



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA

Wprowadzenie do programu SMath Studio Desktop



WYDZIAŁ
**BUDOWNICTWA,
INŻYNIERII ŚRODOWISKA
I ARCHITEKTURY**
POLITECHNIKI RZESZOWSKIEJ

Karol Pereta

Wiesław Bielak

Grzegorz Piątkowski

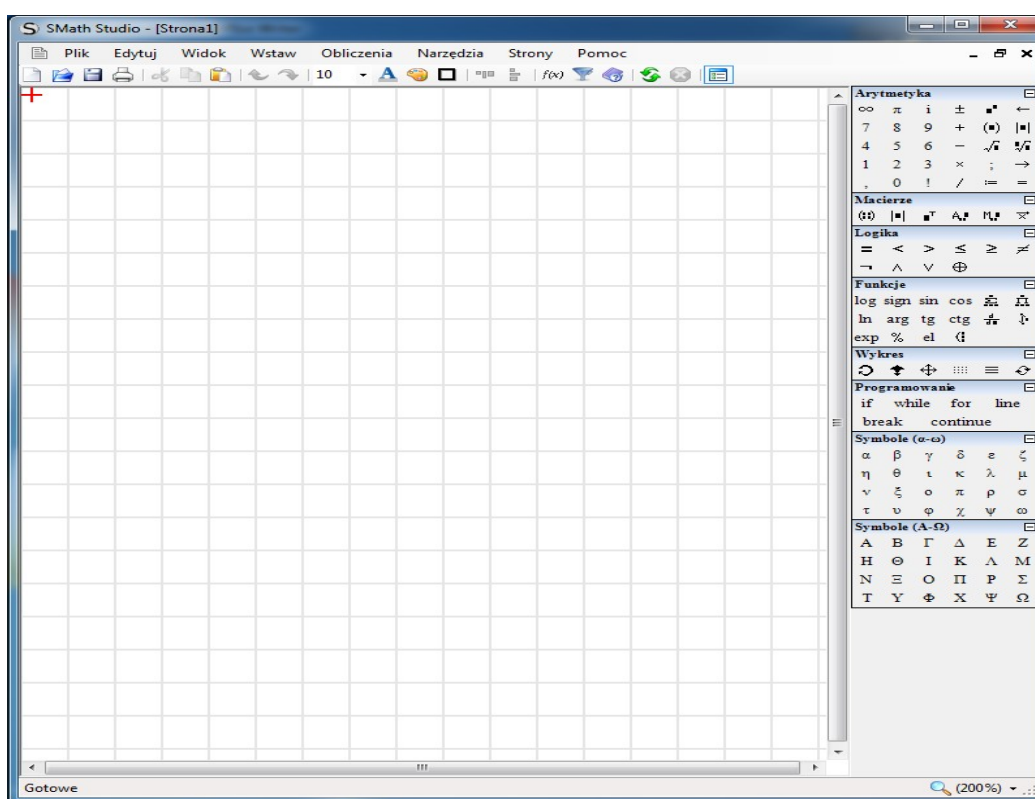
Marzec 2016 r.

1. SMATH STUDIO – ŚRODOWISKO PRACY.....	3
2. OBLICZANIE WARTOŚCI ZMIENNYCH.....	4
3. OBLICZENIA SYMBOLICZNE.....	6
4. DEFINIOWANIE ZMIENNYCH I FUNKCJI.....	7
5. OBLICZENIA NA JEDNOSTKACH.....	10
6. WSTAWIANIE I FORMATOWANIE WYKRESÓW.....	13
7. OBLICZENIA NA MACIERZACH.....	15
8. ROZWIĄZYWANIE RÓWNAŃ I UKŁADÓW RÓWNAŃ.....	16
RÓWNANIA WIELOMIANOWE.....	16
MIEJSCA ZEROWE DOWOLNEJ FUNKCJI.....	17
ROZWIĄZANIE UKŁADU RÓWNAŃ LINIOWYCH.....	18
ROZWIĄZANIE RÓWNAŃ NIELINIOWYCH I UKŁADÓW RÓWNAŃ NIELINIOWYCH.....	20
9. PROGRAMOWANIE.....	21
INSTRUKCJA WARUNKOWA <i>IF</i>	21
PĘTLA ITERACYJNA <i>FOR</i>	24
PĘTLA WARUNKOWA <i>WHILE</i>	32

1. SMath Studio – środowisko pracy

SMath Studio jest darmowym programem wspomagającym obliczenia matematyczne. Jest dobrą alternatywą dla komercyjnego oprogramowania firmy MathSoft znanego pod nazwą Mathcad.

Okno dialogowe programu odzwierciedla kratkowaną kartę z zeszytu formatu A4 (Rys. 1). Wyrażenia zapisane w SMath Studio są zdecydowanie czytelniejsze od tych samych wyrażeń zapisanych w arkuszu kalkulacyjnym. Podobnie jak w arkuszu kalkulacyjnym zmiana danych wejściowych, powoduje automatyczne przeliczenie dalszych części arkusza obliczeniowego, wykorzystujących powyższe dane i wyświetlenie aktualnego wyniku (musi być włączona opcja z menu: **Obliczenia** → **Autoobliczenia**). Program pozwala wykonywać obliczenia zarówno numeryczne jak i symboliczne (niestety w przypadku tych drugich program ma bardzo ograniczone możliwości), wyniki możemy przedstawiać w formie wielorakich wykresów. Dokument programu SMath Studio pozwala także na tworzenie opisów (dla wyrażeń i zmiennych) dzięki czemu program jeszcze bardziej przypomina zwykłą kartkę papieru.



Rys. 1. Okno programu SMath Studio

Z prawej strony okna (Rys. 1) widoczne są paski narzędzi służące do wprowadzania różnego rodzaju znaków: arytmetycznych, macierzy, znaków logicznych, funkcji, wykresów, programowania oraz symboli greckich.

Wszystkie funkcje można odszukać w *Menu głównym* lub na jednym z pasku narzędziowym, dodatkowo większość najczęściej wykorzystywanych funkcji można wywołać za pomocą skrótów klawiszowych.

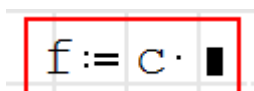
Wszystkie elementy wprowadzane do dokumentu nazywać będziemy **regionami**. Każdy z regionów zajmuje pewien minimalny dla niego obszar dokumentu. W zależności od charakteru wprowadzonego elementu rozróżniamy:

- regiony równań – zawierają definicję zmiennych, równania oraz wyrażenia algebraiczne,
- regiony tekstu – będące komentarzem w dokumencie,

- regiony wykresów – zawierają dwu- i trójwymiarowe wykresy,
- regiony graficzne – zawierają rysunki różnych formatów.

Każdy z regionów w dokumencie można swobodnie przesuwać, kopiować lub kasować w celu uzyskania pożądanej postaci dokumentu. Edycję regionu najwygodniej dokonujemy myszką. Naciskając LPM spowodujemy pojawienie się w tym obszarze pionowej kreski, która określa **punkt wstawienia**. Wewnątrz regionu możemy poruszać się za pomocą klawiszy nawigacji [←], [↑], [→], [↓], [Home], [End]. Klawisz spacja służy do zmiany zakresu edycji regionu.

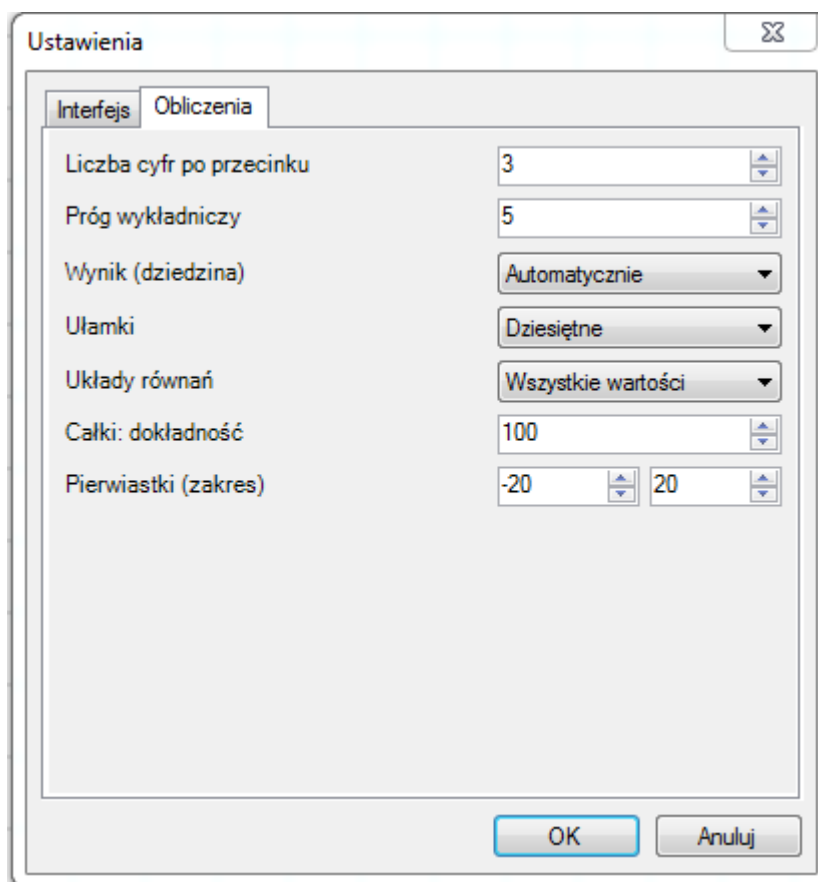
Program interpretuje wprowadzane regiony w naturalnym porządku zapisu, tj. z góry na dół. Cecha ta wymusza zdefiniowanie zmiennej użytej w równaniu powyżej regionu z równaniem. W programie we wszystkich niedokończonych definicjach zmiennych, równań czy na wykresach pojawia się **ramka braku** (Rys. 2) w postaci czerwonej ramki wokół regionu zmiennej. Aby zmienna była zdefiniowana, należy poprawnie uzupełnić wszystkie puste pola w regionie.



