# 1 Hinweise zur Inbetriebnahme des Programms

## Download

Der Download der Software besteht aus einer ausführbaren Setup-Installationsprogramm-Zip-Datei:

Setup\_UR2\_vers\_2.4.xx\_64bit.exe (mit variabler Versionsnummer: ca. 25,0 MB)

Dies kann in Windows-Versionen mit X64-Architektur (64bit) verwendet werden.

Es werden folgende Dateien in den für die UR2-Installation ausgewählten Ordner kopiert:

SetupHinweise.txt

SetupNotes.txt

LizenzHinweise.txt

LicenseNotes.txt

Info\_UR2\_startbat.txt

InfoPath\_Var\_de.txt

InfoPath\_Var\_en.txt

unins000.dat

unins000.exe

Glade.dat

Uncertradio.exe

UR2\_Help\_DE.chm

UR2\_Help\_EN.chm

UR2\_cfg.dat

settings.ini

gtk\_UR2.css

UR2\_SingleAutoRun\_V11.xlsm

UR2\_start.bat

InfoFX1.txt

Die für das GTK-GUI und die PLPLOT-Grafik benötigten DLL-Dateien und andere Dateien werden in den Unterordner GTKuser64 kopiert (entpackt ca. 55 MB).

Die deutsch- und englischsprachigen UncertRadio-Beispielprojekte werden in die Unterordner pros\de und pros\en kopiert.

## Installation

**Von Version 2.1.3 an wird die Software im Wesentlichen mit einem Setup-Installationsprogramm eingerichtet.**

**Schritt 1:**

Es wird empfohlen, den bisher verwendeten Arbeitsordner, z.B. d:\UR2, umzubenennen, aber noch nicht zu löschen, bevor die Installation gestartet wird.

**Schritt 2:**

Ausführen der Anwendung Setup\_UR2\_Vers\_xxx.exe, wobei am Beginn der UR2-Ordner zu selektieren ist. Vor dessen Ausführung ist zu bedenken:

Erfolgte die vorherige Installation schon mit einem Setup-Programm (Setup\_UR2\*.exe), steht davon im UR2-Ordner ein Programm

Bestand der gewählte Ordner schon vor dieser Installation, müssen alle alten Programm- Elemente (inklusive Unterordner) entfernt werden, bevor das obige Setup-Programm gestartet wird.

unins000.exe (verwendet unins000.dat für die zu löschenden Dateien)

zur Verfügung. Mit dessen Ausführung können die Dateien der vorigen Installation komplett entfernt werden, wobei die von Benutzern geänderten oder ergänzten Dateien.

Der Ordner GTKuser64 wird als Unterordner des UR2-Ordners eingerichtet. Weiterhin werden die Unterordner

Icons (icons)

Pros (Beispielprojekte)

Share (Locale-Dateien)

Angelegt.

Während der Installation erfolgen Einträge oder Änderungen von Schlüsseln in der Windows-Registrierung, die aber mit der Ausführung von unins000.exe wieder entfernt werden.

**Schritt 3:**

Die drei Dateien

UR2\_cfg.dat

settings.ini

UR2\_SingleAutoRun\_V11.xlsm

werden NUR DANN in Schritt 2 berücksichtigt, wenn sie im gewählten UR2-Ordner noch nicht vorhanden sind.

**Schritt 4:**

Hinweise zum Editieren der Windows-Umgebungsvariablen finden sich in der Datei InfoPath\_Var\_de.txt.

Mit dem Setup-Installationsprogramm erfolgen auch Einträge in die Windows-Registry.

**Spätere Deinstallation**:

Mit der Ausführung des Setup-Installationsprogramms werden auch folgende zwei Dateien in den UR2-Ordner kopiert:

unins000.exe (verwendet unins000.dat für die zu löschenden Dateien).

Mit dem Ausführen von unins000.exe werden alle zuvor installierten Dateien und Unterordner entfernt. Dateien, die seit der letzten Installation modifiziert wurden oder nicht aus dem Setup-Programm stammen, werden nicht entfernt.

**Manuelle Eintragungen von Umgebungsvariablen**

**Hinweis**:

Die für frühere Programmversionen erforderliche manuelle Eintragung von Windows-Umgebungsvariablen für PLPLOT entfällt, da letztere jetzt temporär im Programm erzeugt werden.

Windows:

Start - Systemsteuerung - Benutzerverwaltung.

In dem Fenster, dass sich dann öffnet, in der Aufgabenleiste (links) "Eigene Umgebungsvariablen ändern" aufrufen.

Windows 7:

Start - Systemsteuerung - System - erweiterte Systemeinstellungen - Umgebungsvariablen.

Hier kann man sich nur mit Admin-Rechten einloggen.

Unter Windows 8.1. muss man mit der rechten Maustaste auf den Startbutton klicken, um die Systemsteuerung aufzurufen.

Windows 10:

„Einstellungen aufrufen“, ins Suchfeld „Umgebung“ eingeben und dann „Systemumgebungsvariablen bearbeiten“ selektieren, dann unter „Erweitert“ „Umgebungsvariablen“ klicken.

Die **Systemvariable "Path"** wird in der Gruppe der Systemvariablen ausgewählt und der Eintrag über die Schaltfläche "Bearbeiten" geöffnet (Ist die Schaltfläche "Bearbeiten" nicht verfügbar, muss man sich mit Administrator­rechten anmelden; dazu siehe unten). Der Eintrag wird um folgenden String erweitert:

....;d:\UR2\GTKuser64\bin;

Es hat sich als sicherer herausgestellt, den hier angeführten String per copy/paste an die Umgebungsvariable anzuhängen. Der Pfad d:\UR2 kann durch selbst gewählte Pfade ersetzt werden.

Falls die Schaltfläche "Bearbeiten" für die Systemvariablen nicht verfügbar war, benötigt man **Administratorrechte**. In diesem Fall kann, **ohne Administratorrechte zu besitzen**, die **Batchdatei** **UR2\_start.bat** für das Starten von UncertRadio benutzt werden; siehe nächster Punkt.

\*\*\* Nach diesen Einträgen muss bei älteren Windowsversionen offenbar ein Neustart des Windows erfolgen. \*\*\*

## Programmstart

**Programm startet nicht wegen fehlender Zugriffsrechte**

Wenn Windows das Starten von Uncertradio.exe nicht erlaubt, kann das Programm mit der Batch-Datei UR2\_start.bat gestartet werden, wie interessierte Anwender herausgefunden haben (vielen Dank dafür!):

UR2\_start.bat:

ECHO OFF

Path = D:\UR2\GTKuser64\bin; %path%

SET LANG=de\_DE

START /min D:\UR2\uncertradio.exe %1

Der Pfadname des Ordners „GTKuser64\bin“ ist damit der erste Eintrag in der PATH-Variablen!

Die Windows-Variable Path wird dabei nur temporär für diesen Programmlauf geändert.

Der verwendete Ordnername, im obigen Beispiel „d:\UR2“, ist der Installationsordner, der beim Setup-Lauf angegeben wurde.

Die (neue) Option /min im Start-Befehl hindert GTK und GFortran daran, viele Warn-/Fehlermeldungen in das cmd-Fenster zu schreiben. Ein optimaler Start von UncertRadio kann durch Doppelklicken auf die UR2\_start.bat erfolgen. Um eine Auswertung eines UR2-Projekts mit einem Doppelklick starten zu können, muss der txp-Dateityp noch mit UR2\_start.bat verknüpft werden; siehe weiter unten.

##### Programm startet nicht oder beendet sich mit einer Runtime-Fehlermeldung

Wenn beim Starten des Programms eine Windows-Fehlermeldung auftritt, die einen Fehlercode (0xc000007b) angibt, bedeutet dies, dass z. B. aufgrund unzureichender Einstellungen in der PATH-Umgebungsvariable eine 32-bit-Version anstelle der 64-bit-Version einer DLL-Datei gefunden wird, so dass ggf. die PATH-Variable angepasst werden muss.

In der Phase des Starts von UR2 wird die kleine Textdatei fort.66 angelegt (nicht verwechseln mit Fort66.txt), die bei nicht erfolgtem Start Aufschlüsse geben kann.

Beispiel für fort.66 (s. Hinweis unten):

uncertradio

ncomargs= 0

work\_path=D:\UR2\

GTK-path=D:\UR2\

Work\_path=D:\UR2\ wpunix= F

Work\_path\_getarg=D:\UR2\

Die erste Zeile zeigt den Kommando-String, der von Windows zum Starten des Programms erzeugt wird, mit oder ohne Pfadangabe. Beim indirekten Start mit Doppelklicken auf eine mit UncertRadio.exe verknüpfte Projektdatei (.txp) enthält sie zusätzlich die Angabe des Projekts mit Pfadangabe. Der Parameter wpunix wird nicht mehr verwendet.

Wenn beim Starten des Programms eine Windows-Runtime-Fehlermeldung das Fehlen einer DLL-Datei anzeigt, ist der GTKuser64-Ordner mit dem vorliegenden Eintrag in die Windows-Umgebungsvariable PATH nicht erreichbar (letzterer fehlt oder ist fehlerhaft).

**Hinweis**: Ab der Version 2.3.07 wird die Datei fort.66 nicht mehr erzeugt; ihr Inhalt wird stattdessen intern gespeichert und (später) am Anfang der Datei fort66.txt ausgegeben.

**Hinweis zur Verknüpfung von Projektdateien (.txp) mit UncertRadio.exe**

Es ist möglich, den Dateityp .txp mit UncertRadio.exe, oder besser noch UR2\_start.bat, zu verknüpfen. Dies kann wie in Windows üblich erfolgen, z.B.:

* Eine txp-Datei markieren, dann rechte Maustaste „Öffnen mit“ klicken, dann „Andere App auswählen“ klicken;
* das Häkchen setzen und „weitere Apps“ klicken;
* Vorschlagsliste nach unten scrollen;
* „andere App auf diesem PC suchen“ klicken;
* Die gewünschte Anwendung, Uncertradio.exe oder besser noch UR2\_start.bat, mit dem Explorer auswählen und Enter geben;
* dann stürzt das UncertRadio schon beim Starten mit einer Laufzeit-Fehlermeldung einmal ab: ignorieren.

Mit dem nächsten Doppelklick auf die txp-Datei startet das Programm dann ordnungsgemäß.

Falls schon Verknüpfungen mit früheren UncertRadio.exe-Versionen existierten, ist die Wahrscheinlichkeit groß, dass trotz Durchführung der obigen Prozedur eine alte Programmversion aufgerufen wird. Das ist ein Windowsproblem und kann im Internet recherchiert werden. Das Öffnen der Textdatei „fort66.txt“ kann zur Klärung beitragen; diese enthält am Beginn den durch die Verknüpfung von Windows erzeugten String des Programmaufrufs (Kommandozeile), z.B.:

D:\test\_UR2\UncertRadio.exe D:\GF\_Pros\UR2\Mehr-Linien-Nuklid-Aktivitaet-V3\_DE.txp

Hierin gibt der ganz links stehende Pfad den Ort der tatsächlich aufgerufenen UncertRadio-Version an.

Wenn man feststellt, dass die falsche UR-Version aufgerufen wird, kann es erforderlich werden, den txp-Dateityp in der Windows-Registrierung zu löschen, um danach die Verknüpfung neu anzulegen. Es wird geraten, für die Bearbeitung der Registrierung die lokale IT-Gruppe heranzuziehen. Der interessierende Pfad in der Registrierung ist:

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\Current version\Explorer\FileExts

**Hinweis zur Verwendung der Excel-gesteuerten Anwendung von UncertRadio**

Die Excel-Datei

UR2\_SingleAutoRun\_V11.xlsm

(ab UR2-Version 2.4.04) ist gegenüber der Vorgänger-Version dahingehend angepasst worden, dass die an UR2 zu übergebenden Dateinamen immer den vollständen Pfad beinhalten. Es wurde ein weiterer Auswerte-Button „Start (autosep)“ integriert; siehe dazu Kapitel 5.2. Es wurden auch weitere Fehlerbehandlungen im dazugehörigen VBA-Code eingeführt.

Sie kann sowohl in der 32-bit- als auch in der 64-bit-Version des MS Office/Excel verwendet werden. Siehe dazu auch den Einleitungsteil im VBA-Modul (Modul\_Auto\_single\_UR) in der Exceldatei.

Das VBA-Modul der Exceldatei reagiert auf den mit der UncertRadio-Version 2.2.03 eingeführten Exit-Code von UncertRadio.exe. Der Exit-Code wird vom VBA-Modul interpretiert; ist dieser gleich 3 (Fehler in UncertRadio), führt dies zum vorzeitigen Abbruch des Moduls.

**Konfigurationsdatei UR2\_cfg.dat:**

|  |  |
| --- | --- |
| [UncertRadio configuration]  [Path]  UR\_path=D:\UR2\  Help\_path=D:\UR2\  Excel\_Auto\_path=D:\UR2\  UR\_AUTO\_output\_path=D:\UR2\  [Local]  Decimal\_point=  List\_separator=;  Language=DE  Monitor#=0  ContrastMode=F  winRelSizeWidth=0.7  winRelSizeHeight=0.9 | Änderbare Pfadnamen:  Pfad in dem die CHM-Hilfe-Datei steht (***lokales LW!***)  Pfad mit der Exceldatei für automatis. Betrieb von UR:  wird in UR derzeit nicht benutzt  Pfad, in den UR seine AUTO-Ausgaben schreibt.  Ist der Wert von UR\_AUTO\_output\_path leer, wird ihm in UR der Pfadname aus dem Argument des Aufrufs aus Excel heraus zugewiesen.  Nur wenn der UR-Output NICHT in diesen Excel-Pfad zurückgeschrieben werden soll, darf der Werte von UR\_AUTO\_output\_path nicht leer sein.  Wenn der Kontrastmodus des Programmfensters temporär benötigt wird, kann dieser auch im Dialog Optionen aktiviert werden  Verhältnis der Fensterbreite zur Monitorbreite  Verhältnis der Fensterhöhe zur Monitorhöhe  Der ältere Eintrag „windowRelSize=X“ wird intern in „winRelSizeWidth=X“ und „winRelSizeHeight=X“ umgewandelt. |

Der Eintrag „Monitor#=0“ wurde mit der Version 2.3.08 eingeführt. Siehe dazu Abschnitt „Verwendung mehrerer Monitore“ am Ende dieses Hilfekapitels.

Das Fenster kann nach dem Programmstart mit der Maus zwar weiter aufgezogen werden, es springt aber nach der nächsten Aktion wieder zurück.

**Länderspezifische Einstellungen**

Die in der Konfigurationsdatei UR2\_cfg.dat unter [Local] aufgeführten Parameter können auf verschiedene Weise festgelegt werden.

1. man lässt die Werte hinter dem „=“ - Zeichen weg:

Decimal\_point=

List\_separator=

Language=

Zunächst wird aus dem Windows-System die Landessprache ausgelesenen, d.h. DE, EN oder FR. Im Falle einer davon abweichenden Sprache wird EN ausgewählt wird.

Daraus werden im Programm die Werte für den Dezimalpunkt und das Listentrennzeichen (für CSV-Dateien) festgelegt:

DE oder FR: Komma als Dezimalpunkt; Semikolon als Listentrennzeichen

EN: Punkt als Dezimalpunkt; Komma als Listentrennzeichen

Hinweis: Die im Hauptmenü – Optionen einstellbare Sprache gilt nur temporär, d.h.

während eines Programmlaufs.

Ab Version 2.2.03 muss dasSprachkürzel, das auf language= folgt, auf jeden Fall eingetragen werden; fehlt es, wird programmintern EN angenommen.

**Ab Version 2.2.07 wird das Sprachkürzel aus der bei der Ausführung des Setup-Programms gewählten Sprache übernommen.**

1. Wenn andere als diese Vorfestlegungen verwendet werden sollen, können die gewünschten Werte in der Konfigurationsdatei hinter den entsprechenden Gleichheitszeichen eingesetzt werden. Unter Umständen braucht nur das gewünschte Listentrennzeichen explizit ergänzt zu werden:

List\_separator=;

1. Die Sprache kann mit Hilfe des Dialogfeldes des Menus „Optionen - Voreinstellungen“ auch umgestellt werden, wenn das Programm schon gestartet wurde (temporär; siehe oben). Mit Hilfe der dort eingestellten Sprache werden dann, wie hier unter a) beschrieben, die Zeichen für Dezimalpunkt und Listentrennzeichen festgelegt. Zusätzlich kann in dem Dialog aber auch das Listentrennzeichen direkt gewählt werden.
2. Für eine grafische Darstellung wird das Dezimalpunkt-Zeichen durch das Sprachkürzel, DE, EN oder FR festgelegt; eine Ländereinstellung in Windows auf diese Sprache ist nicht erforderlich.

Mit der Installation des Programms erfolgen auch **Einträge in die Windows-Registrierung**. Diese werden mit dem Deinstallationsprogramm unins000.exe wieder entfernt.

Ab Version 2.2.07 werden bei der Ausführung von unins000.exe folgende zwei Einträge in der Windows-Registrierung gelöscht:

Root: HKCU; Subkey: "Software\Microsoft\Windows\Currrent version\Explorer\FileExts\

.txp\OpenWithList"; ValueType: RG-Schlüssel; ValueData: "UncertRadio.exe";

Root: HKCU; Subkey: "Software\Microsoft\Windows\Currrent version\Explorer\FileExts\

.csv\OpenWithList"; ValueType: RG-Schlüssel; ValueData: "UncertRadio.exe";

Damit soll sichergestellt werden, dass nach der nächsten Ausführung des Setup-Programms, durch Doppelklicken auf ein txp- oder ein csv-Projekt nicht eine alte Version von UncertRadio.exe gestartet wird.

Die Windowshilfe kann durch Doppelklicken auf den Dateinamen UR2\_help\_de.chm separat vom UncertRadio-Programm gestartet werden.

**Einstellung der Sprache**

Die **Benutzerführung** in UncertRadio ist so eingerichtet, dass sie **sowohl in Deutsch, in Englisch als auch in Französisch** erfolgen kann. Dies lässt sich mit Hilfe des Eintrags „Language“ in der Konfigurationsdatei UR2\_cfg.dat steuern, die bei jedem Programmstart gelesen wird. Die Parameterwerte sind:

DE für Deutsch,

EN für Englisch,

FR für Französisch

Die Änderung der Sprache ist noch einfacher und sicherer durch die Verwendung der Startdatei

UR2\_start.bat:

ECHO OFF

Path = D:\UR2\GTKuser64\bin; %path%

SET LANG=de\_DE

START /min D:\UR2\uncertradio.exe %1

Ein Eintrag SET LANG=en\_EN darin stellt sicher, dass die Sprache in Windows (temporär) ebenfalls auf Englisch geändert wird; nach Beendigung des Programms ist Windows wieder auf Deutsch eingestellt.

