

Komputerowe wspomaganie eksperymentu

3

Dr Piotr Sitarek

Katedra Fizyki Doświadczalnej, Politechnika Wrocławska

Temat na dziś

Dokumentacja, wizualizacja danych

ni.com



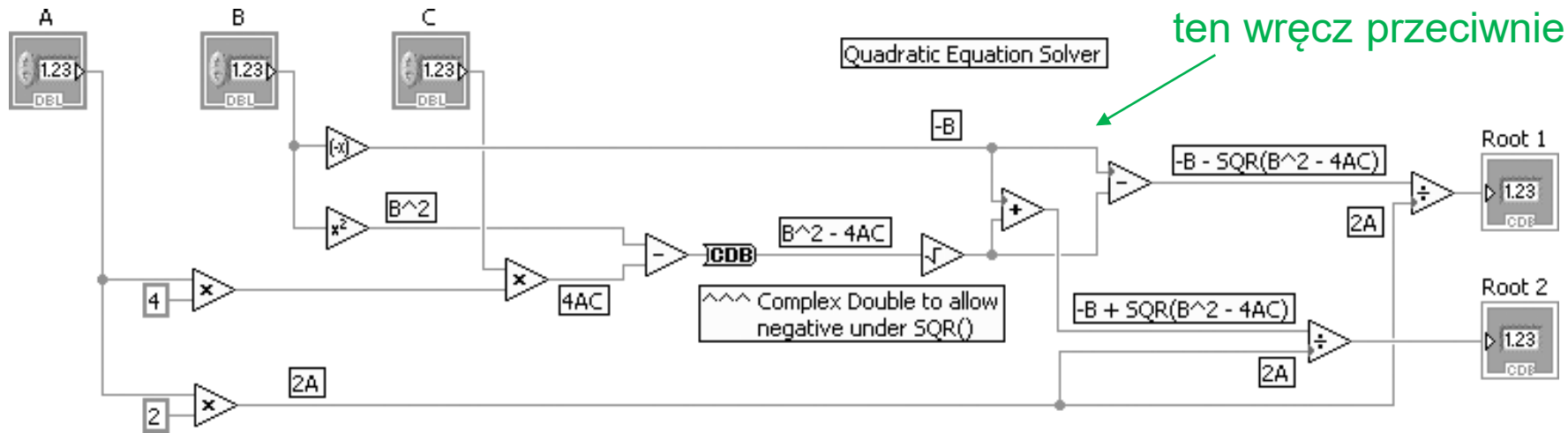
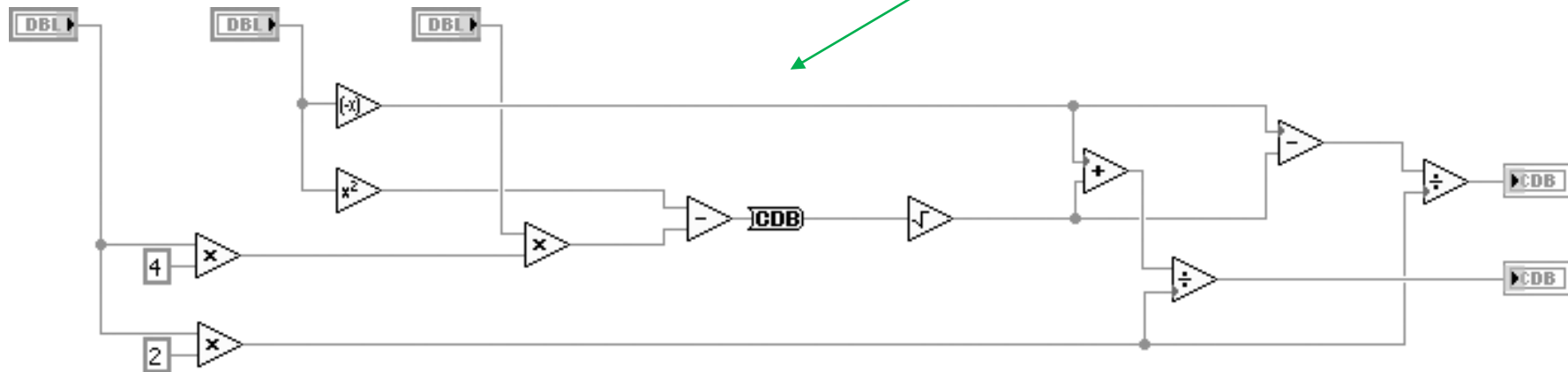
(część materiałów zaczerpnięta ze strony
producenta)

RONALD W. LARSEN, *LabVIEW for Engineers*

Dokumentacja

Opis BD

ten kod niewiele mówi



ten wręcz przeciwnie

Dokumentacja

Opis FP

taki Front Panel jest nieczytelny

1	6	4	-5.23607 +0 i	-0.763932 +0 i
---	---	---	---------------	----------------

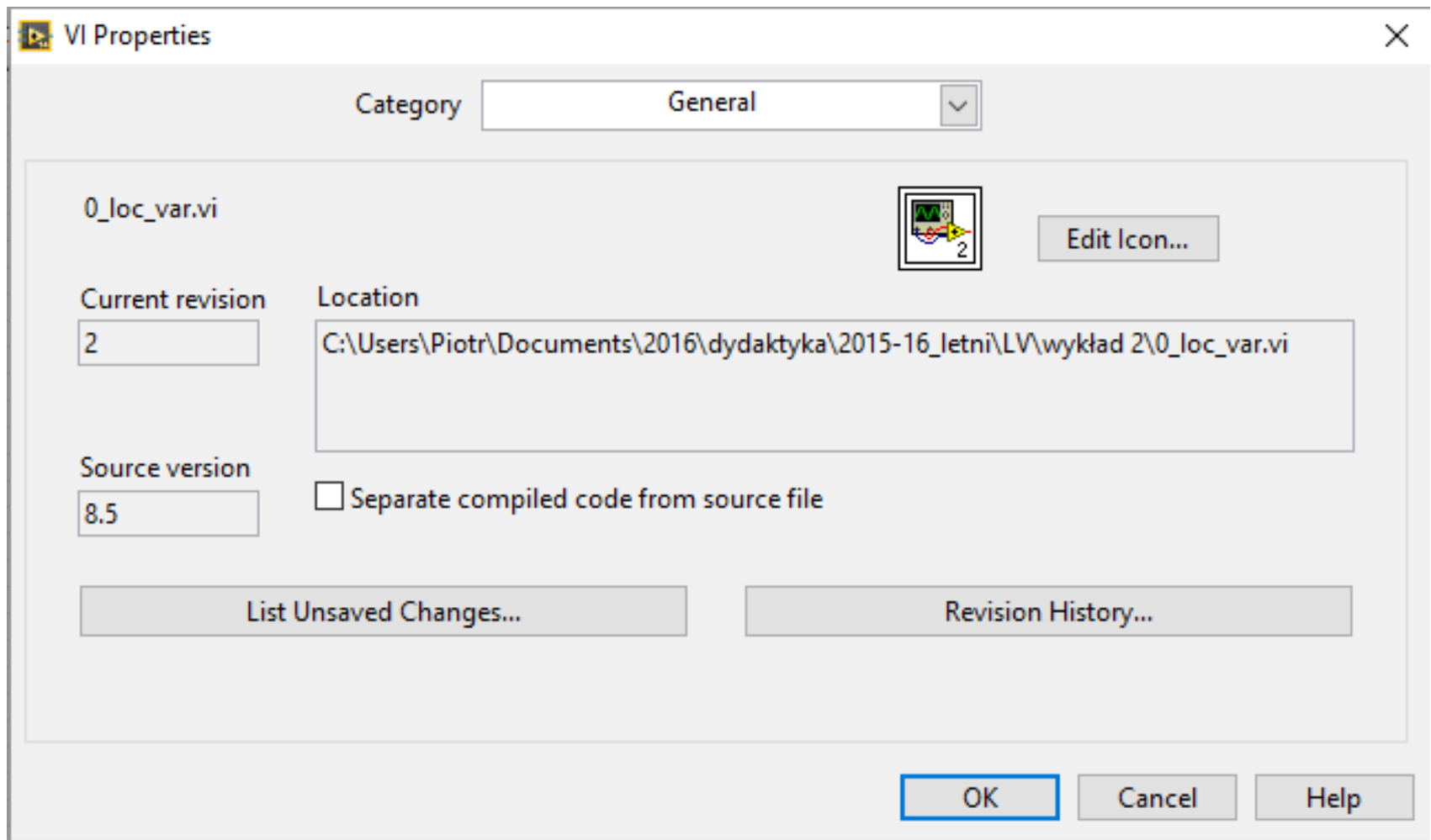
A	B	C	Root 1	Root 2
1	6	4	-5.23607 +0 i	-0.763932 +0 i

Quadratic Equation Solver				
Quadratic Coefficients			Calculated Results	
A	B	C	Root 1	Root 2
1	6	4	-5.23607 +0 i	-0.763932 +0 i

$$A x^2 + B x + C = 0$$

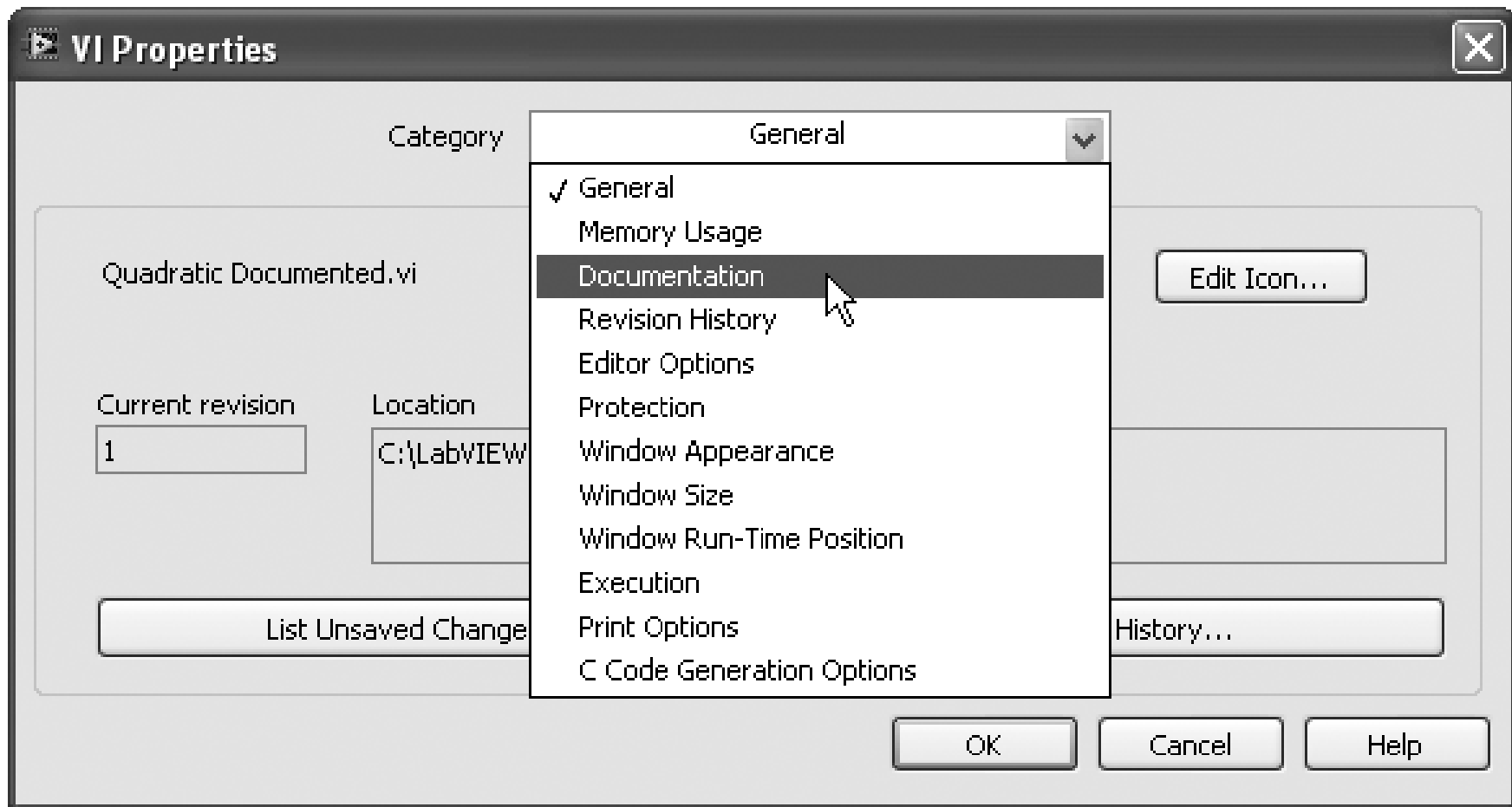
Dokumentacja

■ Informacje opisowe - **File / VI Properties**



Dokumentacja

■ Informacje opisowe - **File / VI Properties**



Dokumentacja

■ Documentation:

- Tytuł
- Autor i data
- składnia i co VI ma robić
- informacje o typach danych
- edycja tekstu

■ **Help tag** - nowe określenia do indeksu (związane z VI)

■ **Help path** – ścieżka do pliku HTML help file (**Detailed Help** pojawi się w Context Help window)

Dokumentacja

■ Documentation:

- Tytuł
- Autor i data

VI Properties

Category: Documentation

VI description

Quadratic Equation Solver
by Ron Larsen, June 2009

Solves quadratic equation of form $Ax^2 + Bx + C = 0$ for both roots.
Roots are of type complex double so that both real and complex roots can be calculated.

Help tag:


Help path:

Dokumentacija

■ Context Help


Context Help

Quadratic Documented.vi



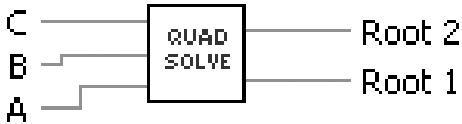
Quadratic Equation Solver
by Ron Larsen, June 2009

Solves quadratic equation of form $Ax^2 + Bx + C = 0$ for both roots.
Roots are of type complex double so that both real and complex roots can be calculated.