Rys. 2. Znacznik braku

2. Obliczanie wartości zmiennych

Przy obliczaniu wyrażeń matematycznych wpisujemy w regionie treść interesującego nas wyrażenia. Domyślnym separatorem dziesiętnym w programie SMath Studio jest przecinek. Dokładność i forma wyświetlanych wyników może być modyfikowana w *Menu Narzędzia* → *Ustawienia* widoczne na Rys. 3.



Rys. 3. Modyfikowanie wyświetlania wyników

$$\sqrt[3]{\frac{2,113 \cdot 10^4}{331 + 3^6}}$$

Rys. 4. Obliczenie wartości pierwiastka

Aby obliczyć w SMath Studio wartość pierwiastka przedstawionego na Rys. 4. należy wykonać następujące czynności:

1. Z paska narzędzi *Arytmetyka* wybrać pierwiastek n-stopnia
2. Wprowadzić z klawiatury 2,113*10^4 a następnie wcisnąć spację do momentu widocznego na Rys. 5.

$$2,113 \cdot 10^4$$

Rys. 5. Licznik przyszłego ułamka

3. Wstawić kreskę ułamkową klawiszem [/]
4. Napisać z klawiatury 331+3^6
5. LPM wskazać stopień pierwiastka i wpisać 3
6. Ostatnim krokiem jest wcisnięcie znaku równości [=]

Przy pomocy skrótów klawiszowych można znacznie przyspieszyć wprowadzanie równań. Poniżej zebrane zostały podstawowe skróty klawiszowe i operatory:

[Ctrl] + [=]	Równe w sensie logicznym	Wprowadzamy [a] następnie [shift]	Wartość bezwzględna
[Ctrl] + [9]	Mniejsze lub równe	[Ctrl] + [Shift] + [P]	Liczba Pi
[Ctrl] + [.]	Symbolicznie równa się	[Ctrl] + [3]	Różne od (lewa strona nie równa prawej)
[Ctrl] + [6]	Indeks górny	[Ctrl] + [0]	Większe lub równe
[.]	Tekstowy indeks dolny	Wpisujemy int	Całka oznaczona
[Ctrl] + [1]	Transponowanie	Wpisujemy sum a następnie [TAB]	Suma wyrażeń
[Ctrl] + [\]	n-ty pierwiastek	[\]	Iloczyn wyrażeń
[Ctrl] + [Z]	nieskończoność	[^]	Do potęgi


Ćwiczenie 2.1: Obliczyć wartości następujących wyrażeń i porównać ich wyniki.

1.	$\sqrt[5]{\frac{123^4 + 3!}{10 \cdot \pi + 4^6}}$	8,888
2.	$\int_0^{2 \cdot \pi} \frac{2 \cdot x \cdot \sin(x) }{4 \cdot \pi + x^2} dx$	0,928
3.	$\ln\left(\sqrt{\frac{76}{23+5^4}} + 100\right) = 4,609$	4,609
4.	$\sum_{n=1}^5 \left(\frac{5 \cdot n + 5^n}{(n+1)!} \right)$	26,417

3. Obliczenia symboliczne

W omawianym programie można wykonywać obliczenia symboliczne tj. przekształcenia i obliczenia na wzorach bez podstawiania wartości numerycznych. Można obliczać pochodne oraz różnego typu równania. Żeby otrzymać rozwiązanie, wynik obliczenia należy zastosować symbol [→] z belki *Arytmetyka* lub nacisnąć [Ctrl] + [.].

Aby obliczyć pochodną należy postępować wg poniższych punktów:

- Do obliczania pochodnej pierwszego rzędu należy wpisać polecenie *diff*, a następnie zatwierdzić klawiszem [TAB]
 - Do obliczania pochodnej pierwszego rzędu można również wybrać symbol pochodnej z belki *Funkcje* .
- Do obliczenia pochodnych wyższych rzędów należy wpisać polecenie *diff* wcisnąć strzałkę w dół wybierając funkcję *diff(3)*, a następnie zatwierdzić wybór klawiszem [TAB]
- Wpisać funkcję
- Uzupełnić różniczkę
- Wstawić symbol [→] z belki *Arytmetyka* lub alternatywnie [Ctrl] + [.]

Przykład obliczeń symbolicznych pochodnej funkcji x^3 przedstawiono na Rys. 6.

$$\frac{d^2}{dx^2} (x^3) = 6 \cdot x$$

Rys. 6. Obliczenie pochodnych

Ćwiczenie 3.1: Obliczyć wartości następujących wyrażeń i porównać wyniki.

Nr	Wyrażenie	Wynik
1.	$\frac{d}{dx} (\ln(x) \cdot x^2)$	$x \cdot (1 + 2 \cdot \ln(x))$

2.	$\frac{d}{dx} \left(\frac{2}{x^2} (\operatorname{tg}(x) + x^5) \right)$	$\frac{2 \cdot (5 \cdot x^3 \cdot \cos(x)^2 \cdot (2 \cdot \cos(x) - x \cdot \sin(x)) + \sin(x) \cdot (1 + 5 \cdot x^4 \cdot \cos(x)^2))}{\cos(x)^3}$
3.	$\frac{d}{dx} (x^5 + 2 \cdot x^4)$	$x^3 \cdot (x + 4 \cdot (2 + x))$

4. Definiowanie zmiennych i funkcji

W SMath Studio w obliczeniach szczególnie wygodne jest stosowanie zmiennych (Rys. 7). Definicja zmiennej polega na przypisaniu konkretnej wartości liczbowej lub innych zmiennych w postaci wyrażenia.

$$a := 15$$

$$b := 25$$

$$c := a + b$$

Rys. 7. Definicja zmiennych

W programie są rozróżniane wielkości liter, dlatego też zmienne o nazwach *aaa* oraz *Aaa* dla programu są dwiema różnymi zmiennymi. Bardzo wygodną formą zapisywania zmiennych jest stosowanie indeksów dolnych (Rys. 8). Zapisu indeksu dolnego dokonujemy po zastosowaniu [.] (kropki).

$$w_{\max} := 441$$

Rys. 8. Nazwa zmiennej z indeksem

Nazwy zmiennych oraz funkcji nie mogą rozpoczynać się cyfrą. W nazwach zmiennych swobodnie można stosować indeksy dolne, natomiast stosowanie indeksów górnych powinno być zarezerwowane dla wykładników potęg lub innych operatorów. Stosowanie greckich liter alfabetu umożliwia pasek o nazwie *Symbole*. Oprócz zmiennych, którym przypisana jest jedna wartość liczbową można definiować zmienne zakresowe. Zmienna zakresowa definiowana jest jako ciąg arytmetyczny. W definicji zmiennej zakresowej należy podać następujące parametry:

1. Wartość początkową ciągu
2. Skok wartości zmiennej jeżeli chcemy aby była to wartość inna niż 1 (jeżeli chcemy, aby zmienna zmieniała swoją wartość o 2, to w drugim polu musimy podać wartość o dwa większą od początkowej)
3. Wartość końcową

Na Rys. 9 przedstawiono dwa warianty definiowania zmiennej zakresowej. Zmienna definiowana jest za pomocą polecenia **range**. Do wyboru są dwie grupy tej funkcji: **range (2)** – zmienna zakresowa ze skokiem co 1, **range (3)** – zmienna zakresowa, w której wartość skoku możemy zdefiniować.

$$z := 1 \dots 4 \qquad x := 1 ; 4 \dots 10$$

$$z = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \qquad x = \begin{pmatrix} 1 \\ 4 \\ 7 \\ 10 \end{pmatrix}$$

Rys. 9. Zmienne zakresowe

SMath Studio umożliwia definiowanie funkcji dostępnych w pasku narzędzi, widniejących pod znakiem $f(x)$. Program umożliwia definicje własnych funkcji (Rys. 10), także w tym celu należy:

1. Wpisać nazwę funkcji
2. Określić argument (gdy funkcja ma więcej argumentów rozdzielamy ją znakiem [;])
3. Wpisać znak przypisania [Shift] + [=]
4. Wpisać równanie funkcyjne.

$$f_1(x) := \ln(x^3 + 3) \qquad g_1(x ; y) := \ln(x^2 + y)$$

Rys. 10. Definicja funkcji

Aby obliczyć wartość funkcji dla konkretnych wartości argumentów należy powyżej definicji podać wartości tych argumentów. Wartości te podajemy jako zmienne o nazwach argumentów. Można też wstawić bezpośrednio wartości jako argumenty do funkcji (Rys. 11).

$$x := 2 \qquad y := 3$$

$$g_1(x ; y) := \ln(x^2 + y)$$

$$g_1(x ; y) = 1,946 \qquad g_1(5 ; 4) = 3,367$$

Rys. 11. Wartości funkcji g_1

Wartość funkcji można również wyznaczyć dla zmiennych zakresowych (Rys. 12).

$$\begin{array}{l}
 r := 1 \dots 10 \quad W(r) := 2 \cdot \pi \cdot r \\
 r = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{pmatrix} \quad W(r) = \begin{pmatrix} 6,283 \\ 12,566 \\ 18,85 \\ 25,133 \\ 31,416 \\ 37,699 \\ 43,982 \\ 50,265 \\ 56,549 \\ 62,832 \end{pmatrix}
 \end{array}$$

Rys. 12. Wartość funkcji dla zmiennej zakresowej

Ćwiczenie 4.1 Zdefiniować zmienne i obliczyć wartość wyrażenia:

$$\text{alfa} := 12 \quad \beta_1 := \sqrt{3} \quad w_{\min} := 65 \quad \frac{\text{alfa} \cdot \beta_1}{w_{\min}}$$

Ćwiczenie 4.2 Zdefiniować zmienne, obliczyć wartości wyrażeń oraz wyświetlić i porównać wyniki:

$$\begin{array}{l}
 z_{\text{dop}} := 1 \dots 6 \quad \varphi_2 := 11 ; 22 \dots 66 \\
 z_{\text{dop}} \cdot \varphi_2 \quad \varphi_2 \cdot z_{\text{dop}} \\
 \varphi_2 + z_{\text{dop}} \quad z_{\text{dop}} + \varphi_2
 \end{array}$$

5. Obliczenia na jednostkach

Program umożliwia nadawanie wartościom niemianowanym sensu fizycznego przez dodanie im standardowych jednostek. Wyniki działań na zmiennych z jednostkami podawany jest przez program w postaci wartości z jednostką. Jednostkę wyniku możemy ustalić dowolnie. Możemy także stosować własne, zdefiniowane wcześniej jednostki.