Der in der Startdatei enthaltene **Befehl „SET LANG=de\_DE“** wird **ab der Version 2.3.07**, basierend auf dem in der UR2\_cfg.dat eingestellten Sprachkürzel, **direkt im Programm** verwendet. Damit kann auf diese Startdatei verzichten werden, wenn damit nur ein Wechsel der Sprache erreicht werden soll. Ist das Sprachkürzel EN, wird der interne Befehl „SET LANG=en\_EN“ verwendet und die Bedienerführung erfolgt in Englisch. Diese Sprachumstellung ist nicht temporär; beim Beenden erfolgt die Rücksetzung auf die Sprache, die vor dem Start von UncertRadio vorlag. Bei einem Programmabbruch unterbleibt diese Rücksetzung; sie ist ggf. mit „SET LANG=de\_DE“ in einem Windowseingabefenster nachzuholen.

Die Sprachumstellung kann im laufenden Programm im Dialog Menü – Optionen – Voreinstellungen vorgenommen werden. Diese umfasst **ab Version 2.3.07 auch diejenigen „GTK stock buttons“**, mit Symbol und Textlabel versehen, die bisher nur bei Neustart des Programms übersetzt wurden. Dies wurde erreicht, indem alle „stock buttons“ in normale Buttons umgewandelt wurden; dabei änderten sich die Form der Button-Symbole geringfügig. Beim komplexen GTK-Dialog für die Dateiauswahl kann die Sprachumstellung weitgehende, aber nicht komplett erfolgen.

**Konfigurationsdatei settings.ini:**

[Settings]

gtk-theme-name = win64

gtk-font-name = Sans Normal 10

In dieser sich auf das GTK3-GUI beziehenden Datei darf nur der Eintrag

gtk-font-name = Sans Normal 10

geändert werden, der Schrifttyp und -größe definiert.

##### Verwendung mehrerer Monitore

Bisher wurde in UncertRadio das Arbeiten mit nur einem an den PC angeschlossenen Monitor unterstellt. Intern wurde der für den „Screen“ ausgelesene Koordinatenbereich (Pixel) genau einem vorhandenen Monitor zugeordnet. Für diesen Fall kann in der oben eingeführten Konfigurationsdatei UR2\_cfg.dat der Eintrag „Monitor#=0“ auch weggelassen werden.

Mittlerweile ist das Arbeiten mit mehr als einem Monitor nicht mehr unüblich. In diesem Fall umfasst der Koordinatenbereich des Screens die entsprechenden Bereiche der einzelnen Monitore. Den Screen muss man sich dann als äußeres, die Gesamtheit der Monitor-Rechtecke einhüllendes Rechteck vorstellen. Die Zuordnung zu einem der Monitor-Rechtecks erfolgt durch Vorgabe der Koordinaten dieses Monitor-Rechtecks als Screen-Koordinaten. Man ordnet der Breite die Koordinate x und der Höhe die Koordinate y zu. x=0 kennzeichnet den linken Rand des Screens und y=0 dessen oberen Rand; die beiden Achsen zeigen also von links nach rechts (x) und von oben nach unten (y). Programmintern wird ein Monitor-Rechteck aus der Koordinate der linken oberen Ecke und den Breiten- und Höhenangaben des Monitors konstruiert.

Die folgende Abbildung zeigt zwei mögliche Screen/Monitor-Anordnungen.

|  |  |
| --- | --- |
| Ein Bild, das Uhr enthält.  Automatisch generierte Beschreibung | screen:  2560 x 1656  (Monitor 2 nicht aktiv) |
|  |  |
| Ein Bild, das Screenshot enthält.  Automatisch generierte Beschreibung | screen:  5760 x 1200 |
|  |  |

In UncertRadio, ab Version 2.3.08, werden beim Programmstart mit GDK-Befehlen die Größe des Screens, die Anzahl vorhandener Monitore und deren Rechteck-Koordinaten abgefragt. Diese Angaben findet man in der Datei fort66.txt in folgender Form wieder (beispielhaft für mehrere Monitorbereiche).

------------------------------------------------------------------------------

coordinates: width x height

\*\*\* Screen: 3200 x 1776

PixelxZoom=120 PixelyZoom=120

\*\*\* Monitors:

\*\*\* 1 Ranges (geom): (1920 - 2560) x (0 - 480) scaling fact= 1.25

\*\*\* 1 Ranges (geom): (0 - 1920) x (480 - 1680) scaling fact= 1.25

\*\*\* 1 Ranges (geom): (1920 - 2560) x (0 - 480) scaling fact= 1.25

\*\*\* # of monitors: 3

\*\*\* Monitor number selected as given in UR2\_cfg.dat: 2

\*\*\* Primary monitor # = 1

\*\*\* Selected monitor: 1; Screen min-max horiz.: 1 - 1919 min-max vertical: 482 - 1642

(und etwas weiter unten)

\*\*\* Main window: first Show: upper-left pos: mposx,mposy= 96 532

\*\*\* Main window: Monitor# at mposx+10,mposy+10= 1

\*\*\* Main window: width= 983 height= 713

------------------------------------------------------------------------------

Dabei gibt es aber noch zwei Probleme.

Die sich aus der Tabelle ergebenden Breiten- und Höhenangaben können einen in Windows 10 eingestellten Skalierungsfaktor enthalten: z.B. 1,25 für eine Windows 10-Skalierung von 125 %. Dieser Faktor kann derzeit von UncertRadio noch nicht sicher aus Windows 10 ermittelt werden. Die Skalierungsangaben in Windows basieren auf einem Wert von 96 DPI (dots per inch), der dort als 100 % der Skalierung ausgewiesen wird. Für eine vom Anwender eingestellte Skalierung von 125 % wird intern jedoch der um das 0,96fache kleinere Wert 120 % gespeichert, z.B. als „LogPixels“ in der Registrierung.

Zum anderen können sich die von UncertRadio den Monitoren zugeordneten Nummern von den in Windows 10 vergebenen Nummern unterscheiden. Windows 10 erlaubt überdies die grafische Umordnung der Monitore.

Die Zuordnung zu einem der Monitore erfolgt durch Vorgabe von dessen Nummer. Die Nummer kann **in der Datei UR2\_cfg.dat** mit folgendem Eintrag vorgegeben werden (hier auf 1 gesetzt):

Monitor#=1

Es kann sein, dass durch Vorgabe der Monitornummer das UR2-Fenster nicht an der gewünschten Stelle des Screens erscheint. Dann ist das Testen der Monitornummern in dem Eintrag Monitor#= erforderlich.

Nachdem das UncertRadio-Fenster vollständig aufgebaut ist, kann man die vom Programm verwendete Monitornummer mit Hilfe des neuen Eintrags „Monitor#“ im Menü – Optionen abfragen. Für diese Abfrage wird derjenige Monitor ermittelt, in dessen Koordinatenbereich die linke obere Ecke des UncertRadio-Fensters liegt.

Ein oder zwei Versionen vor 2.3.08 erfolgte programmintern gelegentlich eine Re-Positionierung oder ein Resize. Da die dafür erforderlichen Koordinaten aus den größeren Screen-Koordinaten abgeleitet wurden, konnte es im Falle von mehr als 1 vorhandenen Monitor vorkommen, dass das UR2-Fenster breiter als die des aktuellen Monitors war. Dieses Fenster konnte auch nicht kleiner gezogen werden.

**Anmerkung**: Der **Umgang mit mehreren Monitoren** in UncertRadio befindet sich noch **in der Erprobungsphase**.

**Aus dem bisherigen Stand ergibt sich folgende Empfehlung für das Vorgehen.**

Ist nur ein Monitor an den PC angeschlossen, gibt es drei Optionen für die UR2\_cfg.dat:

a) den Eintrag „Monitor#=0“ verwenden, oder   
b) diesen ganz weglassen, oder   
c) „Monitor#=1“ verwenden (hierbei werden das UR2-Fenster und das MC-Grafikfenster besser voneinander getrennt);

Bei mehreren Monitoren sollten zunächst die Optionen a) oder b) versucht werden.

Wenn das UncertRadio-Fenster nicht im gewünschten Monitor erscheint, kann die richtige Monitornummer vorerst nur durch Probieren mit dem Eintrag „Monitor#=“ in UR2\_cfg.dat ermittelt werden.

Derzeit versucht UncertRadio **mit folgendem Algorithmus** aus seinen aus Windows ausgelesenen Informationen die passenden Koordinaten des Monitors zu finden. Zunächst wird der Wert der Skalierung gelesen, LogPixels = 96 entspricht 100 %. Weiterhin hält UncertRadio eine Tabelle mit 20 möglichen Breite/Höhe-Kombinationen vor. Mit den von UncertRadio aus Windows gelesenen Werten für die Monitor-Koordinaten wird als erstes versucht, ohne Skalierung aus den 20 vorgegebenen die exakt passende Breite/Höhe-Kombination zu finden. Wenn das nicht gelingt, werden im zweiten Schritt die aus Windows gelesenen Koordinaten um die Skalierung verringert (Teilen

durch (LogPixels/96)) und aus den 20 Kombinationen diejenige Breite/Höhe-Kombination mit der geringsten Abweichung davon verwendet.

## Programm-Version und Anwendungsvermerk

|  |  |
| --- | --- |
|  |  |

Das Windows-Programm UncertRadio wird **kostenfrei** zur Verfügung gestellt.

Das Programm wurde vom Autor nach derzeitigem Stand von Wissenschaft, Normung und Technik geschrieben. Trotzdem wird darauf hingewiesen, dass **keinerlei Garantie für die Richtigkeit** **der** bei der Verwendung **von UncertRadio** **erhaltenen Rechenergebnisse** übernommen werden kann.

Für Hinweise auf mögliche Fehler bin ich sehr dankbar!

Das Programm soll vor allem auch dazu dienen, diejenigen Auswerteverfahren zu behandeln, welche Gegenstand der „Messanleitungen für die Überwachung radioaktiver Stoffe in der Umwelt und externer Strahlung“ (Leitstellen) sowie der entsprechenden Loseblattsammlung „Empfehlungen zur Überwachung der Umweltradioaktivität“, FS-78-15-AKU, sind.

**UncertRadio** eignet sich hierbei **sehr gut für Vergleichsrechnungen,** z.B. zum Vergleich mit Rechnungen mit eigenen Tabellenkalkulationen.

## 1.5 UR- Hilfe und Netzlaufwerke

Die Windowshilfe UR2\_HELP\_DE.CHM zum Programm lässt sich aus Sicherheitsgründen nicht von einem Netzlaufwerk aus betrieben. Daher ist anzuraten, das UR inkl. dazugehöriger Hilfedatei auf einem lokalen Laufwerk zu installieren. Es würde aber auch ausreichen, nur die CHM-Datei in einen Ordner auf einem lokalen Laufwerkt zu verlegen. In der Konfigurationsdatei UR2\_cfg.dat kann dieser Pfad mit dem Eintrag Help\_path= festgelegt werden.

Folgendes kann ebenfalls empfehlen werden:

* Klicken sie im Windows-Explorer mit der RECHTEN Maustaste auf die CHM-Datei und wählen aus dem Kontextmenü den Menüpunkt »Eigenschaften«
* Im Fenster »Eigenschaften« finden den Reiter »Sicherheit«; darauf wird unten der Text »Die Datei stammt von einem anderen Computer. Der Zugriff wurde aus Sicherheitsgründen geblockt.« angezeigt;
* Klicken sie auf den Button »Zulassen«.

## 1.6 Bisher erfolgte Änderungen am Programm (Updates)

***UncertRadio* - Version 2.4.32 2023/09**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 23.09.2023

**Überarbeitung der Windows-Hilfe:**

* Es wurde ein neues Kapitel 5.8 erstellt, in dem der Hintergrund zum Test unter Menüpunkt Optionen – QC-Batch-Test beschrieben wird.

**Hinzufügung eines Beispiel-Projekts:**

* Es erfolgte eine Datenkorrektur im Projekt iso11929-4\_Example-11\_EN.txp; die korrigierte Fassung der Datei lautet: iso11929-4\_Example-11\_V2\_EN.txp.

**Überarbeitung des Programms:**

* Das Layout der Grafiken der gefitteten Abklingkurven wurde verbessert, auch hinsichtlich der Unterscheidung einzelner Kurven.
* Mit dem Schließen des Texteditorfensters zur Ansicht des Fit-Zwischenergebnisses wird auch das dazugehörige Grafikfenster verborgen.
* Der Dialog zur Dateneingabe einer Abklingkurve wurde kürzlich hinsichtlich seines Layouts geändert. Er wurde nochmals überarbeitet, um die Bedienbarkeit einzelner Eingabefelder zu verbessern. Dieser Dialog wurde wieder auf die Betriebsart „nicht modal“ umgestellt, damit darin mit der Maus markierte Datenzeilen gelöscht werden können.

**Überarbeitung der Excelanwendung:**

* keine

***UncertRadio* - Version 2.4.31 2023/09**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 14.09.2023

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Ein kürzlich eingeführter Fehler in einer Callback-Funktion von Eingabefeldern des GUI führte u.a. dazu, dass die Werte der Quantile im Optionen-Dialog nicht mehr geändert werden konnten. Dieser Fehler wurde beseitigt.
* Bisher wurden explizit eingegebene, oder durch Formeln definierte Kovarianzwerte nicht daraufhin geprüft, ob der Betrag der daraus berechneten Korrelation größer als eins wird. Diese Prüfung erfolgt ab dieser Version.
* Es konnte zu dem nicht immer verständlichen Hinweis führen, dass die relative Unsicherheit von w zu groß wäre (< 1/kbeta), so dass die Nachweisgrenze nicht mehr berechenbar wird. Die bisher dafür verwendete Schätzung von urel(w), ohne dass eine separate Gleichung für den Faktor w bzw. den internen Faktor Flinear existiert, wurde nun durch eine einfache Unsicherheitsfortpflanzung ersetzt.

**Überarbeitung der Excelanwendung:**

* keine

***UncertRadio* - Version 2.4.30 2023/08**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 21.08.2023

**Überarbeitung der Windows-Hilfe:**

* Im Abschnitt 1.3 wird auf eine Änderung bei der UR2\_cfg.dat aufgeführt.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Im Falle einer linearen Entfaltung mit der Bedingung, dass nur 1 Messkanal vorliegt, nur eine einzige Ergebnisgröße definiert wird, aber 2 bis 3 Fitparameter angepasst werden, wird der Plot der Zerfallskurve erweitert. Neben der gefitteten Summenkurve werden auch die beteiligten Komponentenkurven geplottet.
* Bei der Verwendung eines Datensatzes zur Berechnung und Verwendung eines Mittelwerts trat bei der MC-Simulation des MW-Typs 3 („Klassisch“, s. Kapitel 6.12.1 der CHM-Hilfe) der Effekt auf, dass die MC-Verteilungen zu schmal waren. Dazu wurde ein überflüssiger Faktor entfernt.
* Der früher eingeführte Faktor windowRelSize zur Einstellung der Fenstergröße wurde durch je einen Eintrag winRelSizeWidth und winRelSizeHeight für Breite und Höhe ersetzt. Siehe dazu Abschnitt 1.3 zur UR2\_cfg.dat. Für den Fall, dass der Screen mehrere Monitore enthält, wurde ein Fehler beseitigt, zu Fehlern führte, dass das Programmfenster größer als der Bildschirm wurde.
* Im Dialog für die Eingabe der Daten zur Berechnung eines Mittelwerts einer Eingangsgröße wurde die Combobox für die Selektion eines Referenzdatensatzes geändert. Es kann auch ein „leerer“ Datensatz selektiert werden, wenn aus Versehen ein Referenzdatensatz selektiert wurde, der keiner ist.
* Wenn ein Symbolname einen Teilstring „ln“ bzw, „LN“ enthielt, versuchte das Programm, diesen als eine Funktion ln() zu interpretieren. Der Fehler wurde behoben. Ein weiterer Fehler wurde behoben, der bei der MC-Simulation eines Projekts mit linearer Entfaltung, mit genau 1 Ergebnisgröße, aber mehr als 1 Fitparameter, dazu führte, dass die Nettozählrate konstant blieb.
* Für die Zellen der Standardabweichungsformeln wurde ein noch verbliebene Begrenzung auf 100 Zeichen behoben.
* Ab dieser Version (2.4.30) ist es möglich, im Textfenster zur Strukturierung für die Gleichungen auch Leerzeilen einzusetzen. Diese Leerzeilen treten nur in diesem Textfenster auf; intern werden sie jedoch ausgelassen. Wenn der Function Parser für eine Gleichung n einen Fehler anzeigt, sind in n, passend zum Textfenster, die Leerzeilen enthalten, d.h. sie sind darin mitzuzählen.

**Überarbeitung der Excelanwendung:**

* keine

***UncertRadio* - Version 2.4.29 2023/08**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 07.08.2023

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* Das Projekt PearsonYork\_mit\_KALFIT\_DE.txp (WTLS für eine Polynom-Kalibrierkurve) wurde ergänzt.

**Überarbeitung des Programms:**

* Weitere 6 Entry-Felder verschiedener Dialoge wurden zur besseren Sichtbarkeit mit Rahmen versehen.
* Ein bestimmte Callback-Funktion verschiedener Bedienelemente in Dialogen wurde geändert, um das ungewollte Verlassen des betreffenden Dialogs zu unterbinden.
* In der Datei InfoFX1.txt wurde ein Eintrag für die UR-Funktion fd (Formel für einen Abklingfaktor) ergänzt.
* Das Paket für WTLS wurde überarbeitet und ist um eine weitere Funktionen ergänzt worden. Mit deren Hilfe kann mit WTLS ein Polynom an eine Kalibrierkurve mit nur einer unabhängigen Variable X angepasst werden, bei der den unabhängigen X-Werten ebenfalls Unsicherheiten zugeordnet sind. Dies erweitert die Funktionalität der Funktion KalFit.

**Überarbeitung der Excelanwendung:**

* keine

***UncertRadio* - Version 2.4.28 2023/07**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 26.07.2023

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Seit der Version 2.4.24 ist ein Fehler im Modell-Dialog der linearen Entfaltung aufgetreten, der die Bedienung des Textfensters für die Xi-Formeln praktisch unmöglich machte, weil nach jedem dort ein- oder ausgefügten Zeichen das Programm diesen Dialog sofort verließ. Dieser Fehler wurde beseitigt.
* Derselbe Dialog wurde im oberen Bereich überarbeitet, so dass mit einer neuen Checkbox explizit definiert wird, ob Zeitdifferenzen als Differenz von vollständigen Datum/Zeit-Angaben ermittelt werden, oder aber in der Tabelle darunter direkt als Zeitdifferenzenwerte in Sekunden eingetragen werden.
* In der Tabelle im TAB „Werte, Unsicherheiten“ wurde im Falle der linearen Entfaltung mit mehr als einer Ergebnisgröße die Bedeutung der Hintergrundfarben auch auf die Fitparameter Fitp1, Fitp2 und Fitp3 angewendet. Die Hintergrundfarbe ist nur weiß, wenn der dazugehörige Fitparameter fixiert werden soll.

