

# LabView

## 5

**Dr Piotr Sitarek**

**Katedra Fizyki Doświadczalnej, Politechnika Wroclawska**

# Temat na dziś

---

Macierze, tablice, file i/o, konwersja typów

ni.com



(część materiałów zaczerpnięta ze strony  
producenta)

RONALD W. LARSEN, *LabVIEW for Engineers*

---

# Macierze i tablice

---

## **Kolekcja danych w LabView:**

klastry (Clusters)

tablice (Arrays)

macierze (Matrices) – 2D

---

# Macierze i tablice

---

## Klaster (cluster)

- Struktura danych grupująca dane różnych typów
- Podobne do *struct* w C i *record* w Pascalu
- Elementy muszą być (wszystkie) typu „control” lub „indicator”
- **Ważna jest kolejność**

### Przykład:

Komunikacja błędów – grupuje: „error flag” typu logicznego, numeryczny - „error code” i łańcuch opisujący błąd.

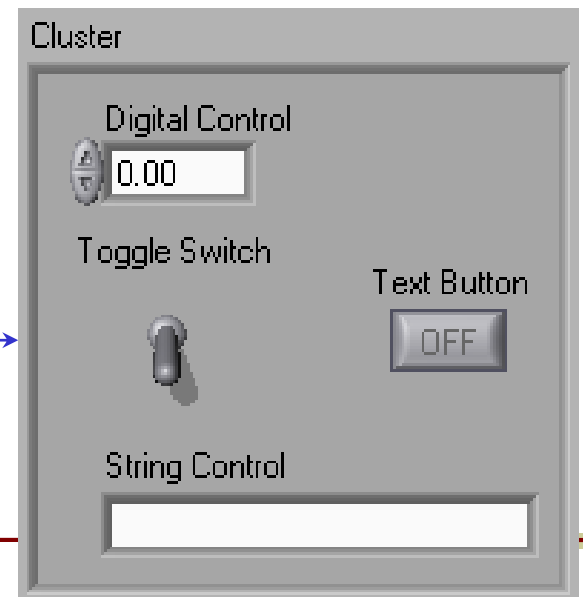
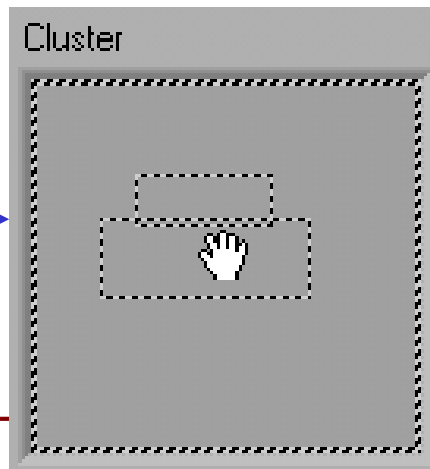
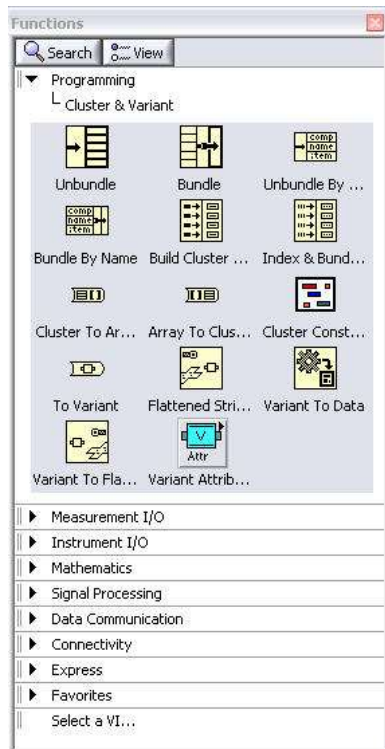
---

# Macierze i tablice

Wybierz „**Cluster**” shell.

Controls»Modern»Array, Matrix & Cluster

Wstaw obiekty do shell'a.



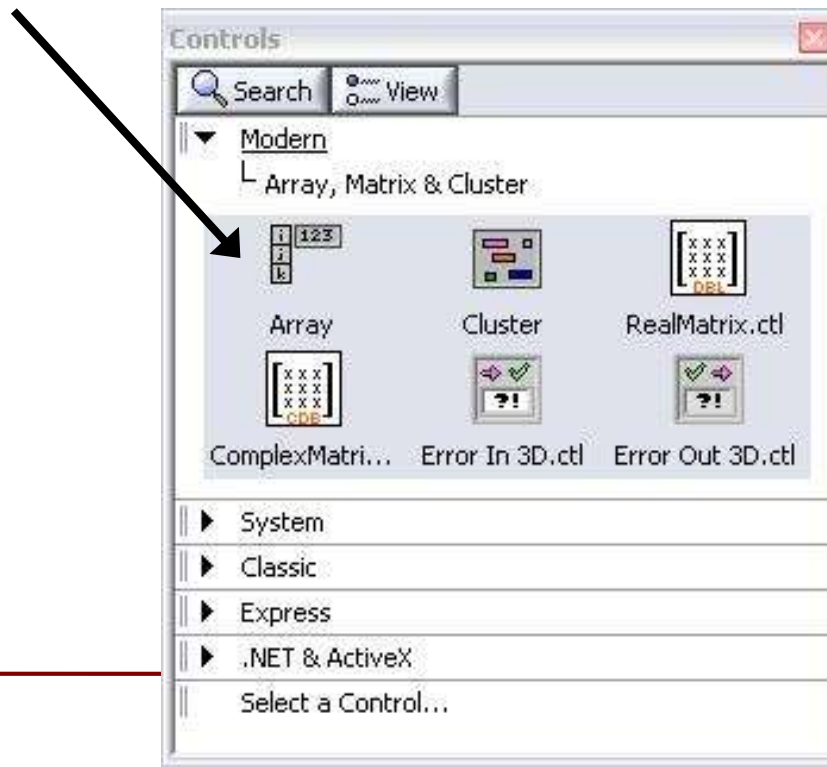
# Macierze i tablice

## Tablice:

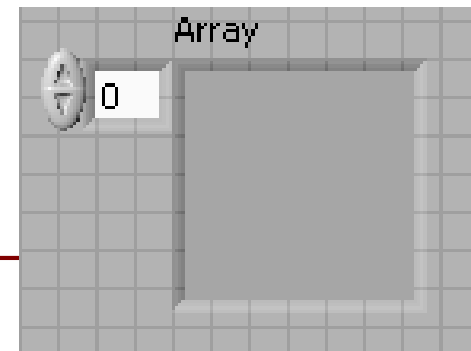
zbiór „kontrolerek” lub „indykatorów” tego samego typu

Budowanie:

**Controls»Modern»Array, Matrix, and Cluster - Array**

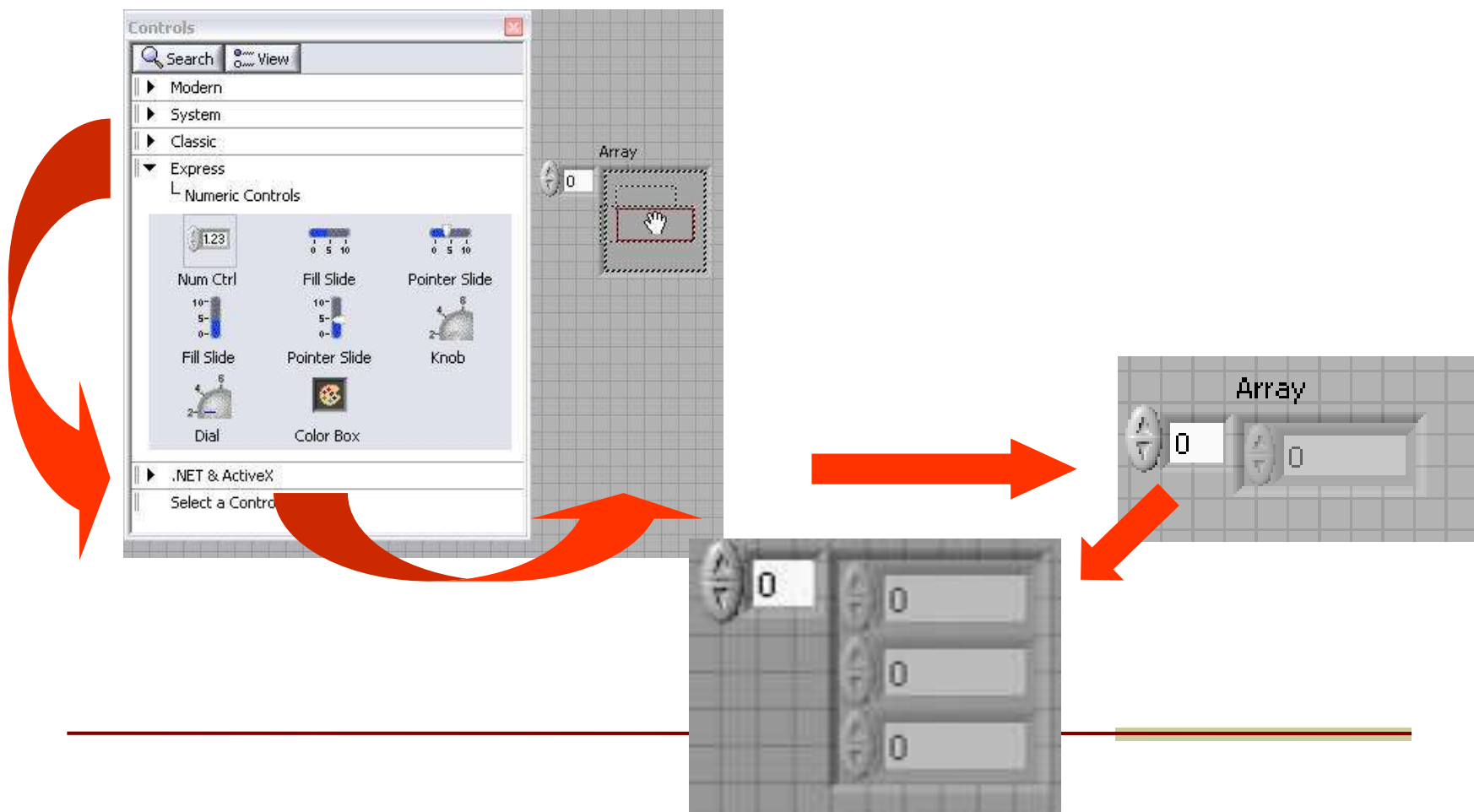


Umieść na Front Panelu



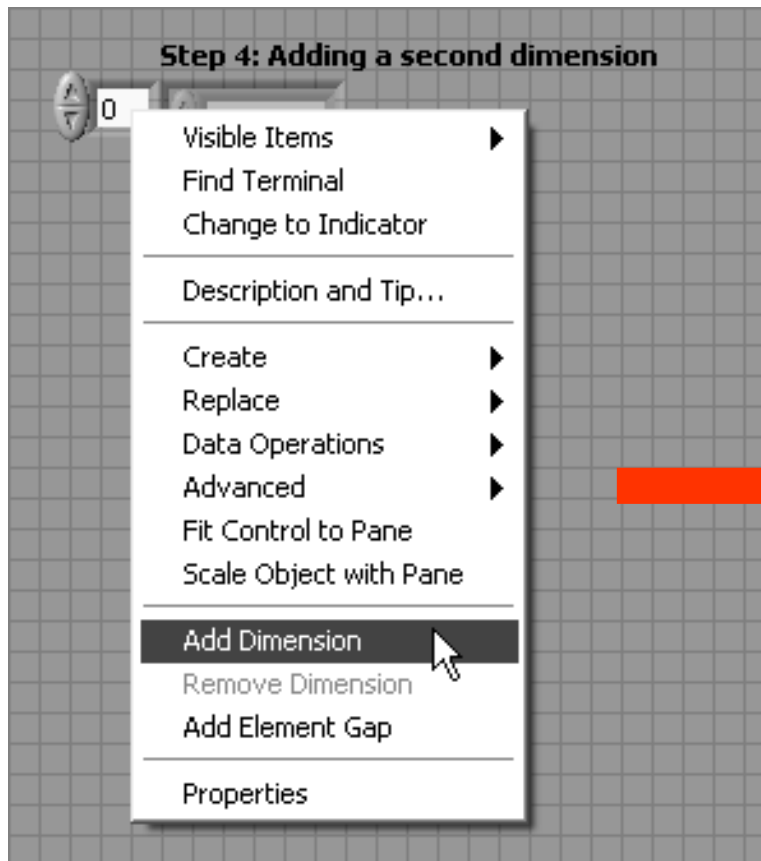
# Macierze i tablice

Wstaw typ danych (np. Numeric Control)  
Zmień (na BD) na „indicator”



# Macierze i tablice

Dostajemy tablicę 1D – 1 kolumna.  
Aby dodać wymiar:





# Macierze i tablice

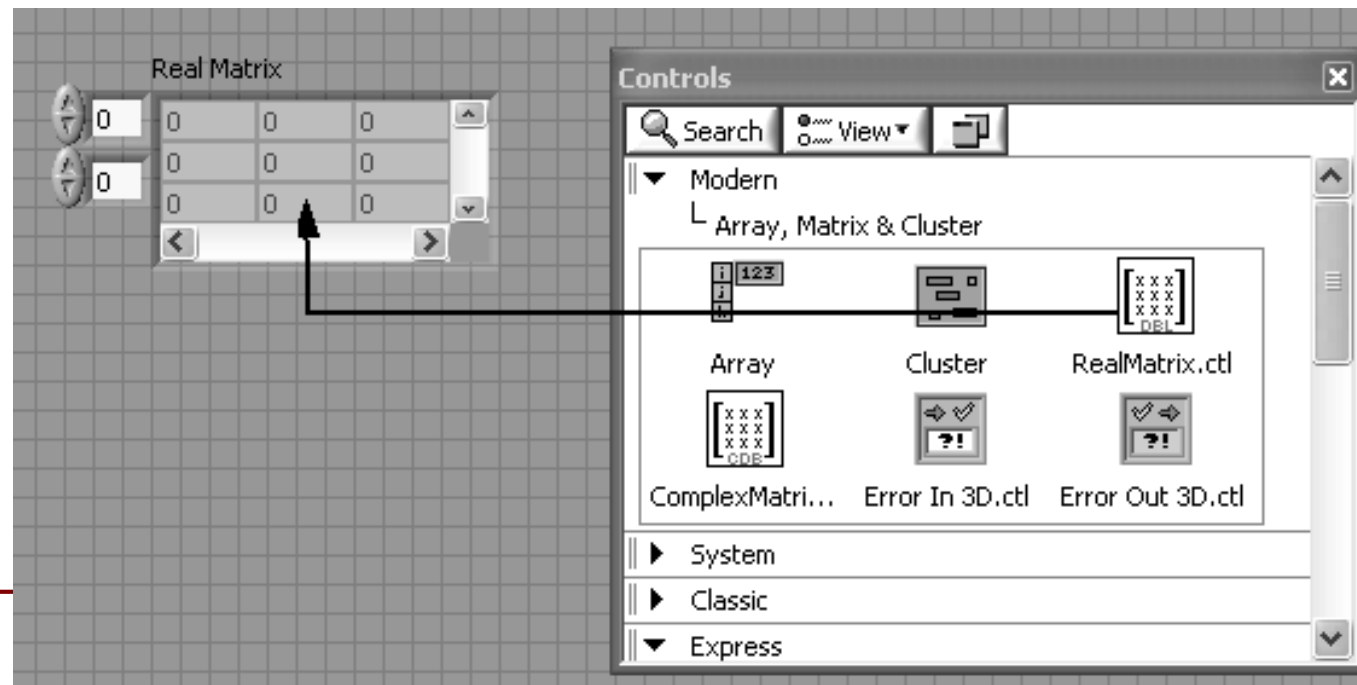
## Array functions (Functions Palette / Programming Group / Array Group):

- Array Max & Min
- Array Size
- Array Subset
- Search 1D Array
- Sort 1D Array
- Split 1D Array
- Reverse 1D Array
- Rotate 1D Array
- Delete From Array
- Insert Into Array
- Replace Array Subset
- Reshape Array
- Transpose 2D Array
- Array To Cluster
- Matrix to Array
- Array to Matrix

# Macierze i tablice

**Macierz** – pojedyncza „kontrolka” przechowująca dane jednego typu. Zwykle 2D (można używać pojedynczych wierszy i kolumn). Indeksowanie zaczyna się od 0.

