Katalóg požiadaviek

Autori: Martina Bodišová, Tomáš Bordáč, Michal Chamula, Patrik Fašang

Zadávateľ: Mgr. Peter Čermák, PhD.

# 1. Úvod

## 1.1 Podstata dokumentu

Tento dokument popisuje požiadavky  zadávateľa na softvér vyvíjaný v projekte Spracovanie spektroskopických dát.

## 1.2 Určenie dokumentu

Tento dokument je určený stakeholderom, ktorí na základe popísaných požiadaviek budú softvér vyvíjať. Finálna verzia tohoto dokumentu je odsúhlasená zadávateľom.

## 1.3 Rozsah Systému

Projekt je dynamicky linkovaná knižnica (DLL), ktorá slúži na rátanie komplexných, neanalytických funkcií. Neobsahuje grafické užívateľské rozhranie.

## 1.4 Slovník pojmov

* **DLL** -(angl. Dynamic Link Library) je množina malých programov, ktorá môže byť použitá viac ako jedným programom v tom istom čase. Táto množina je zväčša uložená v súboroch so suffixom “.dll“
* **LabVIEW** – (angl. Laboratory Virtual Instrument Engineering Workbench) je vývojové prostredie určené na vizuálne programovanie s podporou čítať DLL knižnice napísané v jazyku c++.
* **Konvolúcia** – matematický operátor spracovávajúci dve funkcie. Je definovaný vzťahom:

## 1.5 Referencie

1. *"Decay time integrals in neutral meson mixing and their efficient evaluation"* - Till Moritz Karbach, Gerhard Raven, Manuel Schiller (CERN - Switzerland, NIKHEF - The Netherlands)
2. *"An isolated line-shape model to go beyond the Voigt profile in spectroscopic databases and radiative transfer codes"* - N.H. Ngo, D. Lisak, H. Tran, J.-M. Hartmann
3. *"Efficient computation of some speed-dependent isolated line profiles"* - H. Tran, N.H. Ngo, J.-M. Hartmann

# 2. Všeobecný popis

## 2.1 Perspektíva projektu

Projekt bude súčasťou väčšieho celku, ktorý má za úlohu analyzovať spektroskopické dáta. Optická Spektroskopia je oblasť fyziky, zaoberajúca sa štúdiom elektromagnetického žiarenia emitovaného alebo pohlteného vzorkou. Získané informácie sa dajú použiť buď kvalitatívne (charakteristika vnútornej štruktúry vzorky, poprípade prostredia kde sa nachádza), alebo kvantitatívne (určenie koncentrácie známej vzorky)

## 2.2 Funkcie produktu

Náš softvér by mal byť schopný v optimálnom čase aplikovať rôzne transformácie na vstupné hodnoty – spektrá a modelovať ich tvar použitím funkcií opisujúcich žiarenie(absorpciu) vzoriek. Medzi tieto funkcie patria: Lorentzova, Gaussova, Voigtova alebo Hartmann–Tran.

## 2.3 Charakteristika používateľov

Finálny produkt bude využívať oddelenie experimentálnej Fyziky FMFI UK. Z používateľského hľadiska bude produkt použiteľný iba pod vývojovým prostredím LabVIEW.

# 3. Špecifické požiadavky obsahu DLL

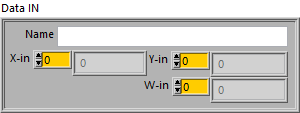
Požiadavky na systém sú, implementovanie vybraných funkcií profilov spektrálnych čiar. Úprava funkcie na transformáciu Y-ovej osi v existujúcej knižnici a výstupu z knižnice do prostredia Labview pridaním výpočtu inverznej funkcie k  Yout.

## 3.0 Načítanie Vstupu a Výstupu

### 3.0.1 Vstup

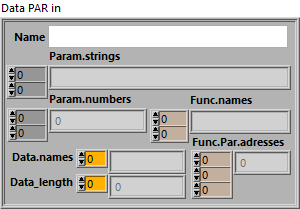
Používateľ po otvorení programu má k dispozícii niekoľko tabuliek, do ktorých môže zadať vstupné hodnoty. Tieto hodnotu vstupujú do DLL, kde sa spracujú. Vstupné formuláre sú zobrazené na obr.1 a obr.2

* Popis vstupného formuláru, Spektrum (obr.1):
  + Polia:
    - **Name:** meno dát
    - **X-in:** x-ová súradnica
    - **Y-in:** y-ová súradnica
    - **W-in:** neistota
  + z Labview: štruktúra (cluster) obsahujúca X,Y,W a meno dát (string)



obr. 1

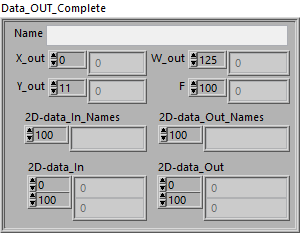
* Popis vstupného formuláru, Parametre (obr. 2)
  + Polia:
    - **Param.strings**: dim 3xM of string: charakterizujúce mená parametrov a ich vlastnosti (meno, model, nezávislé parametre)
    - **Param.values**: dim 3xM of double: charakterizujúce hodnoty parametrov (hodnota, neistota, škála pre GUI)
    - **Func.names**: dim 2xL of string (L je počet transformácií a funkcií modelu – max 6: XT, YT, BL, PK, RF, MC, ktoré treba vypočítať), prvý stĺpec obsahuje meno funkcie (XT, YT, BL, PK, RF, MC), druhý mená „skupín=groups“ v rámci danej funkcie zoradených do jedného stringu: Menno1@Meno2@...@MenoN
    - **Func.par.adresses**: 3xLxQ (Q je max počet skupín nachádzajúci sa niektorej z funkcií), prvý inde(page) definuje funkciu podľa poradia v poli Func.names následne každá skupina má jeden riadok, kde prvá hodnota hovorí koľko hodnôt je v danom riadku (adries – poradie v Param.strings a Param.values), poradie adries parametrov je pevne definované pre každú funkciu-skupinu
    - **Data.names:** Mená vektorov predsimulovaných dát pre funkciu REF
    - **Data.length:** Zodpovedajúca dĺžka(počet bodov) pre vektory predsimulovaných dát (uložené v binárnom súbore na disku)
  + z Labview: štruktúra (cluster) obsahujúca