**Überarbeitung der Excelanwendung:**

* keine

***UncertRadio* - Version 2.4.27 2023/07**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 17.07.2023

**Überarbeitung der Windows-Hilfe:**

* Kleine Ergänzungen zum Kapitel 7.4.3 (Chi-Quadrat Optionen)

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Zur Verwendung der linearen Entfaltung mit Linfit wurde mit der Version 2.4.24 die Liste der Argument-Parameter verkürzt. Das programm-interne Auffinden der tatsächlich von Linfit benutzten Eingangsgrößen wurde überarbeitet, da in diesem Bereich auch noch ein Fehler festgestellt wurde, wenn nur die verkürzte Argumentliste verwendet wurde.
* Die restlichen Überarbeitungen des Programms können unter „Tuning“ zusammengefasst werden. Dies betrifft den Bereich der linearen Entfaltung, für den hauptsächlich eine Verbesserung der Rechenzeit erreicht wurde. Unnötige Wiederholungen von Rechenschritten wurden entfernt. Das sehr rechenaufwändige WTLS-Verfahren wurde ebenfalls überarbeitet. Einige Wiederholungen von Teilen der Ausgabe in die Datei fort66.txt wurden ebenfalls reduziert.
* Die Rechenzeit für die Ausführung des Tests „QC-Batch-Test“ wurde insgesamt um etwa 25 % verringert.

**Überarbeitung der Excelanwendung:**

* keine

***UncertRadio* - Version 2.4.26 2023/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 30.06.2023

**Überarbeitung der Windows-Hilfe:**

* Das Kapitel 7.4.3 (Chi-Quadrat Optionen) wurde mit Ergebnissen eines statistischen Testverfahrens deutlich erweitert; kleine Ergänzungen in den Kapiteln 2.6 und 5.2.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Im Dialog für das lineare Modell der Auswertung einer Abklingkurve wurde ein Fehler im GUI-Handling festgestellt. Der bestand darin, dass bei jeder Änderung in Feldern dieses Dialogs, auch wenn nur 1 Zeichen geändert wurde, der Dialog sofort verborgen wurde, d.h. vor dem Betätigen des OK oder Cancel-Buttons, und die Berechnungen bis zu den „Resultaten“ aktualisiert wurden. Dieses Verhalten wurde abgestellt, so dass Änderungen erst nach dem Verlassen des Dialogs (über OK oder Cancel) umgesetzt werden.
* Für die Auswertung einer Abklingkurve wurde im dazugehörigen Grafikfenster die bisher als stückweise-linear dargestellte Fitkurve durch eine höher aufgelöste Kurve ersetzt.
* Beim Dialog für das Öffnen oder Speichern einer Datei erfolgte eine kleine Änderung. Diese betrifft die Option „Speichern unter“. In das Dateinamenfeld ganz oben wird jetzt nur der voreingestellte Dateiname übertragen, ohne den dazugehörigen Ordner darin anzugeben. Wie bisher wird in dem darunter liegenden Teil des Dialogs der gewünschte Ordner selektiert. Mit „Speichern unter“ werden beide Namensteile für das Speichern zusammengeführt.

**Überarbeitung der Excelanwendung:**

* Im VBA-Modul Modul\_auto\_single\_UR, in der Routine Init\_pathnames, wurde eine andere Excelfunktion eingesetzt, mit der die für das Starten von UncertRadio wichtige PATH-Variable gesetzt wird. Diese Funktion heißt SetEnvironmentA; siehe dazu auch das Hilfe-Kapitel 5.2.

Die aktuelle Version der Excelanwendung ist

UR2\_SingleAutoRun\_V11.xlsm

***UncertRadio* - Version 2.4.25 2023/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 06.06.2023

**Überarbeitung der Windows-Hilfe:**

* Ein kleiner Zusatz zu Kapitel 2.3

**Hinzufügung eines Beispiel-Projekts:**

* J-ALUFT\_Sr-89\_Sr-90\_Linf\_DE.txp, welches die Version mit linearer Entfaltung von J-ALUFT-Sr89-Sr-90\_V2\_DE.txp ist

**Überarbeitung des Programms:**

* Beim Menü „Bearbeiten – Selektieren der Ergebnisgröße“ wurde im ersten Feld ein GUI-Fehler beseitigt, der für die Anzeige einen Unterstrich im Namen eines Symbols entfernen konnte.
* Ein weiterer Fehler, der mit der letzten Programmversion auftrat und interne Symbole-bezogenen Arrays inkorrekt modifizieren konnte, wurde beseitigt. Dieser konnte dazu führen, dass es beim Menüpunkt zur Prüfung der physikalischen Einheiten zu einem Fehler kommen konnte.
* In der Datei BatListRef\_v05.txt wurde das obige neue Projektbeispiel ergänzt.
* Es werden für GTK zwei weitere Umgebungsvariablen temporär erzeugt, XDG\_DATA\_HOME und XDG\_DATA\_DIRS, die auf den Unterordner GTKuser64\share verweisen.

**Überarbeitung der Excelanwendung:**

Keine

***UncertRadio* - Version 2.4.24 2023/04**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 18.04.2023

**Überarbeitung der Windows-Hilfe:**

* Kapitel 3.4, 6.3

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Der Dialog „Optionen“ wurde hinsichtlich seiner Bedienung bei der Eingabe von Quantilen oder dazugehörigen Wahrscheinlichkeiten überarbeitet und dabei auch verkleinert.
* In die Toolbar wurden zur Gruppierung vertikale Trennelemente eingesetzt.
* Es wurde der Fehler beseitigt, dass es bei einem neu zu erstellenden Projekt zu einem Absturz kommen konnte, wenn das Projekt schon gespeichert wurde, bevor überhaupt in einem Folgeschritt Symbole aus den Gleichungen extrahiert wurden. Der einzige Menüpunkt der das zu frühe Speichern noch erlaubte, wurde für die Phase der Erstellung von Gleichungen ebenfalls deaktiviert.
* Der Aufruf der speziellen UncertRadio-Funktion Linfit zur Bearbeitung einer Abklingkurve wurde so vereinfacht, dass nur noch drei Parameter/Größensymbole übergeben werden:

Rd = Linfit(1, Rbl, tmess, tstart)

Die Übergabe weiterer Argument-Symbole ist nicht länger erforderlich.

**Überarbeitung der Excelanwendung:**

Keine

***UncertRadio* - Version 2.4.23 2023/04**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 02.04.2023

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurde ein Fehler beim Ausdrucken einer Matrix mit der Routine MatWrite behoben. Bei einem eingeschränkten Druckbereich wurden die Dimensionen falsch gesetzt, so dass die Matrixelemente in falscher Reihenfolge erschienen.
* Bei der Unsicherheitsfortpflanzung mit Upropa wurde ein selten auftretender Fehler beseitigt, der dazu führte, dass die Erkennungsgrenze null wurde. Die Sammlung der Beispielprojekte war davon nicht betroffen.
* In der Datei UR2\_cfg.dat wurde ein neuer Parameter windowRelSize eingeführt, der die Größe des UR2-Fensters in der Weise setzt, dass es anschließend seine Größe beibehält. Dieser Parameter muss schon in der cfg-Datei festgelegt werden, der in das GUI so eingesetzt wird, bevor der GtkBuilder das GUI aufbaut. Bei schon laufendem GUI kann diese Option offenbar nicht mehr eingestellt werden.
* Es wurde ein neues Icon ***f*(*x*)** in der Toolbar eingeführt, mit dem Kurzinformationen zu den speziellen UR2-Funktionen abgerufen werden können. Hierfür wird die neue Textdatei InfoFX1.txt benötigt.

**Überarbeitung der Excelanwendung:**

Keine

***UncertRadio* - Version 2.4.22 2023/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 01.03.2023

**Überarbeitung der Windows-Hilfe:**

* Die Kapitel 5.1 und 5.2 wurden überarbeitet

**Hinzufügung eines Beispiel-Projekts:**

* Von den vorhandenen Beispielprojekten wurden die folgenden unter dem Aspekt der verwendeten Einheiten geändert und unter höherer Versionsnummer eingestellt:

Calibration-of-weight-Cox-2001\_V2\_EN.txp, Neutron-Dose-Cox-2006\_V2\_DE.txp

**Überarbeitung des Programms:**

* Bei der MC-Simulation von Projekten des Typs „nur GUM, ohne Nachweisgrenzen“ wurde ein Fehler beseitigt, dass zwei für die Berechnung von Nachweisgrenzen erforderliche Variable in diesem Fall nicht auf null gesetzt waren.
* Im Dialog für Graphiken wurde schon bei der vorigen Version von UncertRadio ein Reiter entfernt. Dies führte zu einer Verschiebung innerhalb der verbleibenden Reiter, so dass das Kopieren einer Grafik in die Windows-Zwischenablage nicht funktionierte. Der Fehler wurde behoben.
* Im Zusammenhang mit der Option, physikalische Einheiten zu testen, wurden weitere Änderungen vorgenommen. Für das Laden der beiden dazugehörigen Dateien wurde zur Vermeidung eines Programmabsturzes intern der Arbeitspfad ergänzt. Aus der Datei UnitsTable.txt wurden die Einheiten für m² und m³ entfernt, da sie ja nur Potenzen von m darstellen. Das Ersetzen von angewendeten Einheiten wie z.B. m², m2, cm², cm2 wurden in die Datei Units\_other.txt verlegt.
* Zwei vorhandene Projekte wurden hinsichtlich der darin verwendeten Einheiten überarbeitet; ihre Versionen wurden in die Referenzwerte-Datei übernommen, ohne deren Namen zu ändern:

BatListRef\_v05.txt

**Überarbeitung der Excelanwendung:**

* Die Excelanwendung wurde überarbeitet, um Probleme mit Antivirussoftware zu vermeiden. Die in der Excelanwendung erforderliche Einstellung bestimmter Ordner wurde so geändert, dass die Ordner genommen werden, die bisher schon in der Konfigurationsdatei UR2\_cfg.dat standen. Die für Antivirussoftware kritische Datei UR2\_start\_xls.bat entfällt ganz. Stattdessen nimmt Excel-VBA die Modifikation der Windows-PATH-Variable direkt vor und startet UncerRadio.exe ebenfalls direkt. Zu den Änderungen wird auf die Kapitel 5.1 und 5.2 der CHM-Hilfe verwiesen.

Die aktuelle Version der Excelanwendung ist

UR2\_SingleAutoRun\_V10.xlsm

***UncertRadio* - Version 2.4.21 2023/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 20.02.2023

**Überarbeitung der Windows-Hilfe:**

* Das Kapitel 6.12.3, in dem es um die Varianzen bei zufälligen Einflüssen geht, wurde deutlich erweitert, um die Gleichungen und Tests der beiden Verfahren A („nicht Impulse“) und B („Impulse, mit Einfluss“) ausführlicher darzustellen. Die beiden Fälle entsprechen den Beispielen 13 und 14 in ISO 11929-4.
* Das Kapitel 7.21 zur Option zum Testen der Berechnung von Einheiten wurde überarbeitet und deutlich erweitert.

**Hinzufügung eines Beispiel-Projekts:**

* Es wurden 13 Beispielprojekte hinzugefügt, die für die in ISO 11929-4:2022 behandelten Anwendungsbeispiele erstellt wurden:

Iso11929-4\_Example\_6\_EN.txp bis Iso11929-4\_Example\_17\_EN.txp

* Von den vorhandenen Beispielprojekten wurden die folgenden unter dem Aspekt der verwendeten Einheiten geändert und unter höherer Versionsnummer eingestellt:

Janszen-Sr-89-Sr-90\_V4\_DE.txp, Moreno-Sr90\_IAEA-135\_V3\_EN.txp,

sumEval\_mitteln\_V3\_DE.txp, sumEval\_summe\_V3\_DE.txp

**Überarbeitung des Programms:**

* Bei der Erstellung eines Reports wurde ein Ausgabefehler der erweiterten Unsicherheit des primären Werts beseitigt.
* Die Grafikausgabe für die MC Simulation wurde angepasst. Der Fehler wurde beseitigt, dass Zahlen der Achsenbeschriftung abgeschnitten wurden. Der Min-/Max-Wertebereich der X-Achse wurde etwas erweitert.
* In den Textviews im TAB Verfahren und im TAB Texteditor werden zur besseren Lesbarkeit andere Schriften verwendet.
* In dem Dialog zur Aufnahme der Daten einer Abklingkurve wurde bei der Wahl der Zeitbasis für Messdauern und Zählraten ein Tooltip-Text ergänzt, der auf den Unterschied hinweist, dass die in der Spalte Startdatum verwendete Einheit immer die Sekunde sein muss.
* Eine fehlerhafte Eingabe von Werten führte zufällig zu einem Programmfehler bei einer partiellen Ableitung. Dieser bisher nur einmal gefundene Fehler wurde beseitigt.
* Bei der MC-Simulation von t-verteilten Werten im Fall der Mittelwertberechnung mit zufälligen Einflüssen trat ein Fehler auf, wenn die zu mittelnden Werte „keine Impulsanzahlen“ waren. Bei einer Division durch sqrt((m-1)/(m-3)) wurde fälschlicherweise durch das Quadrat dieses Faktors dividiert. Der Fehler wurde beseitigt.
* Die Option zum Testen der Berechnung von Einheiten wurde überarbeitet und für die Anwendung deutlich verbessert. So konnte diese Option das erste Mal (nur zum Testen) in den „QC-Batch-Test“, d.h. in den Lauf über die Beispielprojekte integriert werden. Dabei wurden einige Projekte identifiziert, deren gefundene Abweichungen oder Besonderheiten in dem Hilfethema, Kapitel 7.21.6, besprochen werden.
* Für die Datei mit den Referenzwerten der Beispielprojekte wurde eine neue Version erstellt:

BatListRef\_v05.txt

***UncertRadio* - Version 2.4.20 2022/08**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 14.08.2022

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Mit der vorangegangenen Version trat bei der grafischen Darstellung in dem dafür vorgesehenen Fenster nur ein schwarzer Hintergrund auf, oder der rechte Rand dieser Grafikflächen wurde nicht vollständig dargestellt. Als Ursache dafür wurden zwei Einstellungen in GTK-Fortran gefunden. Nach deren Anpassungen werden die Grafiken wieder vollständig dargestellt.
* Ein weiterer zufällig gefundener Fehler betraf die Anwendung von Poisson MLE bei der linearen Entfaltung einer Abklingkurve. Für eine interne Ausgabeoption dazu wurde aus Versehen ein linearer Fit einbezogen, der aber das nachfolgende PMLE-Ergebnis verfälschte. Diese fehlerhafte Option wurde deaktiviert.

***UncertRadio* - Version 2.4.19 2022/08**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 01.08.2022

**Überarbeitung der Windows-Hilfe:**

* Die folgenden Kapitel wurden bearbeitet: 3.6.1 (Änderungen der Radiobuttons beim Gammaspektrometriedialog), 7.10, 7.11.1 und 7.16 (Abbildungen von Dialogen hinzugefügt). 7.16 beschreibt die Änderungen bei genannten Radiobuttons.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Die Verwendung des Dialogs „Werte der Spektrumsauswertung“ (Gammaspektrometrie) wurde hinsichtlich der Eingabewerte geändert. Alle Werte der Messgrößen sind ab dieser Programmversion nur als absolute Werte einzugeben. Nur die dazugehörigen Unsicherheiten können mit Hilfe der Radiobuttons als relative (in %) oder als absolute Werte eingegeben werden. Zu den Konsequenzen der Verringerung von bisher 7 auf 5 solcher Radiobuttons wird auf das Hilfekapitel 7.16 verwiesen.

***UncertRadio* - Version 2.4.18 2022/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 26.06.2022

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Ein Programmfehler wurde korrigiert, der bisher dazu führte, dass genau der letzte Buchstabe in der letzten Zeile im Textfeld für Gleichungen weggelassen wurde, wenn dahinter keine weiteren Leerzeichen folgten. Dadurch erzeugte das um einen Buchstaben gekürzte Symbol einen Warnhinweis über ein neues Symbol.
* Im TAB „Gleichungen“ wurde über dem Gleichungen-Textfenster ein Feld mit dem Label „Funktionen“ eingesetzt. Es dient dazu, bei Berührung eine Tooltip-Hilfe zu Namen spezieller Funktionen, wie zum Beispiel für die Quadratwurzel, anzuzeigen. Das Programm wurde ergänzt, indem für den natürlichen Logarithmus auch ln(x) anstelle von log(x) verwendet werden kann.

***UncertRadio* - Version 2.4.17 2022/03**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 31.03.2022

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Der Unterordner „share\locale\“ wird ab dieser Programmversion vom Ordner GTKUser64 in den bei der Installation angelegten UR2-Ordner verschoben. Dies dient der verbesserten Sprachenanpassung von Buttons.
* In einem Textfenster für Gleichungen bzw. Formeln konnte es vorkommen, dass durch das Editieren/Bearbeiten des Inhalts alleinstehende Linefeed-Zeichen (LF) vorkamen, d.h. ohne vorangehendes carriage return-Zeichen (CR); dies entspricht dem Zeilenende-Zeichen in Unix, währen in Windows aber CR+LF als Zeilenende-Kennung verwendet wird. Dies verursachte Probleme/Fehler bei der korrekten Interpretation der Formel in der betreffenden Zeile. Dieses Problem wurde dadurch gelöst, dass jedes alleinstehende LF-Zeichen (ohne vorangehendes CR-Zeichen) durch CR+LF ersetzt wird. Dies wurde für alle Textfenster umgesetzt, auch für dasjenige im TAB „Verfahren“.

Das Auftreten einiger Zeilen mit alleinstehendem LF als Zeilenende war auch in insgesamt 19 der txp Projektdateien zu finden (ausnahmslos DE-Dateien); die alleinstehenden LF wurden in diesen Projektdateien wie oben beschrieben ersetzt. Die modifizierten Dateien erhielten mit Hilfe von Windows-Powershell-Befehlen danach wieder die ursprünglichen Datum/Uhrzeit-Werte der letzten Änderung.

Falls andere Projektdateien mit diesem Problem auftauchen: diese am besten mit dem Editor Notepad++ öffnen, die Anzeige auf „alle Steuerzeichen anzeigen“ umstellen, das Zeilenende-Format erst auf Unix, danach wieder auf Windows umstellen und das Projekt speichern.