Do obliczeń warto stosować nazwy zmiennych, które są ogólnie przyjęte przy danym zagadnieniu. Definicja zmiennej M będącej celem obliczeń przedstawiona została na Rys. 13.

```
a := 10 m
P := 1000 N
M := P · a
M = 10000 J
```

Rys. 13. Definicja zmiennych mianowanych, wynik mianowany

Wyświetlenie wyniku obliczeń, czyli wartości obliczonej zmiennej M , został przedstawiony w podstawowej jednostce (dla przyjętego systemu jednostek) – w naszym przypadku są to **[J = Dżule]** – podstawowa dla systemu **SI** jednostka energii. Czarne pole za literą **[J]** służy do wprowadzenia innej jednostki – wbudowanej lub zdefiniowanej przez użytkownika. Po wciśnięciu znaku równości obliczenia zostaną przeprowadzone automatycznie. Aby zmienić jednostkę należy kliknąć w czarnym polu i z klawiatury wpisać N , zatwierdzić klawiszem **[TAB]** następnie wprowadzić $*$, wybrać z klawiatury m a następnie zatwierdzić klawiszem **[TAB]**. Ostatnim krokiem jest zatwierdzenie całości przyciskiem **[ENTER]** (Rys. 14). Dla nazw jednostek można stosować przedrostki jako mnożniki zmniejszające (litery: d , c , i i inne) lub zwiększające (litery: h , k , M i inne). Na Rys. 15 przedstawiono przykład zastosowania przedrostka zwiększającego k równego 1000.

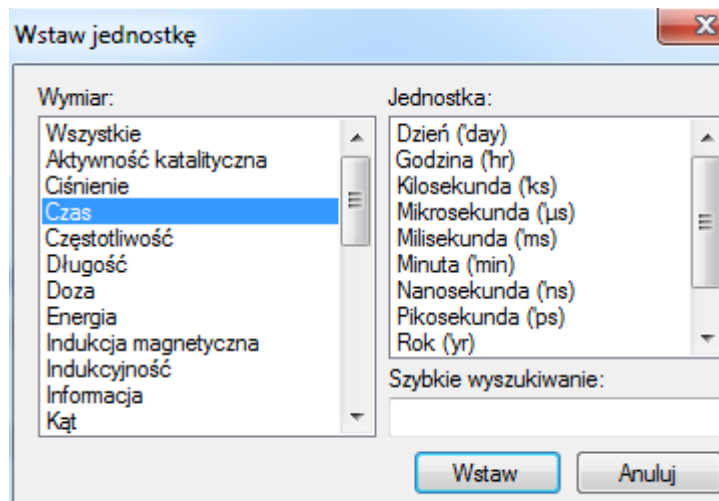
```
M = 10000 N m
```

Rys. 14. Wynik ze zmienioną jednostką

```
M = 10 kN m
```

Rys. 15. Wynik z jednostką poprzedzoną przedrostkiem

Jeżeli użytkownikowi nie jest znana jednostka wbudowana, to może on posłużyć się poleceniem Wstaw→Jednostki. Doskonałym przykładem może być jednostka czasu: **[godzina]**, którą w programie SMath Studio zapisuje się jako **[hr]** (Rys. 16).



Rys. 16. Okno służące do wyboru wbudowanych jednostek

Przy korzystaniu z jednostek należy pamiętać o stosowaniu jednostek zmiennych we wzorach, zwłaszcza przy operacjach addytywnych. Program w przypadku niezgodności informuje nas o błędzie przy próbie wyświetlenia wartości zmiennej (Rys. 17).

```

a := 10 m
P := 1000 N
M := P + a
M = ■

```

Jednostki nie pasują.

Rys. 17. Niezgodność jednostek sygnalizowana przez program

Przy wprowadzaniu argumentów funkcji trygonometrycznych program interpretuje wpisane przez nas wartości liczbowe jako podane w mierze radialnej. Wartości dla miar stopniowych uzyskujemy podając miarę kąta z jednostką stopnie (z ang. *deg*), (Rys. 18).

$$\text{tg}(45 \text{ deg})$$

Rys. 18. Jednostki w funkcjach trygonometrycznych

ĆWICZENIE 5.1: Zdefiniować własną jednostkę jak na przykład [MPa].

ĆWICZENIE 5.2: Przeliczyć prędkość auta wynoszącą 90km/h na jednostki używane w USA czyli mila/h – zastosować wbudowane jednostki.

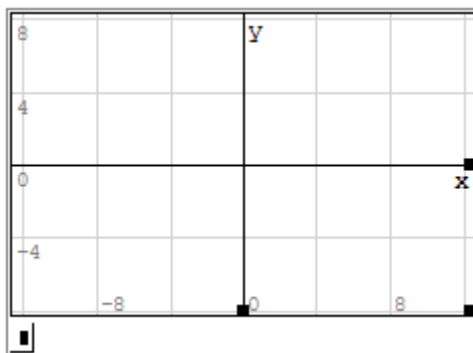
ĆWICZENIE 5.3: Obliczyć moment zginający w środku rozpiętości belki wolno podpartej obciążonej ciężarem własnym z wykorzystaniem jednostek.

ĆWICZENIE 5.4: Obliczyć energię potencjalną ciała sztywnego o konkretnej masie na znanej wysokości.

ĆWICZENIE 5.5: Obliczyć siłę wyporu na ciało zanurzone w cieczy.

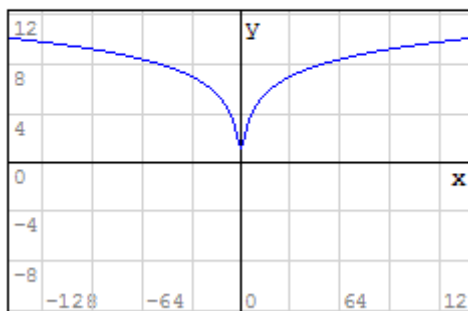
6. Wstawianie i formatowanie wykresów

Oprócz obliczeń numerycznych i symbolicznych program umożliwia użytkownikowi graficzną prezentację wyników w postaci wykresów. W programie można wykorzystać dwa rodzaje wykresów – 2D i 3D. Przed uruchomieniem kreatora wykresów poleceniem z paska narzędzi, (*Wstaw*→*Wykres*) należy zdefiniować funkcje, której wykres chcemy obejrzeć. Dobrze jest też zdefiniować zakres argumentów funkcji chociaż nie jest to konieczne. Po wstawieniu obszaru wykresu należy w aktywnym polu wpisać nazwę funkcji (Rys. 19).



`x:=-100 ; -90 .. 100`


`F(x):=ln(x2+3)`



`F(x)`

Rys. 19. Tworzenie wykresu funkcji

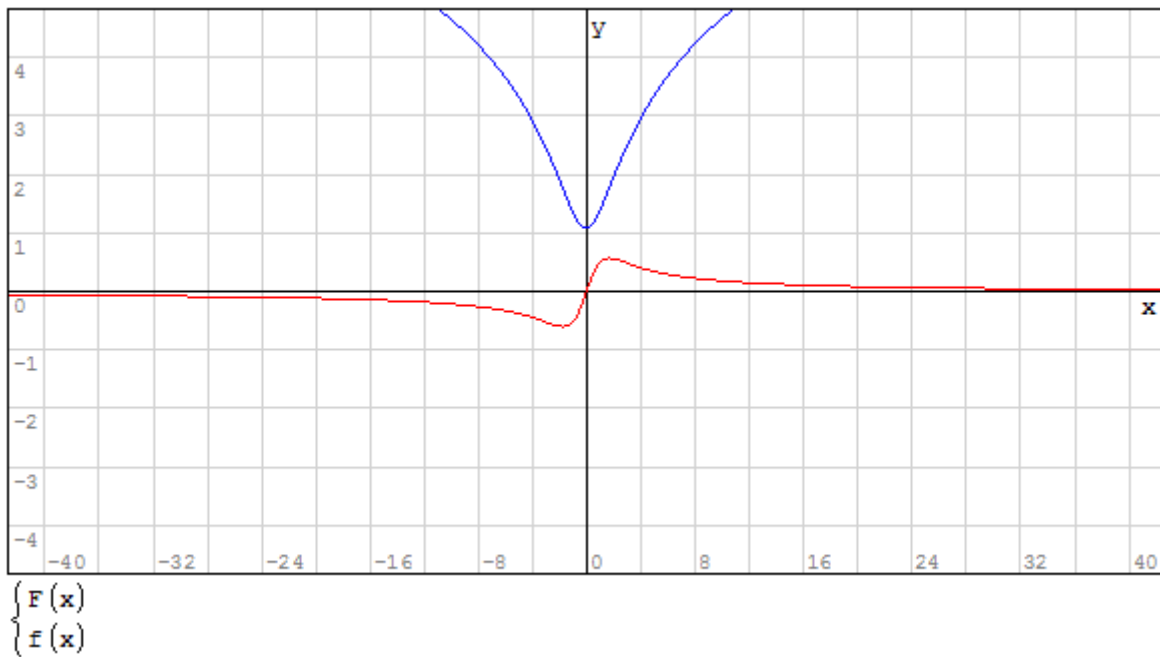
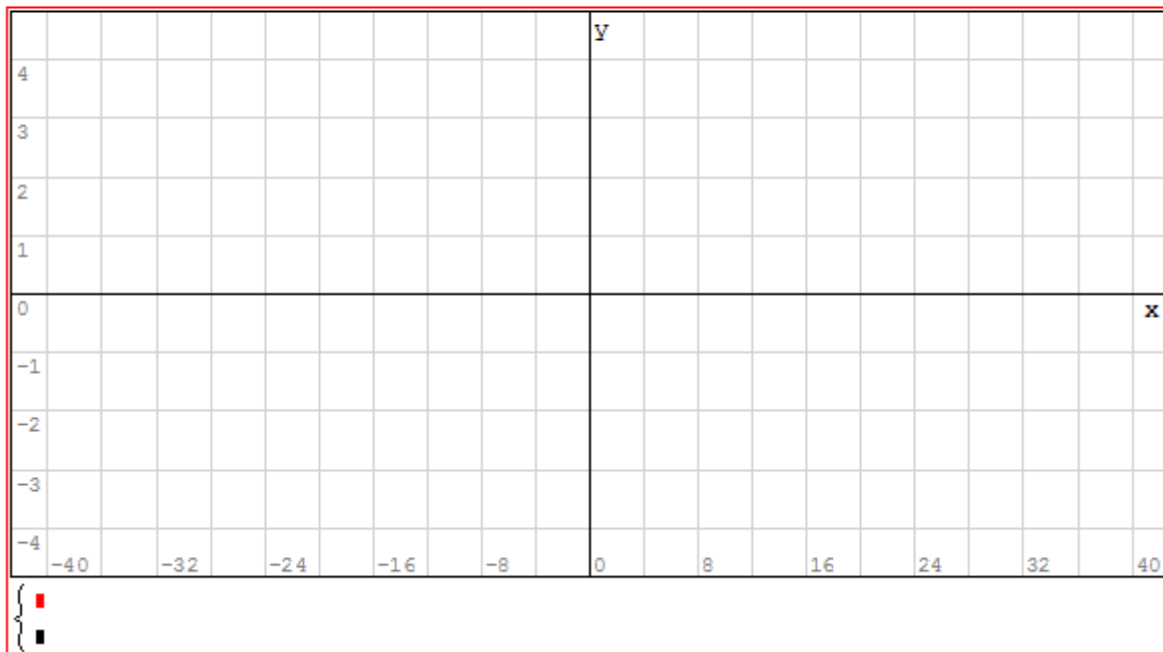
W przypadku programu SMath Studio zmiana zakresu osi jest możliwa za pomocą klawisza [Ctrl] lub [Shift] z pomocą [scroll'a].

