
   d) meshc(XX,YY,ZZ)

Zad. 7*

1. Wykreśl kulę, walec i stożek.

Ćwiczenie 11 - Matlab

Funkcje wielomianowe, operacje na macierzach, pętle, instrukcje warunkowe.

Funkcje wielomianowe

Za pomocą funkcji *polyval* można obliczać wartość funkcji wielomianowej o zadanych współczynnikach.

y = polyval(p,x)

gdzie: p – wektor współczynników

x – argument funkcji wielomianowej

Zad. 1

1. Zdefiniuj
   a. x=0:0.25:pi;
   b. y=sin(x);
2. Wyplotuj y=f(x)
3. Aproksymuj krzywą za pomocą wielomianu stopnia n=3 i wyplotuj razem z poprzednim wykresem.

   p_n3=polyfit(x,y,3)
   y_n3=polyval(p_n3,x)
   y_n3a=polyval(p_n3,linspace(0,pi,1000))

Operacje na macierzach

Żeby utworzyć macierz $A = \begin{bmatrix} 1 & 8 & 9 \\ 3 & 2 & 2 \\ 0 & 1 & 1 \end{bmatrix}$ należy wpisać komendę:

A = [1 8 9; 3 2 2; 0 1 1]

Elementy danego wiersza są oddzielane spacjami lub przecinkami, natomiast kolumny za pomocą średnika.

Zad. 2

1. Zdefiniuj zmienne x = [1 2 3]; y = [0 1 0] oraz A;
2. Pomnóż zmienne x i y "element przez element".
   a. x+y
   b. x.*y
3. Wykonaj mnożenie bez kropki

   x * y

oraz

a. x*y’
b. y*x’
c. x’*y

Gwiazdka jest operatorem mnożenia macierzy. Aby mnożenie, lub potęgowanie było wykonywane dla każdego elementu macierzy osobno należy użyć znaku kropki przed symbolem mnożenia lub potęgowania Brak kropki oznacza operację macierzową.

4. Oblicz wartości macierzy B = A^2 i C=A.^{2} (z kropką).
5. Wykonaj działania:
   a. B/2
   b. 120/C i 120./C

Zad. 3

1. Utwórz zmienną A = [1 0 1; 2 1 0; 1 1 1]. Odwróć macierz A funkcją *inv*. Wynik przypisz do zmiennej Ainv.

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$Ainv = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

2. Wykonaj następujące operacje na macierzy A:
   a. sprawdź rozmiar: length(A), size(A)
   b. oblicz wyznacznik (det), rząd (rank) i stopień uwarunkowania (cond)
   c. problem własny (eig)
   d. norma (norm)
   e. sortowanie (sort, sortrows)
   f. wartości największe i najmniejsze (min, max)

3. Funkcja *inv* pozwala na szybkie i wygodne rozwiązywanie układów równań w formie Ax = b. Za pomocą operacji odwracania macierzy znajdź rozwiązanie układu równań:

$1x_1 + 0x_2 + 1x_3 = 7$

$2x_1 - 1x_2 + 0x_3 = 6$

$1x_1 + 1x_2 + 1x_3 = 11$

W równaniu Ax = b, macierz A zawiera współczynniki, wektor $x = [x_1\ x_2\ x_3]$, natomiast b = [7 6 11]'.

Pętle

Pętla for wykonuje dany blok poleceń określoną liczbę razy.

Zad. 4

1) Zapisz poniższą pętlę w nowym m-pliku. Polecenie tic – toc pozwala na zmierzenie prędkości wykonania się kodu pomiędzy znacznikami.

```
maxVal=200

tic
for i = 1: 1: maxVal
        wynik(i)=i^2;
end
        t1=toc
```

2) Porównaj prędkość wykonania operacji z podejściem wektorowym:

```
tic
wynik2=(1: maxVal)^2
t2=toc
```

3) Sprawdź, jak zmienia się różnica w czasach wykonania obu instrukcji w zależności od wielkości problemu (200 < maxVal < 10000). W tym celu zmodyfikuj kod, tak aby obliczał wartość tr=t1-t2.

```
idx=0;
for k = 200:1e5
        idx=idx+1;
        .
        .
        tr(idx)=t1-t2
end
```

4) Narysuj wykres zmienności liczby tr.

Pętla *while* wykonuje dany blok poleceń jeżeli zadany warunek jest spełniony (podczas gdy). Jest to bardzo przydatne narzędzie wszędzie tam, gdzie obliczenia prowadzone są iteracyjnie, do momentu gdy uzyskana odchyłka wyniku spadnie poniżej założonej wartości. Przykład pętli *while*:

```
a = 10;
while( a < 20 )
  fprintf('wartość a to: %d\n', a);
  a = a + 1;
end
```

Pętla wykona się 10 razy, za każdym razem wyświetlając zadany napis i dodając 1 do wartości zmiennej.