* Bei der linearen Entfaltung einer Abklingkurve konnte es vorkommen, dass beim erstmaligen Berechnen im „TAB Werte, Unsicherheiten“ eine viel zu große Unsicherheit der Ergebnisgröße angezeigt wurde. Dies konnte auch zu einer entsprechenden Fehleranzeige führen, dass die Nachweisgrenze nicht berechnet werden kann. Nach Wiederholung dieser Berechnung (Button Berechnung der Unsicherheiten) verschwand dieser Fehler wieder. Die Ursache war ein Umrechnungsfaktor von 60 (Minuten 🡪 Sekunden), der bei einer Zeitbasiseinheit von Minuten im Bereich des Eingabe-Dialogs der Daten der Abklingkurve doppelt angewendet wurde. Dieser Fehler wurde beseitigt.
* Bei der Anwendung von WTLS zur Auswertung einer Abklingkurve kam es zu Fehlern in der Berechnung, wenn der erste oder der zweite Fitparameter nicht gefittet oder fixiert werden sollte. Das Problem wurde dadurch gelöst, dass in den WTLS-Routinen das System von Gleichungen auf die Dimension tatsächlich zu fittender Parameter verkürzt wurde. Das Fixieren eines Parameters ist bei WTLS nicht erlaubt, so dass nun eine Fehleranzeige erfolgt, wenn „Fixieren“ für einen der drei Paramater selektiert wird.

***UncertRadio* - Version 2.4.16 2022/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 22.02.2022

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurde ein Fehler beseitigt, der die MC-Simulation derjenigen Projekte betraf, mit denen die Aktivität aus mehreren Gammalinien eines Radionuklids berechnet wird. In der MC-Routine des Programms wurde eine falsche Indexvariable verwendet.
* Der Menüpunkt „LSQ Export nach R?“ ließ sich mittlerweile nicht mehr aktivieren. Der Fehler wurde behoben.

***UncertRadio* - Version 2.4.15 2022/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 08.02.2022

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurde ein Fehler beim beseitigt, der bei der Ausgabe eines Reports zu einem Abbruch führte. Er entstand dadurch, dass dabei überlange Formeltext-Zeilen in zwei oder mehr Formeltext-Zeilen umgewandelt wurden, so dass das Formeltext-Array länger wurde als gestattet.
* Es wurde das kleinere Problem beseitigt, dass das Teile des Programmfensters unter dem TAB „Resultate“ vertikal auseinandergerückt erschienen.
* Für den Fortran-Compiler und das GTK3 wurden die jüngsten Updates berücksichtigt.

***UncertRadio* - Version 2.4.14 2022/01**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 18.01.2022

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* Das Beispiel-Projekt La140\_REMSPEC-4Linien-V2\_DE.txp enthielt im TAB Verfahren einen komplett veralteten Text. Dieser wurde aktualisiert und das Projekt in **La140\_REMSPEC-4Linien-V3\_DE.txp** umbenannt. Die Referenzdatei **BatListRef\_v04.txt** wurde entsprechend angepasst, ihr Name aber wurde nicht geändert.

**Überarbeitung des Programms:**

* Es bestand der Fehler, dass nach dem Schließen eines Projekts, ohne das Programm zu beenden, wurden die beiden Textfenster im TAB „Verfahren“ und im TAB „Gleichungen“ nicht zurückgesetzt. Dieser Fehler wurde behoben; es werden beim Schließen des Projekts weitere Variable zurückgesetzt.
* Der Programmcode wurde überarbeitet und inzwischen überflüssiger Code von nicht mehr benötigten Testroutinen entfernt. Die ausführbare Datei wurde damit um etwa 165 kB gekürzt.
* Im Menüpunkt „Report“ wurde ein inzwischen aufgetretener Fehler beseitigt, dass null Zeilen des Texts des TABs „Verfahren“ ausgegeben wurden.
* Der Datenaustausch mit den GTK-Textviews (TAB Verfahren, Felder für Gleichungen) wurde überarbeitet, um überflüssige Leerzeilen am Ende der jeweiligen Texte zu vermeiden.

***UncertRadio* - Version 2.4.13 2021/09**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 07.09.2021

**Überarbeitung der Windows-Hilfe:**

* Der Hilfe wurde folgende neue Kapitel hinzugefügt:

2.2.5 Vermeidung von „verdeckten“ Kovarianzen

2.2.6 Verwendung von Schaltvariablen in Gleichungen

2.2.7 Berechnung physikalischer Einheiten für abhängige Variable

3.3.2 Überarbeitung der physikalischen Einheiten in den Beispielen

7.21 Behandlung physikalischer Einheiten

7.21.1 Zusammenstellung von Basiseinheiten und abgeleiteter Einheiten

7.21.2 Erläuterung zur Berechnung von Einheiten abhängiger Größen

7.21.3 Ausführung des Tests der Einheitenberechnung

**Hinzufügung eines Beispiel-Projekts:**

* Mit der Einführung eines Tests von physikalischen Einheiten abhängiger Variablen wurden alle Beispielprojekte damit getestet. Für die danach erfolgten Änderungen der Projekte wird auf Kapitel [3.3.2](#URH_Ueberarbeitung_Einheiten_Projekte) verwiesen.

**Überarbeitung des Programms:**

* Im Zusammenhang mit physikalischen Einheiten wurde die Verwendung sogenannter **Schaltvariable** (oder **Trigger-Variable**) eingeführt. Deren Bedeutung wird in Kapitel 2.2.6 erläutert.
* Es wurde ein Menüpunkt eingeführt, der dazu dient, die **physikalischen Einheiten abhängiger Variable nach einem Algorithmus zu „berechnen“**. Dies kann im Anschluss an eine normale Auswertung als ein Test auf die Verträglichkeit der Einheiten abhängiger Variable verwendet werden, um Hinweise auf Einheitenfehler zu bekommen. Das Kapitel [2.2.7](#URH_Vorschau_phys_Einheiten) gibt eine kurze Einführung, die Details sind dann im dreiteiligen Kapitel [7.21](#URH_Behandlung_Einheiten) beschrieben.

Der Test beruht auf zwei neuen Tabellen über Basiseinheiten und abgeleitete Einheiten, die in Form von zwei neuen CSV-Dateien bereitgestellt werden. Die eigene Verwendung dieser Tabellen wird im Kapitel [7.21.1](#URH_Basiseinheiten_DE) beschrieben.

Dieser Test wird im jetzigen Stand sicherlich noch nicht perfekt funktionieren. Weitere Arbeiten daran sind geplant.

* Für die Datei mit den Referenzwerten der Beispielprojekte wurde eine neue Version erstellt:

BatListRef\_v04.txt

***UncertRadio* - Version 2.4.12 2021/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 18.06.2021

**Überarbeitung der Windows-Hilfe:**

* Die Hilfe wurde in den Kapiteln 3.4, 3.5, 3.63 (neu), 4.2 und 4.3 ergänzt. Die Ergänzungen betreffen Details zur Eingabe von Eingangsunsicherheiten im Falle coverin=2 sowie Hinweise zur Entfernung von Variablen (Symbolen), die nach erneuter Editierung der Gleichungen überflüssig geworden sind.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Bei den Buttons wurde das Problem beseitigt, dass die Zustände sensitiv oder nicht-sensitiv zuletzt optisch kaum unterscheidbar geworden waren.

***UncertRadio* - Version 2.4.11 2021/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 10.06.2021

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurde der Fehler beseitigt, dass bisher beim Speichern eines TXP-Projektes als CSV-Datei der Parameter coverin (Erweiterungsfaktor für die Unsicherheiten der Eingabewerte) fälschlich als coverf ausgegeben wurde.
* Durch ein Versehen beim Einführen des Kontrastmodus waren das Unsicherheitsbudget und die Ausgabefelder im TAB „Resultate“ wieder editierbar geworden. Dieser Nachteil wurde abgestellt.
* Bei der MC-Simulation war der Spezialfall für eine Impulsanzahl N verloren gegangen, intern N=1 zu setzen, wenn N=0 eingegeben wurde. Diese Ersetzung wurde wiederhergestellt.
* Es wurde eine Fehler beseitigt, der dazu führte, dass bei der Ausgabe eines Reports (Menü Bearbeiten) ausgerechnet der Satz von Bestimmungsgleichungen dabei unterdrückt wurde.

***UncertRadio* - Version 2.4.10 2021/04**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 16.04.2021

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Für die mit der Version 2.4.09 eingeführten Option des Kontrastmodus wurden noch einige auf Farben bezogene Änderungen ausgeführt. Das Farblayout wurde auch so angepasst, dass es dem vor der Version 2.4.09 vorhandenen wieder entspricht, solange kein Kontrastmodus-Eintrag in die Konfigurationsdatei UR2\_cfg.dat oder ContrastMode=F eingetragen wird. Durch einen Programmfehler bei einem bestimmten Farbcode konnten mehr offenbar als eine Zelle für die Unsicherheitsformel der Bruttozählrate grün gefärbt werden.

***UncertRadio* - Version 2.4.09 2021/04**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 06.04.2021

**Überarbeitung der Windows-Hilfe:**

* Im Kapitel 3.6.1 wurden die Erläuterungen zum Punkt mit der Bezeichnung @Means: der Struktur der txp-Projektdatei erweitert. Das damit in Verbindung stehende Kapitel 6.9.1 wurde ebenso erweitert, so dass die Verwendung der verschiedenen Mittelwerttypen klarer wird.
* Im Kapitel 6.9.2 wird die geänderte Eingabe von Werten beschrieben, die in die Tabelle der Einzelwerte für einen Mittelwert einzugeben sind.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Der für die Mittelwertbehandlung von Datensätzen verwendete Dialog „Mittelwerte von Eingangsgrößen“ wurde überarbeitet und ist damit anwendungssicherer geworden. Die Eingabe von Einzelwerten wurde verbessert, indem nach der erfolgten Eingabe eines Wertes die jeweils nachfolgende Zelle schon für die Eingabe geöffnet wird.

Die Anzahl der Eingabewerte wurde von 40 auf 200 erhöht.

* Für alle Dialoge des Programms wurde die Eingabe eines Escape-Zeichens unterbunden. Damit wird der folgende fatale Fehler behoben. Ein im Dialog eingegebenes ESC, ggf. auch unbeabsichtigt, führte bisher dazu, dass der Dialog intern von GTK gelöscht wurde, ohne dass die zum Dialog gehörige interne GTK-Loop beendet wird. Das Weiterlaufen dieser Schleife führt dann dazu, dass das Programm nicht mehr bedienbar ist, und auch nicht korrekt beendet werden kann.
* Das Programm wurde um eine Option „Kontrastmodus“ im Dialog „Optionen“ erweitert. Bei Aktivierung des Kontrastmodus wird das gesamte Farblayout umgestellt, es wird mit weißer Schrift auf dunklem Untergrund kontrastreicher. Zusammen mit dem schon implementierten Zoomen von Grafiken und der Möglichkeit, die Schriftgröße zu vergrößern, stellt dies einen kleinen Beitrag zur Barrierefreiheit dar. Dazu kann in der Konfigurationsdatei UR2\_cfg.dat unter [Local] zur Voreinstellung eine Zeile mit dem Eintrag ContrastMode=T angehängt werden (T: true, F: false).

***UncertRadio* - Version 2.4.08 2021/03**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 25.03.2021

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Das GUI zur Seite im TAB „Resultate“ wurde intern neu organisiert; inhaltlich ändert sich dabei aber nichts.
* Das beim Aufruf der MC-Simulation erscheinende Grafikfenster wurde um eine Option zum Zoomen (in drei Stufen) ergänzt.
* Es wurden noch wenige Fehler im Programm gefunden und beseitigt. Darunter ein bisher erst einmal aufgetretener Fehler im Bereich der Interpretation der Gleichungen.

***UncertRadio* - Version 2.4.07 2021/03**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 11.03.2021

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es erfolgte eine Überarbeitung der Anwendung des Erweiterungsfaktors coverf für die Ausgabe der Unsicherheit der Ergebnisgröße. Üblicherweise wird coverf=1 verwendet. Wenn coverf=2 (d.h. k=2) gesetzt wird (Dialog Optionen), werden im TAB „Resultate“ die mit coverf multiplizierten Unsicherheiten ausgegeben. Das war bisher auch schon der Fall, es wurden aber weitere Ausgabemöglichkeiten daraufhin angepasst, wie z.B. bei Batch-Auswertungen die Ausgabe in die dazugehörigen Dateien. Auch im Aufruf eines Reports wird dieser Faktor berücksichtigt. Die Unsicherheit der Ergebnisgröße in den TABS „Werte, Unsicherheiten“ und „Unsicherheitsbudget“ wird dagegen nicht mit coverf multipliziert. Auch bei der mit dem Menüpunkt „QC-Batch-Test“ erzeugten Ausgabedatei wird coverf nicht angewendet.
* Für die Eingabe von Unsicherheitswerten aus einer Projektdatei steht ab dieser Version der neue Parameter coverin zur Verfügung. Dieser dient dazu, bei Unsicherheiten von Eingangsgrößen, die bereits mit einem Erweiterungsfaktor behaftet sind, diesen Erweiterungsfaktor beim Einlesen wieder herauszunehmen, bevor diese Unsicherheiten in die internen Berechnungen in UncertRadio eingehen. Unsicherheiten, die im TABS „Werte, Unsicherheiten“ eingegeben werden, müssen weiterhin Standardunsicherheiten sein.
* Die Werte der Parameter coverf und coverin können im Dialog „Optionen“ unter „Erweiterungsfaktor Ausgabe“ oder „Erweiterungsfaktor Eingabe“ geändert werden. Standardwerte sind 1.0 für beide Parameter.
* Es wird angenommen, dass Erweiterungsfaktoren bei solchen Eingangsdaten, die Bestandteil von Kurvendaten sind, auf die das Least squares-Verfahren angewendet wird, gleich 1 sind. Dies betrifft die Eingangsdaten, wie z.B. Zählraten-Unsicherheiten im Dialog für die Daten einer Abklingkurve, oder eine Kalibrierkurve oder die Eingangsdaten im Gammaspektrometrie-Dialog für die Auswertung von mehreren Gammalinien mittels gewichtetem Mittelwert.
* Wenn im TAB „Werte, Unsicherheiten“ in der Tabelle für Kovarianzen eine erste Zeile eingegeben wurde, mit einer Formel für die Kovarianz, aber ohne Wert dazu, kam es zu einem Fehler. Dies Problem wurde behoben.

***UncertRadio* - Version 2.4.06 2021/01**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 31.01.2021

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurden noch wenige weitere Allocation-bezogene Fehler beseitigt, die das Programm zum Abbruch bringen konnten. Im Zusammenhang mit nachträglichem Editieren der Gleichungen wurde ein Fehler beseitigt, der beim Löschen von nicht mehr benötigten Zeilen in der Symboltabelle dazu führte, dass für nachfolgende Zeilen die Einträge des Verteilungstyps nicht mehr zu den Symbolen passten.
* Für die Funktion SumEval wurde die Ausgabe von Warnungen ergänzt, wenn die Eingabe der Argumente dieser Funktion hinsichtlich der Anzahl von Variablen nicht konsistent ist (Anzahl der eingegebenen Variablen ungleich dem zweiten Argument).

***UncertRadio* - Version 2.4.05 2020/12**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 12.12.2020

**Überarbeitung der Windows-Hilfe:**

* Im Hilfe-Kapitel 7.13 wurde ein Abschnitt ergänzt, in dem auf die Besonderheit bei partiellen Ableitungen hingewiesen wird.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Die im Sommer 2020 begonnene Einführung von zuordnungsfähigen (allocatable) Arrays führte bisher im Bereich der Monte Carlo Simulation noch zu Problemen mit unzureichender Dimensionierung. Die hierbei betroffenen Programmteile, insbesondere bei Vorliegen von Kovarianzen die Vorbereitung zum Erzeugen von Zufallswerten korrelierter Größen, wurden eingehend überarbeitet und angepasst. Danach ist es nun möglich, die Monte Carlo-Simulationen von z.B. fünfzig Beispielprojekten in einem Durchlauf ohne Programmabsturz rechnen zu lassen, wofür sich der Menüpunkt „Batchauswertung Projekte“ gut eignet. Bei der Gelegenheit wurde eine geringe Anzahl anderer Probleme, auch auf spezielle Projekte bezogen, ebenfalls beseitigt.
* Das bei der iterativen Berechnung von Erkennungs- und Nachweisgrenzen im Verlaufe der MC Simulationen benutzte Verfahren nach Brent kam des Öfteren nicht zu einer Lösung, besonders bei MC-Simulationen. Dessen Anwendung erfordert die Suche nach zwei den Lösungswert sicher einschließenden Anfangswerten (Startintervall). Hierzu wurde eine Verbesserung erreicht, so dass das Brent-Verfahren dann konvergiert.
* Bei der Anwendung des gewichteten Mittelwerts (UR-Funktionen Gamspk1(), SumEval()) wird die Unsicherheit der damit berechneten Aktivitätswerte zweifach berechnet: a) bei der Anwendung dieser beiden UR-Funktionen und b) durch die von UncertRadio unabhängig davon immer durchgeführte Unsicherheitsfortpflanzung für die Ergebnisgrößen. Bei a) werden die inversen Werte der Einzelwert-Varianzen als Gewichte verwendet. Diese Gewichte müssen dabei als konstant betrachtet werden, das bedeutet, die Gewichte werden dabei einmal als Funktion der Eingangsparameter berechnet und dürfen dann nicht mehr variiert werden. Das letztere aber passiert bei der von UR immer verwendeten Unsicherheitsfortpflanzung. Daraufhin wurde programmintern geregelt, dass diese Variation der Gewichte bei partiellen Ableitungen verhindert wird.

Diese Änderung betrifft die Beispiele (maximale Abweichungen nachher/vorher in %);

La140\_REMSPEC-4Linien-V2\_DE.txp 0.25 %

Mehr-Linien-Nuklid-Aktivitaet-V3\_DE.txp 0,46 %

sumEval\_mitteln\_DE.txp 0,49 %

**Überarbeitung der Excelanwendung:**

* Die aktuelle Version der Excelanwendung ist

UR2\_SingleAutoRun\_V9.xlsm

***UncertRadio* - Version 2.4.04 2020/11**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 23.11.2020

**Überarbeitung der Windows-Hilfe:**

* Die Kapitel 6.13 und 7.20 sind neu hinzugekommen.
* Im neuen Hilfe-Kapitel 7.20 wird beschrieben, wie in UncertRadio die für ISO 11929 Teil 2 benötigten Gammaverteilungen der Zählraten erzeugt werden, sowohl im Falle der Messzeitvorwahl (bisher schon möglich) als auch für Impulsvorwahl (neu).