Na jednym wykresie można przedstawić przebieg kilku różnych funkcji w tym samym zakresie odciętych (Rys. 20). W takim przypadku w aktywnym polu należy z belki *Funkcje* wybrać ikonę , a następnie w puste pola wpisać nazwy wcześniej zdefiniowanych funkcji.

x := -100 ; -90 .. 100

$$F(x) := \ln(x^2 + 3)$$


$$f(x) := \frac{d}{dx} \ln(x^2 + 3)$$



Rys. 20. Wykres dwóch funkcji

Ćwiczenie 6.1: Narysować wykres funkcji $F(x)$ i jej pochodnej oraz sformatować jego wygląd tak, tak jak pokazano to na Rys. 20.

7. Obliczenia na macierzach

Wprowadzanie wektorów i macierzy można wykonać z menu *Wstaw*→*Macierz*, kombinacją klawiszy [Ctrl] + [M] albo wybrać ikonę  z belki *Macierze*. Po wykonaniu jednej z powyższych opcji należy określić rozmiar macierzy (domyślnie 3x3) a następnie wprowadzić elementy macierzy (Rys. 21).

$$K := \begin{pmatrix} 5 & 20 & -30 \\ 120 & 15 & 30 \\ 1 & 3 & 4 \end{pmatrix} \quad L := \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

Rys. 21. Przykłady macierzy

Wartości poszczególnych elementów macierzy możemy uzyskać wykorzystując indeksację elementów macierzy Rys. 22. Indeks macierzowy wprowadza się przy pomocy funkcji *el(2)*, *el(3)*. W pierwszej funkcji, w pierwszym polu wprowadzamy nazwę macierzy natomiast w drugim nr elementu, który chcemy wywołać (numeracja elementów rozpoczyna się od lewego górnego rogu i rośnie aż do prawego dolnego rogu – od lewej do prawej, od góry do dołu). Funkcja *el(2)* umożliwia wyświetlenie żądanego elementu przez wprowadzenie nr wiersza i kolumny, w których znajduje się dany argument.

$$K := \begin{pmatrix} 5 & 20 & -30 \\ 120 & 15 & 30 \\ 1 & 3 & 4 \end{pmatrix} \quad L := \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 9 & 4 & 10 \end{pmatrix}$$
$$K_1 = 5 \quad L_7 = 4$$
$$K_5 = 15 \quad L_8 = 10$$
$$K_{12} = 20 \quad L_{21} = 5$$
$$K_{22} = 15 \quad L_{22} = 9$$

Rys. 22. Elementy macierzy

Program umożliwia wykonywanie działań na macierzach, oczywiście zgodnych z zasadami rachunku macierzowego. Oprócz podstawowych działań dostępnych w pasku *Macierze* można wykorzystać kilka dodatkowych funkcji:

- *col(K;n)* – wyświetli n – tą kolumnę macierzy K,
- *cols(K)* – zwraca liczbę kolumn macierzy/wektora,
- *identity(n)* – zwraca macierz jednostkową (n x n – jedynki na przekątnej, zera poza),
- *length(K)* – zwraca liczbę elementów w macierzy, zwraca skalar,
- *matrix(x;y)* – zwraca zerową macierz o podanym rozmiarze – x – liczba wierszy, y – liczba kolumn,
- *minor(K;i;j)* – zwraca dopełnienie elementu [i;j] macierzy,
- *rank(K)* – zwraca rząd macierzy,
- *rows(K)* – zwraca liczbę wierszy macierzy,
- *submatrix(K;i;j;k;n)* – zwraca macierz K od i-tego do j-tego wiersza i od k-tej do n-tej kolumny,
- *minor(K;i;j)* – zwraca podmacierz danej macierzy, powstałą z usunięcia podanego wiersza i podanej kolumny.

Ćwiczenie 7.1: Zdefiniuj macierze i wykonaj na nich działania.

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{pmatrix} \quad B := \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$

$$A^{-1}, A^T, A^{-1} \cdot A, A^{-1} \cdot B, A \cdot B$$

8. Rozwiązywanie równań i układów równań

SMath Studio umożliwia między innymi rozwiązywanie równań oraz układów równań.

Równania wielomianowe

Funkcja `polyroots()` poszukuje pierwiastków, tj. miejsc zerowych wielomianu zdefiniowanego w następujący sposób:

$$c_0 \cdot x^0 + c_1 \cdot x^1 + c_2 \cdot x^2 + \dots + c_n \cdot x^n.$$

Jako argument funkcji musimy podać wektor ze współczynnikami c_i wielomianu (Rys. 23).

$$V := \begin{pmatrix} 5 \\ -4 \\ -3 \\ 2 \end{pmatrix}$$
$$\text{polyroots} \left(\begin{pmatrix} 5 \\ -4 \\ -3 \\ 2 \end{pmatrix} \right) = \begin{pmatrix} 1 \\ -1,351 \\ 1,851 \end{pmatrix} \quad \text{polyroots}(V) = \begin{pmatrix} 1 \\ -1,351 \\ 1,851 \end{pmatrix}$$

Rys. 23. Pierwiastki wielomianu

Ćwiczenie 8.1 Znajdź pierwiastki wielomianów:

$$2 \cdot x^4 + 2 \cdot x^3 + 20 \cdot x^2 + 53 \cdot x - 60$$

$$5 \cdot d^6 + 5 \cdot d^5 + 4 \cdot d^4 + 3 \cdot d^3 + 2 \cdot d^2 + d + 1$$

Miejsca zerowe dowolnej funkcji

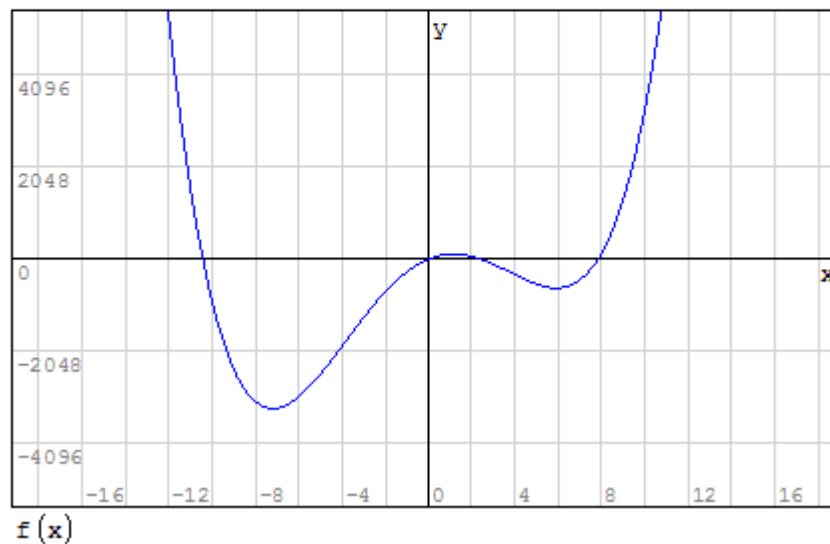
Polecenie `solve()` pozwala w sposób numeryczny znaleźć pierwiastki rzeczywiste zadanej funkcji (Rys. 24). W ten sposób można wyznaczyć miejsca zerowe dowolnej funkcji, w odróżnieniu od polecenia `polyroots()` rozwiązującego tylko równanie wielomianowe.

Argumentami polecenia `solve(2)` są nazwa funkcji i nazwa zmiennej, dla których nastąpi poszukiwanie rozwiązania. W bardziej rozbudowanej wersji `solve(4)` są cztery argumenty, dodatkowo początek i koniec zakresu poszukiwania rozwiązania.

Często, dobrze jest wygenerować dodatkowo wykres analizowanej funkcji, w celu określenia liczby miejsc zerowych w danym zakresie.

$$f(x) := x^4 - 87 \cdot x^2 + 202 \cdot x + \sin(x)$$

$$\text{solve}(f(x); x) = \begin{pmatrix} -10,323 \\ 0 \\ 2,505 \\ 7,82 \end{pmatrix}$$



Rys. 24. Zastosowanie funkcji `solve()`

Ćwiczenie 8.2: Znajdź wszystkie miejsca zerowe funkcji:

$$z(x) := x^3 - e^x + 200$$

$$f(x) := -x^4 + 20 \cdot x^3 + 300 \cdot x^2 + 20 \cdot x + 15$$

Rozwiązanie układu równań liniowych

Do rozwiązania układów równań liniowych należy stosować metodę macierzową. Wymaga to zdefiniowania macierzy współczynników przy niewiadomych oraz wektora (macierzy jednokolumnowej) wyrazów wolnych, a następnie wykonania operacji na tych macierzach.