**Controls Palette / Modern Group / Array, Matrix & Cluster Group / Real Matrix**



# Macierze i tablice

---

Kopiowanie fragmentu macierzy lub tablicy:

Functions Palette / Programming Group / Array Group /  
Array Subset

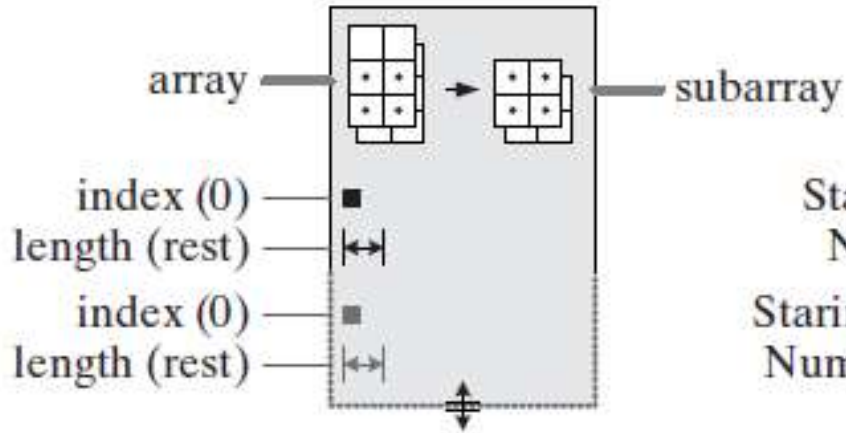
lub gdy potrzebujemy wiersza lub kolumny

Functions Palette / Programming Group / Array Group /  
Index Array

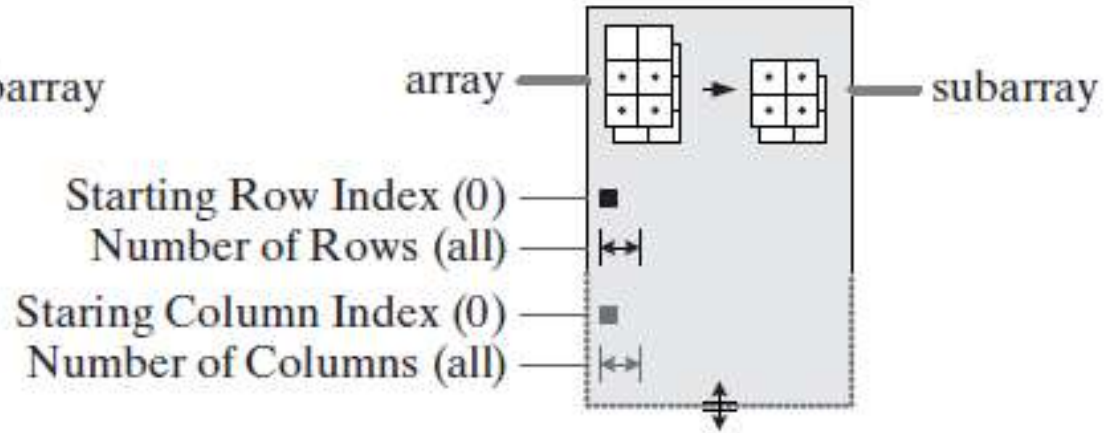
---

# Macierze i tablice

Functions Palette / Programming Group / Array Group /  
Array Subset



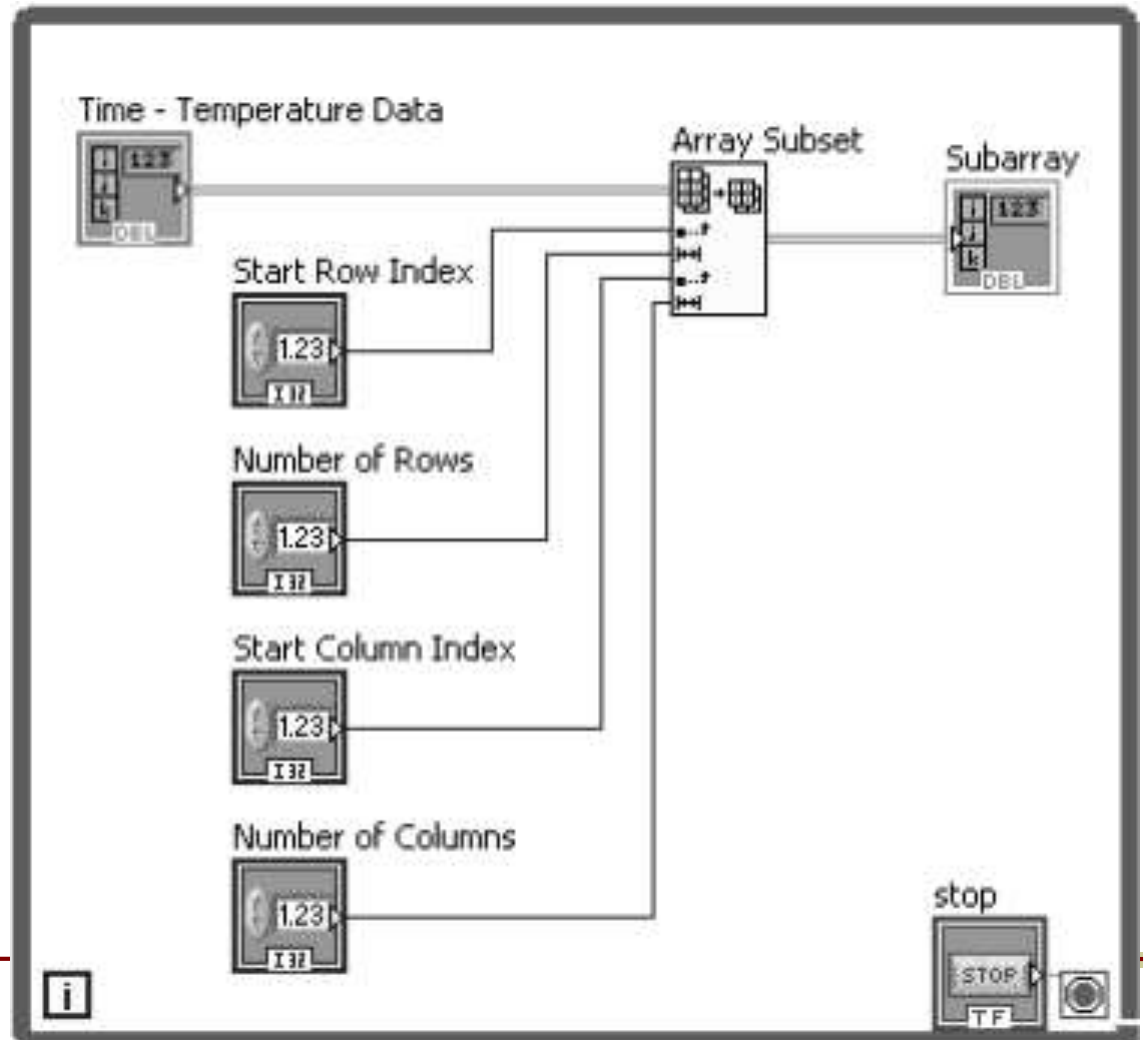
LabVIEW Nomenclature



Descriptive Nomenclature

# Macierze i tablice

Functions Palette / Programming Group / Array Group /  
Array Subset



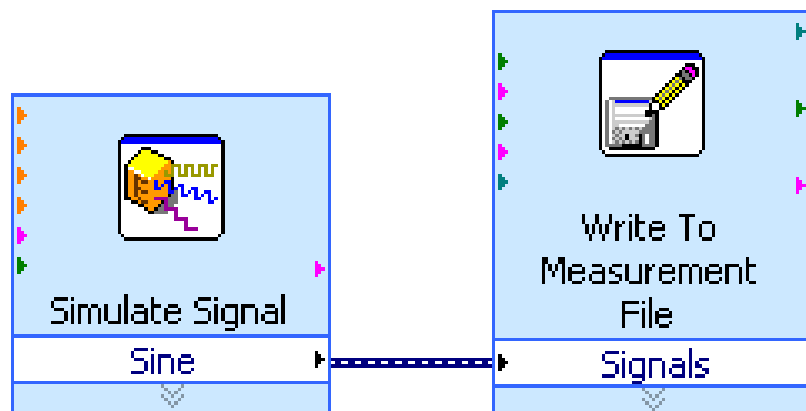
# File I/O

---

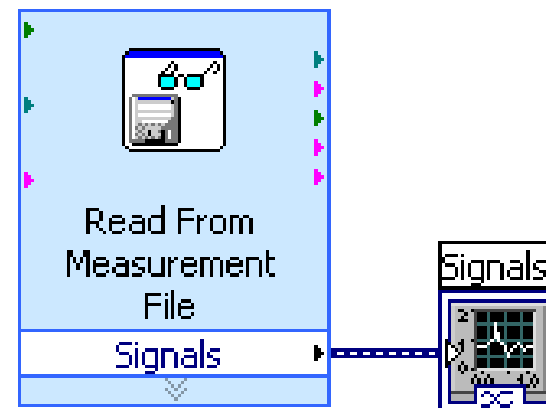
- File I/O – zapis danych do pliku i odczyt danych z pliku.
  - Podstawowe formaty zapisu danych:
    - Binary: liczbowy
    - ASCII: tekstowy
    - LVM: typ LabVIEW zapisu pomiarów (.lvm), plik tekstowy
-