Zad. 5*

1. Napisz funkcję, za pomocą której obliczysz rozwiązanie równania kwadratowego metodą Newtona, z zadaną dokładnością. Użyj pętli *while*.
2. Do funkcji dodaj opcję zliczania liczby powtórzeń funkcji i zadbaj, aby *output* z funkcji zawierał historię iteracji wyniku.
3. Utwórz wykres pokazujący, jak zmieniało się rozwiązanie pomiędzy iteracjami.

Instrukcje

Instrukcja *if* pozwala na wykonanie określonych części skryptu, jeżeli spełnione są zdefiniowane w instrukcji warunki. Przykładowa instrukcja *if*:

```
if  z  < 0
        disp(‘Liczba jest ujemna’)
else
        disp(‘Liczba jest dodatnia, lub zero)
end
```

Opcjonalne polecenie *else*, pozwala na zdefiniowanie alternatywnego zestawu instrukcji, jeżeli warunek jest niespełniony. W powyższym przykładzie, jeżeli z>0 oraz brak byłoby opcji *else*, to żadna instrukcja wewnątrz instrukcji warunkowej *if* nie zostałaby wykonana.

Zad. 6

1. Wykorzystaj funkcję kwadrat, plotującą kwadrat o zadanym boku i położeniu. Kwadrat o boku a=1 umieść w punkcie x=5, y=0.5.
2. Napisz skrypt, który na podstawie zadanej prędkości wyrysuje ruch kwadratu Vk=10 w przestrzeni. Przyjmij interwał czasowy delta_t=0.1 i czas maksymalny maxTime=10
3. W pętli *for* plotuj kolejne położenia kwadratu:

```
for i=1:maxTime
        x=x+Vk*delta_t
        kwadrat(a,x,y)
        pause(0.2)
end
```

4. Dodaj do skryptu warunek *if*, który sprawdzi, czy nastąpiła kolizja kwadratu z granicą obszaru, którą ustaw na x=0. Jeżeli kolizja nastąpi w danym kroku czasowym zapisz czas, po jakim to nastąpiło i wyświetl komunikat to kraksie.

Instrukcja *switch* pozwala na wykonanie określonej akcji w zależności od przewidywanego scenariusza. Scenariuszy może być więcej niż jeden, jak w przypadku instrukcji *if*. Dzięki czemu możliwa jest obsługa wielu różnych wariantów w wygodny, kompaktowy i czytelny sposób.

Zad. 7

1. Napisz funkcję, która za argument przyjmie skalar a oraz zwróci wartość

$$b=\begin{cases} a^2; a > 3 \\ a^3; a < 0 \\ a^4; a = 1 \\ \quad 0; \end{cases}$$

```
switch true
   case a>3 %warunek1
      b=a^2
      disp('odpalono warunek 1')
   case a<0 %warunek2
      b=a^3
      disp('odpalono warunek 2')
   case a==1 %warunek2
      b=a^4
```

```
        disp('odpalono warunek 3')
    otherwise
        b=0
        disp('odpalono warunek 4')
end
```

Czyli, jeżeli wynik z pola *case* odpowiada wynikowi komendy za *switch*, to instrukcje wewnątrz tego *case'a* zostaną wykonane.

2. Dodaj jeszcze jeden *case*. Jeżeli zostanie aktywowany, to wyświetl odpowiedni komunikat.

```
Case a<7
```

3. Sprawdź działanie nowej instrukcji *switch* dla a = 4 i a =-9

Ćwiczenie 12 - Matlab

Budowanie prostych numerycznych modeli fizycznych (rzut ukośny, rozszerzalność temperaturowa).

Rzut ukośny

Zad. 1

1. Przygotuj arkusz symulujący strzelanie pocisku z katapulty podobny do tych, przygotowanych wcześniej w programach *Excel* i Mathcad. Równania współrzędnych pocisku:
    - bez oporu powietrza:

$$x(t) = v_{0x}\, t$$

$$y(t) = v_{0y}\, t - \frac{g\, t^2}{2}$$

    - z oporem powietrza zależnym od prędkości $F_x = -k\, v_x$, $F_y = -k\, v_y$:

    gdzie:

    $v_{0x} = v_o \cos(\alpha)$ - składowa x prędkości początkowej

    $v_{0y} = v_o \sin(\alpha)$ - składowa y prędkości początkowej

    $g = 9{,}81 \frac{m}{s^2}$ - przyspieszenie ziemskie

    Dane:

    $m = 10\, kg$ - masa pocisku

    $k = 1 \frac{kg}{s}$ - opór powietrza

    $\alpha = 40°$ - kąt wystrzału pocisku

    $v_o = 55 \frac{m}{s}$ - prędkość początkowa

    $\Delta t = 0{,}05\, s$ - krok czasowy

2. Wygeneruj i sformatuj wykres, który trajektorię lotu pocisku przy uwzględnieniu oporu powietrza i bez, a także kierunek wyrzutu. Znajdź maksymalną wysokość pocisku i odpowiadającą mu współrzędną x, a także zasięg pocisku. Posłuż się pętlą *while*, która obliczać będzie kolejne prędkości i położenia pocisku, do momentu, gdy wysokość y jest dodatnia.

```
i=1
while min(y)> -.001          %y=0 poziom gruntu
```

```
    ax=-(k/m)* vx;             %
    ay=-g-(k/m)*vy;   % zmiany prędkości pocisku zależne od oporu
    vx=vx+ax*delta_t;          %
    vy=vy+ay*delta_t;          % uaktualnione prędkości
    x(i+1)=x(i)+vx*delta_t+.5*ax*delta_t^2;          %
    y(i+1)=y(i)+vy*delta_t+.5*ay*delta_t^2;          % pozycja pocisku
    t(i+1)=t(i)+delta_t;                             % czas całkowity
    i=i+1;                                           % aktualizacja licznika
  end
```

Bez uwzględnienia oporów ruchu

ax=0 – prędkość się nie zmienia, zerowe przyspieszenie

ay=-g – zmiana prędkości wynika tylko z działania siły grawitacji

Zad. 2

Dla danych z poprzedniego zadania i przypadku uwzględniającego opór powietrza oblicz i zaprezentuj na wykresie trajektorię lotu pocisku wystrzelonego pod różnymi kątami: α_1=30°, α_2=45°, α_3=50°. Wszystkie tory loty powinny znajdować się na wspólnym wykresie współrzędnych x, y.

Zad. 3*

Analogicznie do zad. 2 stwórz wykres trajektorii lotu pocisku dla różnych wartości oporu powietrza: $\mathrm{k}_1=0{,}1\,\frac{\mathrm{kg}}{\mathrm{s}}$, $\mathrm{k}_2=1\,\frac{\mathrm{kg}}{\mathrm{s}}$, $\mathrm{k}_3=5\,\frac{\mathrm{kg}}{\mathrm{s}}$. Pozostałe dane przyjmij jak w zad. 1.

Zad. 4*

Bazując na modelu rzutu ukośnego z oporem powietrza (z zad. 1) wyznacz maksymalną wysokość osiąganą przez pocisk dla wszystkich kombinacji wartości kąta i oporu powietrza z zakresu: $\alpha=30°, 32°.. 60°$, $\mathrm{k}=0.5, 1.. 2.5\,\frac{\mathrm{kg}}{\mathrm{s}}$. Stwórz trójwymiarowy wykres $\mathrm{y_{max}}(\mathrm{k},\alpha)$.

Zad. 5*

Wybierz jeden ze scenariuszy rzutu (kąt, opór) i przeprowadź obliczenia zmieniając jedynie wielkość kroku czasowego. Przyjmij wartości $\Delta t = 0{,}001\,; 0{,}01;\ 0{,}1\ i\ 1$. Pokaż na wykresie zależność wysokości maksymalnej, i obliczony zasięg, od kroku czasowego.

Zderzenia sprężyste

Zadanie polega na obliczeniu liczby zderzeń dwóch klocków o zadanych masach, przy założeniu, że warunki początkowe zadane są, jak na rysunku, a zderzenia między klockami są doskonale sprężyste.

Dane:

M_1 = 10 kg

M_2 = 1 kg

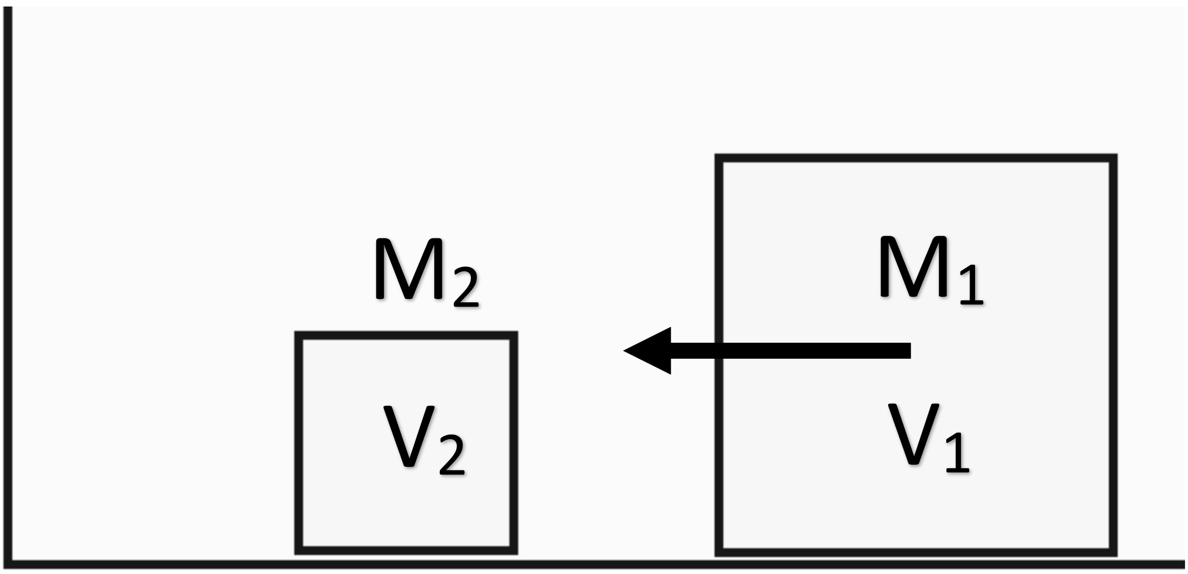
V_{01} = 2 m/s

V_{02} = 0 m/s

x_{10} = 10 m

x_{20} = 1 m

d_t = 0.01 s



Zad. 2

1. Początkowo klocek 2 jest w spoczynku, a klocek 1 posiada prędkość V_0
2. Po zderzeniu sumaryczna energia kinetyczna obu klocków pozostaje niezmienna. Podobnie ma zastosowanie zasada zachowania pędu.
3. Klocek 2, po uderzeniu w ścianę po lewej (x = 0) odbija się, również bez utraty energii. Jego pęd zachowuje wartość, lecz zmienia znak.

4. Narysuj na wykresie (plot) oba kwadraty w pozycjach początkowych. Załóż długość boku klocka 1 – a = 1 m, a klocka 2 – b = 3 m
5. Stwórz model powyższego zjawiska. W pętli oblicz kolejne prędkości i położenia obu klocków. Na początku każdego kroku czasowego ustal, czy obiekty są ze sobą zderzone i czy klocek 2 dobił do ściany w x=0.
   (1) Jeżeli doszło do zderzenia klocków to prędkości:
   $V_1^{nowe} = (V_1\ (m_1\text{-}m_2) + 2M_2\, V_2) / (M_1+M_2);$
   $V_2^{nowe} = (V2\ (m_2\text{-}m_1) + 2M_1\, V_1) / (M_1+M_2);$
   (2) Jeżeli nie doszło do zderzenia, prędkości pozostają bez zmian
   (3) Jeżeli klocek 2 zetknął się ze ścianą, to jego nowa prędkość ma przeciwny znak do dotychczasowego
   $V_2^{nowe} = - V_2^{stare}$
6. Dla każdego klocka nowa pozycja (po upływie czasu dt) będzie obliczona wzorem:
   $x^{nowe} = x\ Vdt$

7. W każdym przebiegu pętli wyplotuj kwadraty na ich aktualnych pozycjach. Dla czytelności użyj funkcji pause(t), gdzie t to czas w sekundach, na jaki zawieszone będzie wykonywanie skryptu. Dzięki temu na wykresie pojawić się powinien płynny, ruchomy obraz prezentujący ewolucję rozwiązania w czasie.
8. Zapisz liczbę zderzeń małego klocka z dużym i ze ścianą. Podaj sumaryczną wartość
9. To samo zrób dla masy dużego klocka:
   a) M1 = 100
   b) M1 = 152
   c) M1 = 10 000
   d) M1 = 1 000 000
10. Zmień krok czasowy na większy, co obserwujesz na animacji?

Ćwiczenie 13 - Matlab

Optymalizacja: metoda Newtona, metoda gradientu sprzężonego.

