

**LabWindows/CVI -  
Advance analysis library**  
(použité informačné zdroje National Instruments)

**doc. Ing. Ján Šaliga, PhD.**  
**KEMT FEI TU Košice**

# Čo je Advance analysis library

- Rozšírenie základnej matematickej knižnice Analysis library zo základného balíka CVI o ďalšie matematické funkcie (knižnice nebudú ďalej rozlišované)
- Umožňuje rýchle a efektívne naprogramovanie zložitých matematických algoritmov najmä v oblasti spracovania signálov (Signal processing) a merania (Measurement)
- Funkcie pracujú zväčša s poľami (jedno a dvojrozmernými) reálnych alebo komplexných čísel
- Mnohé umožňujú vykonať operáciu „na mieste“ („in place“, t.j. pole so vstupnými hodnotami ako aj výstupnými hodnotami je to isté = pôvodné hodnoty sa pri výpočte prepíšu výsledkom)
- Väčšina funkcií vracia hodnotu Status, ktorý indikuje úspešnosť vykonania alebo príznak chyby výkonu funkcie

# Prehľad funkcií v triede Signal Generation

ArbitraryWave - generuje ľubovoľný priebeh daný tabuľkou

ChirpGenerates - chirp signál.

GaussNoiseGenerates pole náhodných čísel s Gausovským rozdelením, nulovou strednou hodnotou a zadanou smerodajnou odchýlkou

ImpulseGenerates – pole s jediným nenulovým prvkom.

PulseGenerates – pole so skupinou nenulových prvkov.

RampGenerates – pole s jedinou pílou (priamka).

SawtoothWaveGenerates – pole s periodickou pílou.

SincGenerates – vzorky sinc funkcie.

SinePatternGenerates – sínusový priebeh s daným počtom periód.

SineWaveGenerates - sínusový priebeh s danou normalizovanou frekvenciou.

SquareWaveGenerates – pravouhlý priebeh s danou normalizovanou frekvenciou.

TriangleGenerates - trojuholníkový priebeh s daným počtom periód.

TriangleWaveGenerates - trojuholníkový priebeh s danou normalizovanou frekvenciou.

UniformGenerates – pole s náhodným rovnomerným rozdelením hodnôt v rozsahu 0-1

WhiteNoiseGenerates pole s náhodným rovnomerným rozdelením hodnôt v rozsahu -amp, +amp

# Funkcie z triedy Array operations

- Podtriedy:
  - 1D operations – operácie s jednorozmernými poľami (vektory), napr. kopírovanie, aritmetické operácie, škálovanie, triedenie a vyhľadávanie a ďalšie
  - 2D operations – dvojrozmerné polia (matice), napr. aritmetické operácie, škálovanie, vyhľadávanie a ďalšie
-

# Funkcie z triedy Complex operations

## Podtrieda Complex Numbers

- Complex Addition CxAdd
- Complex Subtraction CxSub
- Complex Multiplication CxMul
- Complex Division CxDiv
- Complex Reciprocal CxRecip
- Complex Square Root CxSqrt
- Complex Logarithm CxLog
- Complex Natural Logarithm CxLn
- Complex Power CxPow
- Complex Exponential CxExp
- Rectangular to Polar ToPolar
- Polar to Rectangular ToRect

## Podtrieda 1D Complex Operations

- 1D Complex Addition CxAdd1D
- 1D Complex Subtraction CxSub1D
- 1D Complex Multiplication CxMul1D
- 1D Complex Division CxDiv1D
- 1D Complex Linear Evaluation CxLinEv1D
- 1D Rectangular to Polar ToPolar1D
- 1D Polar to Rectangular ToRect1D

# Funkcie z triedy Signal processing

Podtriedy:

- Frequency Domain Analysis (priame a spätné transformácie)
  - FFT Tables
- Time Domain Analysis (konvolúcia, korelácia, integrál, derivácia, ...)
- IIR Digital Filters
  - Cascade Filter Functions (výpočet koeficientov s následným aplikovaním na signál cez `IIRCascadeFiltering(x, n, filterInfo, y);` )
    - Filter Information Utilities
  - One-step Filter Functions (jedna spoločná funkcia pre výpočet koeficientov aj samotnú aplikáciu na signál)
  - Old-Style Filter Functions
- FIR Digital Filters – výpočet koeficientov, aplikácia na signál cez funkciu `Convolve(coef, n, x, m, y);`
- Windows