# File I/O - LVM

## Zapis do pliku LVM



## Odczyt z pliku LVM



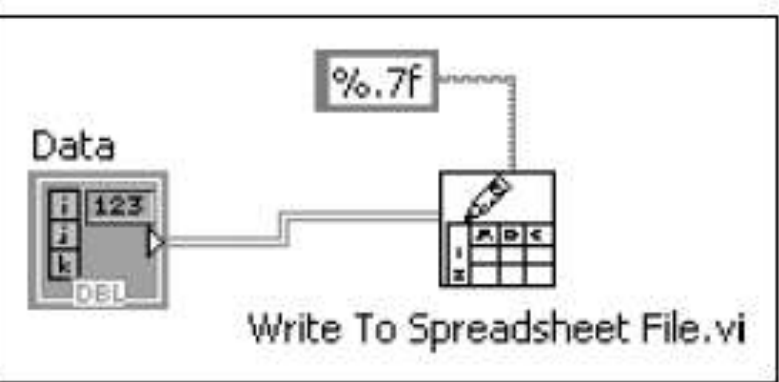
# File I/O - LVM

## Formaty zapisu

A format string for writing floating point numbers:

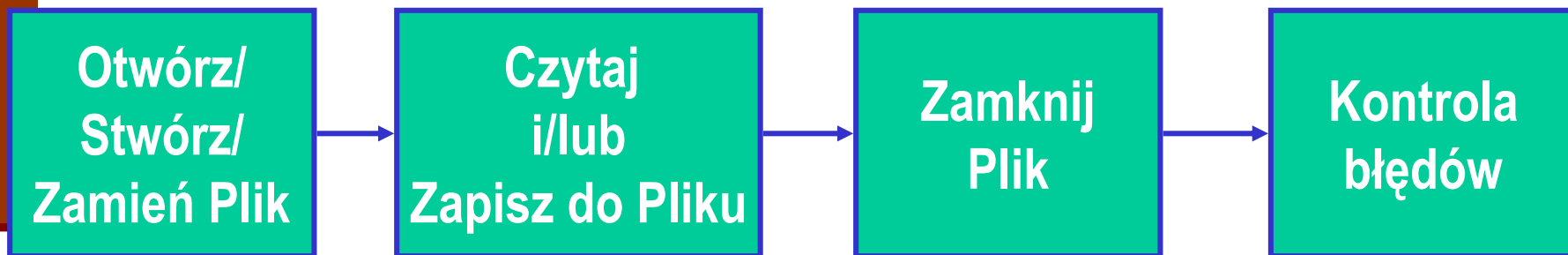
- Begins with the “%” character.
- Sometimes includes “^” (caret, [Shift 6]) to force *engineering notation* (*scientific notation* in multiples of three, e.g., e3, e6.)
- Sometimes a “#” symbol—instructs LabVIEW to drop trailing zeros.
- Sometimes includes a period and a number (e.g., “.3”) indicating the number of decimal places to show.
- Ends with a final letter indicating the notation style.
  - f—floating point notation (decimal point)
  - e—scientific notation
  - g—LabVIEW uses “f” or “e” depending on the value

%.3f (default)	12.345
%.7f	12.3450000
%#.7f	12.345
%e	1.234500e1
%.3e	1.235e1 (rounded)
%^.3e	12.345e0
%g	12.345000
%#g	12.345





# File I/O - ASCII



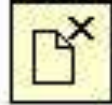
Open/Create/Replace File



Read from Text File



Close File



Simple Error Handler.vi

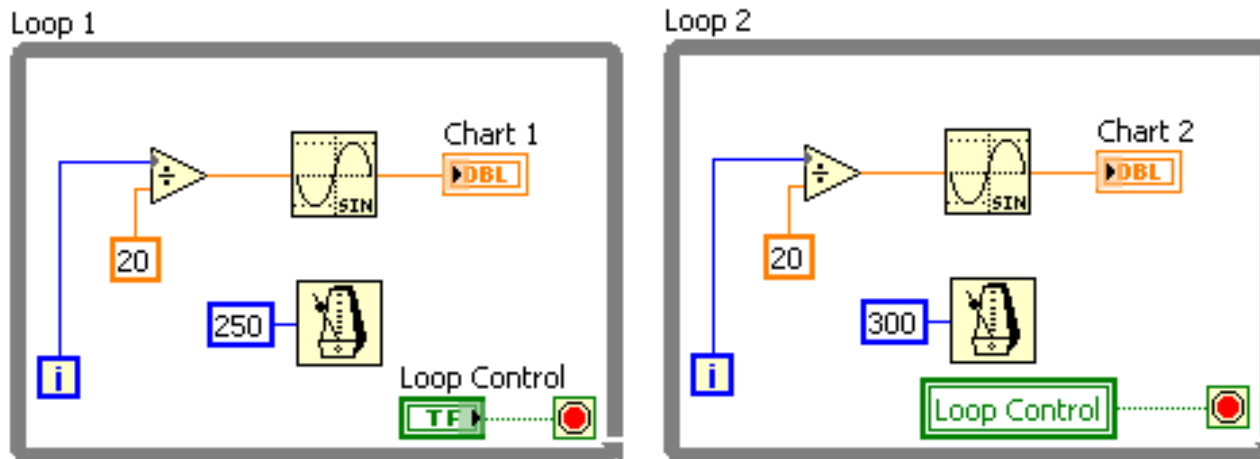


Write to Text File



# Komunikacja pomiędzy pętlami

- Komunikacja pomiędzy pętlami w „standardowy” sposób nie jest możliwa
  - Najpierw wykonywana jest lewa pętla a później prawa
- Do komunikacji pomiędzy pętlami konieczne są zmienne



# Zmienne

---

- *Zmienne (Variables)* – elementy BD, które umożliwiają dostęp do danych zdefiniowanych w innym miejscu.
  - *Zmienne Lokalne (Local variables)* przechowują dane na FP (controls i indicators).
  - Zmienne pozwalają na wymianę danych bez potrzeby użycia „drutów”.
-

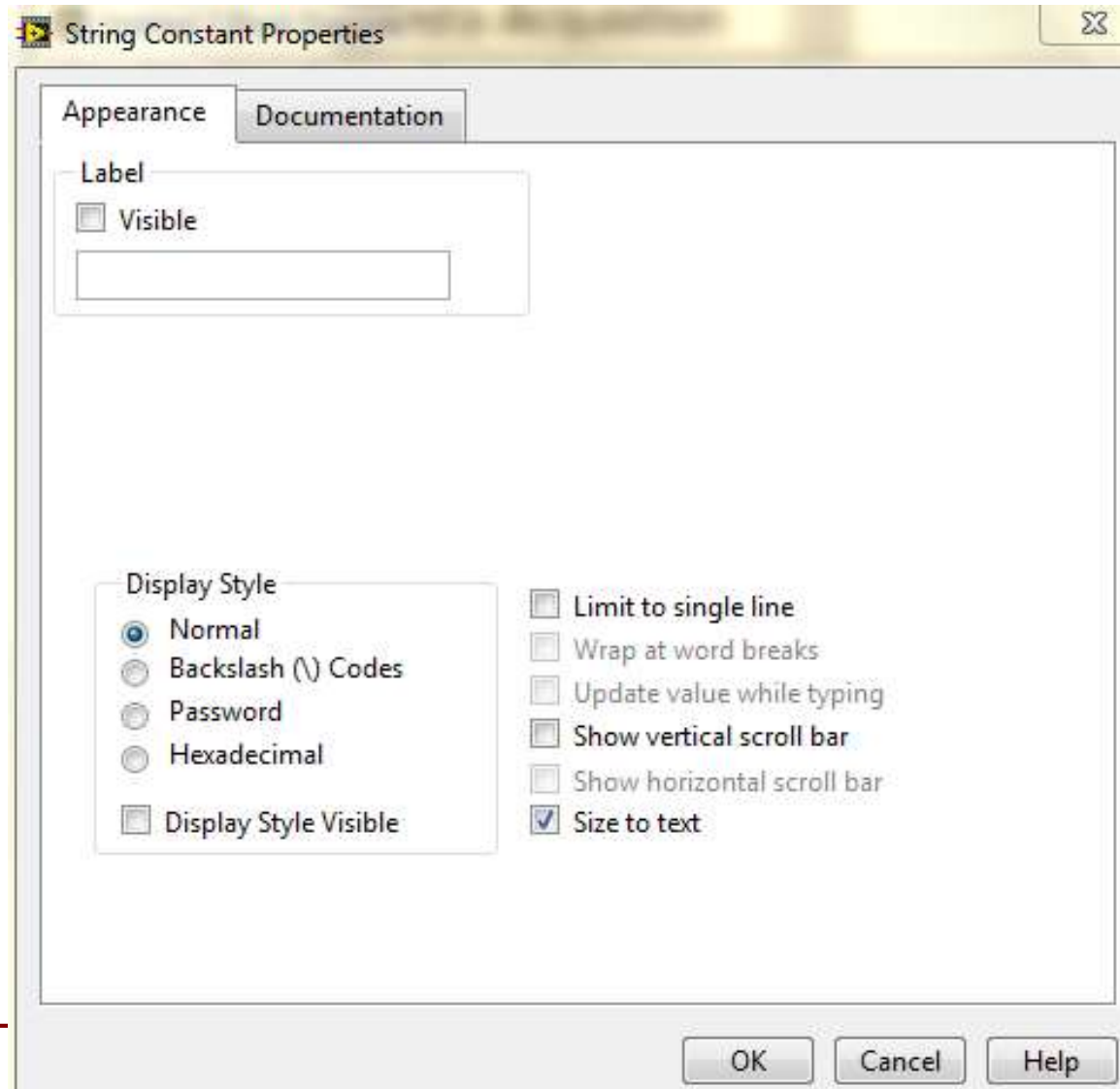
# Zmienne współdzielone

---

- Zmienne współdzielone (Shared Variables) są używane do wymiany danych pomiędzy procedurami (VIs).
  - Typy zmiennych:
    - Single Process: współdzielili dane pomiędzy Vis na tym samym komputerze.
    - Network-published: pomiędzy komputerami wykorzystując „Shared Variable Engine”.
  - Shared Variable musi być zadeklarowana w „project library”.
-