W prezentowanych zadaniach celem jest odnalezienie ekstremów zadanych funkcji na danym przedziale. Wymagać to będzie wielokrotnego obliczania wartości optymalizowanej funkcji, w badanych punktach. Dlatego do każdej badanej funkcji twórz osobne m-pliki - funkcje, które będą zwracały wynik obliczenia i liczbę wywołań, na przykład:

```
function [n,f] = mojaFunkcja(n,x1)

f = 10*x1.^2-16;

n=n+1;
```

Szukanie ze stałym interwałem

Najprostszą ideologicznie metodą odnajdywania optimów funkcji jest obliczenie jej wartości w badanym zakresie

Zad. 1

1) Oblicz wartości funkcji mojaFunkcja na przedziale x1 od -4 do 4 z krokiem co 0.1. Pamiętaj o aktualizowaniu licznika wywołań funkcji. Wszystkie obliczone wartości przechowaj w wektorze y.
2) Za pomocą funkcji min, sprawdź która wartość y jest najmniejsza. Znajdź argument x1 dla którego funkcja przyjęła wartość minimalną.
3) Zmodyfikuj skrypt, tak aby jego wykonywanie zakończyło się, gdy spełniony zostanie warunek, że y(i) > y(i-1). Jaka oszczędność wywołań funkcji nastąpiła dla warunków zadania?

Metoda Newtona

Metoda ta jest używana, do odnajdywania miejsc zerowych funkcji f(x)=0. W przypadku poszukiwania ekstremów należy znaleźć zero funkcji pochodnej, do optymalizowanej f'(x)=0.

Pochodną dowolnej funkcji, którą potrafimy obliczyć w *Matlab* można zróżniczkować numerycznie w punkcie, poprzez obliczenie

a) jej wartości w punkcie x1
b) wartości w punkcie x1+h
c) wartości w punkcie x1-h

W ten sposób pierwsza i druga pochodna to:

$$f'(x^k) = \frac{f(x^k + h) - f(x^k - h)}{2h} = \frac{p1 - p2}{2h}$$

$$f''(x^k) = \frac{f(x^k + h) - 2f(x^k) + f(x^k - h)}{h^2} = \frac{p_3 - 2p_4 + p_5}{h^2}$$

Kolejne przybliżenia argumentu x wyznacza się następująco:

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$$

Zad. 2

1. Napisz skrypt, w którym znajdziesz minimum funkcji *mojaFunkcja.* Ustaw punkt początkowy $x^0 = -2$ i tolerancję rozwiązania tol = 10^{-4} i h=10^{-3}.

```
%% Newton
tol=10e-4;
h=0.001;
f = 1000;
n=0;
x=10;

while abs(f1)>tol

[n,p1]= mojaFunkcja (n,x+h); [n,p2]= mojaFunkcja (n,x-h);
[n,p3]= mojaFunkcja (n,x+h);  [n,p4]= mojaFunkcja (n,x);  [n,p5]= mojaFunkcja (n,x-h);

f1 = ( p1 - p2 ) / (2*h);           %pierwsza pochodna funkcji mojaFunkcja w punkcie x
f2 = ( p3 - 2*p4 + p5 ) / h^2;  %druga pochodna w punkcie x

x=x-f1/f2;                                %aktualizuj x

end
```

2. Porównaj czas działania algorytmu Newtona i metod z zadania 1.

Zad. 3

1. Wykorzystaj skrypt do obliczania trajektorii rzutu ukośnego. Napisz funkcję, która obliczy tor lotu pocisku, ale zwracać będzie jedynie skalar – zasięg pocisku. Przyjmij stałe wartości masy, prędkości początkowej, oporu.

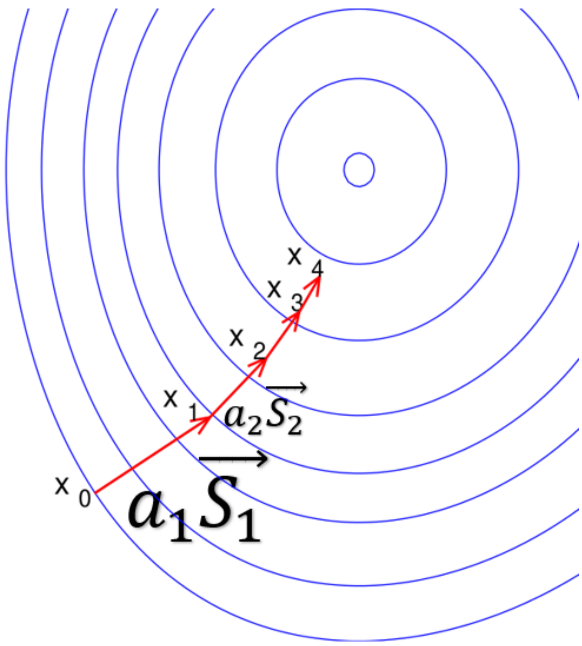
   function L = zasieg(kąt)

2. Zoptymalizuj kąt rzutu, aby osiągnąć maksymalną odległość stosując metodę Newtona i zaczynając od kąta:
   a. 10 stopni
   b. 85 stopni

Metoda gradientu prostego

W metodzie tej również posługiwać się będziemy pierwszą i drugą pochodną. Zauważyć można, że dotychczasowe metody optymalizacji realizowały następującą metodę poszukiwań:

$$\overrightarrow{x_{i+1}} = \overrightarrow{x_i} + a\vec{S}$$



Gdzie:

a – długość wektora **S**

S – kierunek pomiędzy położeniem współrzędnych **x** w obecnej i następnej iteracji.