Na Rys. 25 przedstawiono układ równań liniowych zapisany z użyciem operatora logicznego `=` (CTRL+=). Dodatkowo, zastosowanie polecenia `sys()` umożliwia zapisanie równań spiętych klamrą, co przedstawia Rys. 26.

$$\begin{cases} 5 \cdot x + 6 \cdot y - 4 \cdot z = 20 \\ 3 \cdot x + 2 \cdot z = 12 \\ x - y + z = 0 \end{cases}$$

Rys. 25: Układ równań liniowych

$$\left\{ \begin{array}{l} 5 \cdot x + 6 \cdot y - 4 \cdot z = 20 \\ 3 \cdot x + 2 \cdot z = 12 \\ x - y + z = 0 \end{array} \right.$$

Rys. 26: Układ równań liniowych zapisany poprzez polecenie `sys()`

Zapisanie układu równań nie jest konieczne z punktu widzenia działania programu i dalszych czynności mających doprowadzić do ustalenia wartości niewiadomych. Oznacza to, że niezależnie od tego w jakiej formie zostanie zapisany (lub nie zapisany) układ równań liniowych konieczne jest zdefiniowanie odpowiednich macierzy. W przykładzie z Rys. 27 zdefiniowano zmienną `WN` jako macierz współczynników przy niewiadomych oraz zmienną `WW` jako wektor (macierz jednokolumnową) wyrazów wolnych.

$$\text{WN} := \begin{pmatrix} 5 & 6 & -4 \\ 3 & 0 & 2 \\ 1 & -1 & 1 \end{pmatrix} \quad \text{WW} := \begin{pmatrix} 20 \\ 12 \\ 0 \end{pmatrix}$$

Rys. 27: Macierze definiujące układ równań liniowych

Rozwiązanie, zmienna `R`, uzyskuje się obliczając macierz odwrotną do macierzy współczynników i mnożąc ją przez wektor wyrazów wolnych, co pokazuje Rys. 28.

$$\text{R} := \text{WN}^{-1} \cdot \text{WW}$$
$$\text{R} = \begin{pmatrix} 1 \\ 5,5 \\ 4,5 \end{pmatrix}$$

Rys. 28: Rozwiązanie układu równań liniowych

Macierz odwrotną można również obliczyć używając polecenia `invert()`, patrz Rys. 29.

$$WN^{-1} = \begin{pmatrix} 0,125 & -0,125 & 0,75 \\ -0,0625 & 0,5625 & -1,375 \\ -0,1875 & 0,6875 & -1,125 \end{pmatrix}$$

$$\text{invert}(WN) = \begin{pmatrix} 0,125 & -0,125 & 0,75 \\ -0,0625 & 0,5625 & -1,375 \\ -0,1875 & 0,6875 & -1,125 \end{pmatrix}$$

Rys. 29: Sposoby obliczania macierzy odwrotnej

Niekiedy zachodzi konieczność użycia w dalszych obliczeniach wartości wyznaczonych niewiadomych. Można na podstawie uzyskanego rozwiązania przechowywanego w zmiennej **R** zdefiniować zmienne **x**, **y** i **z** poprzez użycie polecenia *el(2)*. Kolejne etapy takiego definiowania pokazano na Rys. 30.

$x := R_1$	$x = 1$
$y := R_2$	$y = 5,5$
$z := R_3$	$z = 4,5$

Rys. 30: Deklaracja zmiennych na podstawie rozwiązania **R**

Ćwiczenie 8.3: Rozwiąż układy równań liniowych:

$$\begin{cases} 12x + 20y - 5z = 37 \\ -5x - y + 20z = 53 \\ 3x + 7y - 2z = 11 \end{cases}$$

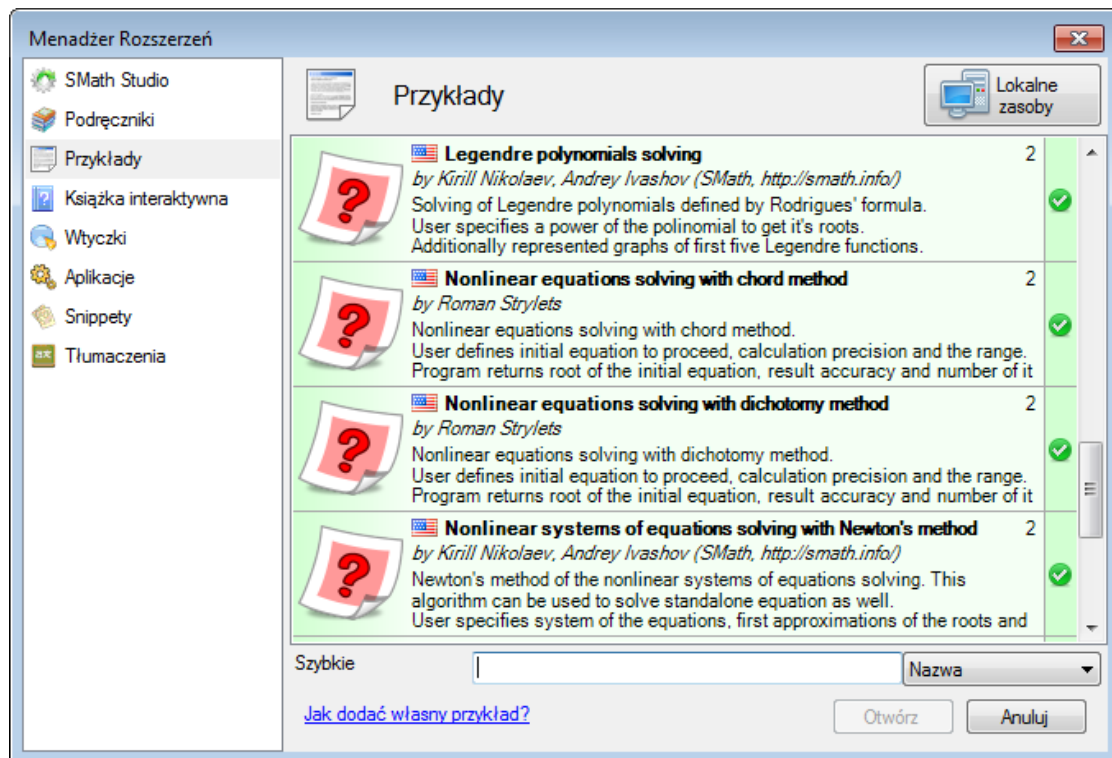
$$\begin{cases} 5N_1 + 6N_2 - 2N_3 = 65 \\ 6N_1 + N_2 = 36 \\ -2N_1 + 8N_2 + 3N_3 = 32 \end{cases}$$

$$\begin{cases} a + b - 4c - 20 = 0 \\ 16a + 10b - 36z = 0 \\ 6a + b + c = 0 \end{cases}$$

Rozwiązanie równań nieliniowych i układów równań nieliniowych

Rozwiązanie układów równań nieliniowych jest możliwe poprzez użycie wielu procedur numerycznych. W przykładach dostępnych we wbudowanej pomocy programu SMath Studio zawarto trzy z takich metod – patrz Rys. 31. W bardzo łatwy sposób można zaadaptować do swoich potrzeb te przykłady poprzez zmianę definicji funkcji $f(x)$.

Realizacja odpowiednich procedur numerycznych wymaga użycie funkcji programistycznych wbudowanych do programu SMath Studio.



Rys. 31: Przykładowe metody rozwiązywania układów równań nieliniowych

9. Programowanie

Instrukcja warunkowa *if*

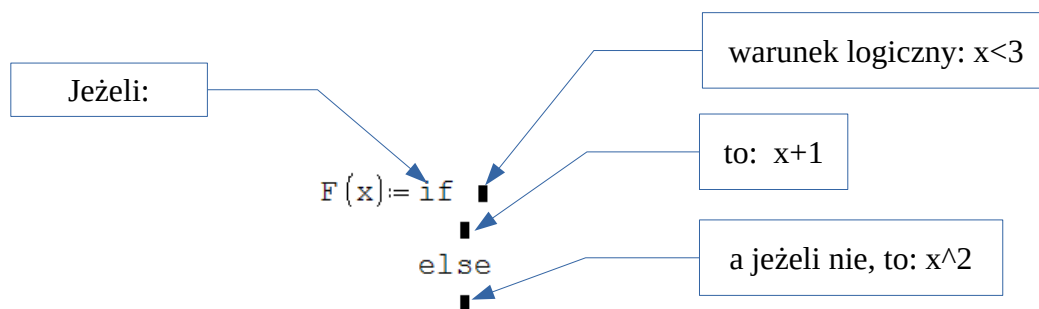
Niektóre funkcje określone są w sposób przedziałowy, tj. w zależności od tego w jakim zakresie znajduje się ich argument (Rys. 32). Wartości takiej funkcji można obliczyć wykorzystując instrukcję warunkową „jeżeli” (polecenie *if*).

Budowa i działanie instrukcji warunkowej jest analogiczne jak w arkuszu kalkulacyjnym.

Na Rys. 33 przedstawiono definicję funkcji $F(x)$ za pomocą polecenia *if* w programie SMath Studio.

$$F(x) := \begin{cases} x + 1 & x < 3 \\ x^2 & x \geq 3 \end{cases}$$

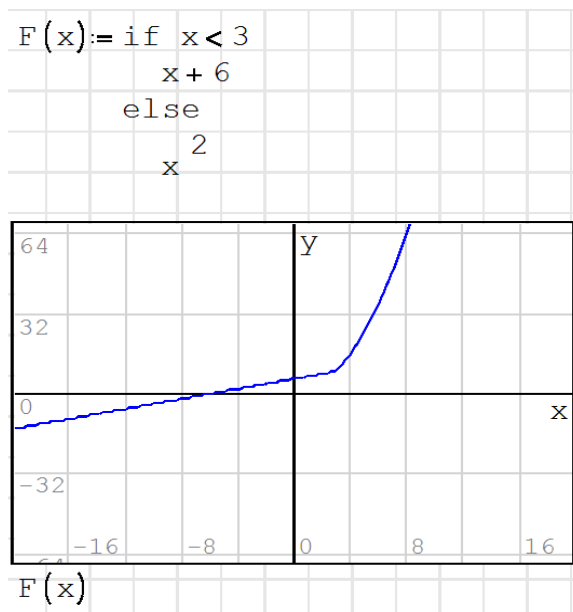
Rys. 32. Przykład funkcji o zmiennej postaci



Rys. 33. Zastosowanie polecenia *if* do definicji funkcji

Przykład 9a

Na Rys. 34 przedstawiono schemat działania polecenia *if* – jeżeli argument funkcji $F(x)$ jest mniejszy od 3 to wartości funkcji obliczane będą zgodnie z formułą $x+1$, jeżeli argument funkcji będzie większy bądź równy 3 to funkcja przyjmować będzie wartości x^2 .