# Łańcuchy znaków - Strings

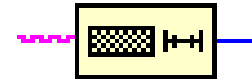
## Własności



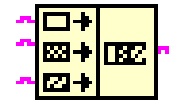
# Łańcuchy znaków - Strings

Podstawowe funkcje:

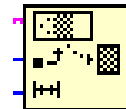
- długość łańcucha – **String length**



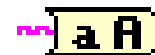
- łączenie łańcuchów – **Concatenate Strings**



- podłańcuch – **String Subset**





- duże/małe litery – **To Upper/Lower Case**



# Konwersja

Conversion

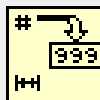
<b>EXT</b>	<b>DBL</b>	<b>SGL</b>	<b>FXP</b>
To Extended ...	To Double Pr...	To Single Pre...	To Fixed-Point
<b>I64</b>	<b>I32</b>	<b>I16</b>	<b>I8</b>
To Quad Inte...	To Long Inte...	To Word Inte...	To Byte Inte...
<b>U64</b>	<b>U32</b>	<b>U16</b>	<b>U8</b>
To Unsigned...	To Unsigned...	To Unsigned...	To Unsigned...
<b>CXT</b>	<b>CDB</b>	<b>CSG</b>	<b>#[...]</b>
To Extended ...	To Double Pr...	To Single Pre...	Number To ...
<b>[...]</b>	<b>?1:0</b>	<b>#→0</b>	<b>[Us]</b>
Boolean Arra...	Boolean To (...)	To Time Sta...	String To Byt...
<b>[Us]</b>	<b>[UNIT]</b>	<b>[...]</b>	
Byte Array T...	Convert Unit	Cast Unit Bas...	Color to RGB...
			
RGB to Color...			

# Konwersja

▼ Programming

└ String

└ String/Number Conversion



Number To ...



Number To ...



Number To ...



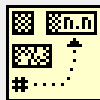
Number To ...



Number To ...



Number To ...



Format Value



Decimal Stri...



Hexadecimal...



Octal String ...



Fract/Exp Stri...

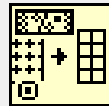


Scan Value

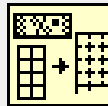


# Konwersja

---



Spreadsheet ...



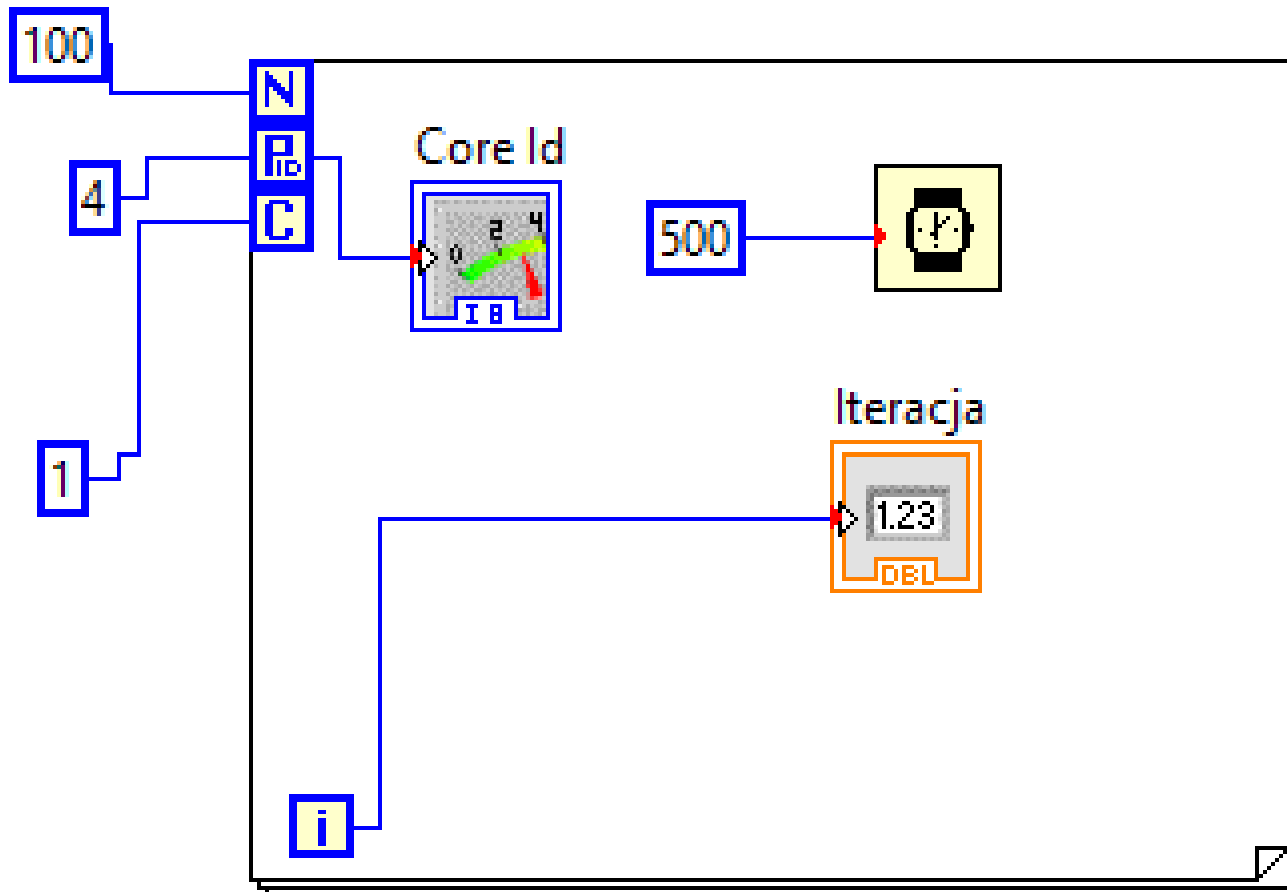
Array To Spr...

---

# Funkcje matematyczne

## Obliczenia równoległe

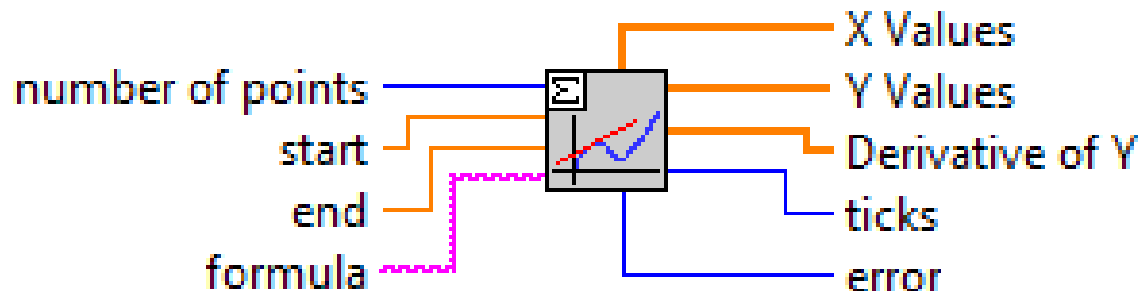
- configure iteration parallelism



# Funkcje matematyczne

- zaawansowane funkcje matematyczne

## NI\_Gmath.lvlib:Differentiation.vi



Calculates both function values and the values of the derivative of a given 1D function defined by a formula at equidistant points in an interval.