**Hinzufügung eines Beispiel-Projekts:**

* 5 neue Beispiele wurden hinzugefügt:

sumEval\_summe\_DE.txp, sumEval\_mitteln\_DE.txp, Tritium\_4Bubbler\_used\_1-3\_DE.txp,

Tritium\_4Bubbler\_used\_2-3\_DE.txp, ImpulsVorwahl\_DE.txp

**Überarbeitung des Programms:**

* In bestimmten Fällen muss die Messung einer Aktivität so erfolgen, dass die Ergebnisgröße z.B. aus mehreren verschiedenartigen Messungen oder Messungen verschiedener Kompartimente einer Probe zu bestimmen ist. Für diesen Zweck wurde für UncertRadio eine **neue Funktion SumEval** geschaffen. Mit einer Option können die Einzelaktivitäten entweder addiert oder gemittelt werden. Die Anwendung dieser Funktion wird im neuen Kapitel 6.13 beschrieben. Dafür wurde auch zwei Beispielprojekte erstellt.
* Neu ist in UncertRadio die Bearbeitung des Falles der Impulsvorwahl mit einer Gammaverteilung der dazugehörigen Zählrate(n). Näheres: siehe das neue Hilfe-Kapitel 7.20.
* Die vier neuen Beispielprojekte wurden in die für die QC-Batch-Auswertung vorgesehene Referenzdatei aufgenommen, von der jetzt die neue Version BatListRef\_v03.txt verwendet wird.
* Die Darstellung des Dialogs zur Eingabe von Einzelwerten für die Mittelwertberechnung wurde geringfügig erweitert.
* Die Ausgabe des Reports wurde überarbeitet, da es vorkommen konnte, dass bei Projekten mit mehreren Ergebnisgrößen die Ergebniswerte nicht immer in der richtigen Reihenfolge ausgegeben wurden. Ein Formatfehler bei der Ausgabe einer Intervalllänge wurde beseitigt.
* Für die gewichtete totale least squares-Analyse wird im Programm jetzt das besser passende Kürzel WTLS (anstelle von bisher TLSQ) verwendet.
* Von PLPLOT wurde die Version 5.15.0 implementiert.
* Die die Kommunikation zwischen UncertRadio und der Excelanwendung betreffenden Teile wurden überarbeitet. Siehe dazu auch Kapitel 5.2 der Hilfedatei.
* Für die neuen Tritium-Bubbler-Beispielprojekte (Gleichungen mit zwei Unbekannten) wurde die Linfit Funktion benutzt, obwohl eine Abklingkurve nicht direkt vorliegt. Sie wurde modifiziert, um die Xi-Funktionen durch Setzen der Option, die Xi(t) separat für jede einzelne Messung vorzugeben, berücksichtigen zu können.

**Überarbeitung der Excelanwendung:**

* In der Excelanwendung wurde eine zweite VBA-Routine zum Einlesen von CSV-Dateien ergänzt: Sub GetTextFile. Die Routine RUN\_UR\_AUTOSEP wurde erweitert, indem nach einer mit UR2 erfolgten Auswertung eines CSV-Projekts zusätzlich die Werte und Unsicherheiten der abhängigen Größen aus der Datei fort66.txt ausgelesen werden (VBA Sub read\_text\_data).
* Die aktuelle Version der Excelanwendung ist

UR2\_SingleAutoRun\_V9.xlsm

***UncertRadio* - Version 2.4.03 2020/09**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 30.09.2020

**Überarbeitung der Windows-Hilfe:**

* Die Kapitel 5.2 and 5.3 wurden erweitert.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurde ein Fehler beseitigt, der beim Abfragen der Monitor-Eigenschaften zum Programmabsturz führen konnte.
* Bei der MC-Simulation wird ab dieser Version auch der primäre Messwert der Ergebnisgröße und dessen Unsicherheit ermittelt.
* Es wurden neue Versionen des Fortran-Compilers und von GTK3 implementiert. Dies offenbarte zunächst weitere Programmfehler, zumeist im Zusammenhang mit der korrekten Memory allocation von Feldern; das erleichterte jedoch deren Beseitigung und verminderte damit das Programmabsturz-Risiko.
* Im Zusammenhang mit der vorgenannten Fehlersuche wurde insbesondere das spezielle Gammaspektrometrie-Tool für die Auswertung einer Aktivität aus mehreren Gammalinien komplett überarbeitet und anwendungssicherer gemacht. Dieses Tool ist hinsichtlich des Datenaustauschs zwischen dem GTK3-GUI und dem Fortran Code recht kompliziert, wenn hierbei die Effizienz-Kovarianzen berücksichtigt werden, komplizierter als der Fall der Auswertung von Zerfallskurven.
* Wenn in einem einzulesenden Projekt für eine Größe der Symboltyp ‘p‘ definiert wurde, wurde dieser Wert mit ‘a‘ bzw. ‘u‘ überschrieben. Dieser Fehler wurde behoben.
* Das GUI des Menüs Optionen – Modelltyp funktionierte nicht mehr richtig: es wurde immer der erste Modelltyp (die Standardeinstellung) angezeigt. Intern wurde jedoch der richtige Modelltyp verwendet. Der Anzeigefehler wurde beseitigt.

**Überarbeitung der Excelanwendung:**

* In der Excelanwendung wird die schon vorhandene **Batchdatei** UR2\_Start\_xls.bat standardmäßig zum Starten von UncertRadio aus der Excelanwendung heraus verwendet. Für die im VBA-Teil der Excelanwendung erfolgten Änderungen wird auf die Abschnitte 5.2 und 5.3 verwiesen. Es wurde ein Modul RUN\_UR\_AUTOSEP eingeführt, das im Vergleich zum VBA-Modul SINGLERUN\_UR eine Auswertung ohne Zwischenspeichern der UR-Ergebniswerte in der Datei AutoReport-Result.csv ausführt.
* Die aktuelle Version der Excelanwendung ist

UR2\_SingleAutoRun\_V8.xlsm

***UncertRadio* - Version 2.4.02 2020/07**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 16.07.2020

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurden zwei Fehler beseitigt, die bei der Überarbeitung von Gleichungen im TAB Gleichungen auftreten und zum Programmabsturz führen konnten.
* Es wurde ein Fehler eliminiert, der nach der Bearbeitung von Gleichungen gelegentlich zum Verschwinden eines Eingangswertes und/oder eines Unsicherheitswertes führte. Ein weiterer beseitigter Fehler bestand darin, dass nach Änderung von Werten im TAB „Werte, Unsicherheiten“ gelegentlich bestimmte Werte erst nach zweimaliger Betätigung des „Berechnen…“ Buttons in der Tabelle angezeigt wurde.

***UncertRadio* - Version 2.4.01 2020/07**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 08.07.2020

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* In einer Routine für die Durchführung der MC-Simulation wurden zwei Fehler beseitigt:
* Durch eine begrenzte Länge eines Feldes konnten zur Simulation nicht mehr als 3 Runs ausgeführt werden; ein 4. Run führte zum Programmabsturz;
* Anfang Januar 2020 wurde eine spezielle Behandlung einer als gamma-verteilt anzusetzender Bruttozählrate abgeschaltet. Dies stellte sich erst jetzt als Irrtum heraus und wurde wieder aktiviert. Leider trat dieser Fehler nicht bei der QC-Batch-Auswertung auf, so dass er erst jetzt beseitigt werden konnte. Dieser Fehler trat ab der Version 2.3.05 auf.

***UncertRadio* - Version 2.4.00 2020/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 05.06.2020

**Überarbeitung der Windows-Hilfe:**

* Der Abschnitt 2.2 wurde etwas erweitert und strukturiert;
* Der Abschnitt 2.3 „Gleichungen als Baum-Topologie“ wurde neu aufgenommen; die Nummerierung der in Kapitel 2 nachfolgenden Abschnitte erhöht sich;
* In Abschnitt 6.11 wurde die Liste der implementierten Verteilungen aufgenommen;
* im Abschnitt 5.1 wurde der Hinweis auf die BATSER-Alternative zur „seriellen Auswertung eines Projekts“ aufgenommen.

**Hinzufügung eines Beispiel-Projekts:**

* ein Beispiel mit längeren Symbolnamen, Ra226\_U235-bei-186keV\_DE\_long.txp, das inhaltlich mit dem schon vorhandenen Ra226\_U235-bei-186keV\_DE.txp übereinstimmt.
* Im Sr-89/Sr-90-Projekt der PTB erfolgte eine kleine Umstellung in den Gleichungen (siehe dazu das Projekt selbst), ohne dass die Ergebnisse sich ändern; der neue Projektname ist:

Janszen-Sr-89-Sr-90\_V2\_DE.txp

**Überarbeitung des Programms:**

* Der Fortran-Anteil des Programms wurde komplett überarbeitet, speziell durch die Verwendung von String-Feldern unterschiedlicher Stringlängen; siehe auch Abschnitt 2.4. Damit wurden intern einige Dinge vereinfacht, z.B. das Ein- und Auslesen der Textfelder für die Verfahrensbeschreibung und die Aufnahme der Gleichungen. Es erfolgte eine weitreichende Überarbeitung, in deren Verlauf sicher auch Schwächen des bisherigen Programms beseitigt werden konnten.
* Es stellte sich heraus, dass bei dem Lauf des QC-Batches einerseits einige Projekte mehrfach in der Referenzdatei BatListRef-v01.txt vorkamen, andererseits für Projekte mit mehr als einer Ergebnisgröße nur die erste davon ausgewertet wurde. Dies wurde behoben und die alte Referenzdatei in die neue mit dem Namen BatListRef-v02.txt überführt.
* Zu dem über den UR2-Menüpunkt „Serielle Auswertung eines Projekts“ zu startenden Batch wurde eine Alternative hinzugefügt. Damit wird dies nicht aus UR2 selbst heraus gestartet, sondern über einen Kommandozeilenbefehl. Dazu wurde das Kürzel AUTO um das Kürzel BATSER ergänzt. Der Befehl für das in Abschnitt 5.6 angeführte Beispiel lautet:

>uncertradio.exe BATSER J-ALUFT-Sr89-Sr-90\_DE.txp J\_Aluft\_serial\_DE.csv

Siehe auch Abschnitt 5.1.

* Nach einer neueren Version von GTK3 wurden einige der Icons aus der Toolbar nicht mehr auf dem bisher üblichen Weg gefunden. Die bisher verwenden Icons wurden bei GTK3 durch neue ersetzt, die überwiegend schwarz sind; die bisherigen wurden unter „legacy“ eingeordnet. Das erforderte eine Programmanpassung, indem die betreffenden legacy-Icons jetzt in den kürzlich schon eingeführten Unterordner „Icons“ des UR2-Ordners aufgenommen wurden, nachdem das GUI so eingestellt wurde, dass diese Icons über ihre Grafik-Dateinamen identifiziert werden.
* Mittlerweile funktionierte im Dialog zum Öffnen einer Datei der Ordner „zuletzt geöffnet“ nicht mehr (der recent manager). Hierzu war eine kleine Programmanpassung zur Anpassung an das geänderte Format der Dateischreibweise in der Datei „recently-used.xbel“ in Windows nötig.
* Es wurde ein neues Hilfe-Kapitel 2.3 „Gleichungen als Baum-Topologie“ in die CHM-Hilfe aufgenommen. Darin wird beschrieben, wie programmintern mittels einer neuen Routine aus den vom Benutzer eingegebenen Gleichungen und ihrer hierarchischen Struktur versucht wird, die einzelnen Zählraten-Symbole zu identifizieren, die zum Ausdruck der Nettozählrate beitragen. Bei der Ausführung des QC-Batches wird jetzt eine Kurzfassung der Ergebnisse daraus in die Datei fort.64 geschrieben. Dieses Tool als solches ist sicher noch nicht komplett ausgereift.
* Das Format der für das Saven von Resultaten verwendeten Datei UR-Saved-Results.csv, sowie der in der unten angesprochenen Excelanwendung verwendeten AutoReport-Dateien, wurde leicht geändert (Änderung der Spaltenüberschriften; zusätzlich zwei Spalten für das kürzeste Wahrscheinlichkeitsintervall). Daher sollten bereits vorhandene Autoreport-Dateien umbenannt werden, damit von UncertRadio dann die neuen Versionen angelegt werden können.
* Das Programm ist nun für die Auswertung von Messungen mit Impulsvorwahl vorbereitet, indem der Messdauer als Variable eine Erlangverteilung zugeordnet wird (d.i. eine Gammaverteilung für ganzzahlige Impulse): Auswahl von „Npreset“ als Verteilungstyp. Ein Beispiel dazu steht noch aus.

**Überarbeitung der Excelanwendung:**

* Für die Excelanwendung wurde eine **Batchdatei** UR2\_Start\_xls.bat zum Starten von UncertRadio aus der Excelanwendung heraus eingeführt. In der Tabelle6 wurde den Projektnamen der jeweilige Unterordnername vorangestellt. Das Format der CSV-Ausgabedatei wurde leicht geändert. Für die im VBA-Teil der Excelanwendung erfolgten Änderungen wird auf die Abschnitte 5.1 und 5.2 verwiesen.
* Die aktuelle Version der Excelanwendung ist

UR2\_SingleAutoRun\_V7.xlsm

* Für VBA-Kontrollausgaben wird darin jetzt debug.print verwendet.

***UncertRadio* - Version 2.3.08 2020/04**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 06.04.2020

**Überarbeitung der Windows-Hilfe:**

* ergänzt im Kapitel 1.3 (Verwendung mehrerer Monitore)

**Hinzufügung eines Beispiel-Projekts:**

* keins.

**Überarbeitung des Programms:**

* Werden an einem Arbeitsplatz mehrere Monitore verwendet, kann es passieren, dass das UncertRadio-Fenster nur zum Teil sichtbar ist oder sich über mehr als 1 Monitorbreite erstreckt. Die für diese Version 2.3.08 durchgeführten Erweiterungen und Anpassungen beziehen sich auf die neu eingeführte Unterscheidung zwischen einem „Screen“ und mehreren innerhalb des Screens liegenden „Monitoren“.
* Dazu wurde die Konfigurationsdatei UR2\_cfg.dat um einen Eintrag „Monitor#=“ erweitert.

Es wird auf den neuen Abschnitt „Verwendung mehrerer Monitore“ am Ende von Kapitel 1.3 verwiesen.

***UncertRadio* - Version 2.3.07 2020/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 28.02.2020

**Überarbeitung der Windows-Hilfe:**

* ergänzt im Kapitel 1.3 (Sprachumstellung)

**Hinzufügung eines Beispiel-Projekts:**

* keins.

**Überarbeitung des Programms:**

* Das komplette Übersetzen der Dialogbestandteile bei laufendem Programm wurde bisher dadurch erschwert, dass die „GTK stock buttons“ sich nur beim Start des Programms, vor dem Aufbau des Fensters, übersetzen ließen. Da GTK die bisherige interne Übersetzung langfristig aufgibt, wurde diese Buttons in normale Buttons umgewandelt, die aber nahezu so aussehen wie vorher. Dies erlaubt nun die nahezu vollständige Sprachumstellung über das Menü – Optionen im laufenden Programm. Weiteres siehe Kapitel 1.3.
* Im Zuge der Ersetzung der „stock buttons“ werden zusätzlich neue Icon-Dateien in dem Programmordner installiert.
* Nach dem Programmstart wurde bisher die kleine Datei „fort.66“ erzeugt, die beim Programmstart durch Doppelklick auf eine Projektdatei in den Ordner dieses Projekts geschrieben wurde. Diese fällt ab der Version 2.3.07 weg, indem die darin erzeugten Ausgaben nun in einem internen Array gespeichert und nachträglich am Beginn der Datei „fort66.txt“ mit ausgegeben werden.
* Der neue Button im Tab „Gleichungen“ erhält das Label „Symbol ändern“.
* Für das Arbeiten mit zwei Bildschirmen nebeneinander wird die Anzahl der Pixel größer, ggf. doppelt so groß in der Breite. Um Konflikte zu vermeiden, werden in UncertRadio die maximalen Werte für Breite und Höhe auf 1920 und 1080 beschränkt.
* Es wurde ein Problem beseitigt, dass das Programm bei sehr langen Kommando-Argument-Strings beendet wurde.
* Bei einem Projektbeispiel mit linearer Entfaltung wurde bei der Anwendung des „total least-squares“-Verfahrens eine im Vergleich zur Monte Carlo-Simulation zu kleine Unsicherheit der Ergebnisgröße festgestellt. Als Ursache dafür wurde ein Fehler im Programm identifiziert, der aber nur dann erkennbar wurde, weil die relative Unsicherheit einer bestimmten Nachweiswahrscheinlichkeit recht groß war, größer als 15 % in dem Beispiel. Der Fehler wurde beseitigt.
* Für das Arbeiten mit zwei Bildschirmen nebeneinander wird die Anzahl der Pixel größer, ggf. doppelt so groß in der Breite. Um Konflikte zu vermeiden, werden in UncertRadio die maximalen Werte für Breite und Höhe auf 1920 und 1080 beschränkt.
* Es wurde ein Problem beseitigt, dass das Programm bei sehr langen Kommando-Argument-Strings beendet wurde.
* Bei einem Projektbeispiel mit linearer Entfaltung wurde bei der Anwendung des „total least-squares“-Verfahrens eine im Vergleich zur Monte Carlo-Simulation zu kleine Unsicherheit der Ergebnisgröße festgestellt. Als Ursache dafür wurde ein Fehler im Programm identifiziert, der aber nur dann erkennbar wurde, weil die relative Unsicherheit einer bestimmten Nachweiswahrscheinlichkeit recht groß war, größer als 15 % in dem Beispiel. Der Fehler wurde beseitigt.

***UncertRadio* - Version 2.3.06 2020/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 10.02.2020

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins.

**Überarbeitung des Programms:**

* Beim Aufrufen der Funktion

Gamspk1(E, tlive)

ist es ab der Programm-Version 2.3.06 2020/02 erlaubt, für die Livetime anstelle von tlive auch andere Symbolnamen zu verwenden. Dieses Symbol muss zwingend das zweite Argument von Gamspk1 sein.

* Im Tab „Gleichungen“ wurde oberhalb des Textfensters für Gleichungen zusätzlich der Button „Symbol-Name“ eingesetzt, der dieselbe Funktion hat wie der Menüpunkt „Bearbeiten – Ändern des Symbol-Namens“, aber bei der Bearbeitung von Gleichungen „sichtbarer“ ist. Dies soll dazu beitragen das nachträgliche Ändern von Symbol-Namen NICHT direkt in den Gleichungen vorzunehmen.
* Bisher war es möglich, das Programm UncertRadio mehr als einmal zu starten. Dies kann generell zu Fehlern führen, speziell auch beim Speichern der Projekte. Dies wird jetzt unterbunden, indem nach dem Starten des Programms die Windows Taskliste nach eventuell schon vorhandenen UncertRadio-Prozessen (Instanzen) abgefragt wird. Falls das der Fall ist, beendet sich das später gestartete Programm mit einem Warnhinweis.