Procedura obliczeń za pomocą metody gradientu prostego polega na:

a) wybraniu punktu początkowego x^0 i kryterium tolerancji (*tol*)
b) obliczeniu gradientu funkcji kosztu $\nabla f(x^0)$
c) Jeżeli $|\nabla f(x^0)| < tol$, zatrzymaj. W przeciwnym przypadku idź do d)
d) Określ kierunek poszukiwań $S^0 = -\nabla f(x^0)$
e) Oblicz długość a, dla którego wyrażenie
$$f(x^0 + aS^0)$$
przyjmuje wartość minimalną.

f) Zaktualizuj rozwiązanie $x^1 = x^0 + aS^0$i wróć do punktu b)

Nowością tutaj jest fakt, że można tą metodą szukać optimum funkcji wielu zmiennych.

Zad. 4

1. Użyj metody gradient prostego do wyznaczenia optimum funkcji
$$f(x_1, x_2, x_3) = x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3$$

2. Napisz funkcję myf3.m obliczającą wartość zadanej funkcji f w zadanym punkcie.

```
function [n,f] = myf3(n,x1,x2,x3)
    f = x1^2+2*x2^2+2*x3^2+2*x1*x2+2*x2*x3;
n=n+1;
```

3. W celu rozwiązania punktu e) metody napisz funkcję do odnajdywania ‘a’ metodą Newtona. Zwróć uwagę, że x=[x_1 x_2 x_3].

```
function [n,a]=newton(n,x,a,S,h,tol)
pa=-10;

while abs(pa)>tol

  xa=x+a*S;
  xa1=x+(a+h)*S;
  xa2=x+(a-h)*S;

  [n,pa1]=myf3(n,xa1(1),xa1(2),xa1(3));
  [n,pa2]=myf3(n,xa2(1),xa2(2),xa2(3));
  [n,pa3]=myf3(n,xa1(1),xa1(2),xa1(3));
  [n,pa4]=myf3(n,xa(1),xa(2),xa(3));
```

```
    [n,pa5]=myf3(n,xa2(1),xa2(2),xa2(3));

    pa = ( pa1 - pa2 ) / (2*h);
    fb = (pa3 - 2*pa4 + pa5 ) / h^2;
    a=a-pa/fb;

end
```

4. Znajdź minimum funkcji f. Użyj kroku h = 10^{-6}, tolerancji przed zatrzymaniem algorytmu tol=10^{-5} oraz punkt początkowy x^0=(2, 4, 10) I maksymalną liczbę wywołań funkcji n = 1000. Rozwiązanie: (8.04787e-3, -6.81319e-3, 3.4217e-3).

```
xstart=[2 ; 4 ; 10];
h=10e-6;
n=0;
pa=1;
tol=10e-5;
tol1=10e-5;
a=1;
S=1;
x=xstart;

while sum(abs(S))>tol
        [n,f1] = myf3(n,x(1)+h,x(2),x(3));
        [n,f2] = myf3(n,x(1)-h,x(2),x(3));
        p1=(f1-f2)/(2*h);                    %gradient w kierunku osi x1

        [n,f3] = myf3(n,x(1),x(2)+h,x(3));
        [n,f4] = myf3(n,x(1),x(2)-h,x(3));
        p2=(f3-f4)/(2*h);                    %gradient w kierunku osi x2

        [n,f5] = myf3(n,x(1),x(2),x(3)+h);
```

```
        [n,f6] = myf3(n,x(1),x(2),x(3)-h);
        p3=(f5-f6)/(2*h);                    %gradient w kierunku osi x3

        S=-[p1 ; p2 ; p3];                   %kierunek poszukiwań nowego x

        [n,a]=newton(n,x,a,S,h,tol1); %szukanie odległości
        x=x+a*S                                  %nowe x
    end
```

Metoda gradientu sprzężonego

Jest to rozwinięcie poprzedniej metody. Tutaj do wyznaczenia kierunku w danej iteracji wykorzystywane są informacje z poprzedniej informacji, co powinno poprawić szybkość działania algorytmu. Realizowane jest to w następujący sposób:

Określ kierunek poszukiwań w kroku (k-1):

$$S^{k-1} = -\nabla f(x^{k-1})$$

W kroku (k) dodaj poprawdkę:

$$S^{k} = -\nabla f(x^{k}) + \beta_k S^{k-1}$$

Gdzie:

$$\beta = \left(\frac{|\nabla f(x^{k})|}{|\nabla f(x^{k-1})|}\right)^2$$

Zad. 5

1. Zoptymalizuj funkcji Rosenbrota za pomocą metody gradientu prostego, z wykorzystaniem algorytmu Newtona. Rozpocznij w punkcie x^0=(-1.5, 2.25). Algorytm jest analogiczny do tego z zadania 4, należy jednak pamiętać o podmianie obliczanej funkcji (mojaFunkcja -> rosen(n,x1,x2)).