Context Help

Quad Solve.vi



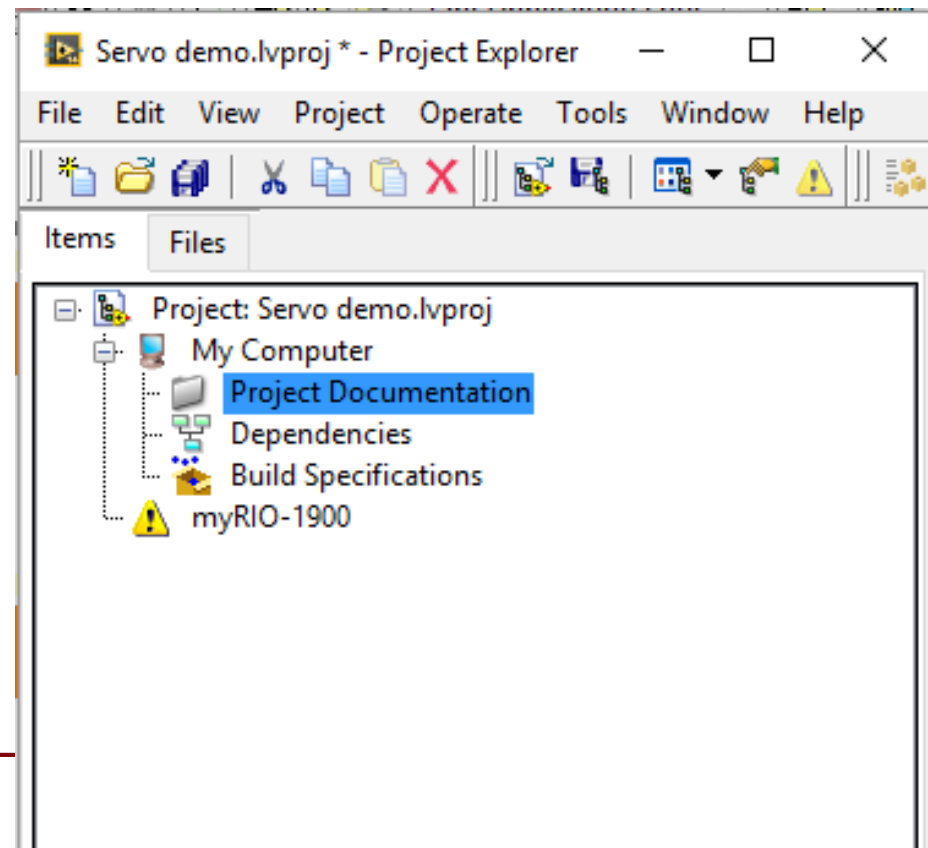
Quadratic Equation Solver
by Ron Larsen, June 2009

Solves quadratic equation of form $Ax^2 + Bx + C = 0$ for both roots.
Roots are of type complex double so that both real and complex roots can be calculated.

Projekty

Projekt – zbiór wszystkich VIs i SubVIs niezbędnych do działania.

Project / New Project . . . - Add



Dokumentacja

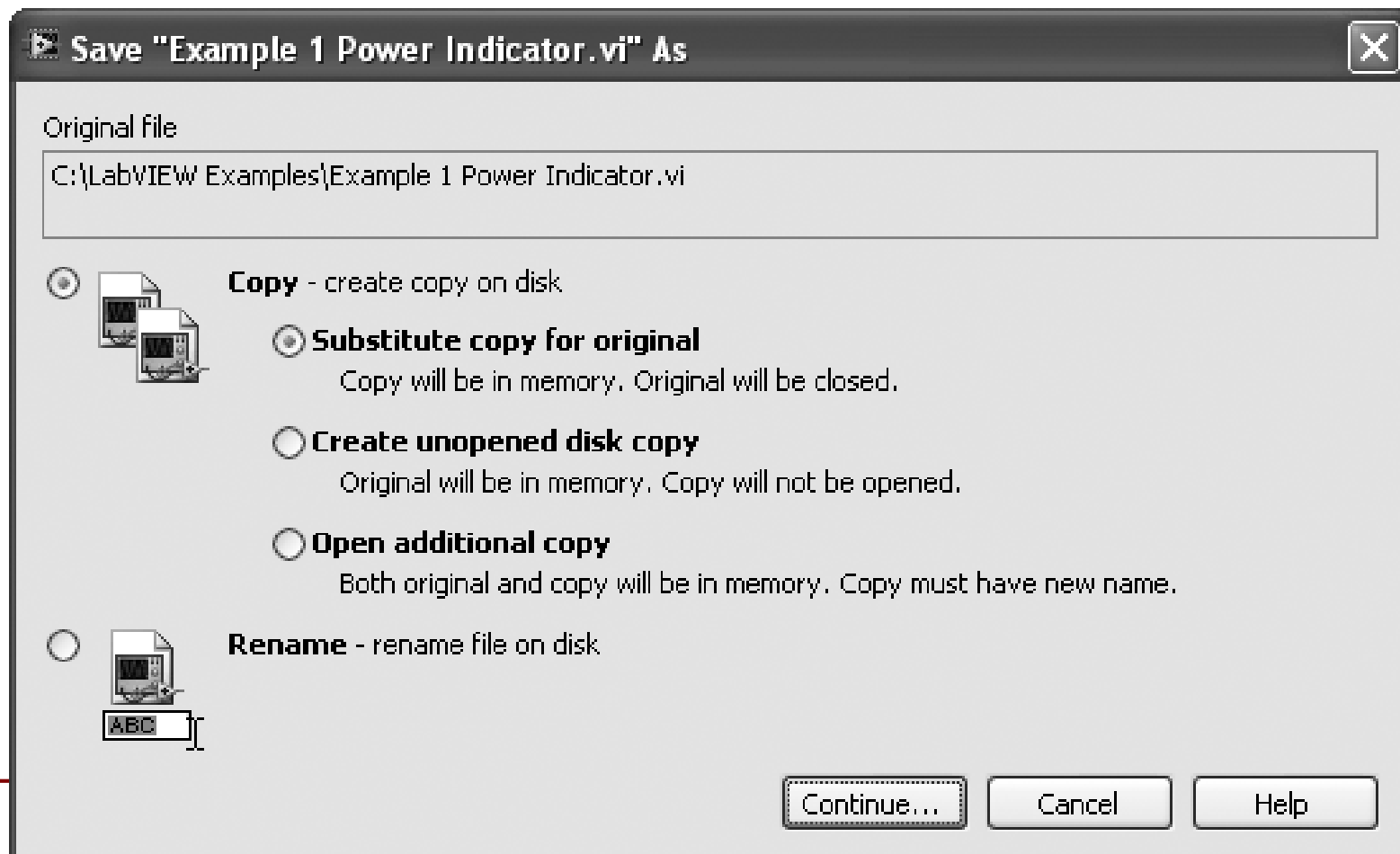
■ Zachowywanie

- **Save** – zachowuje aktualny VI.
- **Save As . . .** (dialog box).
- **Save All** – zachowuje wszystkie otwarte VI
- **Save for Previous Version . . .** zachowuje w poprzedniej wersji

Dokumentacja

■ Zachowywanie

■ Save As . . . (dialog box).

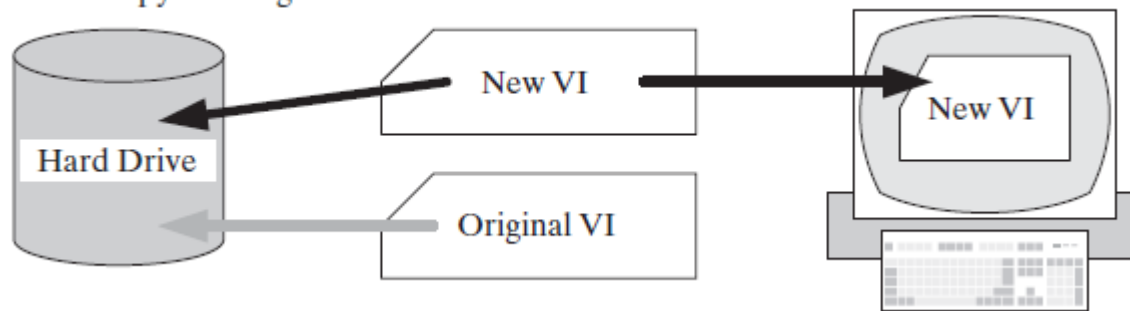


Dokumentacja

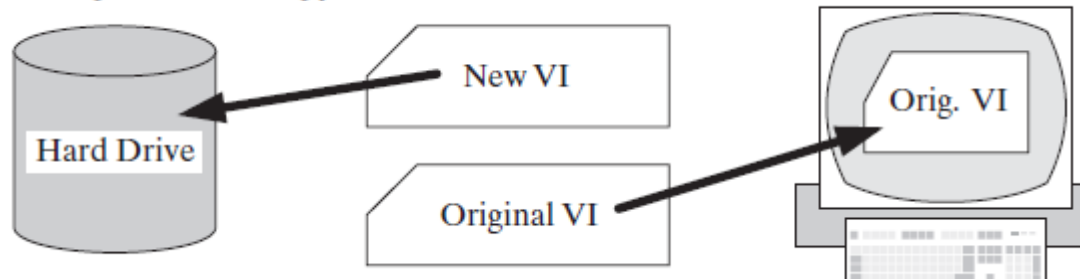
■ Zachowywanie

■ Save As . . . (dialog box).

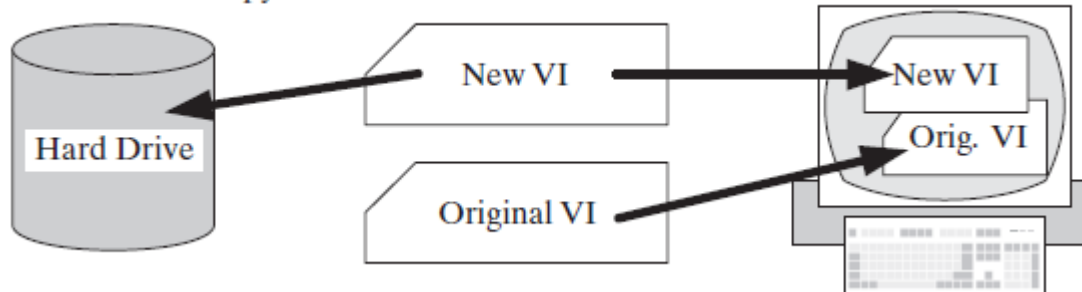
Substitute Copy for Original



Create Unopened Disk Copy



Open Additional Copy



Wielokrotny dostęp do procedur

- normalnie dostęp do VI lub subVI jest jednokrotny – wielokrotne wywołanie tego samego kodu działa krokowo – musi zakończyć się „jeden dostęp” aby rozpoczął się kolejny
 - można to zmienić ustawiając tzw. reentrant execution – jednoczesny dostęp do danego VI lub subVI z różnych miejsc programu – każdy dostęp rezerwuje własny obszar pamięci – przyspiesza działanie programu ale obciąża pamięć
 - File / VI Properties
-

Wielokrotny dostęp do procedur

VI Properties

Category: Execution

Allow debugging

Reentrancy

- Non-reentrant execution
- Shared clone reentrant execution
- Preallocated clone reentrant execution

Reentrancy settings affect memory usage, call overhead, jitter, and state maintained within the VI. Display Context help for guidance with selecting the best setting for your use case.

Inline subVI into calling VIs

Priority: normal priority

Preferred Execution System: same as caller

- Enable automatic error handling
- Run when opened
- Suspend when called
- Clear indicators when called
- Auto handle menus at launch

OK Cancel Help

Biblioteki .LLB

Library Benefits and Drawbacks

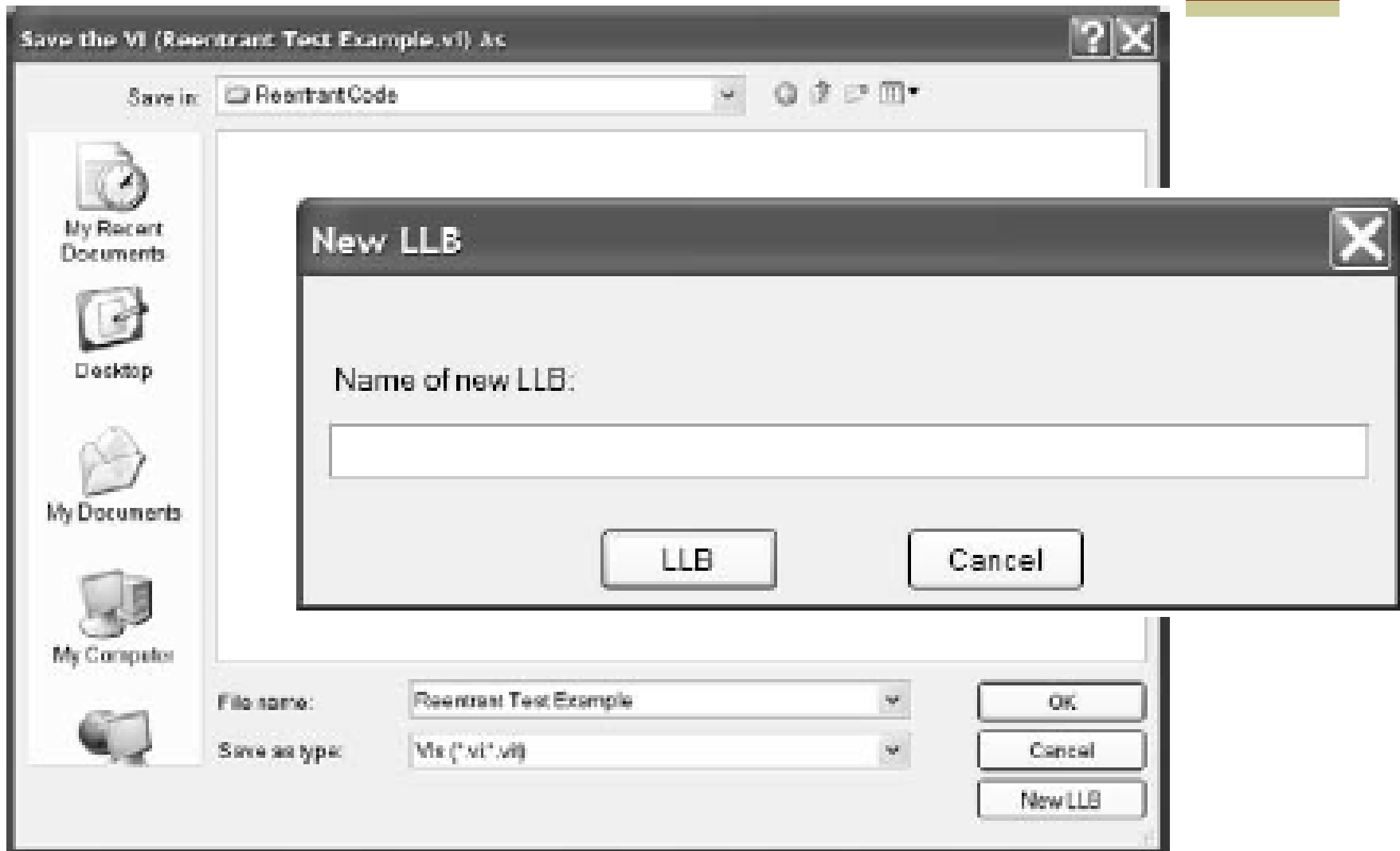
Benefits of Saving as .VI

1. You can use your own file system to manage and store VIs.
2. Libraries are not hierarchical. Libraries cannot contain subdirectories or sublibraries.
3. Loading and saving VIs is faster and requires less disk space for temporary files.
4. More robust than storing entire project in the same file.
5. There is the possibility of a library becoming corrupt.
6. Source Control cannot operate on individual VIs in a LLB, only the entire LLB.