***UncertRadio* - Version 2.3.05 2020/01**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 09.01.2020

**Überarbeitung der Windows-Hilfe:**

* Die Kapitel 6.11 „Spezielle Verteilungen und ihre Eigenschaften“ und 6.12 „Bruttogröße: Varianzinterpolation für einen Mittelwert“ wurden neu hinzugefügt, wobei 6.12 einen Teil des Kapitels 4.3 ersetzt.

**Hinzufügung eines Beispiel-Projekts:**

* Im Zuge der in den neuen Kapiteln 6.11 und 6.12 vorgestellten Weiterentwicklungen des Programms wurde neue Versionen von zwei alten Projektdateien hinzugefügt:

ISO-Beispiel-2a\_V2\_DE.txp and ISO-Beispiel-2b\_V2\_DE.txp

Das alte Projekt ISO-Beispiel-2a\_DE.txp wird in der Datei BatListRef\_v01.txt nicht mehr berücksichtigt, da die darin enthaltenen Interpolationsformel den Beitrag von urel(w) nicht berücksichtigte (-> 5 % Abweichung).

**Überarbeitung des Programms:**

* Bei der „Batch-Auswertung Projekte“ wurde dafür gesorgt, dass dabei erzeugte Plotdateien nicht mehr in den Ordner mit den Projekten geschrieben werden.
* Bei der Ausgabe eines Reports wurde die Angabe der Länge eines Konfidenzintervalls korrigiert.
* Enthält die „grüne Zelle“ in der Tabelle „Werte, Unsicherheiten“ eine recht lange Formel, konnte das Programmfenster recht breit werden; es blieb auch so breit, wenn ein anderes TAB selektiert wurde. Dies wurde geändert: für andere selektierte TABs wird die Fensterbreite wieder reduziert.
* Die Compilierung der CHM-Hilfe wurde so geändert, dass die mathematischen Symbole und Formeln nicht mehr zu klein im Vergleich zur Textgröße sind.
* In dem Dialog, der die Parameter der kombinierten Binomial-/Poisson-Verteilung bei langer Messdauer eines kurzlebigen Radionuklids (Kapitel 6.10) abfragt, wurde ein Hilfe-Button ergänzt.
* Im Vorgriff auf mögliche Anwendungen sind neben der schon vorhandenen Gammaverteilung auch die Beta- und die t-Verteilung implementiert worden. Siehe dazu das neue Hilfe-Kapitel 6.11.
* Für eine Bruttogröße, für deren Wert der Mittelwert einer Messreihe verwendet wird, war es bislang erforderlich, zum Zwecke der Interpolation zwischen zwei bekannten Werten der Varianz eine Formel in die „grüne Zelle“ (Standardabweichung der Bruttogröße) einzutragen. Dies wird ab dieser Programmversion nicht mehr nötig sein, da die dazu nötigen Berechnungen jetzt innerhalb des Programms erfolgen. Dazu wird auf das neue Hilfe-Kapitel 6.12 verweisen.
* In diesem Zusammenhang wurde das Format einer Projektdatei um einen Eintrag „refmean=x“ ergänzt, falls ein **Referenz**-Datensatz verwendet wird.

***UncertRadio* - Version 2.3.04 2019/11**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 02.12.2019

**Überarbeitung der Windows-Hilfe:**

* Das Kapitel 5.7 „Projekte im Batch-Mode auswerten“ wurde neu hinzugefügt.

**Hinzufügung eines Beispiel-Projekts:**

* In folgenden schon vorhandenen Projekten wurde die Zeile „GamDistAdd=1 ergänzt (die bekannten Resultate ändern sich dadurch nicht):

Gamma\_dist\_de.txp und Gamma\_dist\_en.txp

**Überarbeitung des Programms:**

* Der Menüpunkt „**Bearbeiten – Batch-Auswertung Projekte**“ wurde eingeführt, der die Auswertung mehrerer Projekte im Batch-Mode ermöglicht. Dabei kann auch die MC-Simulation einbezogen werden. Als Ergebnis erhält man eine CSV-Datei, die in Form von kleinen Tabellenblöcken die Ergebnisse für jedes Projekt und jede Ergebnisgröße enthält. Zur Beschreibung wird auf den im Kapitel 5 neu erstellten Abschnitt 5.7 verwiesen.
* Bei dem Menüpunkt „**Bearbeiten – Serielle Auswertung**“ wurde die bisher nicht berücksichtige Abhängigkeit des Listentrennzeichens in CSV-Dateien von der Sprache korrigiert.
* Die Handhabung des Dialogs zur Aufnahme von Einzelwerten einer Messreihe (Toolbar-Icon ; Kapitel 6.9.2) wurde verbessert. Die Option des „klassischen“ Mittelwerttyps wurde erweitert, dass er auch für mehr als drei Einzelwerte angewendet werden kann.
* Die Positionierung des Grafik-Fensters der MC-Simulation wurde geändert, so dass es nach Möglichkeit rechts vom Programmfenster erscheinen soll. Das Programmfenster wurde dazu nach links verschoben.
* Die Modifikation des Formelinterpreters für die Anwendung der UncertRadio-internen Funktion uval(x) (siehe Kapitel 7.8) in den vom Benutzer eingegebenen Gleichungen wurde erweitert.
* Bei der Anwendung der MC-Simulation wurde die Anzahl der Iterationen, die im Falle der Erkennungsgrenze erfolgen, um den Mittelwert der betreffenden Verteilung näher an null zu bringen, auf 7 beschränkt. Das reduziert die dafür benötigte Rechenzeit ohne merklichen Genauigkeitsverlust.
* In bestimmten Fällen muss in die grüne Zelle im TAB Werte, Unsicherheiten eine Unsicherheitsfunktion der Bruttogröße in Form einer Interpolationsformel eingegeben werden. Dabei kann es vorkommen, dass in dieser Formel Größen auftreten, die in der Wertetabelle oberhalb der grünen Zelle stehen, die also erst später berechnet werden. Das Programm wurde so erweitert, dass die Werte solcher Größen vorab ermittelt werden und damit für die Berechnung in der grünen Zelle zur Verfügung stehen.
* Der interne Parameter **GamDistAdd**, der bei der Wahl der Gammaverteilung für Impulsanzahlen zum Tragen kommt (siehe Dialog im Menü Optionen – Voreinstellungen), wird gemäß der ISO 11929-2019 ab jetzt mit dem Wert 0,0 vorbesetzt. Der entsprechende Wert aus schon vorhandenen Projekten wird jedoch beim Einlesen des Projekts beibehalten.
* Es gibt zwei ältere Projekte (Gamma\_dist\_de.txp, Gamma\_dist\_en.txp), bei denen die Impulsanzahlen gamma-verteilt benutzt werden und GamDistAdd noch nicht explizit in der Projektdatei enthalten war. Dann wurde bisher beim Einlesen GamDistAdd auf 1 gesetzt. Die bisher in der Datei BatListRef\_v01.txt dokumentierten Werte wurden also mit GamDistAdd=1 gerechnet. Da ab der vorliegenden UncertRadio-Version das GamDistAdd nicht mehr automatisch 1 gesetzt wird, wurde bei beiden Projektdateien die Zeile GamDistAdd=1 ergänzt. Dadurch bleibt die Konsistenz der jetzigen Programmversion mit der Referenzdatei BatListRef\_v01.txt erhalten.

***UncertRadio* - Version 2.3.03 2019/10**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 15.10.2019

**Überarbeitung der Windows-Hilfe:**

* Die Kapitel: 3.9 „Programm-Test“ und 6.10 „Messung eines kurzlebigen Radionuklids mit vergleichsweise langer Messdauer“ wurden neu hinzugefügt.

**Hinzufügung eines Beispiel-Projekts:**

* neues Projekt zum Kapitel 6.10: Ac228\_binomial\_DE.txp
* in folgenden schon vorhandenen Projekten wurde die Zeile „ModelType=GUM\_restricted“

ergänzt (die bekannten Resultate ändern sich dabei nicht):

Calibration-of-weight-Cox-2001\_EN.txp; Neutron-Dose-Cox-2006\_DE.txp;

Kessel-1-2006\_DE.txp; Kessel-2a-2006\_DE.txp; Kessel-2b-2006\_DE.txp;

Wuebbeler-Ex1\_DE.txp; Wuebbeler-Ex2\_DE.txp

**Überarbeitung des Programms:**

* Zur Verbesserung der Qualitätskontrolle wird unter dem Menü Optionen – **QC-Batch-Test** ein Test zur Verfügung gestellt, mit dem alle mit UncertRadio installierten Beispiel-Projekte automatisiert durchgerechnet werden (MC-Rechnungen ausgenommen). Die erhaltenen Ergebnisse werden mit den in einer Datei bereitgestellten Referenzwerten verglichen. Bei Abweichungen werden diese in einer Ausgabedatei festgehalten. Siehe das neue Kapitel 3.9 in der Hilfedatei.
* Das Programm wurde für den Spezialfall erweitert, dass die Verteilung der Bruttoim­puls­anzahl einer Überlagerung von Binomialverteilung (Probenbeitrag) und Poissonver­teilung (Nulleffekt) folgt. Dieser Fall kann bei der Messung der Aktivität eines Nuklids mit kurzer Halbwertszeit auftreten, wenn das Produkt deutlich größer als 0,1 oder gar ist. Näheres dazu ist dem neuen Kapitel 6.10 der CHM-Hilfe zu entnehmen.
* Die Bedienbarkeit von Eingabefeldern wurde verbessert.

**Überarbeitung des Setup-Programms:**

* **Mit der Version 2.3.01 (Jan. 2019) wird eine geänderte Fassung der Batchdatei UR2\_Start.bat installiert, in der der Ordner GTKuser64\bin den ersten Eintrag der darin temporär modifizierten Path-Variablen darstellt; bisher wurde dieser Eintrag an die Path-Variable angehängt.** Siehe auch Abschnitt 1.2.

**Damit soll vermieden werden, dass das Runtime-System nicht auf DLL-Dateien gleichen Namens zugreift, die zu anderen Programmen gehören.**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.3.02 2019/05**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 05.05.2019

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins, aber die Datei J\_Aluft\_serial\_DE.csv als Beispiel für die mit dieser Version eingeführten seriellen Auswertung der Projektdatei J-ALUFT-Sr89-Sr-90\_DE.txp wurde korrigiert.

**Überarbeitung des Programms:**

* Mit der Version 2.3.01 trat der Fehler auf, dass bei der Ersterstellung eines Projekts die Typenzuordnung, „a“ wie abhängig bzw. „u“ wie unabhängig, falsch herauskam. Dieser Fehler wurde behoben.
* Das Layout von Tabellen im UncertRadio GUI wurde im Hinblick auf vollständige Sichtbarkeit von Zahlen in Zellen nochmals überarbeitet, so dass die betreffenden Spalten sich nicht zu sehr horizontal ausbreiten. Insbesondere der Dialog zur Aufnahme der Daten einer Abklingkurve ist deutlich handlicher geworden. Das Optimieren der Spaltenbreiten von Tabellen kann auch durch Doppelklicken auf den vertikalen Trennstrich zwischen zwei Spaltenköpfen erfolgen; zuvor ändert sich dabei die Form des Mauszeigers.
* Es werden neuere Versionen von GTK3+, GTK-Fortran und des Fortran-Compilers verwendet.

**Überarbeitung des Setup-Programms:**

* **Mit der Version 2.3.01 (Jan. 2019) wird eine geänderte Fassung der Batchdatei UR2\_Start.bat installiert, in der der Ordner GTKuser64\bin den ersten Eintrag der darin temporär modifizierten Path-Variablen darstellt; bisher wurde dieser Eintrag an die Path-Variable angehängt.** Siehe auch Abschnitt 1.2.

**Damit soll vermieden werden, dass das Runtime-System nicht auf DLL-Dateien gleichen Namens zugreift, die zu anderen Programmen gehören.**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.3.01 2019/03**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 13.03.2019

**Überarbeitung der Windows-Hilfe:**

* Abschnitt 5.6 ist neu hinzugekommen.

**Hinzufügung eines Beispiel-Projekts:**

* keins, aber die Datei J\_Aluft\_serial\_DE.csv als Beispiel für die mit dieser Version eingeführten seriellen Auswertung der Projektdatei J-ALUFT-Sr89-Sr-90\_DE.txp.

**Überarbeitung des Programms:**

* Bisher hatte das Beenden des UncertRadio die lästige Folge, dass danach für kurze Zeit eine Kaskade von Fenster-Rahmen kurz aufflackerte. Es ist bisher nicht gelungen, dies mit GTK-Mitteln zu verhindern. Dieser Effekt taucht nicht auf, wenn mit Code::Blocks das UncertRadio nicht als eine „GUI application“, sondern als eine „Console application“ erstellt wird. Das wiederum hat den Nachteil, dass in dem nun zusätzlich geöffneten CMD.exe Fenster viele GTK-Warnmeldungen auftauchen, die aber unerheblich sind. Fortran-Fehlermeldungen würden darin ebenfalls auftauchen. Der beste Kompromiss wurde in einer Modifikation der Startbat-Datei mit der Option /min gefunden. Vereinfacht bedeutet dies:

>start uncertradio es wird viel Text im CMD-Fenster ausgegeben;

>start /min uncertradio es erscheint kein weiterer Text im CMD-Fenster.

Näheres dazu siehe Abschnitt 1.3.

* Das Programm wurde auf einen 64-bit-Gfortran-Compiler umgestellt, so dass es unter Verwendung von 64-bit-Versionen von DLL-Bibliotheken nun als eine 64-bit-Anwendung verwendet wird. Die verwendeten Software-Produkte (siehe Abschnitt 2.3) wurden ebenfalls auf 64-bit-Versionen umgestellt.

Hinweis: wenn beim Starten des Programms eine Windows-Fehlermeldung auftritt, die einen Fehlercode (0xc000007b) angibt, bedeutet dies, dass z. B. aufgrund unzureichender Einstellungen in der PATH-Umgebungsvariable eine 32-bit-Version anstelle der 64-bit-Version einer DLL-Datei im System gefunden wird.

* Das gestartete Programm konnte bisher im Hauptspeicher recht groß werden. Dies ist auf große Arrays zurückzuführen, die vom Autor des Programms innerhalb von Bayes-Statistik-bezogener Test-Routinen verwendet werden, die von normalen ISO 11929-Anwendungen aber gar nicht benötigt werden. Die Zuweisung von Hauptspeicher (Allocation) für diese Arrays wurde daher abgestellt. Ebenso wurden die Längen der Arrays, welche die Listen der Symbole; Messwerte, Unsicherheiten usw. beinhalten, maßvoll gekürzt.
* Es wurde beobachtet, dass die Eingabe-Tabellen in den Dialogen für Abklingkurven, für Gammaspektrometrie und für Kalibrierfunktionen dazu neigten, sich auf dem Bildschirm sehr auszubreiten. Dies wurde durch eine Programmanpassung abgestellt, die eine maximale Stringlänge der Zellen einer Spalte zur Einstellung der Spaltenbreite verwendet.
* Im Abschnitt 6.9 wird die Änderung erläutert, dass bei der Verwendung von Datensätzen zum Zwecke der Mittelwertberechnung die dritte Option des Mittwerttyps auf „klassisch“ (nicht-Bayes’sche Interpretation) umgestellt wurde, was für Vergleichszwecke sinnvoll sein kann.
* Ein Fehler wurde beseitigt, der bei der Bearbeitung der Gleichungen zum Verschwinden einzelner Zellen der Spalten Einheit und Bedeutung der Symbol-Tabelle führen konnte.
* Der Menüpunkt „**Bearbeiten – Serielle Auswertung**“ wurde eingeführt, der die mehrfache Auswertung eines Projekts mit partiell geänderten Eingangsgrößenwerten/-unsicherheiten erlaubt. Zur Beschreibung wird auf den im Kapitel 5 neu erstellten Abschnitt 5.6 verwiesen.

**Überarbeitung des Setup-Programms:**

* **Mit der Version 2.3.01 (Jan. 2019) wird eine geänderte Fassung der Batchdatei UR2\_Start.bat installiert, in der der Ordner GTKuser64\bin den ersten Eintrag der darin temporär modifizierten Path-Variablen darstellt; bisher wurde dieser Eintrag an die Path-Variable angehängt.** Siehe auch Abschnitt 1.2.

**Damit soll vermieden werden, dass das Runtime-System nicht auf DLL-Dateien gleichen Namens zugreift, die zu anderen Programmen gehören.**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.2.11 2018/11**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 18.11.2018

**Überarbeitung der Windows-Hilfe:**

* 3 Projekte in Abschnitt 3.3 ergänzt.

**Hinzufügung eines Beispiel-Projekts:**

* Es wurden zwei Varianten eines UR2-Projekts beigefügt, in welchem das beim BSH (Hamburg) betriebene Gesamt-Gamma-Messverfahren in Meerwasser behandelt wird:

BSH\_Gesamt-Gamma\_var1\_DE.txp

BSH\_Gesamt-Gamma\_var2\_DE.txp

**Überarbeitung des Programms:**

* Das bei der vorangegangenen Version des Programms aufgetretene Flackern von Tooltips konnte beseitigt werden, nur beim Dialog zum Öffnen eines Projekts ist es nicht komplett beseitigt, auch nicht nach einem GTK3-Update auf Version 3.24.1.
* Bei der MC-Simulation wurde ein Fehler beim Aufruf einer MATMUL-Funktion beseitigt, der auftrat, wenn eine Kovarianz bei der MC-Simulation zu berücksichtigen war.
* Das bei Iterationen eingesetzte Verfahren nach Ridders für die Nullstellensuche wurde durch das effektivere Verfahren nach Brent ersetzt.
* Das Plotfenster wird jetzt beim erstmaligen Aufruf aus etwa der Mitte des Bildschirms nach rechts oben verschoben. Beim MC-Plot der Ergebnisgröße wurde die Skalierung der vertikalen Achse leicht geändert, was sich aber nur auswirkt, wenn ein großer Teil der MC-Werte kleiner als null ist.
* Die wesentlichen Zwischenergebnisse der MC-Simulation in Form von Tabellen wurden aus der Ausgabedatei fort66.txt in eine neue Datei MC\_Tables.txt verschoben.

**Überarbeitung des Setup-Programms:**

* **Mit der Version 2.2.11 (November 2018) wird eine geänderte Fassung der Batchdatei UR2\_Start.bat installiert, in der der Ordner GTKuser3\bin den ersten Eintrag der darin temporär modifizierten Path-Variable darstellt; bisher wurde dieser Eintrag an die Path-Variable angehängt.** Siehe auch Abschnitt 1.2.