### Terminal Data Type

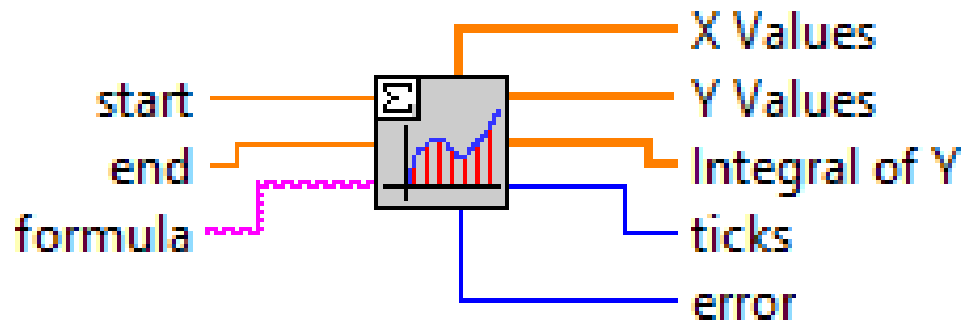
**CA** Derivative of Y (1D array of)

**DBL** (double [64-bit real (~15 digit precision)])

# Funkcje matematyczne

- zaawansowane funkcje matematyczne

## NI\_Gmath.lvlib:Integration.vi

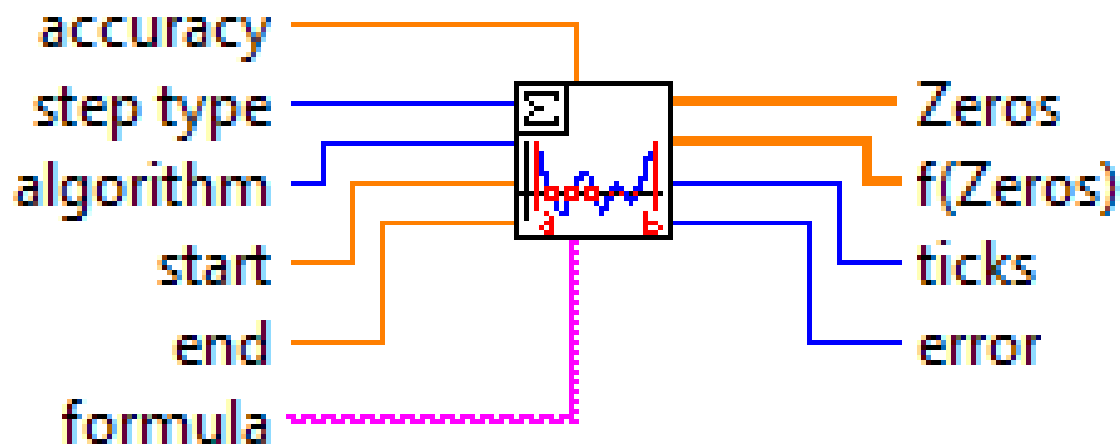


Calculates both the function values and the integral of a 1D function between **start** and **end**. The function is defined by a **formula**. The number of points to be calculated depends on the complexity of the given function.

# Funkcje matematyczne

- zaawansowane funkcje matematyczne

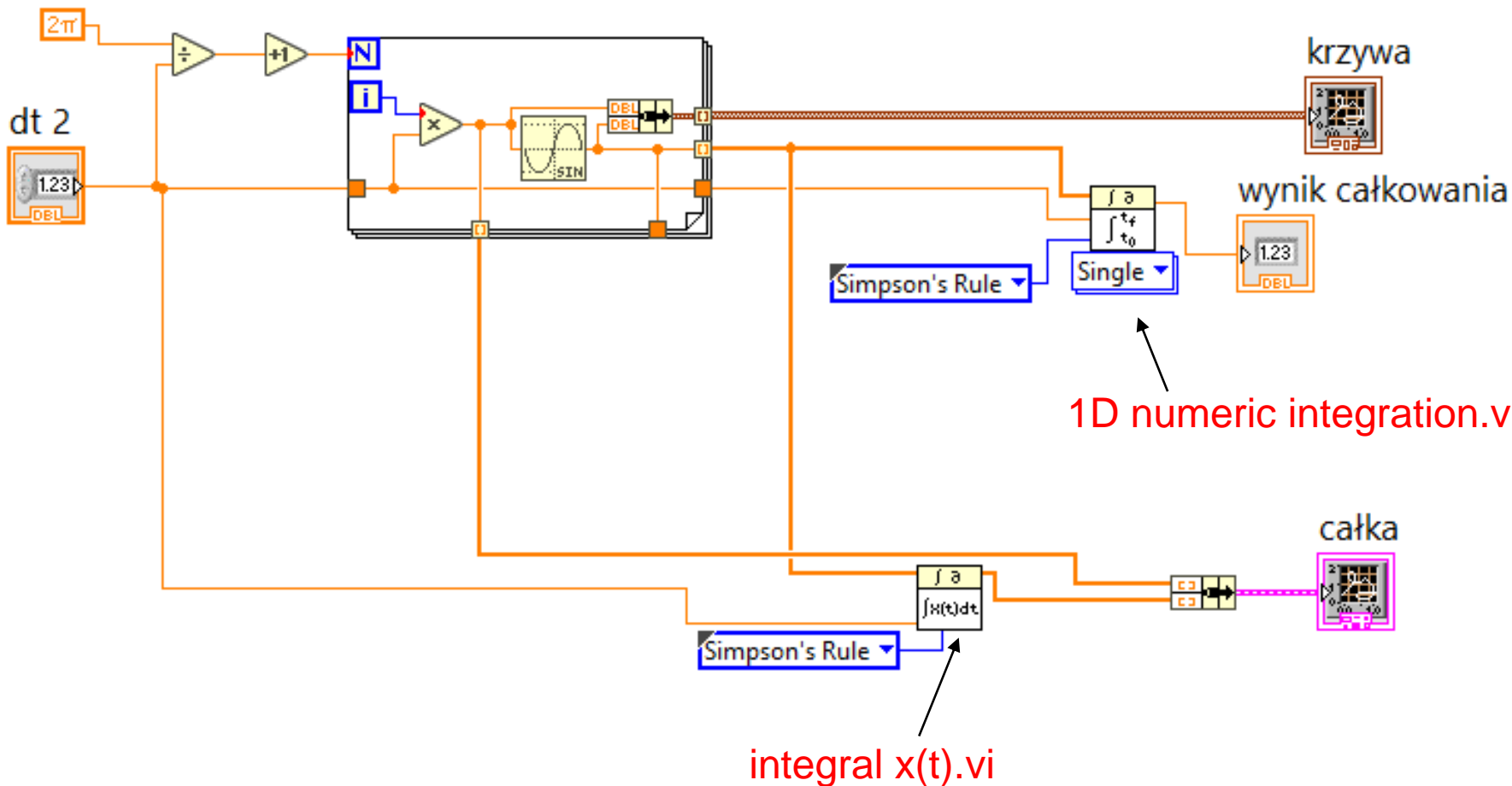
## NI\_Gmath.lvlib:Find All Zeros of f(x) (Formula).vi



Determines all zeros of a 1D function in a given interval. You must manually select the polymorphic instance to use.