Benefits of Saving as .LLB

1. 255 characters can be used for naming files (may be useful for Macintosh, where filenames are limited to 31 characters).
 2. Easier for transporting VIs to different platform or to a different computer
 3. Libraries are compressed and require less disk space.
 4. Can now be viewed in Windows Explorer (starting in LabVIEW 7).
 5. Can set one or several VIs to start when the library is opened (top level VIs).
-

Biblioteki .LLB

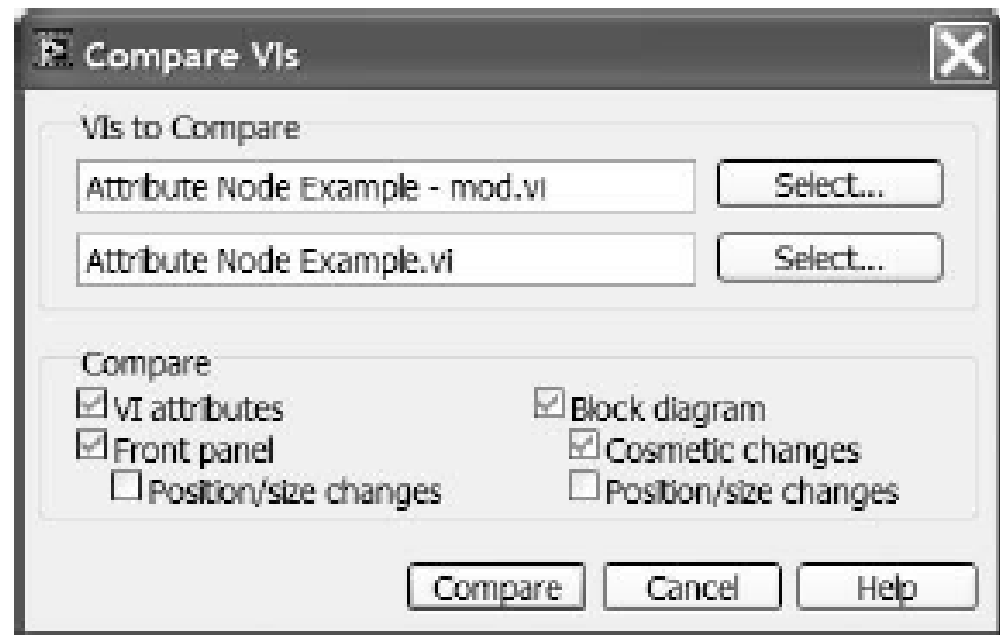


Biblioteki .LLB

- Save As ... VI Library
 - New Library
 - dodać do istniejącej biblioteki – llb są traktowane w Windows jako folder
 - Tools / LLB Menager
-

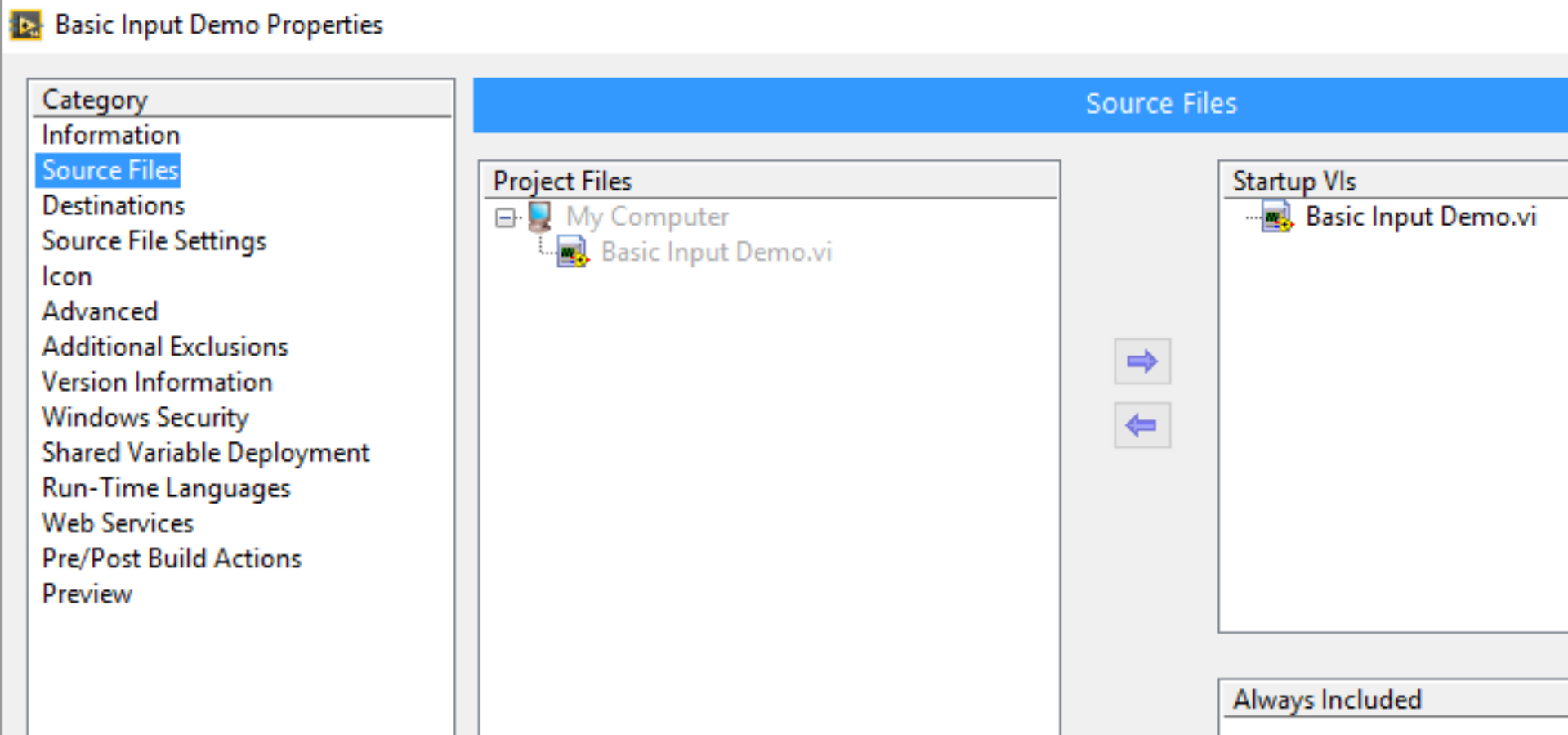
Porównywanie VIs

- Tools / Compare / Compare Vis
- / Show Differences
- / Compare VI Hierarchies



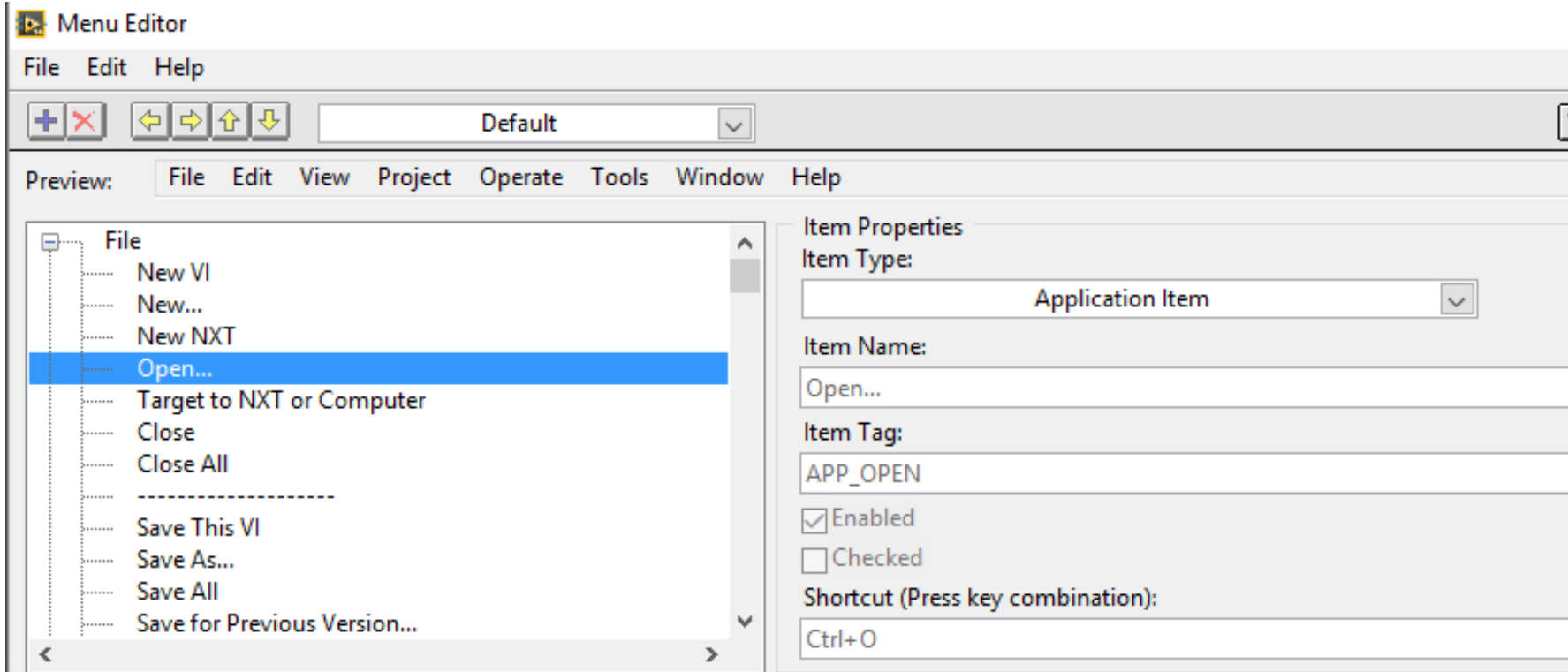
Tworzenie aplikacji

- Tools / Build Application (exe)



Menu Editor

Edit / Run-Time Menu



Wykresy

Chart Types

- Waveform Chart
- Intensity Chart

Graph Types

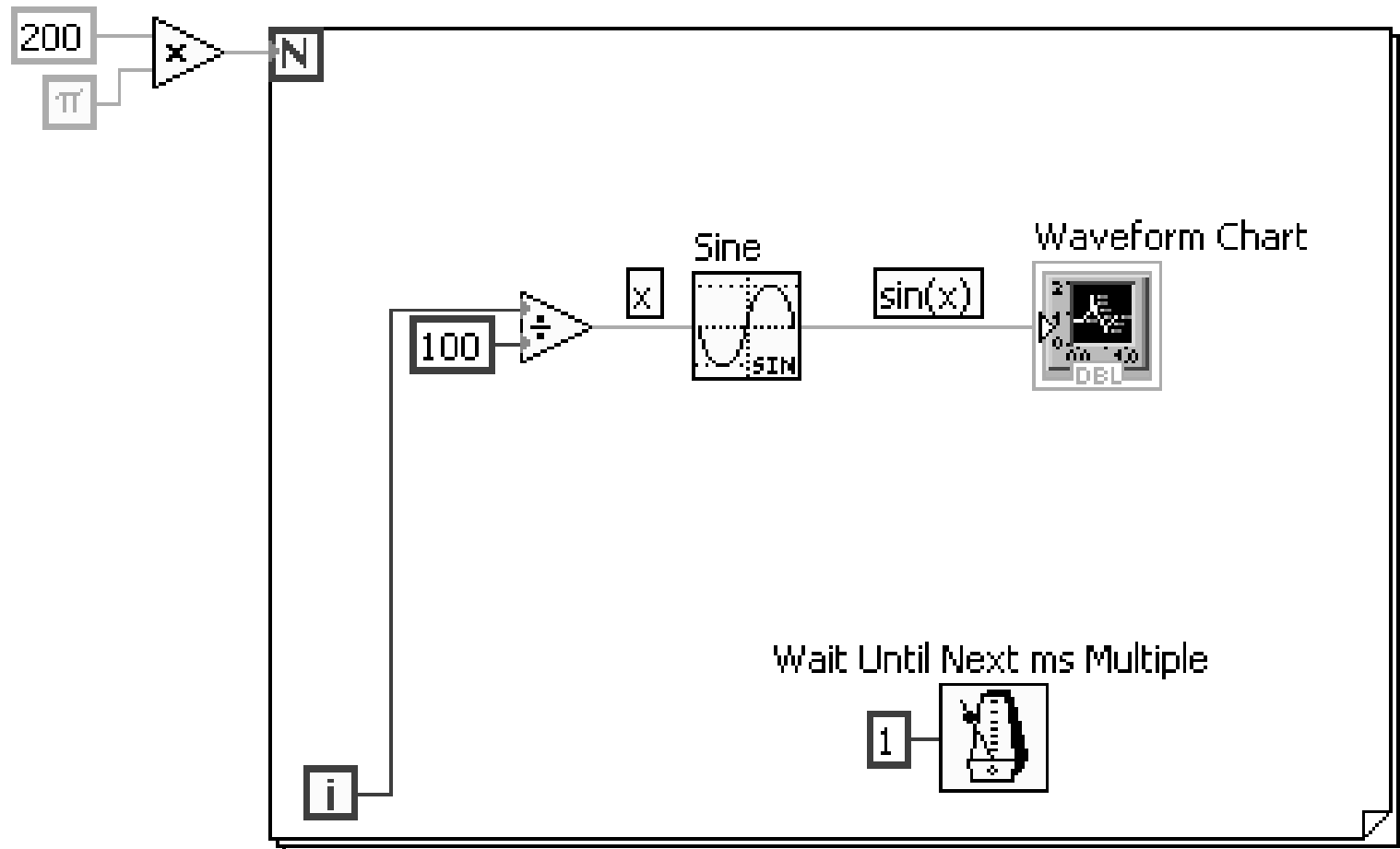
- Waveform Graph
- XY Graph
- Intensity Graph
- Digital Waveform Graph
- Mixed Signal Graph
- 3D Surface Graph
- 3D Parametric Graph
- 3D Curve Graph

Wykresy

- **chart** – pojedyncze punkty ciągle uaktualniane - *Waveform Chart*
- **graph** – otrzymuje całość danych (jako tablicę) zanim je wyświetli
- LabVIEW umożliwia
 - wykresy graph i chart - 1D (stały odstęp pomiędzy x-ami)
 - wykres graph - 2D (XY Graph)
 - wykres graph - 3D