**Damit soll vermieden werden, dass das Runtime-System nicht auf DLL-Dateien gleichen Namens zugreift, die zu anderen Programmen gehören.**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.2.10 2018/09**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 18.09.2018

**Überarbeitung der Windows-Hilfe:**

* keine

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Das bei einer früheren Version des Programms aufgetretene Problem in dem GTK-Dialog zum Öffnen einer Datei konnte nun durch eine Aktualisierung der GTK-Version auf 3.24.0 behoben werden. Dadurch wird vermieden, die GTK-Version auf einem älteren Stand einzufrieren.
* Im Menü Bearbeiten – Ändern des Symbolnamens wurde ein weiterer Fehler behoben. Dieser konnte auftreten, wenn eine Variable Rb in z.B. Rg geändert werden sollte; wenn dabei gleichzeitig eine Variable Rb0 vorlag, wurde letztere durch den Fehler fälschlicherweise ebenfalls umbenannt, in Rg0.

**Überarbeitung des Setup-Programms:**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.2.09 2018/09**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 12.09.2018

**Überarbeitung der Windows-Hilfe:**

* *Kapitel 2.2 (Ende), 2.3, 2.6, 3.3, 3.5 (Ende)*

**Hinzufügung eines Beispiel-Projekts:**

* neues Projekt: dwd\_sr89\_sr90\_TDCR\_Verfahren\_DE.txp

**Überarbeitung des Programms:**

* Die in UR2 verwendete GTK-Version wurde um eine Version zurückgesetzt, um den Dialog zum Öffnen einer Datei wieder wie gewohnt zu gestalten (das freie Manövrieren in dem Dateiverzeichnis war nicht mehr möglich).
* Das für die MC-Simulation verwendete Plotfenster wurde geringfügig vergrößert.
* Ein Fehler wurde beseitigt, bei dem eine recht große Unsicherheit dazu führen konnte, dass Erkennungs- und Nachweisgrenze nicht berechnet wurden.
* Beim Menü Bearbeiten – Ändern des Symbolnamens konnte es zu Programmabstürzen kommen. Der Fehler dazu wurde behoben; er lag in einer Routine, in der aus Versehen noch eine implizite Variablendeklaration auftrat, so dass fehlerhafte Werte-Zuordnungen entstehen konnten.
* Bei einem Programmaufruf des Typs „>uncertradio.exe d:\ur2\projekt.txp“ führte das Weglassen der Erweiterung „.exe“ zum Absturz. Dieser Fehler wurde beseitigt.

**Überarbeitung des Setup-Programms:**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.2.08 2018/06**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 25.06.2018

**Überarbeitung der Windows-Hilfe:**

* *Kapitel 6.7 wurde präzisiert*

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es wurden neuere Versionen von PLPLOT (5.13.0), GTK-Fortran (17.10), GFortran (7.3.0) und NuHelp (v2018.04.23) eingesetzt.
* Es wurde ein Problem bei der internen String-Wandlung (Locale – UTF8) identifiziert und beseitigt. Dies war wahrscheinlich die Ursache dafür, dass das Programm gelegentlich abstürzen konnte. Damit ist die Betriebssicherheit beim Arbeiten im UncertRadio-GUI erheblich besser geworden.
* Das beim Laden eines Projekts bisher gezeigte kleinere Fenster (LoadProject) wird ab dieser Version generell nicht mehr verwendet. Das beruhigt die Ansicht des sich aufbauenden GUI.
* Wenn im TAB „Gleichungen“ Zahlenangaben in Gleichungen direkt verwendet wurden, wurde bisher das „E“ in z.B. 5.0E-5 nicht korrekt interpretiert; dieser Fehler wurde beseitigt.
* Wenn im TAB „Werte, Unsicherheiten“ die „grüne“ Zelle für die Formel der Bruttozählratenunsicherheit im Bereich der unabhängigen Größen lag, wurde die damit berechnete Unsicherheit nicht in die ganz rechts liegende Spalte übertragen. Dieser Fehler wurde korrigiert.
* Die bisher für recht kleine Impulsanzahlen N verwendbare „N+1“-Regel wurde im Vorgriff auf eine neue Version der ISO 11929 in eine „N+x“-Regel geändert. Nach der wird nur für den Fall N=0 das x durch 1 ersetzt wird; für jedes N > 0 wird dagegen x=0 verwendet. Die mit dieser Regel verknüpfte Variable GamDistAdd (im Dialog Optionen – Voreinstellungen) muss dafür auf 0 gesetzt werden. Nur wenn N=0 vorliegt, ersetzt das Programm dies intern durch N=1. Näheres dazu siehe Kapitel 6.7.

**Überarbeitung des Setup-Programms:**

* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.2.07 2018/03**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 21.03.2018

**Überarbeitung der Windows-Hilfe:**

* Kapitel 1.2 wurde im Abschnitt „Länderspezifische Einstellungen“ erweitert
* Kapitel 7.11.3 wurde präzisiert

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Im Dialog des FitDecay-Models für lineare Entfaltung waren bisher im Feld für die Eingabe der Xi-Formel waren bisher Dummy-Einträge der Form „X3 = 1.0“ erforderlich, wenn z.B. weniger als drei Formeln tatsächlich benötigt wurden. Ab Version 2.2.07 können solche Dummy-Einträge entfallen. Die Änderung im Programm wurden so angelegt, dass auch alte Projekte, die solche Dummy-Einträge enthalten, weiterhin bearbeitet werden können.

**Überarbeitung des Setup-Programms:**

* Das Installations-Programm überträgt ab Version 2.2.07 dasjenige Sprachkürzel in die Datei UR2\_cfg.dat, das der Sprache entspricht, die am Beginn der Ausführung des Installations-Programms selektiert wird.
* Ab Version 2.2.07 wird mit der Ausführung der Deinstallation, unins000.exe, der Name „UncertRadio.exe“ in den Einträgen für die .txp und .csv Verknüpfungen in der Windows-Registrierung entfernt, so dass, nach einer späteren Installation, verhindert wird, dass eine alte UncertRadio-Version beim Doppelklicken auf ein Projekt auf gerufen wird. Näheres siehe Punkt „Länderspezifische Einstellungen“ im Kapitel 1.2 der CHM-Hilfe.

***UncertRadio* - Version 2.2.06 2018/02**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 27.02.2018

**Überarbeitung der Windows-Hilfe:**

* Kapitel 7.8: Einführung der Funktion uval(x);

Kapitel 4.3: Erweiterung der Formel für die Zählraten-Varianzen-Interpolation

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* Es gibt Fälle, in denen ein Unsicherheitswert innerhalb einer Gleichung für das Modell der Messung benötigt wird. Ein Beispiel dafür ist die relative Unsicherheit u(w)/w des verfahrensbezogenen Kalibrierfaktors w, wenn das Modell z.B. eine lineare Varianz-Interpolation erfordert. Hierfür wurde die Funktion uval(x) eingeführt, welche den Wert u(x) der Unsicherheit der Größe x zur Verfügung stellt. Im genannten Beispiel kann die relative Unsicherheit von w mit dem Ausdruck uval(w)/w innerhalb einer Gleichung verwendet werden.
* Wenn die Werte der Bruttozählrate Rb und der Nulleffektzählrate jeweils als Mittelwerte mehrerer Einzelwerte behandelt werden, ist eine lineare Interpolation zweier Zählraten-Varianzen zur Berechnung der Unsicherheit eines angenommenen Werts von y erforderlich. Da die Norm ISO 11929 vorschreibt, diese lineare Interpolation auf zwei Ergebnisgrößen-Varianzen anzuwenden, muss die in UncertRadio verwendete analoge Interpolation zwischen Zählraten-Varianzen nicht notwendigerweise linear sein. Im Kapitel 4.3 wird beschrieben, dass die bisher in UncertRadio verwendete Formel zur Zählraten-Interpolation durch Hinzunahme eines einfachen Terms zur vollen Äquivalenz mit der nach der Norm vorgegebenen Interpolation führt.

***UncertRadio* - Version 2.2.05 2018/01**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 31.01.2018

**Überarbeitung der Windows-Hilfe:**

* Kapitel 2.3

**Hinweis**: Die Schriftgröße in der CHM-Hilfe kann jetzt durch wiederholtes Klicken auf das Schriftart-Icon variiert werden.

**Hinzufügung eines Beispiel-Projekts:**

* keins

**Überarbeitung des Programms:**

* In der grafischen Darstellung der Wahrscheinlichkeitsverteilungen bei der MC-Simulation wurde die Skalierung der jeweils zwei Kurven („blau“ und „rot“) so abgeändert, dass die Integrale darüber jeweils den Wert 1 ergeben. Diese Anpassung hat jedoch keinen Einfluss auf die aus der Simulation abgeleiteten Kennwerte.
* Die Versionen der bisher für UncertRadio verwendeten Programmier-Tools wurden aktualisiert (GTk3+, Glade, Gfortran; siehe Abschnitt 2.4). Die bisher beim Kopieren eines Dialog-Screenshots miterfasste zusätzliche Umrandung tritt nun nicht mehr auf.

***UncertRadio* - Version 2.2.04 2018/01**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 15.01.2018

**Überarbeitung der Windows-Hilfe:**

* Überarbeitete Kapitel: 1.2: Manuelle Eintragungen von Umgebungsvariablen;
* Das Kapitel 5.2 „Automatisierter Betrieb mit Excelanwendung“ wurde hinsichtlich sprachabhängiger Variablen ergänzt.
* Datei UR2\_AutoRun\_V5.xlsm durch UR2\_AutoRun\_V6.xlsm ersetzt.

**Hinzufügung eines Beispiel-Projekts:**

* keines

**Überarbeitung des Programms:**

* Die Installation beinhaltet ab Version 2.2.01 eine Batchdatei

UR2\_start.bat,

die das Starten von UncertRadio bei eingeschränkten Windows-Nutzerrechten erleichtert (siehe Abschnitt [Installation](#_Installation)).

Der PATH-Befehl in dieser Batchdatei wurde mit der Version 2.2.03 geändert.

* Im automatischen Modus, gesteuert durch Excel, interpretiert UR2 seit dieser Version die sprachenabhängigen Werte des Dezimalpunkts und des Listentrennzeichens.

**Modifikation in der Exceldatei UR2\_AutoRun\_V6.xlsm:**

* In dem Modul Modul\_Auto\_single\_UR wurde die Routine Init\_pathnames um sprachabhängige Variablen sowie einer Variablen für Unterpfad der UncertRadio-Projekte erweitert.

Für das Betreiben von UncertRadio aus Excel heraus wurde der Kommandostring um ein weiteres Argument, das LC-Argument, erweitert: z. B.: LC=DE,;

Damit werden dem UncertRadio die in der Excelanwendung gültigen sprachabhängigen Variablen übergeben, die UncertRadio für die Ausführung übernimmt. Damit sollen Konflikte hinsichtlich der Werte des Dezimalpunkts und des Listentrennzeichens in beiden Programmen im Vorwege vermieden werden: UR2 verwendet dieselben wie Excel.

Siehe dazu das Kapitel 5.2 dieser Hilfe. Die Änderungen im Excel-Modul sind mit dem Datum 12.1.2018 und 13.1.2018 markiert.

***UncertRadio* - Version 2.2.03 2018/01**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 03.01.2018

**Überarbeitung der Windows-Hilfe:**

* Überarbeitete Kapitel: 1.2;
* Dateiname UR2\_AutoRun\_V4.xlsm durch UR2\_AutoRun\_V5.xlsm ersetzt.

**Hinzufügung eines Beispiel-Projekts:**

* keines

**Überarbeitung des Programms:**

* Die Installation beinhaltet ab Version 2.2.01 eine Batchdatei

UR2\_start.bat,

die das Starten von UncertRadio bei eingeschränkten Nutzerrechten erleichtert (siehe Abschnitt [Installation](#_Installation)).

Der PATH-Befehl in dieser Batchdatei wurde mit der vorliegenden Version geändert.

* Das Programm wurde um eine dritte Sprache, französisch, erweitert. Die Texte von GUI-Elementen und Message-Dialogen wurden mit Google-Translate in Französisch übersetzt. Das gewünschte Sprachkürzel (DE, EN oder FR) muss ab dieser Programmversion explizit in der Datei UR2\_cfg.dat angegeben werden. Dies reicht ohne Verwendung von sprachabhängigen Batchdateien aus, dass für die Label aller GUI-Elemente, die Message-Dialoge und Grafiken dieselbe Sprache verwendet wird. Es ist dabei nicht mehr nötig, zusätzlich die gewünschte Sprache direkt in Windows einzustellen. Dezimalpunkt und Listentrennzeichen werden Programm-intern in Abhängigkeit vom Sprachkürzel gesetzt. Dies betrifft auch den sprachabhängigen Dezimalpunkt in Achsenbeschriftungen in PLPLOT-Grafiken.
* Für die Programmführung in Französisch steht keine französische Fassung der CHM-Hilfe zur Verfügung, so dass hierfür deren englische Fassung herangezogen wird.
* Die einfache Einstellmöglichkeit der Sprache kann auch zu Problemen führen, wenn z.B. der Dateiinhalt der Datei UR-Saved-Results.csv in Deutsch angelegt wurde, während in einem Programmlauf in Englisch ebenfalls Werte in diese Datei geschrieben werden sollen: die Dezimaltrennzeichen und Listentrennzeichen in der Datei und im Programm passen dann nicht zusammen. UncertRadio führt daher vor einem „Save to CSV“ eine entsprechende Verträglichkeitsprüfung durch; bei Unverträglichkeit erfolgt eine Fehlermeldung und das Programm gibt bei seiner Beendigung einen Exitcode 3 aus. Siehe dazu auch den Hinweis zu UR2\_AutoRun\_V5.xlsm (s.u.).
* Es wurde der Bug beseitigt, dass zwei der dialogbezogenen Hilfethemen der CHM-Hilfe bei der letzten Version nicht erreichbar waren.
* Im Programm wurde nochmals intensiv nach Fehlermöglichkeiten gesucht, wie Division durch null, ungültige Zahlen oder Underflows, die gelegentlich zum Programmabbruch führen konnten. Nur wenige solche Fehlerorte wurden noch gefunden und die Fehler durch Änderung der Programmteile abgefangen. Ein Beispiel ist die MC-Simulation von Poisson-verteilten Impulsanzahlen: eine Tangens Funktion, in einer Exponentialfunktion der Funktion des Zufallszahlengenerators enthalten, konnte nahe π/2 zu sehr großen Tan-Werten im Exponenten führen. Eine bestimmte Art und Weise, Strings in Zellen von GTK-Treeviews („Tabellen“) zu speichern, erwies sich als „fehleranfällig“; die entsprechenden Programmanweisungen wurden angepasst und dieser Fehlertyp beseitigt.

**Modifikation in der Exceldatei UR2\_AutoRun\_V5.xlsm:**

* Das Modul Modul\_Auto\_single\_UR in dieser Datei prüft, ob UncertRadio mit einem Exitcode > 0 beendet wird. Falls in UncertRadio eine Fehlerbedingung erkannt wird, wird der Exitcode von 3 zum vorzeitigen Abbruch des VBA-Moduls verwendet. Dieser Fall tritt dann ein, wenn UncertRadio feststellt, dass in der CSV-Datei UR-Saved-Results.csv, abhängig von der verwendeten Sprache, andere Zeichen für den Dezimalpunkt oder das Listentrennzeichen vorliegen als diejenigen, die UncertRadio für „Save to CSV“ gerade verwendet.

Die Änderungen (wenige Zeilen, <5) in den Modulen von UR2\_AutoRun\_V5.xlsm gegenüber UR2\_AutoRun\_V4.xlsm sind mit der Datumsangabe 1.1.2018 versehen, die als Suchkriterium verwendet werden kann, um die Änderungen aufzufinden.

***UncertRadio* - Version 2.2.02 2017/12**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 18.12.2017

**Überarbeitung der Windows-Hilfe:**

* Überarbeitete Kapitel: keine.

**Hinzufügung eines Beispiel-Projekts:**

* überarbeitetes Projekt: TemperaturKurve\_KALFIT\_V2\_DE.txp (es wurde im TAB „Verfahren“ nur eine Information ergänzt)

**Überarbeitung des Programms:**

* Die Installation beinhaltet ab Version 2.2.01 eine Batchdatei

UR2\_start.bat,

die das Starten von UncertRadio bei eingeschränkten Nutzerrechten erleichtert (siehe Abschnitt [Installation](#_Installation)).

* In der Tabelle im TAB "Werte, Unsicherheiten" wurde zur besseren Lesbarkeit die Anzahl der signifikanten Stellen um 2 verringert.
* Beim Einlesen eines Projekts wurde bisher die Variable ModelType um ein Zeichen verkehrt eingelesen. Dieser Fehler wurde behoben; er betraf nur das Projekt mit negativ-linearem Modell (Rn-222-Emanation\_DE.txp).
* Der Dialog zur Auswahl einer Datei wurde überarbeitet:

beim Speichern einer Datei hat der betreffende Button nun das Label "Speichern" anstelle von "Öffnen";

die Dateifilter im Falle von Text-Dateien (.txt) wurden angepasst, so dass Filter und Dateierweiterungen nun konsistent sind;

* Der Inhalt der Report-Datei wurde erweitert: die probabilistisch-symmetrischen und die kürzesten Grenzen des Überdeckungsintervalls werden getrennt angegeben.
* Aus internen Gründen wurden zwei signifikante Änderungen am Programm vorgenommen:

a) Die recht umfangreiche Routine für die Ausführung der MC-Simulation wurde durch Zerlegen in kleinere Einheiten neu strukturiert, um insgesamt besser interpretierbar zu werden.

b) Die Nachweisgrenzen-Iteration, sowohl analytisch als auch bei der MC-Simulation, wurde auf die Verwendung des Verfahrens nach Ridders umgestellt (siehe Numerical Recipes). Die beim MC-Verfahren erforderlich Iteration für die Erkennungsgrenze (der Mitterlwert der MC-Verteilung soll möglichst nahe null gebracht werden) wurde auf das Sekantenverfahren umgestellt. Damit konnte die MC-Routine deutlich kürzer gefasst werden.

* Um diese beiden Umstellungen ausführlich testen zu können, wurde eine schon vorhandene (interne) Testroutine zur Abarbeitung vieler Projekte im Pseudo-Batch-Betrieb deutlich erweitert. Dies betrifft sowohl die Anzahl der Projekte als auch die Ausweitung auf MC-Simulationen. Das analytische Verfahren wurde damit an 109 Projekten getestet, das MC-Verfahren an etwa 50 Projekten (Laufzeit ca. 90 min; das Programm wir dazu nur einmal gestartet).

Bei diesen Tests traten einige Fehler auf. Bei den meisten der betroffenen Projekte traten sie bei Einzelauswertung der Projekte mit UR2 (mit jeweiligem Neustart des Programms) jedoch nicht auf. Dies deutete auf bestimmte Variablen, die nicht korrekt initialisiert waren (das Programm bekam eine Art "Gedächtnis"), so dass im Batch-Betrieb inkorrekte Werte dieser Variablen verwendet wurden. Es betraf zum Beispiel die Vorbereitung der Kovarianzen für die MC-Simulation.

