# Katalóg požiadaviek

# Spetroskopické dáta

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

**Zadávateľ**: Mgr. Peter Čermák, PhD.

# Úvod

## Podstata dokumentu

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

## Určenie dokumentu

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

## Rozsah Systému

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

## Slovník cudzích 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“
* **Stakeholder-** osoba alebo skupina osôb, ktorá sa podieľa na rovnakom projekte, napr. podnikaní, programovaní, vede a pod.
* **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:

## 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

# Všeobecný popis

## 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).

## Funkcie produktu

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.

## Charakteristika používateľov

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

# Špecifikácia požiadaviek

# Načítanie vstupného súboru

* + 1. Systém prečíta vstupný súbor a dáta poukladá do objektov v DLL a pripraví tak namerané dáta na ďalšie spracovanie. Načítavanie dát je už implementované v knižnici DLL, ktorú sme zdedili. LabView načítané dáta zobrazí vo formulároch a nijak ich nemení. Umožňuje tieto dáta len prezerať, prípadne zvoliť transformačnú funkciu, ktorou sa tieto hodnoty prepočítajú a zobrazí počiatočný stav hodnôt, ktoré sme načítali so súboru v jednom formulári a zmenené dáta zvolenou funkciou v druhom formulári.
  1. Zobrazenie načítaných vstupov
     1. Prostredie LabView ponúka dva formuláre súvisiace s načítanými dátami.
     2. Názov prvého formulára je **Data IN** (takto je nazvaný v LabView), ktorý ponúka používateľovi zadať:
        1. **Name** - názov datasetu (merania) napr. pondelok, utorok , …
        2. Hodnotu **X-in**, kde si zvolíme z ktorého indexu chceme zobraziť X-ovú súradnicu
        3. Hodnotu **Y-in**, kde si zvolíme z ktorého indexu chceme zobraziť Y-ovú súradnicu
        4. Hodnotu **W-in**, kde si zvolíme z ktorého indexu chceme zobraziť W (chybu merania)
     3. Názov druhého formulára je **Data PAR In** (názov v LabView), ktorý ponúka používateľovi zadať:
        1. **Name** - názov datasetu (merania) napr. pondelok, utorok , … (mal by byť rovnaký ako v bode 2.1.1.1)
        2. **Func.names** - umožní používateľovi zvoliť funkciu, ktorou sa majú vstupné hodnoty upraviť. Tieto hodnoty sú z popísané v bodoch 2.1.1.2, 2.1.1.3 a 2.1.1.4
  2. Výstupy do LabView
     1. Používateľ má k dispozícii v LabView ďalší formulár, v ktorom sa zobrazia transformované dáta. Keďže dáta sú uložené v poliach, tak sa zobrazujú iba z jedného indexu, ktorý môže používateľ meniť a tak si prezrieť postupne všetky hodnoty. Po načítaní súboru budú rovnaké ako vo formulári popísanom v časti 2.1.1. Až keď používateľ zvolí funkciu a stlačí tlačidlo na transformovanie dát, tieto dáta vo formulári prepočíta podľa zvolenej funkcie a vypíše ich znovu do tohto formulára. Nasledujúci popis počíta s tým, že už bola zvolená funkcia a stlačené tlačidlo na transformáciu dát.
     2. Názov formulára v LabView je **Data\_OUT\_Fast**, ktorý obsahuje kolónky:
        1. **Name** - názov datasetu (rovnaký ako v bode 2.1.1.1 a nebude sa meniť)
        2. Hodnotu **X**\_**Out**, kde si zvolíme z ktorého indexu chceme zobraziť X-ovú súradnicu. Táto súradnica je už zmenená zvolenou funkciou.
        3. Hodnotu **Y**\_**Out**, kde si zvolíme z ktorého indexu chceme zobraziť Y-ovú súradnicu. Táto súradnica je už zmenená zvolenou funkciou.
        4. Hodnotu **W**\_**Out**, kde si zvolíme z ktorého indexu chceme zobraziť W (chybu merania). Táto chyba merania je stále rovnaká (ako v bode 2.1.1.4). Teda ju DLL nebude nijako meniť.

1. Komunikácia s DLL

DLL bude priamo komunikovať s LabView len cez funkciu fast(), ktorá načíta vstupný súbor. Dáta poukladá do štruktúr a nezmenené ich zobrazí v LabView. Používateľ bude môcť zvoliť funkciu, ktorou bude chcieť dáta transformovať/zmeniť. Ďalej sa budú dáta spracovávať podľa zvolenej funkcie.

1. Funkcie

Funkcie, ktoré si bude môcť používateľ zvoliť sú transformácia x-ovej osi **xt()**, y-ovej osi **yt()**, **gauss()**, **lorentz()**, **voigt()** a hartmann-tran **htp()**. Tieto všetky funkcie budú vždy počítať len s jednou hodnotou x, y, w, ktoré máme načítané v štruktúre (sú zobrazené vo formulári popísanom v bode 2.1.1). Ďalšie parametre, ktoré sú potrebné na výpočet funkcie sú načítané zo súboru a uložené do poľa, obsahujúceho štruktúry Parameters. Štruktúra obsahuje parametre pre všetky funkcie, ale vyplnené sú len tie, ktoré daná funkcia potrebuje (potrebné parametre sú popísané v časti 6. pre jednotlivé funkcie). Výsledné hodnoty X a Y sa zapíšu do nového dvojrozmerného poľa. Výpočet sa vykoná pre každú X-ovú a Y-ovú hodnotu, ktoré vstupujú do funkcií ako vektor.

1. Popis jednotlivých funkcií

Popis funkcií, ktoré si môže používateľ zvoliť. Nie sú tu popísané funkcie **xt()** a **yt()**, pretože už sú implementované v zdedenom DLL.

* 1. Gauss

Gauss() bude počítať postupne s každou hodnotou X, Y. Tieto hodnoty tvoria vektor v, ktorý vstupuje do funkcie. Parametre pre , , načíta so svojej sady parametrov. Výsledkom bude vektor so zmenenými hodnota X a Y, ktoré budú uložený do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšie načítaného X, po najvyššie načítané X.

* 1. Lorentz

lorentz() bude počítať postupne s každou hodnotou X, Y Tieto hodnoty tvoria vektor v, ktorý vstupuje do funkcie. Parametre pre , , načíta so svojej sady parametrov. Výsledkom bude vektor so zmenenými hodnota X a Y, ktoré budú uložený do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšie načítaného X, po najvyššie načítané X.

P(v - v0) =

* 1. Voigt

Voigt() bude počítať postupne s každou hodnotou X, Y. Tieto hodnoty tvoria vektor v, ktorý vstupuje do funkcie.Parametre pre , , w0, načíta so svojej sady parametrov. Výsledkom bude vektor so zmenenými hodnota X a Y, ktoré budú uložený do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšie načítaného X, po najvyššie načítané X.

* 1. Hartmann-tran

Hartmann-tran bude počítať postupne s každou hodnotou X, Y. Tieto hodnoty tvoria vektor v, ktorý vstupuje do funkcie. Parametre pre, načíta so svojej sady parametrov. Pričom je **Z** komplexné číslo, preto do výsledku prepočítaných hodnôt vektora vezmeme do úvahy iba reálnu časť. Výsledkom bude vektor so zmenenými hodnota X a Y, ktoré budú uložený do poľa výsledkov na rovnakom indexe, ako boli pôvodné hodnoty X, Y. Definičný obor je od najmenšieho načítaného X, po najvyššieho načítaného X.

FHTP(v)= Re