Wykresy

Waveform Chart – rysowanie punkt po punkcie



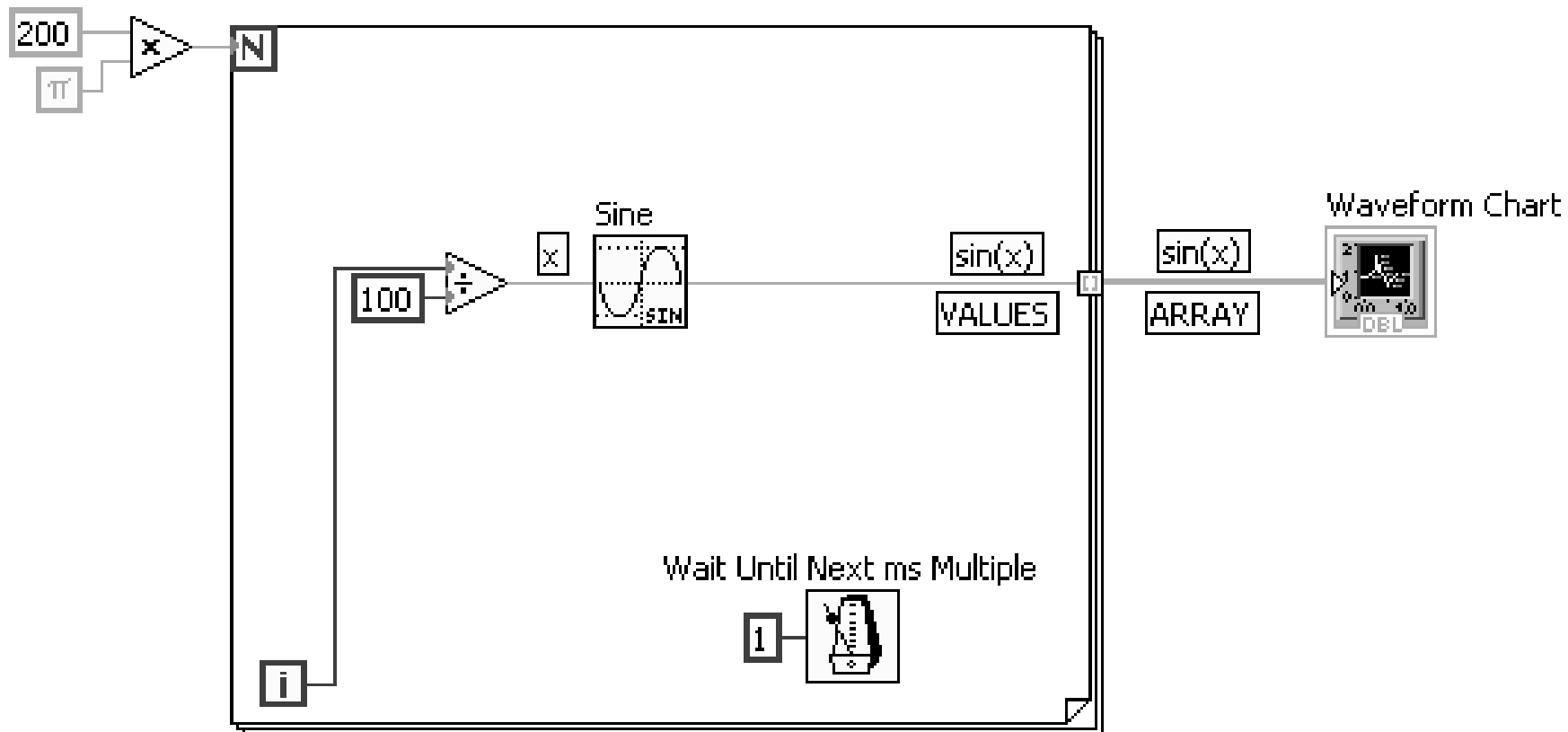
Wykresy

Waveform Chart – rysowanie punkt po punkcie

- **chart history length**
 - **update mode**
 - **Strip Chart (default)**
 - **Scope Chart**
 - **Sweep Chart**
-

Wykresy

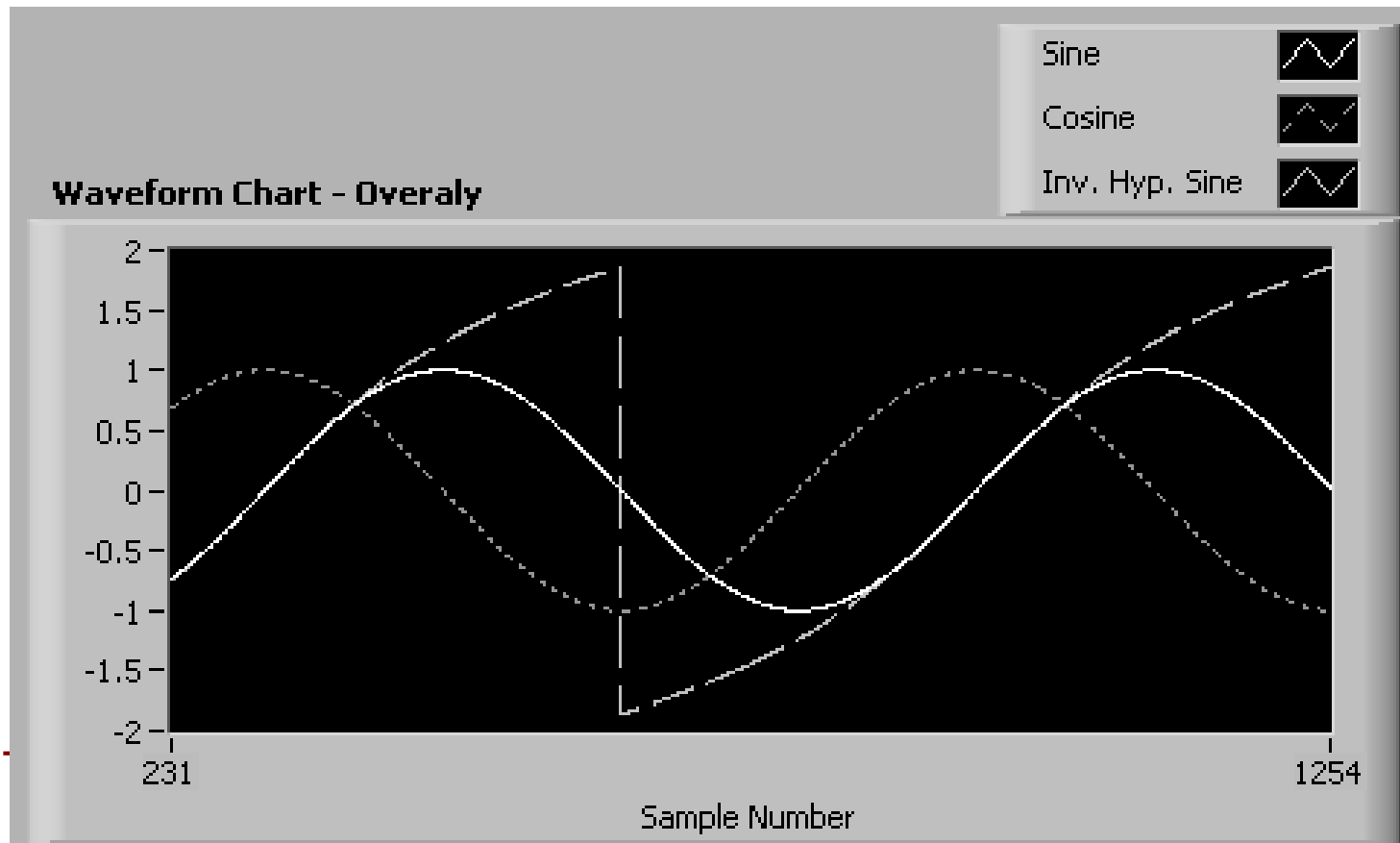
Waveform Chart – rysowanie całości



Wykresy

Waveform Chart – kilka krzywych

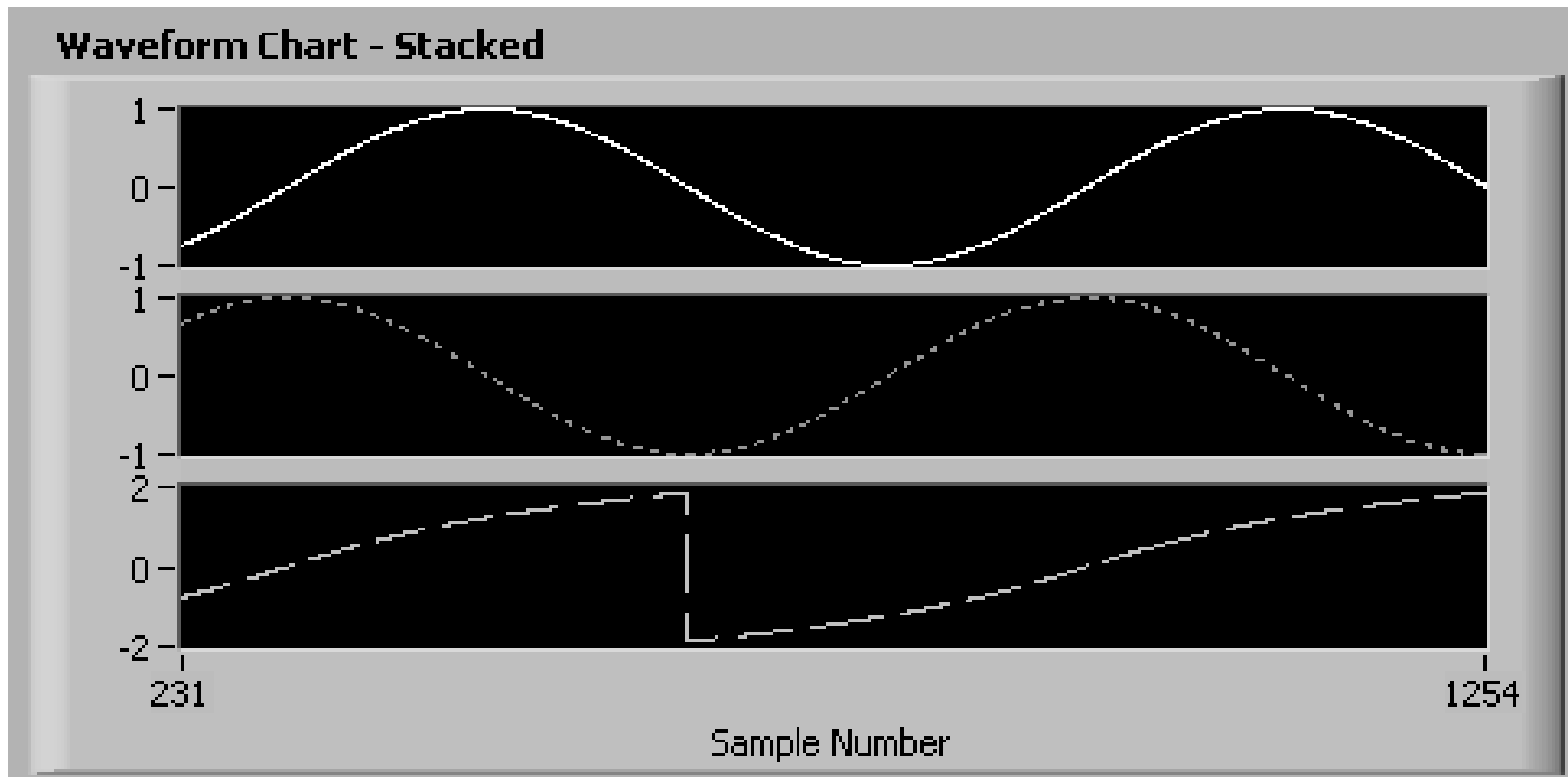
- **Overlay Plots** – wszystkie krzywe na tym samym wykresie



Wykresy

Waveform Chart – kilka krzywych

■ **Stacked Plots – krzywe osobno**



Wykresy

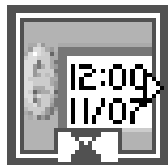
Waveform Graphs

- tablica (1D) do wyświetlenia
 - wszystkie wartości wyświetlane od razu
 - wiele wykresów – trzeba zbudować i przekazać odpowiednią tablicę
-

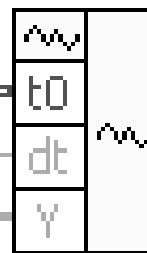
Wykresy

Waveform Graphs i Charts - opcje

Start Time, t_o



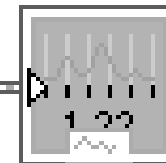
Build Waveform function



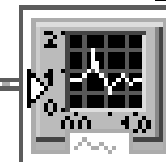
Time Interval, Δt



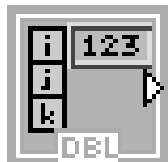
Waveform (bundled)