   ```
   function [n,f] = rosen(n,x1,x2)
       f = 100*(x2-x1.^2).^2+(x1-1).^2;
       n=n+1;
   ```

2. Zmodyfikuj skrypt, aby wykorzystać metodę gradientu sprzężonego:

$xstart = [-1.5\ ;\ 2.25\ ];$

$iter = 0;$
$h = 10e - 6;$
$n = 0;$
$pa = 1;$
$tol = 10e - 3;$
$tol1 = 10e - 5;$
$a = 1;$
$S = 1;$
$x = xstart;$
$x\_plot = [];$

```
% pierwszy wektor kierunkowy S0
[n,f1] = rosen(n,x(1)+h,x(2)); [n,f2] = rosen(n,x(1)-h,x(2));
p1=(f1-f2)/(2*h);
[n,f3] = rosen(n,x(1),x(2)+h); [n,f4] = rosen(n,x(1),x(2)-h);
p2=(f3-f4)/(2*h);
S=-[p1 ; p2];

while max(abs(S))>tol      %właściwa pętla

[n,a]=newton2(n,x,alpha,S,h,tol1);
x_new=x+a*S;                                    %nowe X na baize "poprzedniego" S

%conjugate part
[n,f1] = rosen(n,x_new(1)+h,x_new(2)); [n,f2] = rosen(n,x_new(1)-h,x_new(2));
p1=(f1-f2)/(2*h);
[n,f3] = rosen(n,x_new(1),x_new(2)+h); [n,f4] = rosen(n,x_new(1),x_new(2)-h);
p2=(f3-f4)/(2*h);

S_new=-[p1 ; p2];                               %nowe S

if max(abs(S_new))<tol
   break
   disp('break')
end

beta=((norm(-S_new,2))./(norm(-S,2)));
beat=beta^2;                                    %poprawka S
```

```
S=S_new+beta*S;                          %aktualizacja nowego S

x=x_new
x_plot=[x_plot [x]];                     %zapisanie historii x
iter=iter+1;

end
```

3. Wyrysuj funkcję rosen na przedziale x1=[-3:0.1:3] oraz x2=[-3:0.1:3];

```
x1=[-3:0.1:3 ; -3:0.1:3];
[x1,x2]=meshgrid(x1(1,:),x1(2,:));
[n,f]=rosen(n,x1,x2);
surfc(x1,x2,f); view(180,-90)
hold on
```

4. Dodaj do narysowanej powierzchni historię rozwiązania x_plot dla metody gradientu sprzężonego.

```
plot(x_plot(1,:),x_plot(2,:),'o-r')
plot(x_plot(1,:),x_plot(2,:),'ro-')
```

5. Zrób to samo z historią dla metody gradientu prostego i porównaj wyniki.

Ćwiczenie 14 - Matlab

Kolokwium zaliczeniowe

Cały czas zajęć tj. 1,5 godziny jest przewidziany na kolokwium zaliczeniowe z umiejętności obsługi programu *Matlab* i wykonywania obliczeń. Kolokwium polega na samodzielnym rozwiązaniu przedstawionych zadań i zapisaniu ich w formie pliku.

Zakres kolokwium:

- posługiwanie się różnymi typami zmiennych
- operacje na danych
- zastosowanie wbudowanych funkcji programu
- znajdowanie optimów lokalnych i globalnych funkcji
- rozwiązywanie równań i układów równań
- tworzenie i edycja wykresów dwuwymiarowych i trójwymiarowych
- wykorzystanie instrukcji warunkowych i pętli do obliczeń
- budowanie prostych numerycznych modeli zjawisk i zagadnień z fizyki

Ćwiczenie 15 – Prezentowanie wyników

Zasady prezentacji wyników obliczeń, raportów i sprawozdań w procesorze tekstu i prezentacjach multimedialnych.

Wszelkie modele, obliczenia, czy projekty powinny zostać prawidłowo udokumentowane.

Procesor tekstu

Procesor tekstu jest zatem narzędziem, które pozwala na:

- projektowanie poprawnej struktury dokumentu
- sposobów formatowania tekstu z użyciem stylów
- definiowanie podpisów i odsyłaczy oraz automatycznej numeracji obiektów.