# Funkcie z triedy Measurement

- AC/DC Estimator [ACDCEstimator](#)
- Amplitude/Phase Spectrum [AmpPhaseSpectrum](#)
- Auto Power Spectrum [AutoPowerSpectrum](#)
- Cross Power Spectrum [CrossPowerSpectrum](#)
- Impulse Response [ImpulseResponse](#)
- Network Functions [NetworkFunctions](#)
- Power Frequency Estimate [PowerFrequencyEstimate](#)
- Scaled Window (Obsolete) [ScaledWindow](#)
- Spectrum Unit Conversion [SpectrumUnitConversion](#)
- Transfer Function [TransferFunction](#)
- Total Harmonic Distortion [HarmonicAnalyzer](#)

# Funkcie a podtriedy z triedy Statistics

- Basics – výpočty základných štatistických charakteristík, napr. stredná hodnota, disperzia, rôzne momenty, medián, modus, ...
- Probability Distributions – výpočet distribučných funkcií a inverzných distribučných funkcií rôznych rozdelení
- Analysis of Variance – testovanie hypotéz (viacrozmerné rozdelenia)
- Nonparametric Statistics – chybové funkcie, kontingenčná tabuľka



# Funkcie z triedy Curve fitting

Curve Fitting – prekladanie bodov rôznymi funkciami (aproximácia) na základe rôznych kritérií

- Podtrieda Intervals obsahuje funkcie na výpočet intervalov spoľahlivosti pre rôzne aproximácie

# Funkcie z triedy Interpolation

- Interpolation – funkcie pre výpočet – odhad hodnôt medzi bodmi nejakej závislosti známej iba v niektorých bodoch

# Funkcie z triedy Vector & Matrix Algebra

- Podtriedy:
  - Real Matrices – operácie s maticami, zloženými iba z reálnych čísel, napr. determinat, inverzná matica, transpozícia, násobenie matíc, rôzne charakteristické hodnoty matíc, ...
  - Complex Matrices – podobné operácie ale s komplexnými maticami

# Funkcie z triedy Additional Numerical Methods

Rôzne ďalšie numerické funkcie ako napr. výpočet komplexných koreňov, numerický integrál, detektor špičky v signále, prahový detektor, ...

- Podtrieda Special Functions obsahuje najmä výpočty hodnôt vyšších transcendentných funkcií ako sú napr. Besselove, Gamma, Integralsinus, ...

# Funkcie Free Analysis Memory a Get Error String

```
main()
{
double *x = NULL, *amplitudes = NULL, *locations = NULL, *secondDerivatives = NULL, err = 0, xSize;
char * opis_chyby;
/* Insert code here to determine xSize. */
x = (double *)malloc (xSize * sizeof(double));
err = PeakDetector(x, xSize, 0.01, 3, 0, 1, 0, &count, &locations, &amplitudes, &secondDerivatives);
/* Memory is allocated internally by the PeakDetector function for the locations, amplitudes and
second_derivatives outputs. Use the FreeAnalysisMem function to free this memory. */
FreeAnalysisMem(locations);
FreeAnalysisMem(amplitudes);
FreeAnalysisMem(secondDerivatives);
opis_chyby = GetAnalysisErrorString (err);
// vypis chyby vo forme stringu
}
```

# Poznámka – formáty komplexných čísel

1. Pole komplexných čísel sú vlastne dve polia obsahujúce reálnu a imaginárnu zložku alebo absolútnu hodnotu a fázu. Používa sa hlavne u jednoduchých funkcií z Signal processing
2. V triede Vector & Matrix Algebra ale aj napr. v nových funkciách pre FFT sú komplexné čísla vo forme poľa obsahujúcu štruktúry v tvare

```
typedef struct {  
    double real;  
    double imaginary;  
} ComplexNum;
```