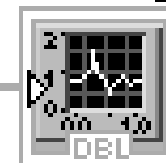
Plotting the Waveform



Data Array

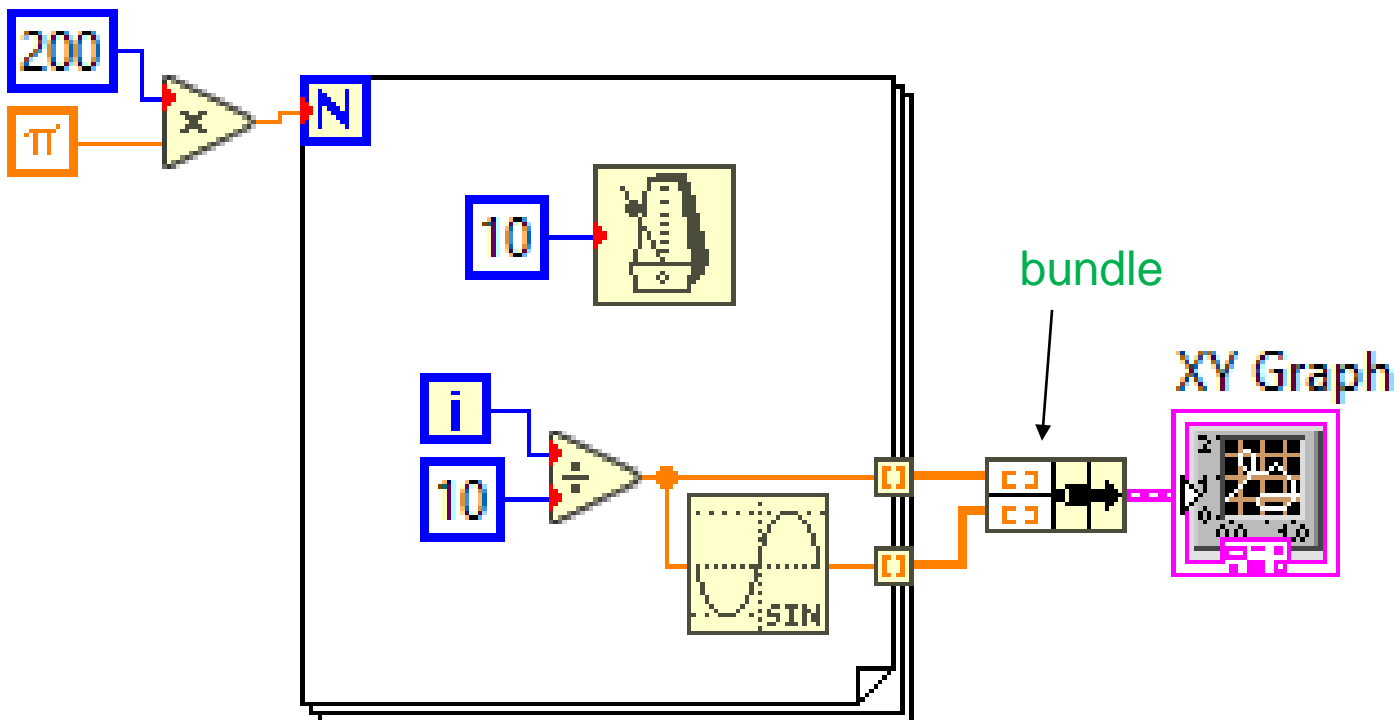


Plotting the Data Array



Wykresy

XY Graphs – wykresy 2D



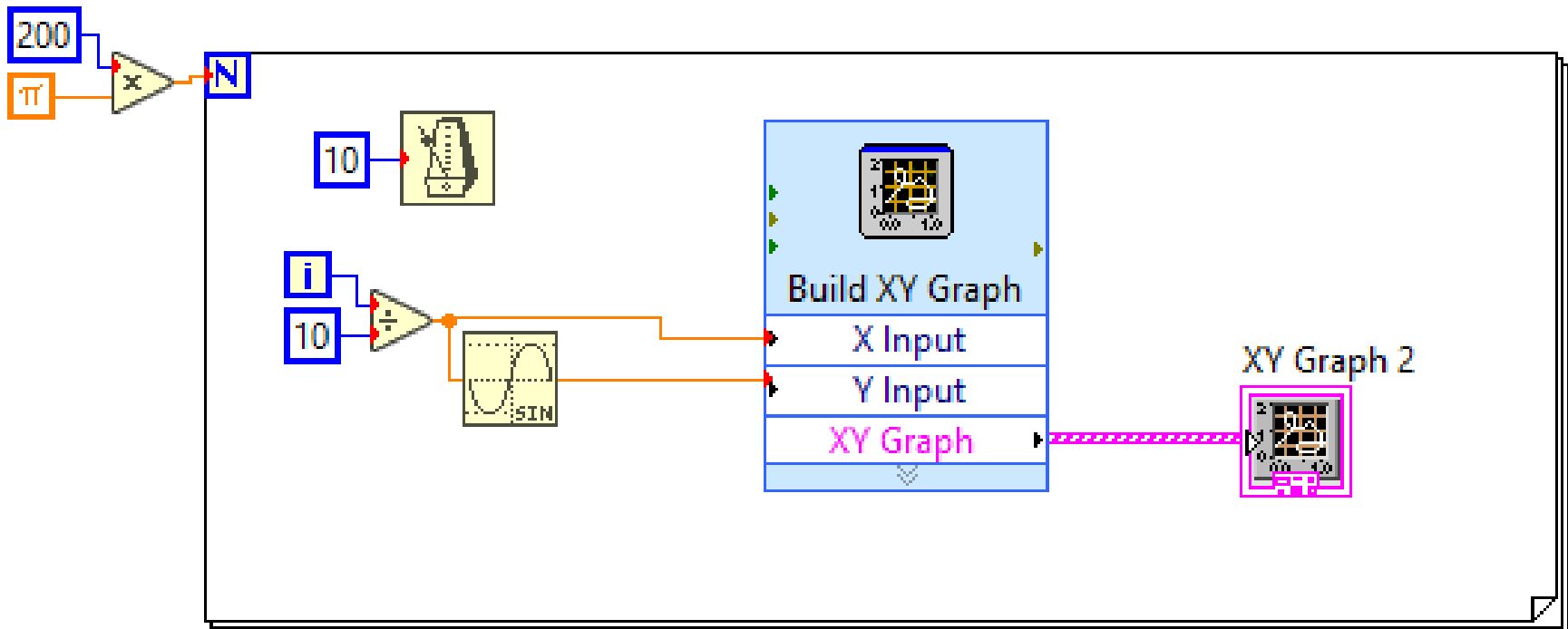
zwykle XY Graph używany jest do konstrukcji całego wykresu – jak wyżej

Wykresy

XY Graphs – wykresy 2D

Można też rysować punkt po punkcie:

- korzystając z XY Graph'u z funkcją Build XY Graph (wstawia się automatycznie gdy wybierzemy XY Graph z zakładki Express FP)

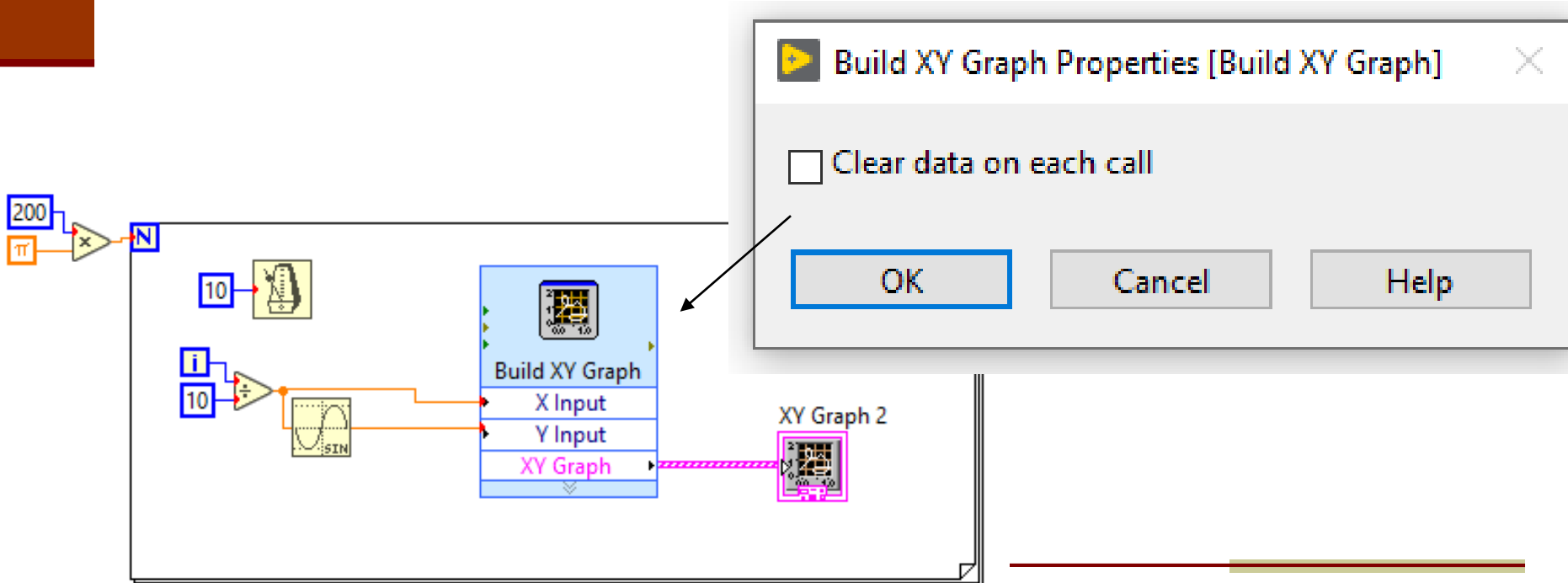


Wykresy

XY Graphs – wykresy 2D

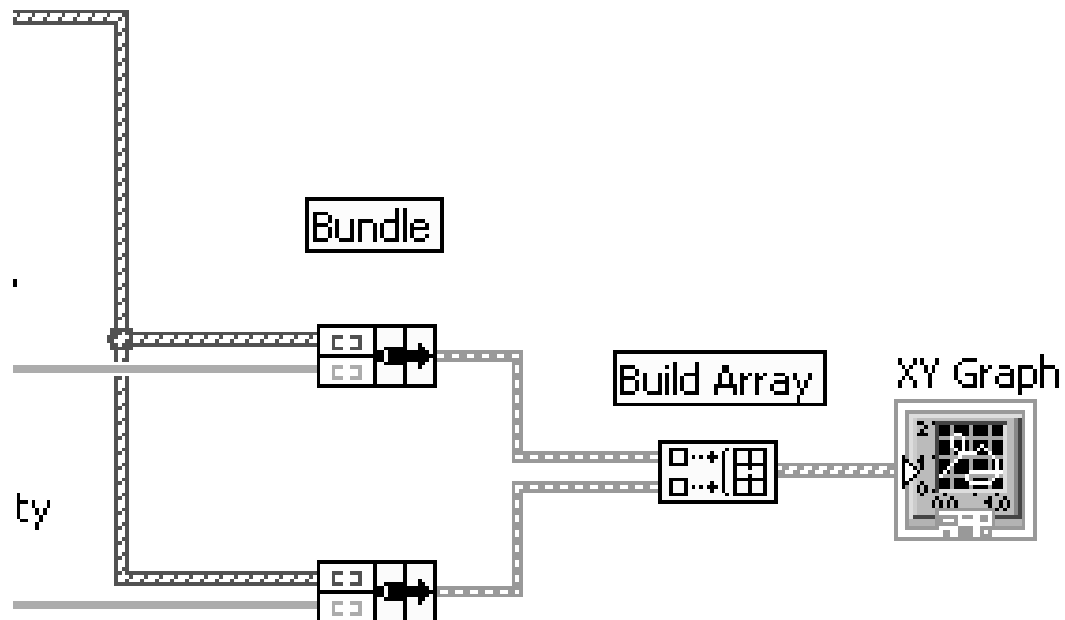
Można też rysować punkt po punkcie:

- trzeba jeszcze w Properties dla „Build XY Graph” odznaczyć „Clear data on each call”



Wykresy

XY Graphs – więcej wykresów



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:

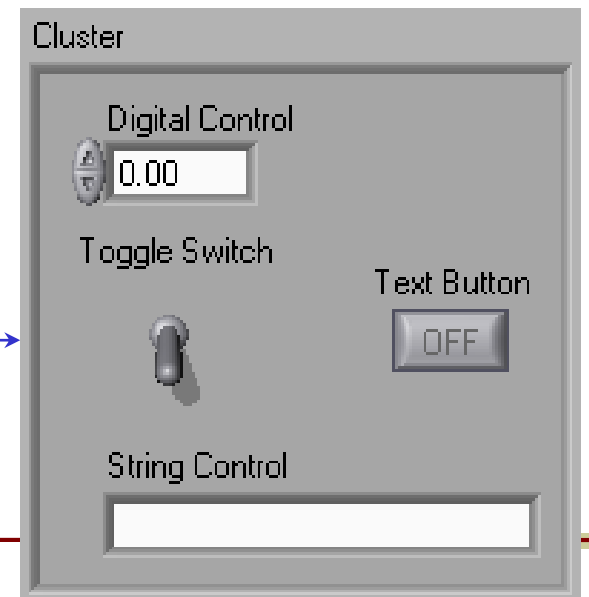
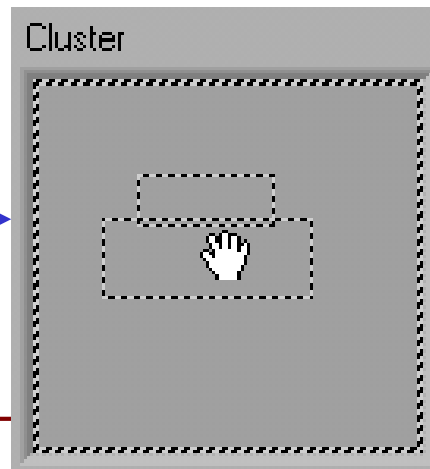
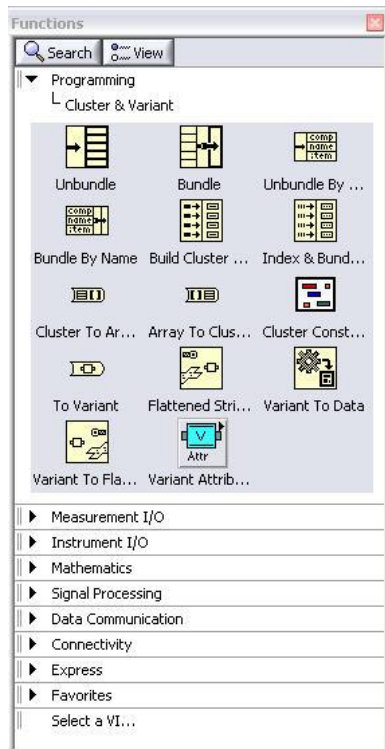
Komunikacja błędów – grupuje: „error flag” typu logicznego, numeryczny - „error code” i łańcuch opisujący błąd.

Macierze i tablice

Wybierz „**Cluster**” shell.

Controls»Modern»Array, Matrix & Cluster

Wstaw obiekty do shell'a.