Zad. 0

1. Utwórz nowy plik tekstowy w programie Word.
2. Ustaw środowisko pracy, tj. style, ustawienia strony, itd. zgodnie z poniższym:
   a. Ustaw nagłówek i stopkę według wzoru:
      i. nagłówek – Raport TI $data %np. 20.06.19r.
      ii. stopka - Twoje imię i nazwisko
   b. Zdefiniuj styl o nazwie Rozdzialy_Inicjały (np. *NazwaStylu_JR*) o następujących cechach: czcionka *Arial 14*, podkreślona, czerwona, równanie do lewej, odstęp przed i po akapicie 6 pkt., numerowanie. Formatuj stylem Rozdzialy_Inicjały tytuły rozdziałów.
   c. Zdefiniuj styl o nazwie podpisy_*Inicjały:* czcionka *Times New Roman 12, pochylona*, wyjustowane, interlinia 1,5 wiersza, wcięcie od lewej całego akapitu 0,5 cm, wypunktowanie. Formatuj stylem *podpisy_Inicjały* podpisy tabel i rysunków.
   d. Nadaj dokumentowi adekwatne właściwości: autora, tytuł, itd.
   e. Ustaw marginesy: górny, dolny, prawy – 25 mm, lewy – 35 mm.

Zad. 1

1. Przedstaw w formie raportu porównanie trzech modeli rzutu ukośnego, wykonanych w programach *Excel*, *Mathcad* i *Matlab*.

2. Utwórz stronę tytułową, na której umieść:
   a. Tytuł, datę, liczbę stron
   b. Adresata
   c. Nazwisko Autora
   d. Streszczenie (2-4 zdania)

3. Dodaj spis treści na drugiej stronie, a po nim umieść nomenklaturę – spis wszystkich symboli użytych w pracy (wraz z jednostkami). Układ właściwej części raportu powinien być następujący:
    a. Wstęp: opis zagadnienia (czego szukamy?), rysunek poglądowy
    b. Opis modelu: prawa fizyczne i ich zapis matematyczny. Pokazanie danych wejściowych i wyjściowych modelu oraz zastosowanych algorytmów obliczeń.
    c. Wyniki: suchy opis uzyskanych rezultatów, w tym:
        i. Tabelę podsumowującą maksymalny zasięg pocisku, maksymalną wysokość i czas lotu dla wybranej wartości kąta i współczynnika oporu powietrza, obliczone w trzech programach
        ii. Wykres trajektorii pocisków obliczony w trzech programach dla kąta wystrzału: 12^o i współczynnika oporu k = 0,75 m/kg.
    d. Dyskusja wyników: propozycje interpretacji wyników i zauważonych korelacji
    e. Podsumowanie

Pamiętaj, że tabele i rysunki muszą być podpisane i ponumerowane (automatycznie), a także ich obecność musi być uzasadniona w tekście i wyraźnie wskazana po numerze. Osie na wykresach muszą być podpisane, a wielkość czcionki na rysunkach musi być zbliżona do rozmiaru pisma raportu.

Prezentacja multimedialna

Profesjonalna prezentacja multimedialna jest pomocą przy dzieleniu się wiedzą i wynikami pracy. Struktura dokumentu stanowi szkielet dla wystąpienia ustnego, a informacje i obiekty zawarte w prezentacji stanowią ilustrację do opowieści snutej przez referenta.

Często prezentacje przygotowywane są na firmowej formatce. Na Politechnice Wrocławskiej obowiązują wzory dostępne na stronie https://pwr.edu.pl/uczelnia/o-politechnice/materialy-promocyjne/logotyp

Zad. 2

1. Utwórz prezentację w programie PowerPoint.
2. Prezentacja powinna być na szablonie PWr.
3. Informacje zawarte w raporcie (Zad. 1) przenieś do prezentacji multimedialnej.
4. Pamiętaj o strukturze dokumentu:
    a. Slajd tytułowy (autor, data, tytuł, adresat)
    b. Plan wystąpienia (powiedz, o czym powiesz)
    c. Rozdziały (powiedz, co masz powiedzieć)
    d. Podsumowanie (powiedz, o czym powiedziałeś)
    e. Slajd z podziękowaniem za uwagę

5. Prezentacja służy do ilustrowania zagadnień, które trudno opowiedzieć słowami. Zatem nie należy umieszczać dużych fragmentów tekstu, bądź przesadnie wielu wzorów na slajdach. Mniej znaczy więcej.
6. Upewnij się, że na wszystkich ilustracjach oznaczenia są duże i widoczne.