Rys. 34. Zastosowanie funkcji „if” wraz z wykresem wartości funkcji $F(x)$

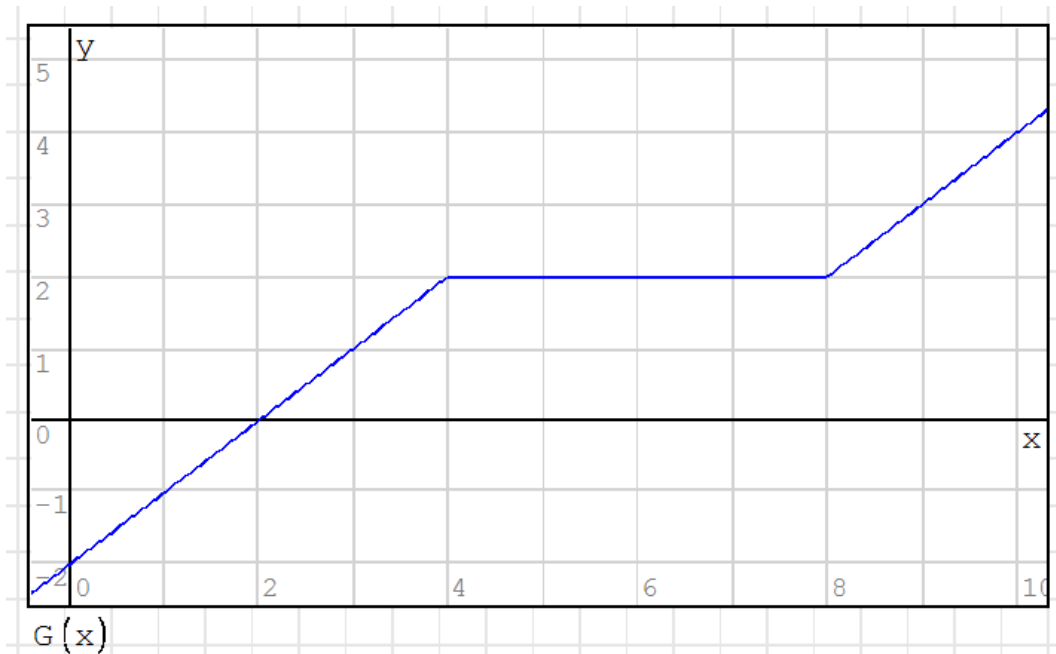
Przykład 9b

W przypadku wielu przedziałów zmienności funkcji definicję funkcji można zrealizować poprzez użycie polecenia warunkowego *if* jako argumentu innego polecenia warunkowego *if*. Przykład realizacji dla funkcji trzy-przedziałowej $G(x)$ przedstawia Rys. 35.

$$G(x) = \begin{cases} x-2 & \text{dla } x < 4 \\ 2 & \text{dla } 4 \leq x \leq 8 \\ x-6 & \text{dla } x > 8 \end{cases}$$

```
G(x) := if x < 4
      x - 2
      else
        if | x ≥ 4
           | x ≤ 8
           2
        else
          x - 6
```

Linia bloku instrukcji z
Menu Programowanie
(line)



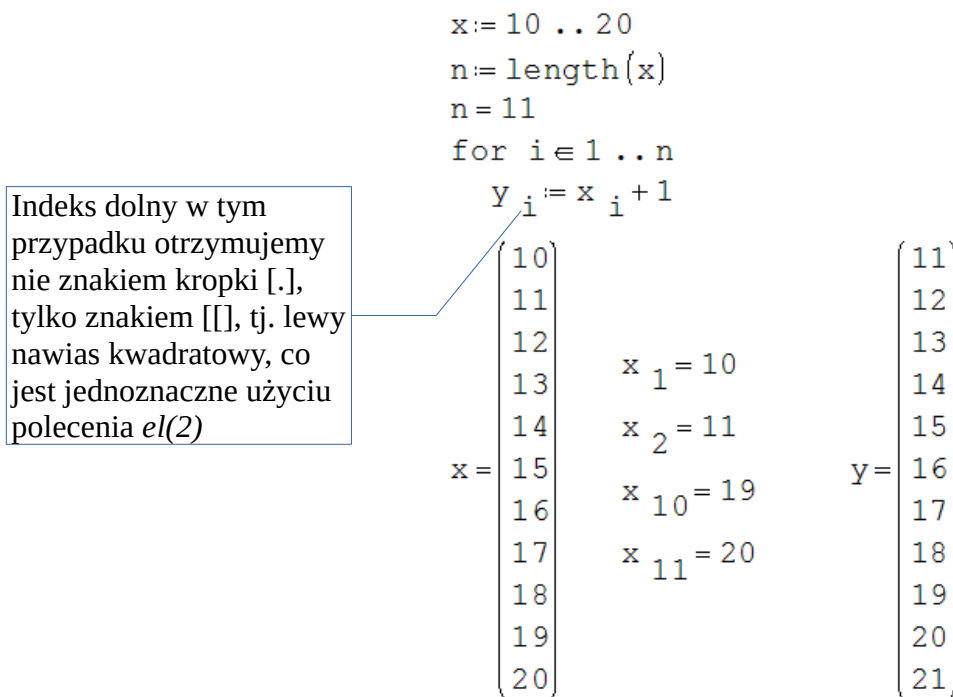
Rys. 35. Funkcja *if* wewnątrz funkcji *if* wraz z wykresem wartości funkcji $G(x)$

Pętla iteracyjna *for*

Użycie pętli iteracyjnej *for* (określanej również mianem pętli zakresowej lub pętli licznikowej) pozwala na wykonanie tych samych operacji ustaloną liczbę razy. Do działania pętli licznikowej *for* konieczna jest zmienna sterująca określająca ile razy polecenia ujęte w pętli zostaną wykonane.

Na przykładzie widocznym na Rys. 36 zmienna sterująca to zmienna *i*. Zmienna sterująca *i* została zdefiniowana jako zmienną zakresową $1 \dots n$, gdzie zmienna *n* określa liczebność zbioru argumentów przechowywanych w zmiennej *x*. Wyznaczenie liczby elementów danej zmiennej jest możliwe dzięki użyciu polecenia *length()*.

W omawianym przykładzie zmienna *x* mieści się w zakresie $\langle 10, 20 \rangle$, w który wchodzi 11 elementów, stąd $n = 11$. Po wprowadzeniu polecenia *for* musimy podać zakres, w którym przeprowadzamy obliczenia, w naszym przypadku $i \in 1 \dots n$. Oznacza to iż pętla obliczy wartości funkcji *y* dla argumentów od x_1 do x_n . Jak widzimy $x_1 = 10$, $x_2 = 11$, itd. Gdy $i = 5$ to pętla obliczy wartość funkcji *y* dla piątego elementu z zakresu zdefiniowanych argumentów *x*.



Rys. 36. Zastosowanie pętli „for”

Przykład 9c

```
x := 1 .. 10
n := length(x)
for i ∈ 1 .. n
  yi 1 := if xi ≤ 5
           xi
         else
           xi + 100
```

$$y = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 106 \\ 107 \\ 108 \\ 109 \\ 110 \end{pmatrix}$$

```
z := 1 .. 10
for j ∈ 1 .. length(z)
  qj := if zj ≤ 5
         zj
       else
         zj + 100
```

$$q = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 106 \\ 107 \\ 108 \\ 109 \\ 110 \end{pmatrix}$$

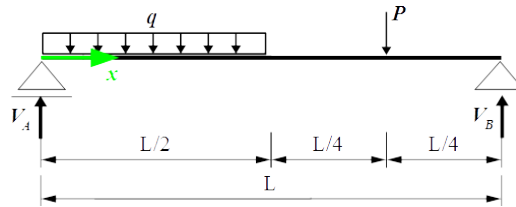
Przykład 9d

Samochód jedzie z prędkością 300km/h i nagle zaczyna hamować z przyspieszeniem równym 5m/s^2 . Prędkość samochodu mierzono od chwili rozpoczęcia hamowania co 4s przez 28s. Podaj prędkość samochodu w 0s, 4s, 8s, 12s i 16s hamowania, kiedy prędkość zacznie przyjmować wartość ujemną przy pomocy odpowiedniej funkcji wstaw wartość 0.

```
t := 0 s ; 4 s .. 28 s      v0 := 300  $\frac{\text{km}}{\text{hr}}$ 
t =  $\begin{pmatrix} 0 \\ 4 \text{ s} \\ 8 \text{ s} \\ 12 \text{ s} \\ 16 \text{ s} \\ 20 \text{ s} \\ 24 \text{ s} \\ 28 \text{ s} \end{pmatrix}$       a := 5  $\frac{\text{m}}{\text{s}^2}$ 
n := length(t)
n = 8
for i ∈ 1 .. n
  vi := if v0 - a · ti ≥ 0
          v0 - a · ti
        else
          0
v =  $\begin{pmatrix} 83,333 \frac{\text{m}}{\text{s}} \\ 63,333 \frac{\text{m}}{\text{s}} \\ 43,333 \frac{\text{m}}{\text{s}} \\ 23,333 \frac{\text{m}}{\text{s}} \\ 3,333 \frac{\text{m}}{\text{s}} \\ 0 \\ 0 \\ 0 \end{pmatrix}$ 
```

Przykład 9e

Wyznacz zmienność (oblicz wartości funkcji i następnie narysuj wykresy) momentu zginającego i siły poprzecznej w belce obciążonej jak na Rys. 37.