# Funkcje matematyczne

2018



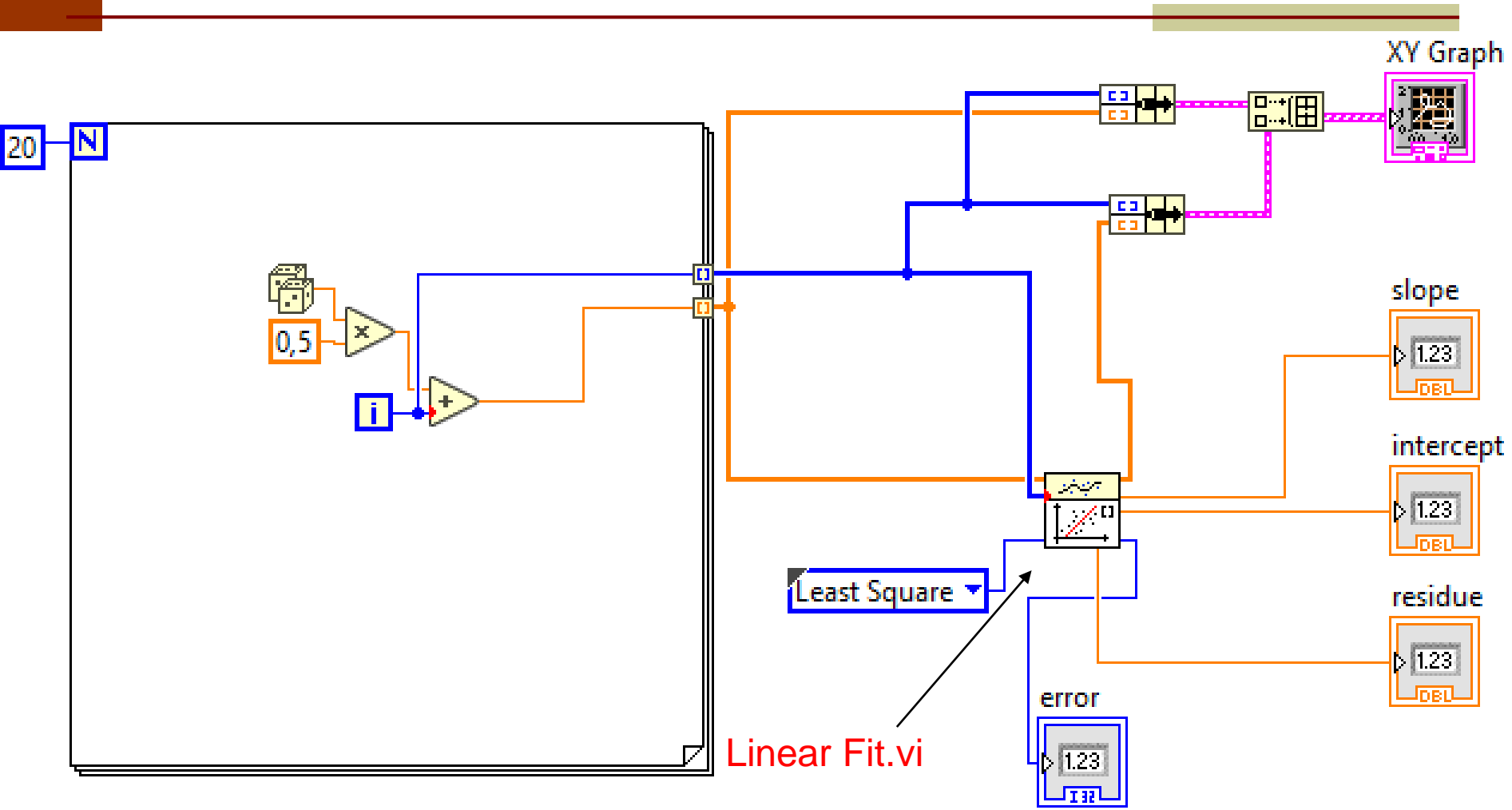
1D numeric integration.vi

integral x(t).vi

stop



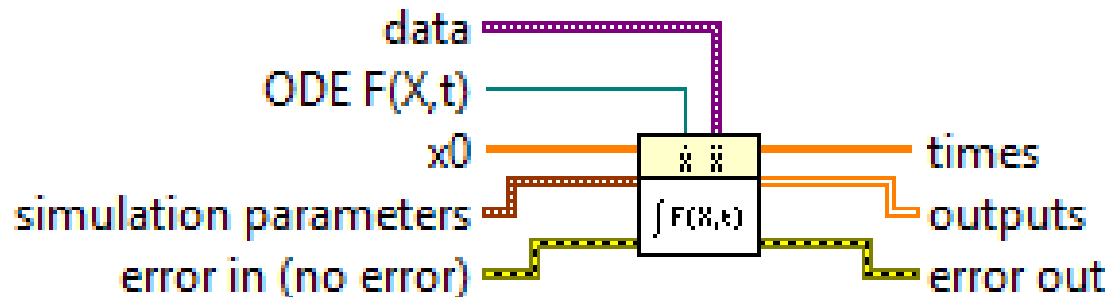
# Funkcje matematyczne



# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Solver.vi



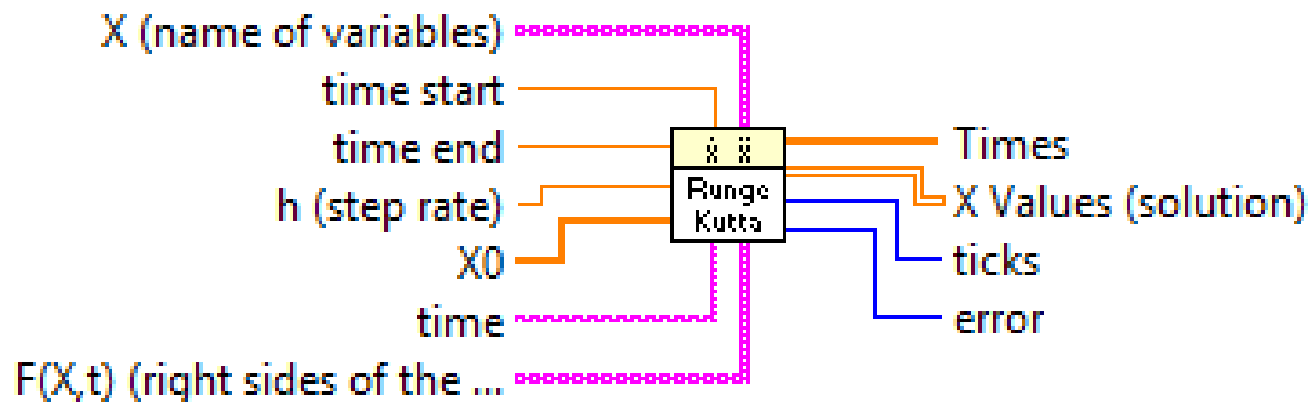
Solves ordinary differential equations with initial conditions of the following form:  $X' = F(X, t)$ . You must manually select the polymorphic instance to use.



# Funkcje matematyczne



- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Runge Kutta 4th Order.vi



Solves ordinary differential equations with initial conditions using the Runge Kutta method.

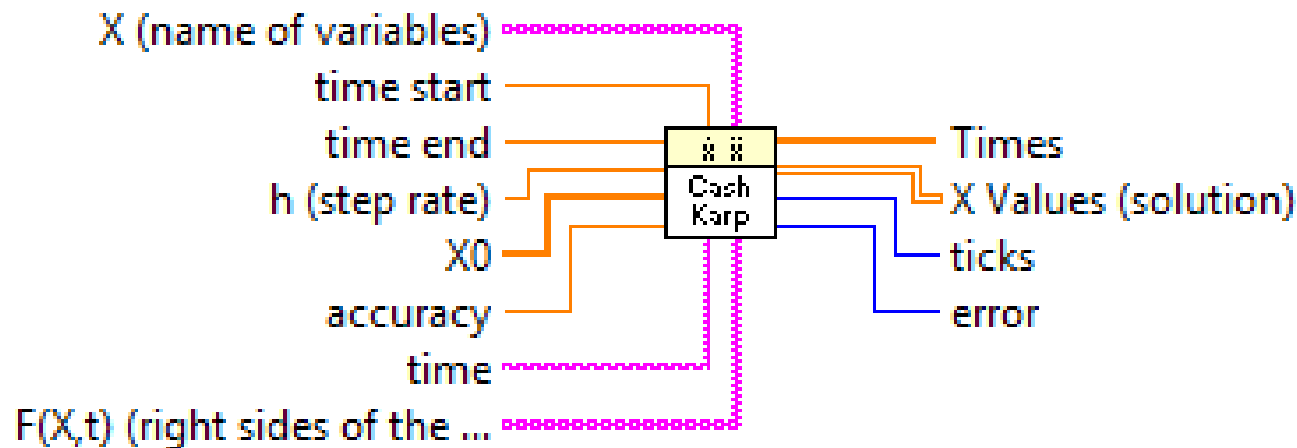
### Terminal Data Type

-  F(X,t) (right sides of the ODE as functions of X and t) (1D array of)
-  (string)

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Cash Karp 5th Order.vi



Solves ordinary differential equations with initial conditions using the Cash Karp method.

### Terminal Data Type

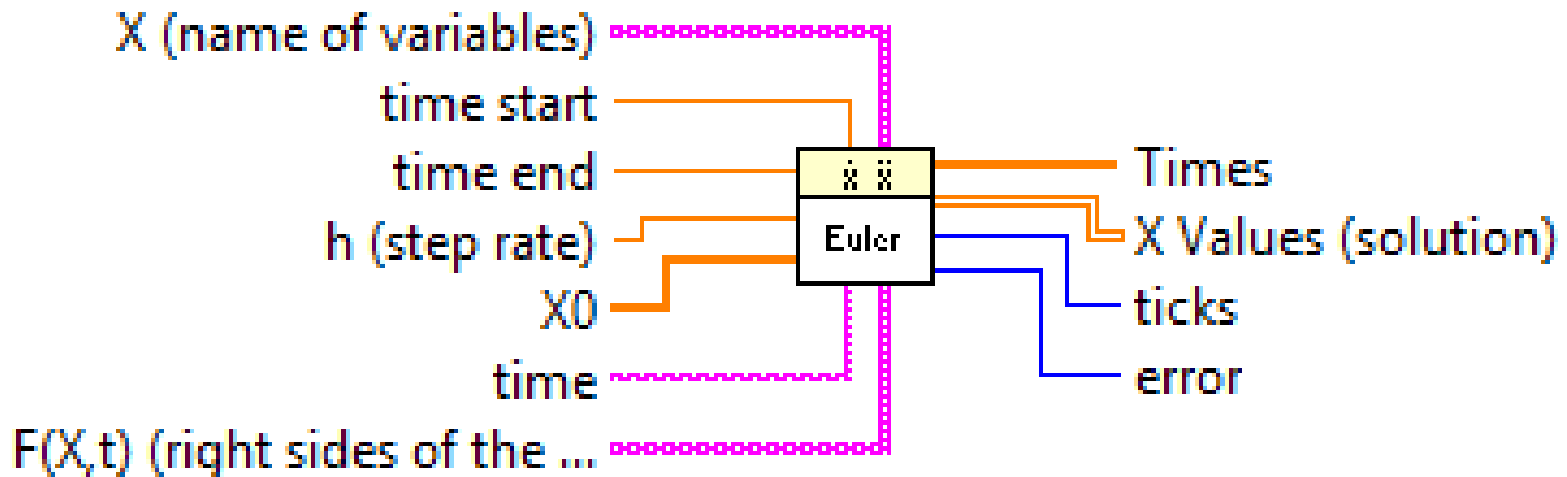
**CA** X0 (1D array of)