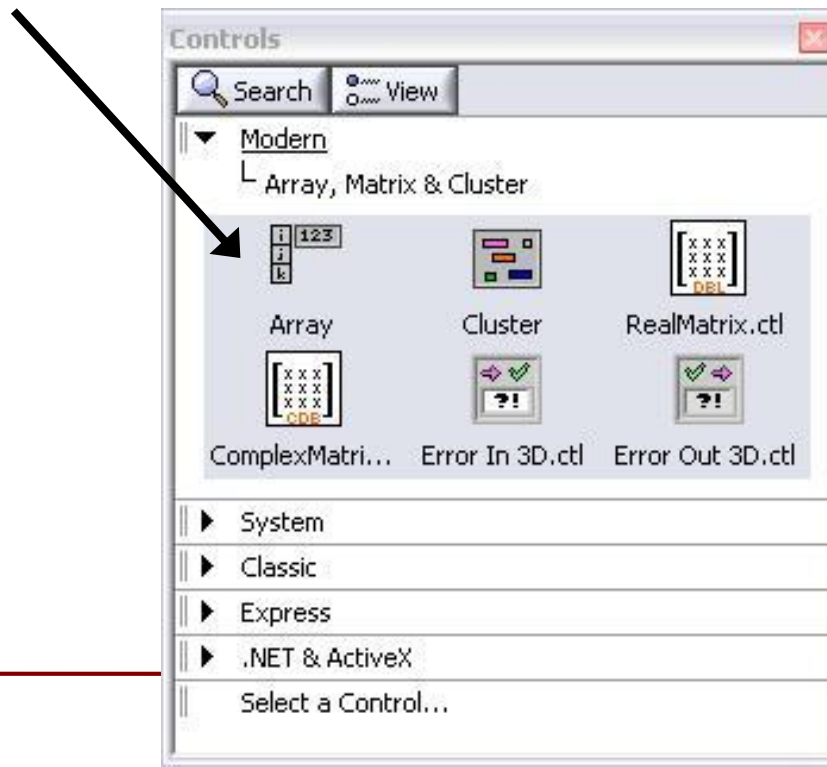
Macierze i tablice

Tablice:

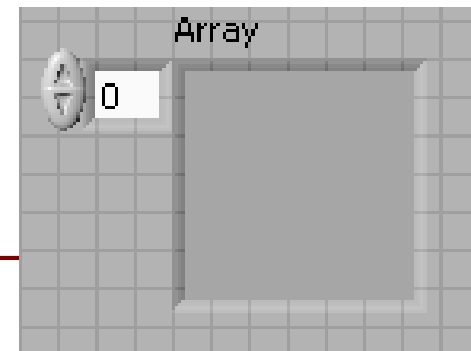
zbiór „kontrolerek” lub „indykatorów” tego samego typu

Budowanie:

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

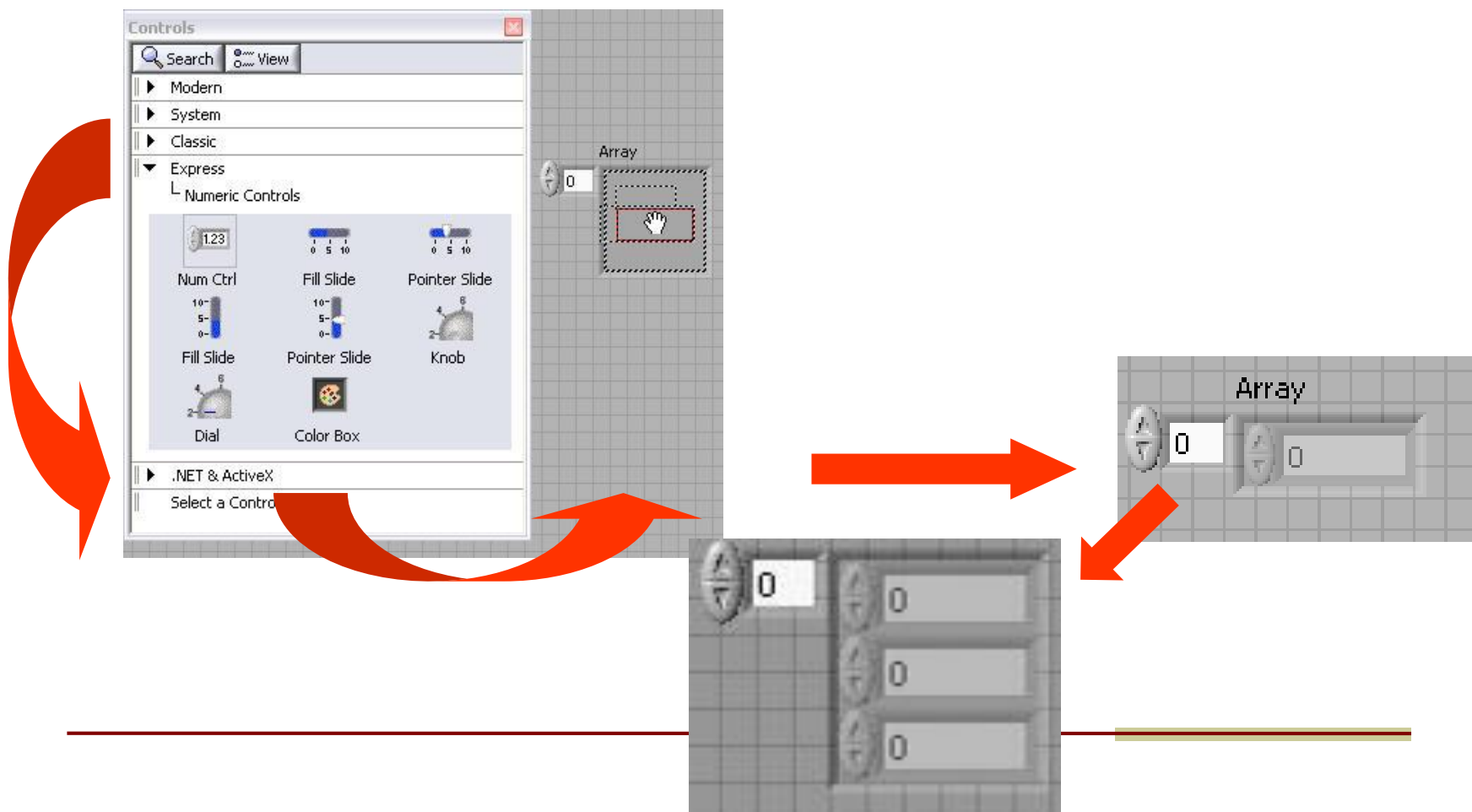


Umieść na Front Panelu



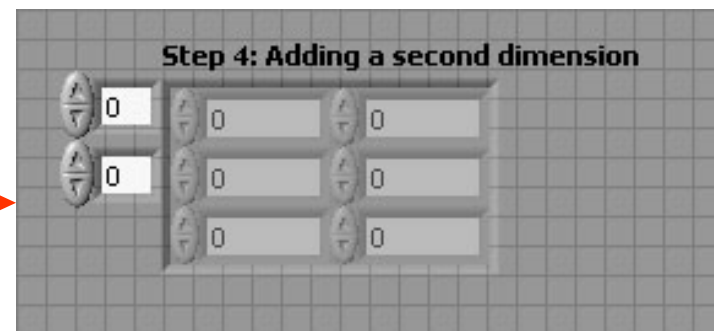
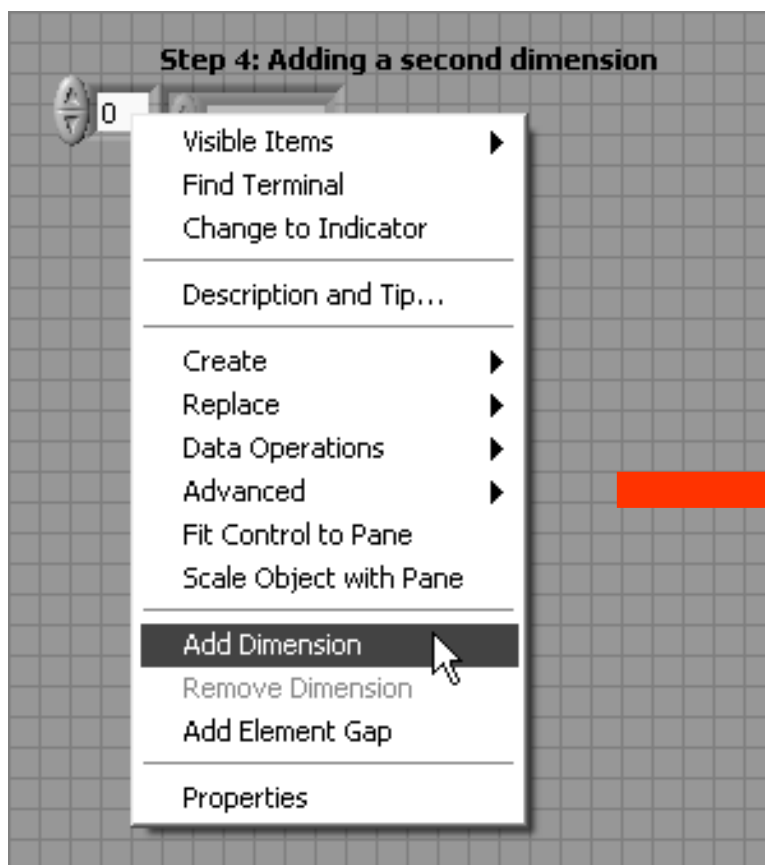
Macierze i tablice

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

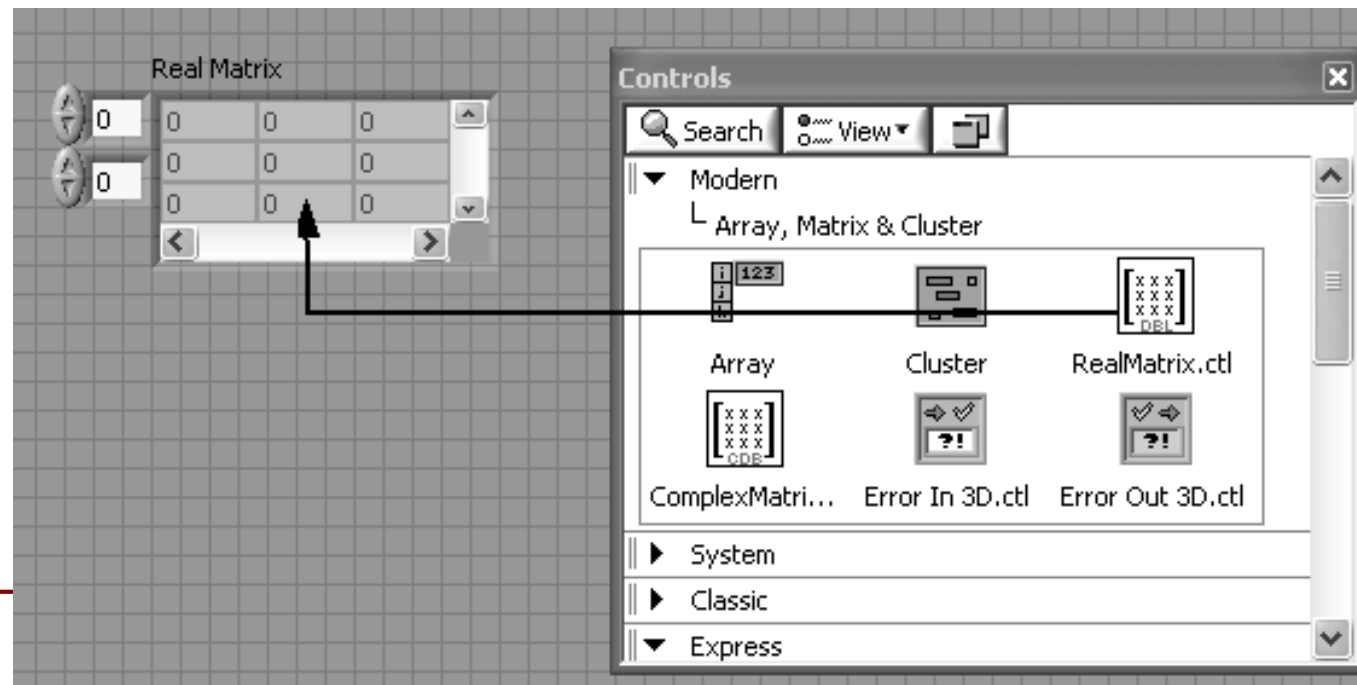
Array functions (Functions Palette / Programming Group / Array Group):

- 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:

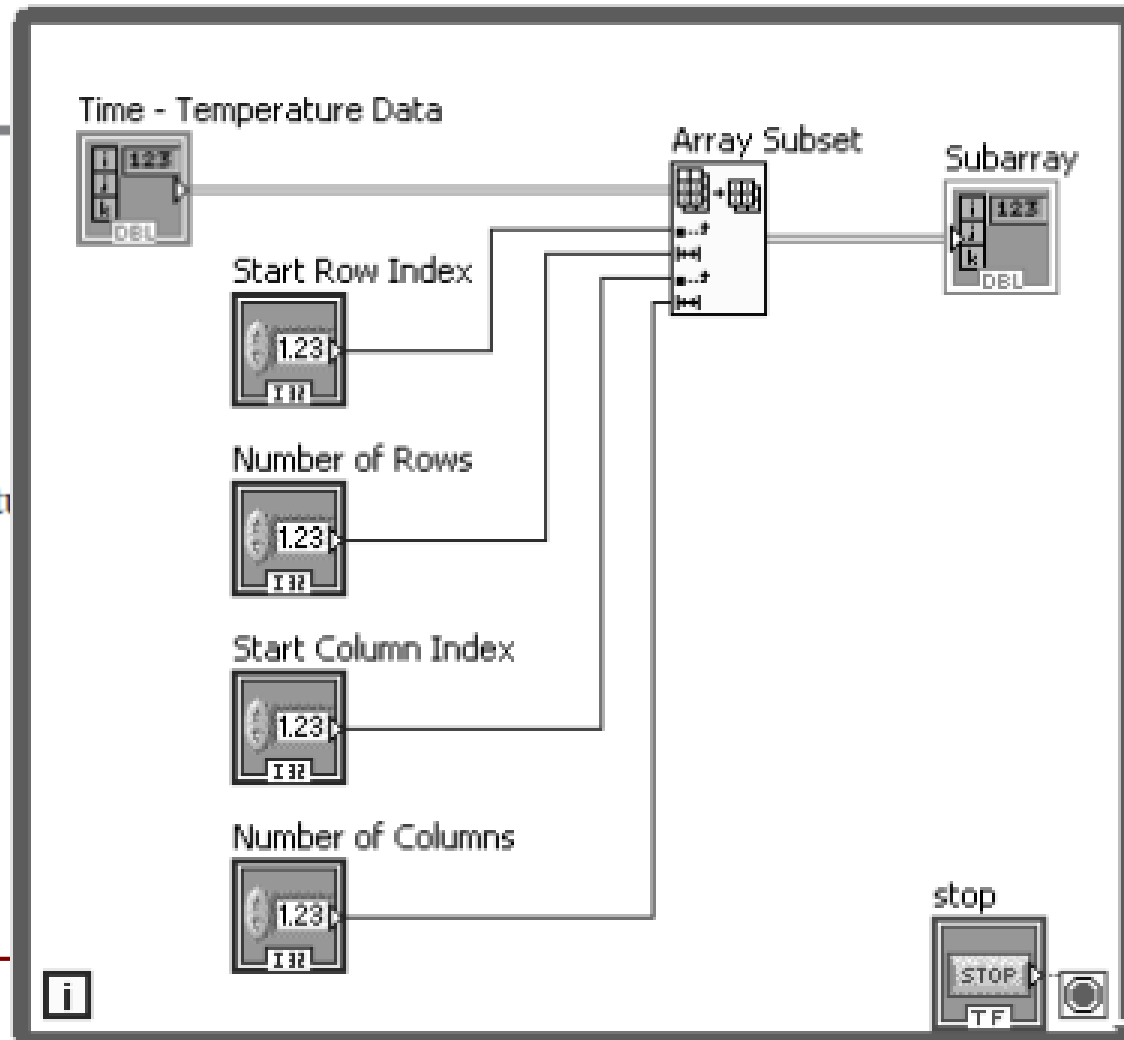
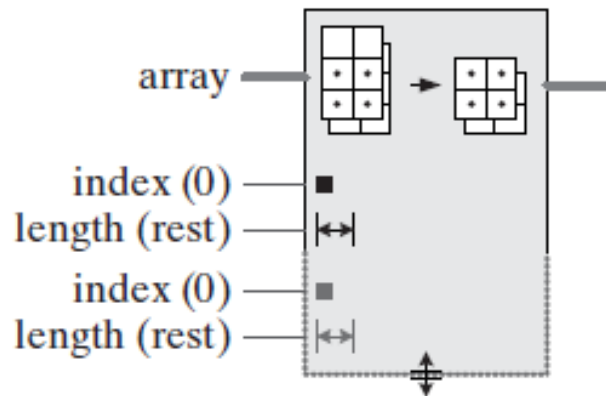
Functions Palette / Programming Group / Array Group /
Array Subset