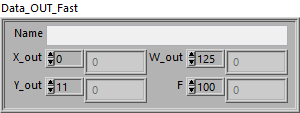


*obr. 2*

### 3.0.2 Výstup

Vstupné dáta boli spracované funkciami napísanými v DLL, ktorých výstupné hodnoty vrátime do výstupných formulárov obr.3, obr.4.

* Upravené spektrum (obr.1) do Data\_out\_Fast (obr.3) Xout , Yout , Wout
* Upravené parametre (obr.2) do Data\_OUT\_Complete
* Výstupné parametre (vráti rovnaké parametre ako dostal na vstupe, s aktuálnymi hodnotami)



*Obr. 3*

*Obr. 4*

## 3.1 spracovanie vstupov

DLL načíta vstupné hodnoty. Následne ich prepočíta podľa zvolenej funkcie a ich výsledky vráti do výstupných tabuliek. (obr.3, obr.4) Funkcie, podľa ktorých DLL vedieť počítať výsledné dáta sú v tejto stati rozpísané podrobnejšie.

### 3.1.1 Transformácia X-ovej súradnice (XT)

Vezme zo vstupu hodnotu zadanú X-in (z obr.1), nasledovným vzorcom ju prepočíta a výsledok vráti používateľovi do tabuľky Data\_OUT\_Fast a Data\_OUT\_Complete

**Xout** = XOff + XScl(Xin)

Kde:

**Xin**= vstupná súradnica

**XOff**= p0konštantná funkcia

**XScl**= polynomická funkcia kde p0 reprezentuje fixný bod transformácie

### 3.1.2 Transformácia Y-ovej súradnice (YT)

Vezme zo vstupu hodnotu zadanú Y-in (z obr.1), nasledovným vzorcom ju prepočíta a výsledok vráti používateľovi do tabuľky Data\_OUT\_Fast a Data\_OUT\_Complet.

**Yout** = **YOff**(Xin) + **YScl**(Xin) \* **YTyp**{Yin, **YPol**(Xin) + **YTrg**(Xin) + **YSpl**(Xin)}

Kde:

* **Yin s**= vstupná premenná
* **YOff** = polynóm typu *p-type* s parametrami pi.
* **YScl**(Xin) = polynóm typu *p-type* s parametrami pi.
* **YPol** = polynóm typu *p-type* s parametrami pi.
* **YTrg** = TRIG(t-type){pi, pi+1, pi+2, Xin} trigonometrická funkcia typu *t-type* s parametrami p3i.
* **YSpl** = 3DSPLINE{n x (pi, pi+1) Xin} kubická krivka prechádzajúca cez body definované ako n párov (pi, pi+1)
* **YTyp** funkcia ktorá definuje typ operácie ktorá ma byť vykonaná na sume YPol, YTrg a YSpl (POL+TRG+SPL=I0; Y\_in\*I0, Y\_in/I0, 1-(Y\_in/I0), -ln(Y\_in/I0))
* **p-type** definuje typ polynomiálnej funkcie:
  + 0 = štandardný polynóm
  + 1 = Ledenrov polynóm 1ho rádu
  + 2 = Chebyshevov polynóm 1ho rádu

1. **t-type** definuje typ trigonometrickej funkcie:
   * 0 = pi\*sin(2\*π\*pi+1 + pi+2)
   * 1 = pi\*cos(2\*π\*pi+1 + pi+2)

### 3.1.3 funkcia doppler počíta dopplerov algorytmus podľa vzorca

FD(v - v0) = exp( -ln(2) ())

* Vstupom funkcie doppler je jeden argument: double v
* Argument môže byť akékoľvek číslo v množine reálnych čísel
* Výstupom funkcie je double

### 3.1.4 funkcia lorentz počíta lorentzov algorytmus podľa vzorca

FL(v - v0) =

* Vstupom funkcie lorentz je jeden argument: double v
* Argument môže byť akékoľvek číslo v množine reálnych čísel
* Výstupom funkcie je double

### 3.1.5 funkcia voigt je konvolúcia Lorentz a Gauss profilu

1. Lorentz profil:

* Vstupom funkcie voigt sú 2 argumenty: double x, double gama
* Argumenty môžu byť akékoľvek čísla v množine reálnych čísel
* Výstupom funkcie je double

1. Gauss profil:

* Vstupom funkcie voigt sú 2 argumenty: double v, double sigma
* Argumenty môžu byť akékoľvek čísla v množine reálnych čísel
* Výstupom funkcie je double

1. Ich konvolúcia sa vypočíta podľa vzorca:

* Vstupom funkcie voigt sú 3 argumenty: double v, double sigma, double gama
* Argumenty môžu byť akékoľvek čísla v množine reálnych čísel
* Výstupom funkcie je double

### 3.1.6 funkcia htp počíta Hartmann-tran algoritmus podľa vzorca:

FHTP(v)= Re()

* Kde A(ν) a B(ν) vieme určiť ako kombinácie funkcie pravdepodobnosti.
* Vstupom funkcie voigt sú 3 argumenty: double v, double sigma, double gama
* Argumenty môžu byť akékoľvek čísla v množine reálnych čísel
* Výstupom funkcie je double