**DBL** (double [64-bit real (~15 digit precision)])

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Euler Method.vi

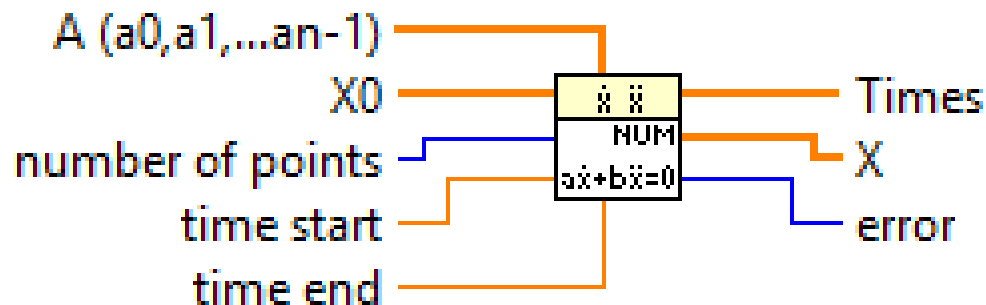


Solves ordinary differential equations with initial conditions using the Euler method.

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Linear nth Order Numeric.vi



Solves an nth-order, homogeneous linear differential equation with constant coefficients in numeric form.

### Terminal Data Type

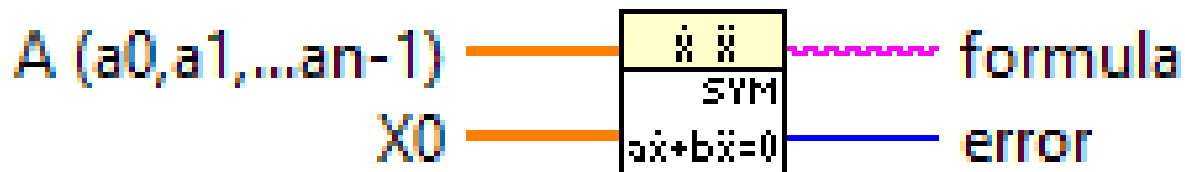
**DBL** X (1D array of)

**DBL** (double [64-bit real (~15 digit precision)])

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Linear nth Order Symbolic.vi

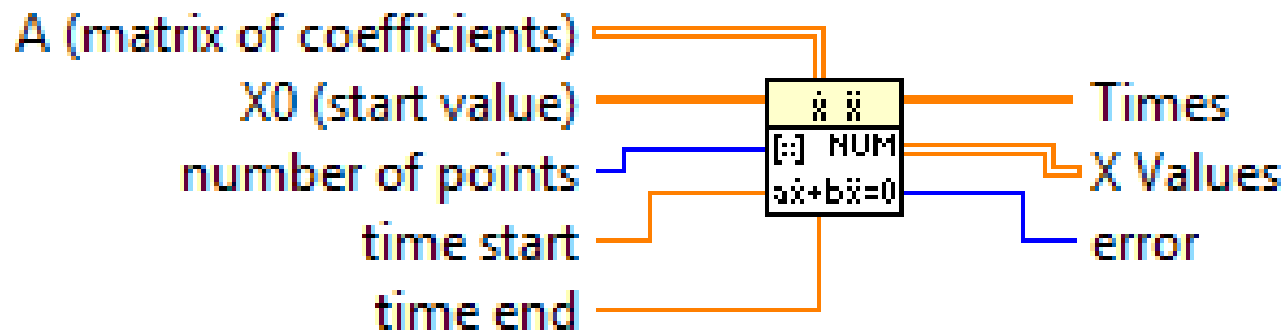


Solves an nth-order, homogeneous linear differential equation with constant coefficients in symbolic form.

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Linear System Numeric.vi

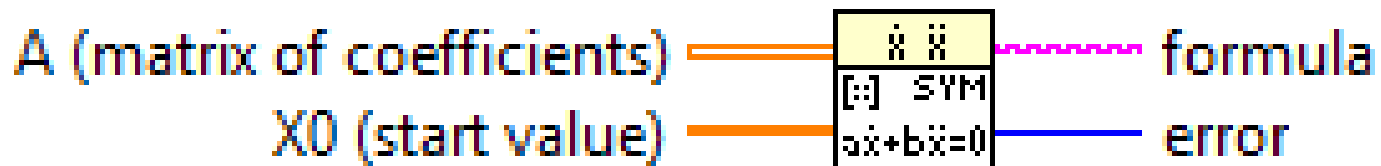


Solves an n-dimension, homogeneous linear system of differential equations with constant coefficients, for a given start condition.

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_Gmath.lvlib:ODE Linear System Symbolic.vi

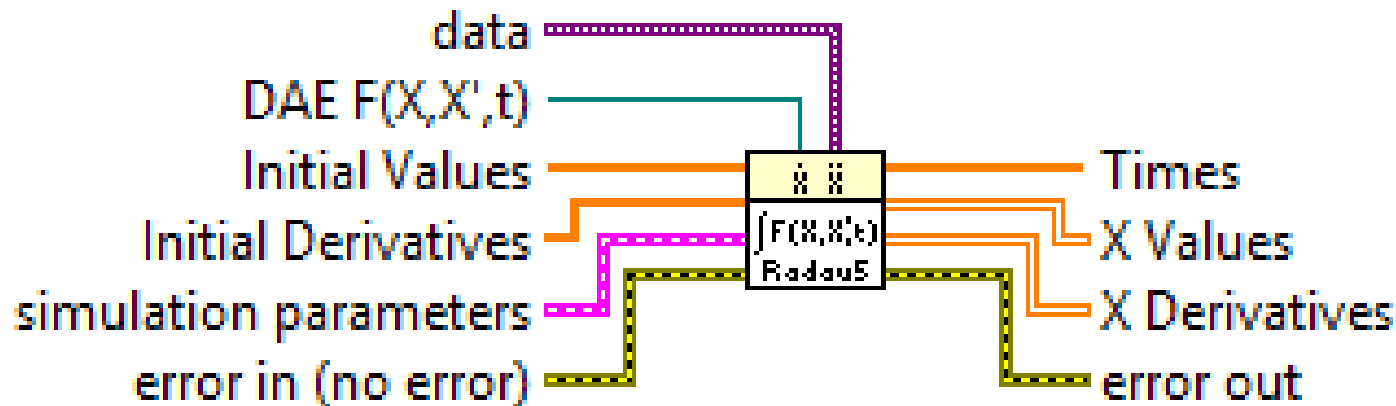


Solves an n-dimension linear system of differential equations with a given start condition. The solution is based on the determination of the eigenvalues and eigenvectors of the underlying matrix. The solution is given in symbolic form.

# Funkcje matematyczne

- Mathematics / Differential Equations
  - Ordinary Differential Equations

## NI\_DAE.lvlib:DAE Radau 5th Order.vi



Solves differential algebraic equations (DAEs) with initial conditions by using the Radau IIA method. You must manually select the polymorphic instance to use.



# Zadanie 1

---

- napisać program wypełniający kwadratową ( $N \times N$ ) tablicę 2D liczbami naturalnymi z zakresu  $\langle 0; 100 \rangle$  dla zadanego  $N$  (ilości wierszy i kolumn)
  - korzystając z „property nodes” dopasować rozmiar indykatora tablicy do rzeczywistej wielkości  $N$  (ilości wierszy i kolumn) – aby nie wyświetlał pustych wierszy i kolumn
  - dodać element konwersji tablicy na macierz – wyświetli macierz
-

# Zadanie 2

---

- korzystając z poprzedniego zadania wygeneruj macierz  $N$  wymiarową
  - wygeneruj macierz transponowaną do powyższej i podziel jedną przez drugą
  - pamiętaj aby korzystając z „property nodes” dopasować rozmiar indykatora macierzy do rzeczywistego rozmiaru  $N$
  - wydziel zadaną kolumnę macierzy jako tablicę 1D
-

# Zadanie 3

---

- korzystając z poprzednich zadań napisz program zamieniający miejscami dwa wybrane wiersze (lub dwie kolumny) danej tablicy
-

# Zadanie 4

---

- do zadania 2 dodaj możliwość zapisania macierzy wynikowej do pliku używając funkcji „Write to measurement file”
-

# Zadanie 5

---

- do zadania 2 dodaj możliwość zapisania macierzy wynikowej do pliku używając funkcji zapisu do pliku ASCII - str. 17
  - napisz inny program, który odczyta zapisane do pliku dane z powrotem do macierzy
-