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

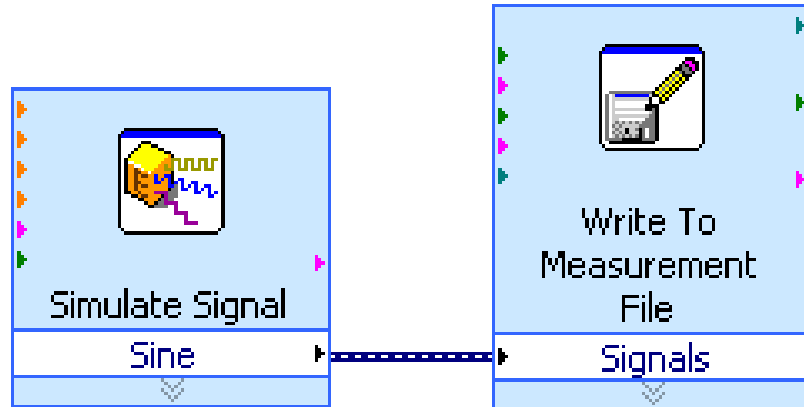


File I/O

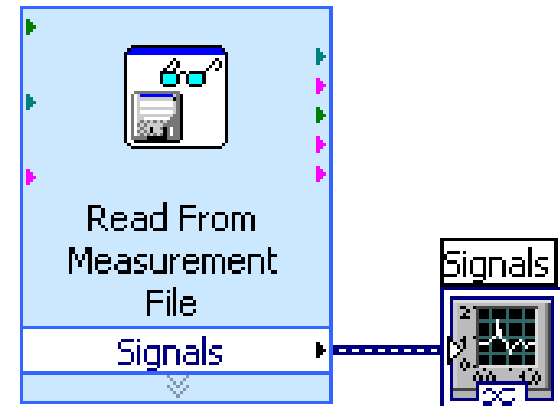
- File I/O – zapis danych do pliku i odczyt danych z pliku.
 - Podstawowe formaty zapisu danych:
 - Binary: liczbowy
 - ASCII: tekstowy
 - LVM: typ LabVIEW zapisu pomiarów (.lvm), plik tekstowy
-

File I/O - LVM

Zapis do pliku LVM



Odczyt z pliku LVM



Przykład

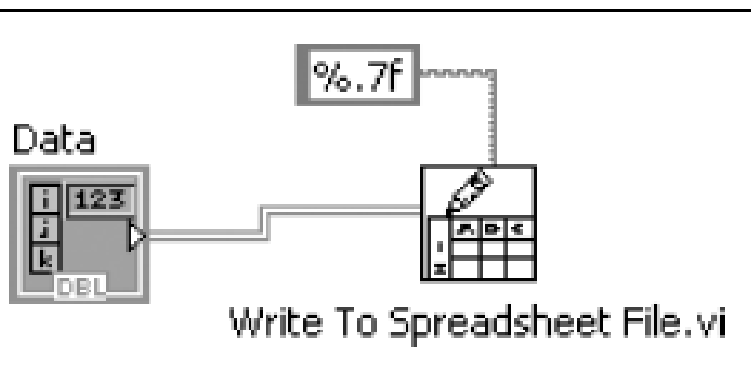
File I/O - LVM

Formaty zapisu

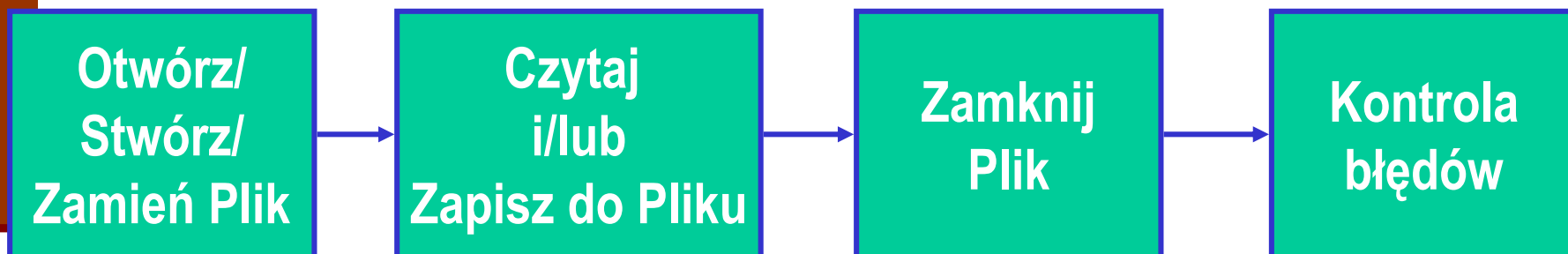
A format string for writing floating point numbers:

- 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) %.3f (default) 12.345
 - e—scientific notation %.7f 12.3450000
 - g—LabVIEW uses “f” or “e” depending on precision %#.7f 12.345

%.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



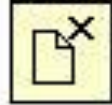
Open/Create/Replace File



Read from Text File



Close File



Simple Error Handler.vi

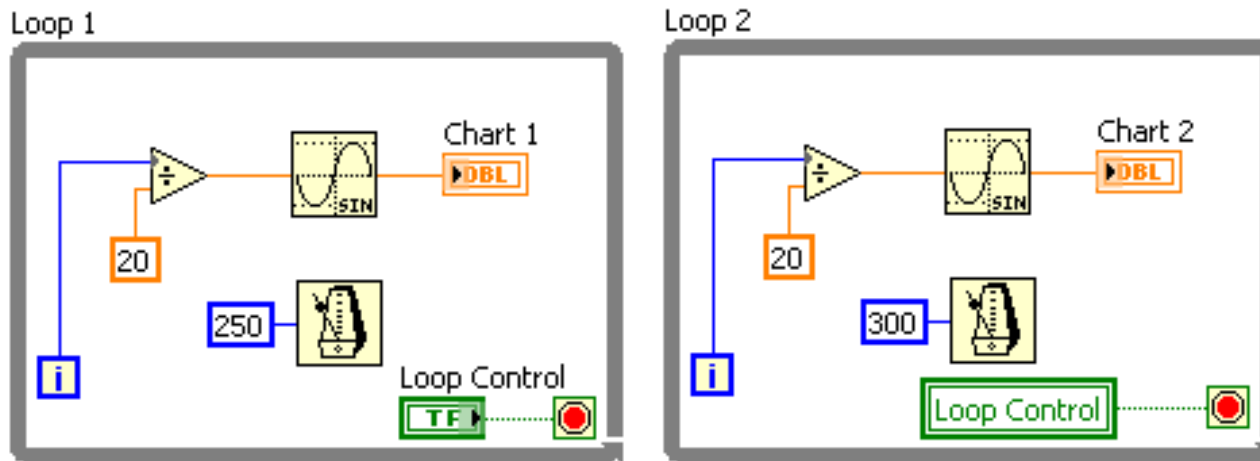


Write to Text File



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

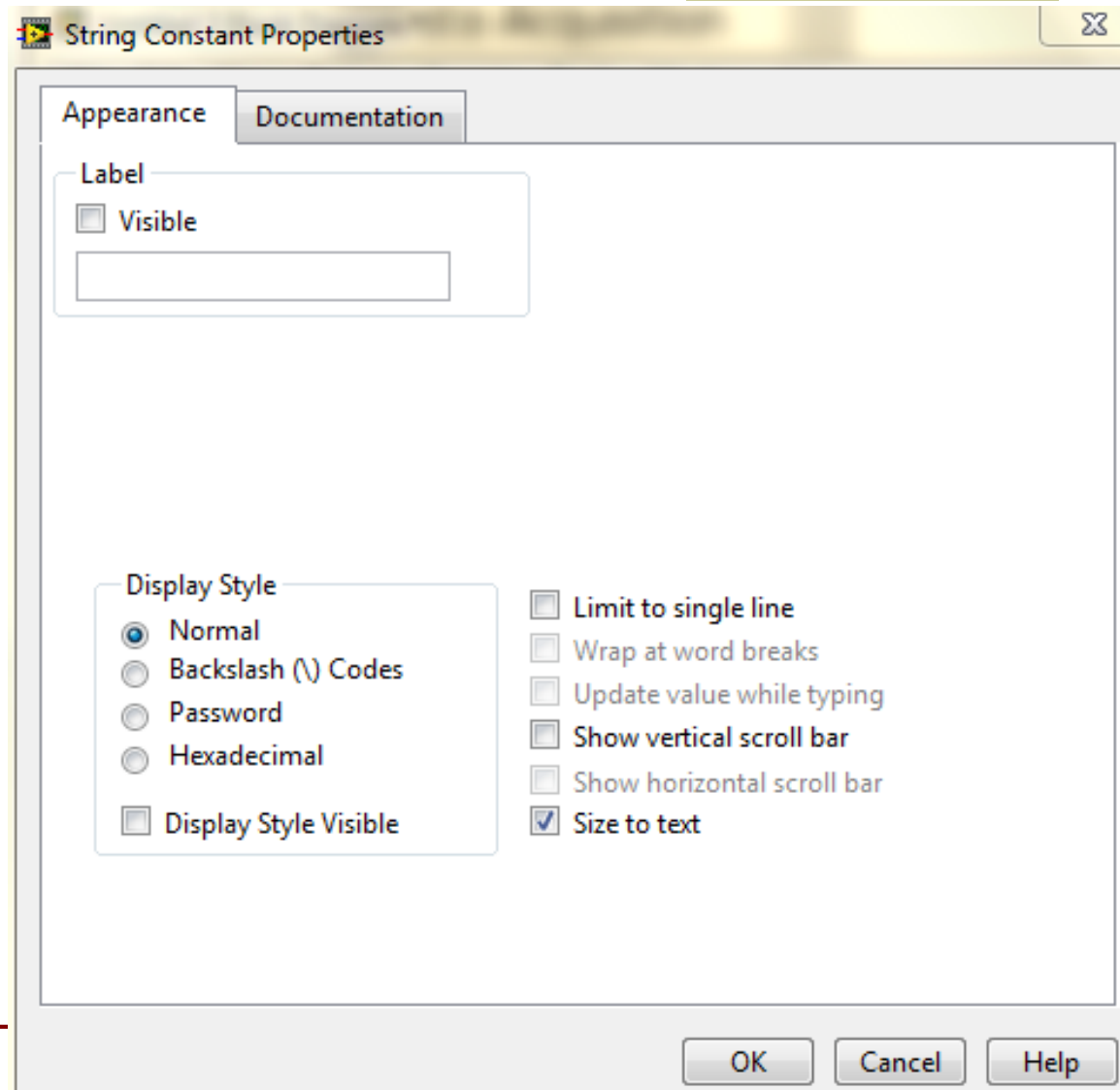
- *Zmienne (Variables)* – elementy BD, które umożliwiają dostęp do danych zdefiniowanych w innym miejscu.
 - *Zmienne Lokalne (Local variables)* przechowują dane na FP (controls i indicators).
 - Zmienne pozwalają na wymianę danych bez potrzeby użycia „drutów”.
-

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



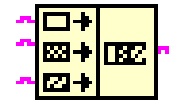
Łańcuchy znaków - Strings

Podstawowe funkcje:

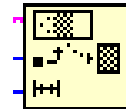
- długość łańcucha – **String lenght**



- łączenie łańcuchów – **Concatenate Strings**



- podłańcuch – **String Subset**





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



Konwersja

Conversion

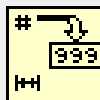
EXT	DBL	SGL	FXP
To Extended ...	To Double Pr...	To Single Pre...	To Fixed-Point
I64	I32	I16	I8
To Quad Inte...	To Long Inte...	To Word Inte...	To Byte Inte...
U64	U32	U16	U8
To Unsigned...	To Unsigned...	To Unsigned...	To Unsigned...
CXT	CDB	CSG	#[...]
To Extended ...	To Double Pr...	To Single Pre...	Number To ...
[...]	?1:0	#→0	[Us]
Boolean Arra...	Boolean To (...)	To Time Sta...	String To Byt...
[Us]	[UNIT]	[...]	
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 ...



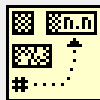
Number To ...



Number To ...



Number To ...



Format Value



Decimal Stri...



Hexadecimal...



Octal String ...

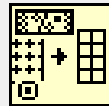


Fract/Exp Stri...