Rys. 37: Rozważana belka

Wartości reakcji obliczamy wg następujących wzorów:

$$V_B = -V_A + q \cdot \frac{L}{2} + P \quad V_A = \frac{q \cdot \frac{L}{2} \cdot \frac{3}{4} \cdot L + P \cdot \frac{L}{4}}{L}$$

DANE :

$$L := 20 \text{ m}$$

$$q := 2 \frac{\text{kN}}{\text{m}}$$

$$P := 4 \text{ kN}$$

ROZWIĄZANIE :

OBLICZENIE REAKCJI

$$V_A := \frac{\left(q \cdot \frac{L}{2} \cdot \frac{3}{4} \cdot L + P \cdot \frac{L}{4} \right)}{L}$$

$$V_A = 16 \text{ kN}$$

$$V_B := -V_A + q \cdot \frac{L}{2} + P$$

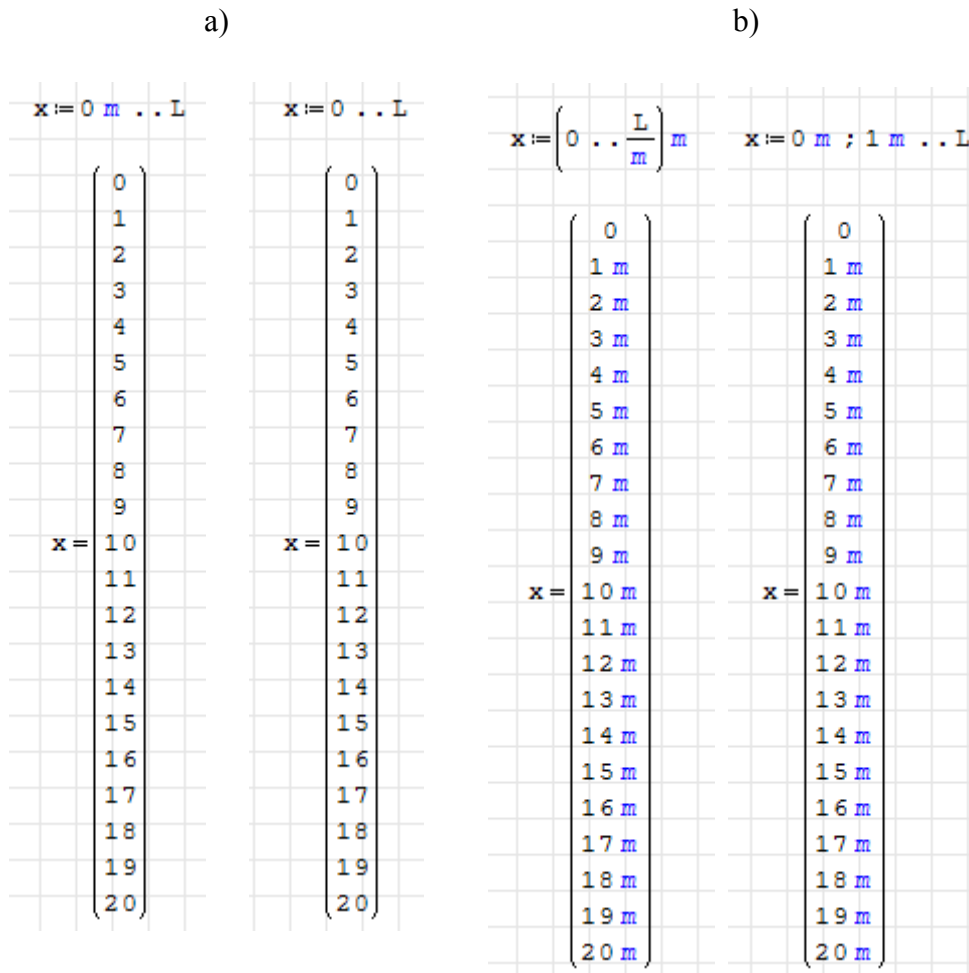
$$V_B = 8 \text{ kN}$$

Funkcja momentu zginającego dla rozważanej belki jest następującą funkcją przedziałową:

$$M_y(x) = \begin{cases} 0 \leq x \leq \frac{L}{2} & V_A \cdot x - \frac{q \cdot x^2}{2} \\ \frac{L}{2} \leq x \leq \frac{3L}{4} & V_A \cdot x - \frac{q \cdot L}{2} \cdot \left(x - \frac{L}{4}\right) \\ \frac{3L}{4} \leq x \leq L & V_A \cdot x - \frac{q \cdot L}{2} \cdot \left(x - \frac{L}{4}\right) - P \cdot \left(x - \frac{3L}{4}\right) \end{cases}$$

stąd zachodzi konieczność zdefiniowania zmiennej x jako zmiennej zakresowej $\langle 0 \ L \rangle$ oraz konieczność zastosowania wielopoziomowej funkcji *if* jako argumentu pętli licznikowej *for*.

Przy definiowaniu zmiennej zakresowej poleceniem *range(2)* przypisane zmiennej L jednostki sprawiają kłopot – zmienna zakresowa x „nie dziedziczy” jednostek po zmiennej L . Na Rys. 38a) pokazano dwa nieudane warianty definicji zmiennej x . Z kolei na Rys. 38b) pokazano dwa prawidłowe warianty definicji zmiennej x . Użycie polecenia *range(3)* umożliwi wygenerowanie zmiennej x z dowolnym przyrostem – w drugim przykładzie z Rys. 38b) zastosowano przyrost jednostkowy by uzyskać ten sam efekt co w trzech poprzednich przykładach.



Rys. 38: Nieprawidłowe a) i prawidłowe b) sposoby definiowania zmiennej zakresowej z jednostką

ZASADNICZE OBLICZENIA

```
x := (0 .. L/m) m
```

Określenie zbioru punktów,
dla których obliczone zostaną wartości funkcji M_y i Q_z

```
n := length(x)
```

Ustalenie liczebności zbioru argumentów

```
n = 21
```

Obliczenia wartości zmiennej M_y w trzech przedziałach zmienności zmiennej x

```
for k ∈ 1..n
```

```
  Myk := if xk ≤ 1/2 · L
```

$$V_A \cdot x_k - \frac{q \cdot (x_k)^2}{2}$$

```
  else
```

```
    if xk > 1/2 · L
```

```
      xk ≤ 3/4 · L
```

$$V_A \cdot x_k - q \cdot \frac{L}{2} \cdot \left(x_k - \frac{L}{4} \right)$$

```
    else
```

```
      if xk > 3/4 · L
```

```
        xk ≤ L
```

$$V_A \cdot x_k - q \cdot \frac{L}{2} \cdot \left(x_k - \frac{L}{4} \right) - P \cdot \left(x_k - \frac{3}{4} \cdot L \right)$$

```
    else
```

```
      error("Błąd danych")
```

WYNIKI :

$x =$	$\begin{pmatrix} 0 \\ 1\text{ m} \\ 2\text{ m} \\ 3\text{ m} \\ 4\text{ m} \\ 5\text{ m} \\ 6\text{ m} \\ 7\text{ m} \\ 8\text{ m} \\ 9\text{ m} \\ 10\text{ m} \\ 11\text{ m} \\ 12\text{ m} \\ 13\text{ m} \\ 14\text{ m} \\ 15\text{ m} \\ 16\text{ m} \\ 17\text{ m} \\ 18\text{ m} \\ 19\text{ m} \\ 20\text{ m} \end{pmatrix}$	$M_y =$	$\begin{pmatrix} 0 \\ 15 \\ 28 \\ 39 \\ 48 \\ 55 \\ 60 \\ 63 \\ 64 \\ 63 \\ 60 \\ 56 \\ 52 \\ 48 \\ 44 \\ 40 \\ 32 \\ 24 \\ 16 \\ 8 \\ 0 \end{pmatrix} \text{ kNm}$
-------	--	---------	---

Zdefiniowanie zmiennej umożliwiającej narysowanie wykresu

$$MW := \text{augment} \left(\frac{x}{m} ; \frac{M_y}{\text{kNm}} \right)$$



MW

ZADANIE KONTROLNE: zrobić obliczenia i narysować wykres $Q_z(x)$

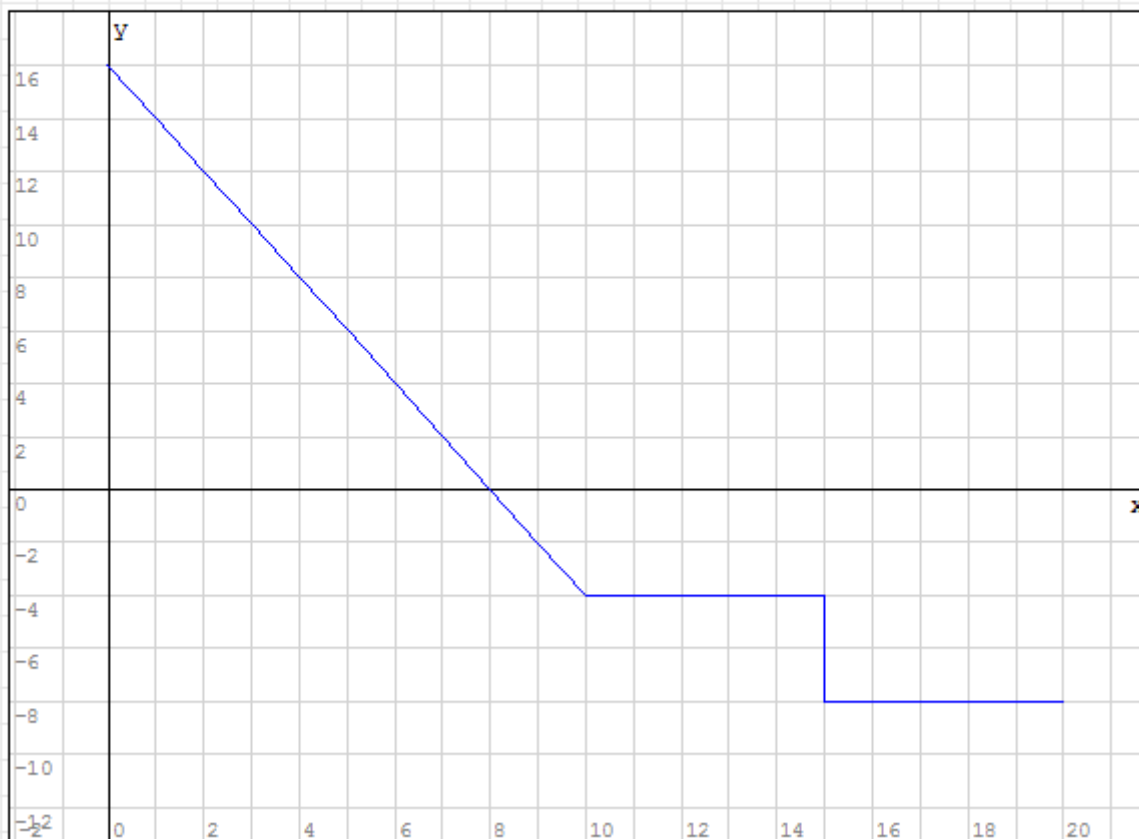
Funkcja siły poprzecznej jest następującą funkcją przedziałową:

$$Q_z(x) = \begin{cases} 0 \leq x \leq \frac{L}{2} & V_A - q \cdot x \\ \frac{L}{2} \leq x \leq \frac{3L}{4} & V_A - q \cdot \frac{L}{2} \\ \frac{3L}{4} \leq x \leq L & V_A - q \cdot \frac{L}{2} - P \end{cases}$$

Wykres zmienności funkcji siły poprzecznej dla rozważanej belki przedstawia się następująco:

Zdefiniowanie zmiennej umożliwiającej narysowanie wykresu

QW:= augment $\left(\frac{x}{m}; \frac{Q_z}{kN} \right)$



Pętla warunkowa *while*

Użycie pętli warunkowej *while* (określanej również mianem pętli repetycyjnej) pozwala na wykonanie tych dopóki określony warunek jest spełniony. Warunek zawarty w definiowanej pętli *while* musi być wyrażeniem logicznym zwracającym albo wartość **1** utożsamianą z **prawdą** albo wartość **0** jako **falsz**. Wartość wyrażenia logicznego jest ustalona na początku pętli. Jeśli wyrażenie logiczne przyjmuje wartość **1**, to instrukcje zawarte w pętli wykonywane są po raz kolejny.

Przykładem dobrze ilustrującym działanie pętli warunkowej *while* jest obliczanie wartości silni liczby naturalnej **n** (**n!**).

Źródło: <https://pl.wikipedia.org/wiki/Silnia>

- Silnia liczby naturalnej n – iloczyn wszystkich liczb naturalnych nie większych niż n
- Oznaczenie $n!$ dla silni wprowadził w 1808 roku Christian Kramp.
- Zapis $n!$, $2!$ itd. odczytujemy „ n silnia”, „ 2 silnia” itd.

Definicja rekurencyjna silni ma postać:

$$n! = \begin{cases} 1 & \text{dla } n = 0 \\ n \cdot (n - 1)! & \text{dla } n \geq 1 \end{cases}$$

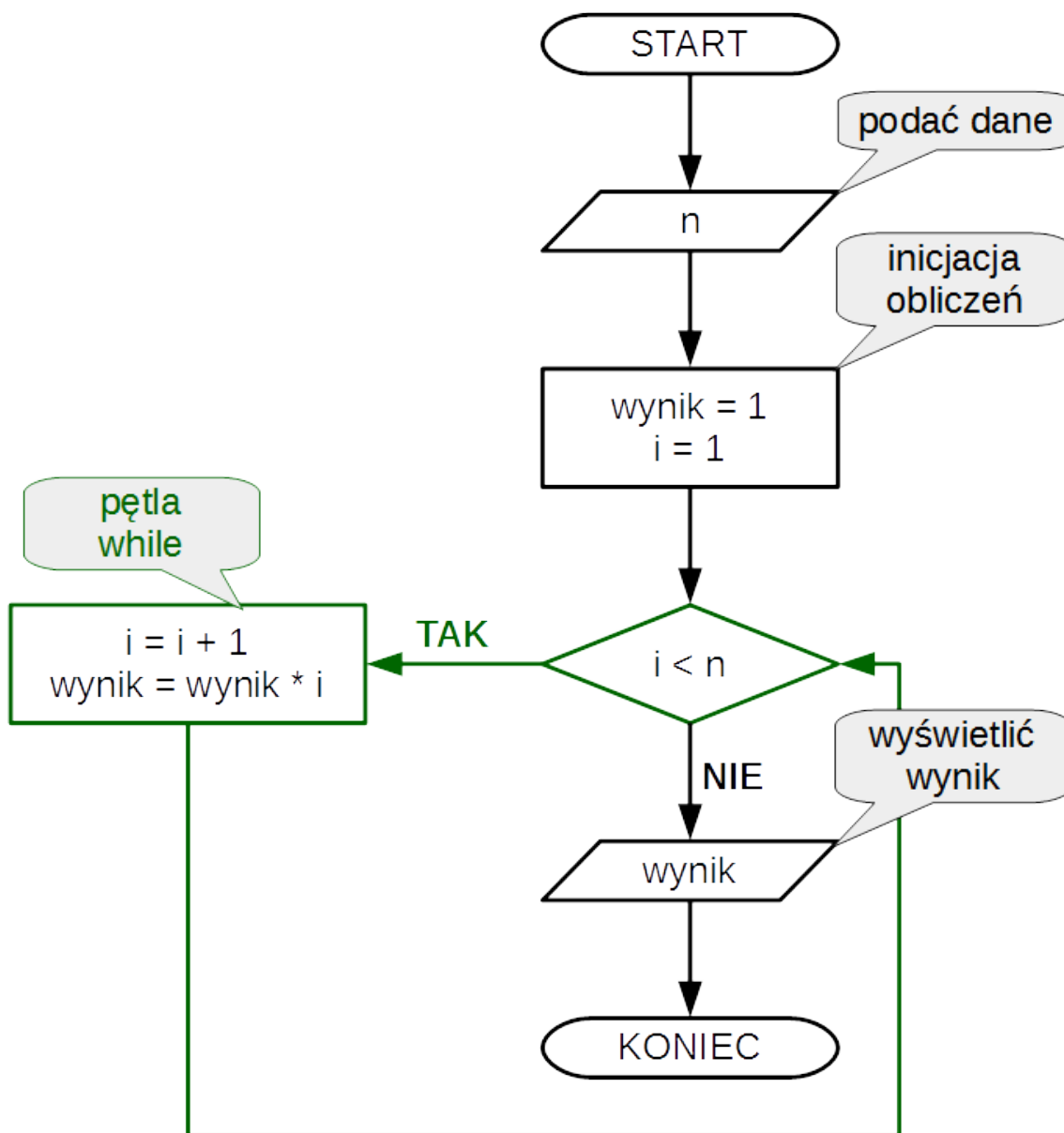
Przykłady:

$$4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$$

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

$$6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$$

Schemat blokowy przedstawiający algorytm obliczania **n silnia** przedstawiono na Rys. 39. Ten i wiele innych algorytmów można znaleźć na stronie: <http://www.algorytm.org/algorytmy-arytmetyczne/silnia.html>.



Rys. 39: Schemat blokowy algorytmu obliczania $n!$

Realizację algorytmu przedstawionego na Rys. 39 w programie SMath Studio przedstawia Rys. 40.

```
n:=5   wprowadź liczbę naturalną
i:=1   inicjacja zmiennej sterującej
ns:=1  wartość początkowa silnia
while i<n
  i:=i+1
  ns:=ns·i
ns=120  wyświetlenie wyniku obliczeń
```

Rys. 40: Obliczanie wartości ***n* silnia** z wykorzystaniem pętli *while*

Oczywiście program SMath Studio posiada wbudowany operator ! (wykrzyknik) umożliwiający obliczanie ***n* silnia**, co widać na Rys. 41.

```
n:=5
ns:=n!
ns=120  wyświetlenie wyniku obliczeń
```

Rys. 41: Obliczanie wartości ***n* silnia** poprzez wykorzystanie operatora !

Przykład 9f

Pętli *while* ma zastosowanie w wielu algorytmach rozwiązania równania nieliniowego. Realizację jednego algorytmu nazywanego **metodą siecznych** (również **metodą Newtona-Raphsona**) przedstawiono na Rys. 42. Jest to jeden z wielu przykładów dostępnych w programie SMath Studio: menu **Pomoc** → **Przykłady**.

```
Nonlinear equations solving with chord method

f(x)=0                                     Nonlinear equation
Input data:
f(x):=x3-8                               Equations left part
ε:=10-5                                   Calculations precision
a:=-10      b:=10                          Range
Calculation:
ddf(x):=d/dx d/dx f(x)
if ddf(a)·f(a)>0
  | xn:=b
  | c:=a
else
  | xn:=a
  | c:=b
n:=0
while |f(xn)|>ε
  | xn:=xn-f(xn)·(c-xn)/(f(c)-f(xn))
  | n:=n+1
Result:
xn=2                                       Answer
f(xn)=-9,0375·10-6                       Left part value
n=109                                     Number of iterations
```

Rys. 42: Przykład zastosowania pętli *while* do rozwiązywania równania nieliniowego

Obliczenia w pętli *while* są powtarzane dopóty dopóki obliczana wartość bezwzględna funkcji $f(xn)$ dla iterowanego argumentu xn jest większa od założonego błędu – od wartości zmiennej ϵ . W przypadku idealnym rozwiązaniem jest takie xn , dla którego $f(xn)$ równa się dokładnie zero.

Widać, że dla zdefiniowanej funkcji $f(x)$ i założonego przedziału wyszukiwania potrzeba było 109 iteracji, aby osiągnąć założony poziom dokładności, tj. $|f(xn)=-9,0375 \cdot 10^{-6}| \leq \epsilon = 10^{-5}$, co oznaczało zakończenie działania pętli warunkowej *while*.

ZADANIE KONTROLNE: sprawdzić ile iteracji potrzeba dla uzyskania rozwiązania zakładając przedział wyszukiwania na $\langle 0 \ 3 \rangle$. Dodatkowo, sprawdzić co da zmiana wymaganej dokładności na 10^{-3} .