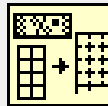


Scan Value

Konwersja



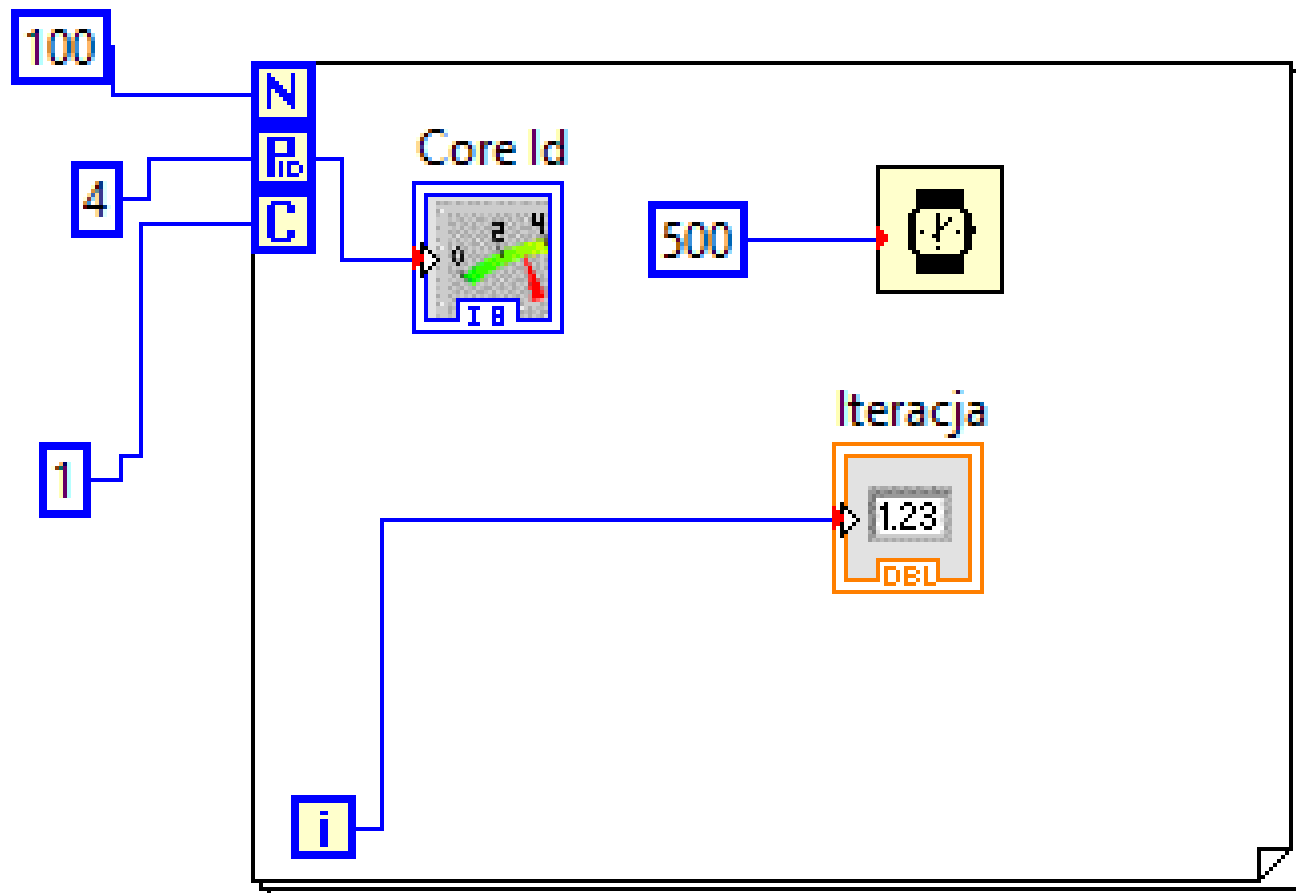
Spreadsheet ...



Array To Spr...

Obliczenia równoległe

- configure iteration parallelism



Funkcje matematyczne



- zaawansowane funkcje matematyczne

NI_Gmath.lvlib:Differentiation.vi

number of points
start
end
formula

Calculates both
derivative of a
at equidistant points

Terminal Data

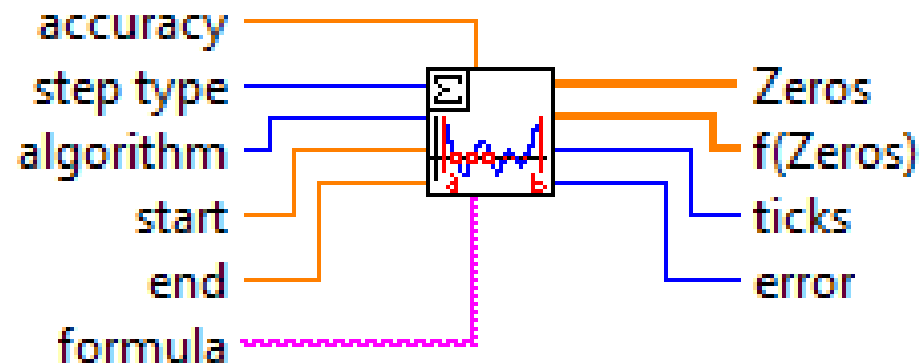
 Derivative of
 (double)

NI_Gmath.lvlib:Integration.vi

start
end
formula

Calculates both
of a 1D function
function is defined
points to be
the given function

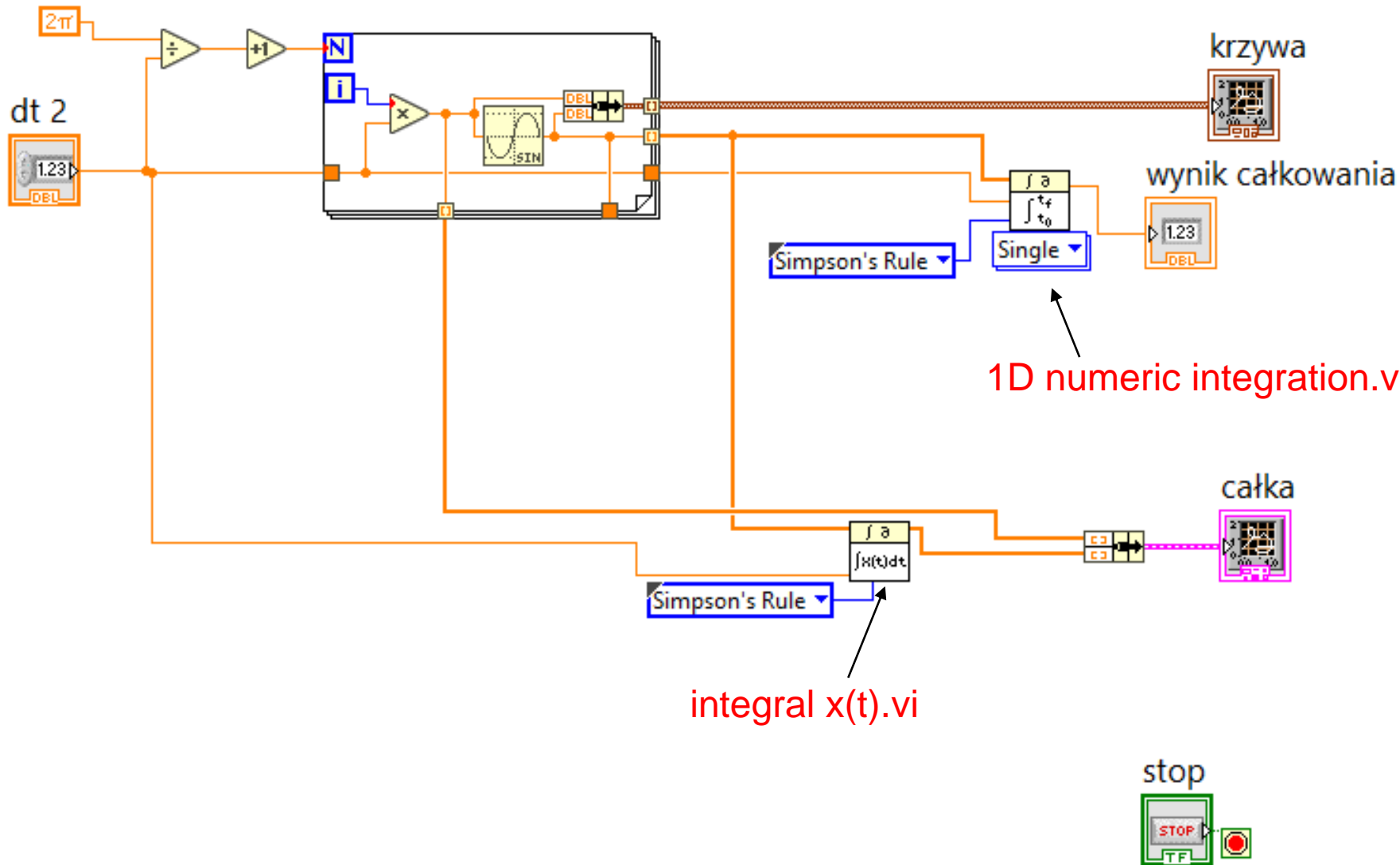
NI_Gmath.lvlib:Find All Zeros of f(x) (Formula).vi



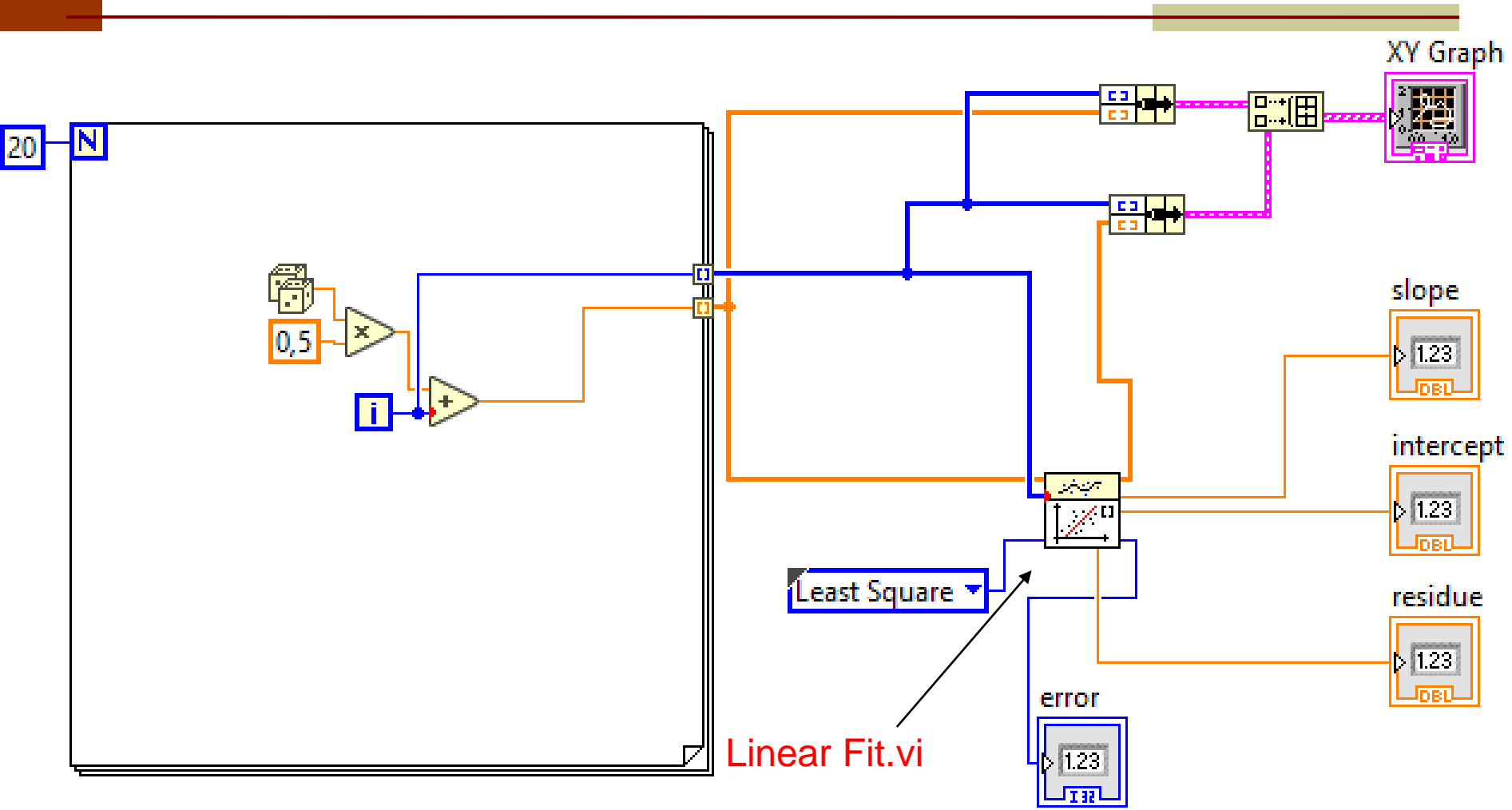
Determines all zeros of a 1D function in a given interval. You must manually select the polymorphic instance to use.

Funkcje matematyczne

2018



Funkcje matematyczne



Funkcje matematyczne

- Mathematics / Differential Equations
 - Ordinary Differential Equations

NI_Gmath.lvlib:ODE Solver.vi

data
ODE F(X,
)
simulation paramete
error in (no erro
h
Solves ordinary differ
conditions of the foll
manually select the p

F(X,t) (right side
Solves ordinary
the Runge Kutta

Terminal Data
ca F(X,t) (right
abc (string)

NI_Gmath.lvlib:ODE Runge Kutta 4th Order.vi

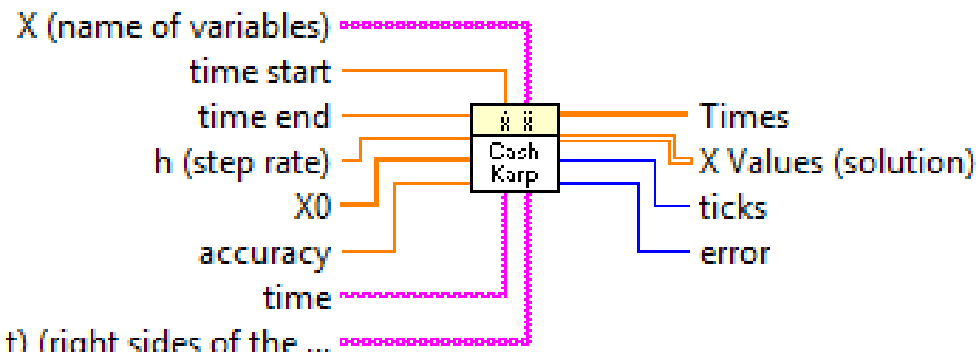
X (name of
h

F(X,t) (right sides of the ...
Solves ordinary differential equations with initial conditions using the

Terminal Data Type

ca X0 (1D array of
dbl (double [64-bit real (~15 digit precision)])

NI_Gmath.lvlib:ODE Cash Karp 5th Order.vi



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

Funkcje matematyczne

- Mathematics / Differential Equations
 - Ordinary Differential Equations

NI_Gmath.lvlib:ODE Euler Method.vi

X (name of variable) NI_Gmath.lvlib:ODE Linear nth Order Numeric.vi

time step

time error

h (step rate)

A (a0,a1,...,an-1)

NI_Gmath.lvlib:ODE Linear nth Order Symbolic.vi

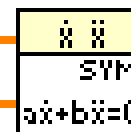
number of points

time

A (a0,a1,...,an-1)

time

X0



formula

error

F(X,t) (right sides of the

Solves ordinary differential equations using the Euler method.

Solves an nth-order differential equation in numeric form.

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

Terminal Data Type

CA X (1D array of

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

Funkcje matematyczne

- Mathematics / Differential Equations
 - Ordinary Differential Equations

NI_Gmath.lvlib:ODE Linear System Numeric.vi

A (matrix of
X
num

NI_Gmath.lvlib:ODE Linear System Symbolic.vi

A (mat

NI_DAE.lvlib:DAE Radau 5th Order.vi

Solves
equati
Solves an n-
differential e
given start c
eigenv
matrix.

Solves
equati
Solves an n-
differential e
given start c
eigenv
matrix.

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