Das umfangreiche Testen und die erfolgte Richtigstellung der Initialisierung dieser Variablen tragen dazu bei, das Risiko unkorrekter Auswertungen deutlich zu verringern.

* Einer der Fehler trat beim Projekt ISO-Beispiel-2a\_DE.txp auf, Erkennungs- und Nachweisgrenze bei der MC-Simulation waren deutlich größer als beim analytischen Verfahren. Dieser erforderte eine Anpassung des Programms: die zu iterierende Bruttozählrate darf in diesem Fall nicht aus der Bruttoimpulsanzahl Nb abgeleitet werden, da u(Nb), als sd/sqrt(mb) gegeben, nicht gleich sqrt(Nb) ist.

***UncertRadio* - Version 2.2.01 2017/12**

**Informationen zu erfolgten Änderungen**

G. Kanisch, ehemals TI-FI, Hamburg, 05.12.2017

**Überarbeitung der Windows-Hilfe:**

* Überarbeitete Kapitel: siehe die Links unter „Überarbeitung des Programms“.

**Hinzufügung eines Beispiel-Projekts:**

* neues Projekt: Mittelwert-theta\_DE.txp

**Überarbeitung des Programms:**

* Die Installation beinhaltet ab dieser Version eine Batchdatei

UR2\_start.bat,

die das Starten von UncertRadio bei eingeschränkten Nutzerrechten erleichtert (siehe Abschnitt [Installation](#_Installation)).

* Neu ist ab dieser Version, dass es möglich ist für eine Eingangsgröße, die mehrfach gemessen wurde, die Einzelwerte dazu als Datensatz einzugeben. Daraus werden Mittelwert und Standradunsicherheit des Mittelwerts berechnet, die als Werte in die Tabelle unter dem TAB „Werte, Unsicherheiten“ eingetragen werden. Diese Option wird im Projekt dadurch aufgerufen, dass in der Symboltabelle (TAB „Gleichungen“) der Typ für diese Größe, üblicherweise „a“ bzw. „u“, auf „m“ (Mittelwert) gesetzt wird. Die Behandlung des Datensatzes (Eingabe, Varianz-Option) erfolgt in einem neuen Dialog, der über ein neues Icon  in der Toolbar aufgerufen wird. Siehe [Abschnitt 6.9.2](#_Anwendung_von_Mittelwerten).

In Ergänzung dazu wurde noch die Typ-Angabe „p“ für eine solche Größe eingeführt, die als Parameter ohne Unsicherheit verwendet werden soll.

Die txp- und csv-Formate der Projektdatei wurde um die Angaben zu Mittelwert-Datensätzen ergänzt (s. [Abschnitt 3.6.1](#_3.6.1_Projektdatei_im)).

* Das bisherige Verfahren zur Berechnung des kürzesten Überdeckungsintervalls wurde vereinfacht und lässt nun auch den Wert null als untere Grenze zu (s. [Abschnitt 6.2](#URH_BestBayes_DE)). Mit der Checkbox „min. Coverage-Intervall“ im TAB „Resultate“ kann zwischen der Anzeige der Intervallgrenzen als „probabilistisch symmetrisch“ und „kürzestes Überdeckungsintervall“ gewechselt werden.
* Das Plotfenster wurde zur Verbesserung der Lesbarkeit etwas vergrößert und die Strichstärke von Linien verringert.
* Bei der durch den Button view-refresh.png angestoßenen Folge von Neuberechnungen wurde ein bisher fehlender TAB-Wechsel-Schritt ergänzt, der ebenfalls Berechnungen ausführt. Das macht die Anwendung dieses Buttons sicher.
* Während des Arbeitens an einem geöffneten Projekt konnte es passieren, dass manchmal eine Wiederholung der Berechnungen zu einem anderen Ergebnis führen konnte als vorher. Aber, nach Sichern des Projekts, Neustart des Programms und Laden des Projekts stimmte das Ergebnis wieder. Eine spezielle interne Programmvariable wurde als Ursache dafür identifiziert, die offensichtlich nicht richtig initialisiert war. Dieses Problem wurde behoben.
* Ein anderer behobener Fehler bestand darin, dass gelegentlich die beiden Comboboxen für Netto- und Bruttozählrate den Zustand „nicht-selektiert“ hatten.

# Inhalt des Programms

## 2.1 Einleitung

Das Windows-Programm **UncertRadio** erlaubt es, mit Hilfe von einzugebenden Gleichungen für die **Auswertung einer Messung aus dem Bereich der Radioaktivitätsmessung** die komplette Berechnung der Ergebnisgröße und ihrer **kombinierten Standardunsicherheit der Ergebnisgröße (nach ISO GUM)** als auch der damit eng verknüpften Werte der **Erkennungsgrenze und der Nachweisgrenze nach DIN EN ISO 11929-1:2021** durchzuführen.

Im Programm wird die **Interpretation des ISO GUM nach der** **Bayes’schen Theorie der Messun­sicherheit (Bayes-Statistik)** vorausgesetzt. Anders als in der konventionellen Statistik besitzt eine „Bayes’sche Messunsicherheit“ weder eine statistische Unsicherheit noch mit ihr verknüpfte statistische Freiheitsgrade. Die Erfassung von Freiheitsgraden ist unter dieser Voraussetzung im Programm nicht vorgesehen.

Eine Einschränkung in der Anwendung besteht darin, dass bisher nur Messprobleme bearbeitet werden konnten, bei denen die **Messungen** von Impulsanzahlen bzw. Zählraten **durch Vorwahl der Messdauern beendet werden**. Der statistisch anders zu behandelnde Fall der Begrenzung durch Vorwahl von Impulsanzahlen, bei dem nicht die Impulsanzahlen, sondern die Messdauern zu Zufallsgrößen werden, wird ab der Programmversion 2.4.04 ebenfalls behandelt (s. Kapitel 7.20).

Derzeit ist das Programm dafür eingerichtet, **bis zu drei Ergebnisgrößen** zu verwalten.

Die Schritte, die im Programm nacheinander abgearbeitet werden, sind:

|  |  |
| --- | --- |
| 1. | Eingabe einer kurzen textlichen Beschreibung des Messproblems; |
| 2. | **Eingabe der** das Messproblem, d.h. **die Ergebnisgrößen *y***, definierenden **Gleichungen**;diese definieren das **„Modell der Auswertung“**; die ersten der Gleichungen müssen die(se) Ergebnisgröße(n) definieren;  Anmerkung: Wenn mehr als eine Ergebnisgröße zu behandeln sind, erfolgt die Berechnung von Ergebnisgröße, Unsicherheiten-Budget, Erkennungsgrenze und Nachweisgrenze nur für eine dieser Ergebnisgrößen; bei der erstmaligen Definition der Gleichungen ist die erste von ihnen „aktiviert“, zu einem späteren Zeitpunkt kann im Hauptmenü des Programms eine andere ausgewählt werden; |
| 3. | automatische **Extraktion der** darin enthaltenen **Formelsymbole** und **Eingabe von** deren **Beschreibung** (Einheit, Bedeutung), automatische Einteilung in abhängige und unabhängige Symbole; manuelle Ergänzung von Symbolen, die nicht explizit in den Gleichungen auftreten; |
| 4. | Auswahl der Symbole, welche die **Nettozählrate (*Rn*)** und die **Bruttozählrate** darstellen, allein für den Zweck der späteren Berechnung von Erkennungs- und Nachweisgrenze; mit der Nettozählrate ist hier die „**verfahrensbezogene Nettozählrate**“ gemeint, in welcher auch der auf Interferenzen störender Radionuklide zurückgehende Zählratenbeitrag berücksichtigt wird, der sich im Allgemeinen aus einer Rechnung ergibt; |
| 5. | tabellarische **Eingabe von Messwerten und Standardunsicherheiten** für die Eingangsgrößen (unabhängige Symbole); |
| 6. | bei der **Eingabe von Messunsicherheiten** kann für die **Verteilungsform** einer unabhängigen Messgröße zwischen a) **Normalverteilung**, b) **Rechteckverteilung** und c) **Dreiecksverteilung** sowie d) **Gammaverteilung** sowie weitere gewählt werden; Halbbreiten der letzten beiden Verteilungsformen werden vom Programm gemäß GUM in Standardunsicherheiten konvertiert; für eine komplette Liste siehe [andere Verteilungsformen](#URH_SpezielleVertlg_DE)  Anmerkung: Im Falle von Low-Level-Anwendungen mit sehr kleinen Impulsanzahlen kann zur Verbesserung [**die sogenannte „(*N*+x)-Regel“**](#URH_NP1REGEL_DE) (d) für diese ausgewählt werden, wodurch die daraus berechneten Zählraten als Gamma-verteilt zu betrachten sind. |
| 7. | für die Eingabe kann zwischen **absoluten und relativen Unsicherheiten** gewählt werden; |
| 8. | für Zählraten oder Impulsanzahlen können **Unsicherheiten auch als Formeln** eingegeben werden; |
| 9. | für die Bruttozählrate muss eine Formel für ihre Standardabweichung eingegeben werden; dies ist die „**Unsicherheits-Funktion“ (Standardunsicherheit) der Bruttozählrate**, auf der die Berechnung von Erkennungs- und Nachweisgrenze basiert; |
| 10. | tabellarische **Eingabe von Kovarianzen zwischen Eingangsgrößen**, die als Formeln oder als Korrelationskoeffizienten eingegeben werden; |
| 11. | numerische Berechnung von Werten und kombinierten Standardunsicherheiten für die abhängigen Größen einschließlich der Ergebnisgröße und des **Unsicherheiten-Budgets;** die Einbeziehung von Kovarianzen/Korrelation von Eingangsgrößen ist möglich; Berechnung der "[**besten Schätzer und Vertrauensgrenzen nach Bayes**](#URH_BestBayes_DE)" für den Wert und die Standardunsicherheit der Ergebnisgröße; |
| 12. | numerisch-iterative Berechnung von **Erkennungs- und Nachweisgrenze** der Ergebnisgröße auf der Basis der numerischen Berechnung ihrer kombinierten Standardunsicherheit unter Einbeziehung von Kovarianzen; |
| 13. | es kann zur Überprüfung eine [**Monte Carlo Simulation**](#URH_MC_SIM_DE) ausgeführt werden, die ebenfalls den Wert der Ergebnisgröße und deren kombinierten Standardunsicherheit ermittelt; partielle Ableitungen sind dabei nicht erforderlich: für jeden als unabhängig (u) charakterisierten Messwert bzw. Parameter wird ein seiner Verteilung entsprechender Wert „gewürfelt“ und damit der Wert der Ergebnisgröße berechnet. Aus der vielfachen Wiederholung wird dann eine statistische Verteilung der Ergebnisgröße erhalten und daraus ihr bester Schätzwert als Mittelwert und aus der Streuung die kombinierte Standardunsicherheit berechnet; Konfidenzgrenzen werden als Quantile dieser Verteilung bestimmt, während für Erkennungsgrenze und Nachweisgrenze separate MC-Verteilungen mit modifizierten Werten der Ergebnisgröße erzeugt und daraus als Quantile ermittelt werden; |
| 14. | am Ende kann ein **vollständiger Report als Textdatei** erzeugt werden, die alle Gleichungen und Eingangswerte, das Unsicherheiten-Budget und die Endergebnisse enthält - einschließlich der Monte Carlo-Ergebnisse. |

**UncertRadio** eignet sich **sehr gut für Vergleichsrechnungen** zu solchen Lösungen, die man selbst auf der Basis von Tabellenkalkulationen entwickelt hat. Letztere können je nach Anwendung recht komplex werden und sind gelegentlich, gerade im Hinblick auf die richtige Fortpflanzung von einzelnen Unsicherheiten, nicht leicht überschaubar.

## 2.2 Numerische Verfahren

### 2.2.1 Struktur der Gleichungen

Die Bestimmungsgleichungen zur Berechnung des Wertes der **Ergebnisgröße(n) *y,*** die das Auswertemodell definieren, können in einem Dialogtextfeld des Programms direkt eingegeben werden. Dabei können zur Verbesserung der Übersicht auch Gleichungen für Hilfsgrößen eingeführt werden. Aus den Gleichungen werden automatisch die verwendeten **Formelsymbole** extrahiert, die danach in einer Tabelle einzeln mit einer **Einheit** sowie ihrer **Bedeutung** ergänzt werden können. Nach der Eingabe von Werten der primär gemessenen Größen und ihrer Unsicherheiten wird ein sogenannter "Function Parser" verwendet, um mit Hilfe der zuvor definierten Gleichungen den Wert der Ergebnisgröße ***y*** zu berechnen.

Wichtig bei der Aufstellung der Gleichungen ist, auf die Einführung einer hier im Programm **Nettozählrate *Rn***genannten Größe zu achten, von der die **Ergebnisgröße** linear abhängt:

***y = Rn \* FL + FC*** (1)

Hierin stellt der Proportionalitätsfaktor *FL* den verfahrensbezogenen Kalibrierfaktor dar. *FC* berücksichtigt ggf. weitere Interferenzbeiträge, wie z.B. den, der durch Zugabe einer Traceraktivität vor Beginn einer radiochemischen Analyse entsteht.

Die Nettozählrate *Rn* soll als diejenige Nettozählrate (genauer: verfahrensbezogene Nettozählrate), verstanden werden, für deren Berechnung aus der Bruttozählrate *Rb* alle möglichen darin enthaltenen Beiträge abgezogen werden, die nicht von der Ausgangsprobe der durch ein Analysenverfahren hergestellten Messprobe herrühren. Dies sind nicht nur der Detektor-Nulleffekt *R0*, sondern auch der Blindwert *Rbl*, sowie, z.B. im Falle der Alphaspektrometrie mit Tracern, zusätzlich auftretende Blindwerte (in diesem Fall aus Verunreinigungen der Tracer herrührend). Ebenso kann ein berechneter Beitrag *Rint* enthalten sein, der eine Störung (engl.: interference) durch ein anderes Radionuklid berücksichtigt. **Die verfahrensbezogene Nettozählrate kann z.B. lauten:**

*Rn* = *Rb* – *R0* – *Rbl* – *Rint* . (2)

Die Konstanten FL und FC lassen sich für beliebige Arten von Bestimmungsgleichungen, die linear bezüglich der Nettozählrate sind, ganz leicht programmintern ermitteln. Diese Darstellung gestattet UncertRadio, die obige Gleichung für einen modifizierten Wert y‘ leicht nach der Zählrate aufzulösen:

*Rn‘ = (y‘ – FC) / FL.* (3)

Ebenso kann die Gleichung für die Nettozählrate *Rn* in verallgemeinerter Form als eine lineare Funktion der Bruttozählrate Rb dargestellt werden:

*Rn = FB \* Rb - R0total*. (4)

Der Faktor FB ist in den meisten Fällen gleich 1; *FB* kann aber auch ungleich 1 sein. R0total stellt die Summe aller von der Bruttozählrate abzuziehenden Untergrundbeiträge dar; siehe Gl. (2). Am Beginn der Rechnungen ermittelt UncertRadio den Wert von *FB* und daraus den festen Wert R0total:

*R0total = FB\*Rb - Rn*. (5)

Zu einer im Verlaufe der Nachweisgrenzen-Berechnung variierten Nettozählrate *Rn’* ergibt sich daraus die modifizierte Bruttozählrate:

*Rb‘ = (Rn‘ + R0total) / FB*. (6)

**Besonderheit**

In Gleichung (2) können alle Zählraten jeweils auch mit Faktoren *g* multipliziert werden, die auch mit Unsicherheiten behaftet sein können, z.B.

*Rn* = *gb*\**Rb* – *g0*\**R0* – *gbl*\**Rbl* – *gint*\**Rint* (7)

**Nicht-lineare Abhängigkeit**

Es kann Fälle geben, in denen der Zusammenhang zwischen der Ergebnisgröße und der Nettozählrate, oder zwischen der Ergebnisgröße und der Aktivität bei linearer Entfaltung, nicht linear ist. Das hat zur Folge, dass die Kennwerte *FC* und *FL* in Gl. (1) nur noch Näherungswerte sind und die in Gl. (3) angegebene Umkehrung von Gl. (1) nicht mehr korrekt ist.

Daher werden in UncertRadio zusätzlich zu Gl. (1) zwei neue interne Funktionen verwendet:

* Alternativ zu Gl. (1) wird unter Verwendung des Unterprogramms **RESULT** (s. weiter unten) eine Funktion **ActVal**(*Rn*) zur Berechnung des Werts der Ergebnisgröße verwendet;
* für die Umkehrung nach Gl. (3) wird alternativ eine Funktion **RnetVal**(*y‘*) eingesetzt; hierfür wird das numerische Sekanten-Verfahren verwendet; es benötigt Startwerte für die untere und obere Grenze des zu suchenden Nettozählratenwerts, die aus *y‘*, *FC* und *FL* leicht berechnet werden können.

### 2.2.2 Typen von Modellen

Betrachtet man die Beziehung zwischen der Ergebnisgröße und der Nettozählrate, oder der Aktivität des Messpräparats im Falle der linearen Entfaltung, wird in UncertRadio zwischen drei Typen von Modellen der Messung unterschieden:

* positiv lineares Modell, mit Nachweisgrenzenberechnung
* negativ lineares Modell, mit Nachweisgrenzenberechnung
* nur GUM, ohne Nachweisgrenzenberechnung

Die im vorigen Abschnitt beschriebene Beziehung stellt den überwiegend anzutreffenden Fall des **positiv linearen Modells** dar. Dabei nimmt die Nettozählrate *Rn* mit zunehmender Bruttozählrate zu. Die Aktivität wird erkannt, wenn die Bruttozählrate die Nulleffektzählrate signifikant übersteigt.

Ein **negativ lineares Modell** zeichnet sich dadurch aus, dass der Bruttoeffekt den „Untergrund“ signifikant unterschreiten muss, um den damit verbundenen Effekt als nachgewiesen betrachten zu können. Hier wird die Differenzbildung im Ausdruck der Nettogröße ***Rn*** aus Gl. (1) umgekehrt:

***y = (R0 – Rb)\*FL + FC***  mit *Rb* < *R0* (8)

*Rb* und *R0* sind bei diesem Modell nicht notwendigerweise Zählraten. Das Projekt Rn-222-Emanation\_DE.txp ist ein Beispiel dafür.

Der als **„nur GUM“** bezeichnete Fall wurde nur dafür eingeführt, dass man für die Messung zwar eine Unsicherheit nach GUM, jedoch keine Nachweisgrenze berechnen will oder kann, weil die Messung z.B. eine Wägung darstellt.

### 2.2.3 Kombinierte Standardunsicherheit der Ergebnisgröße

Die Berechnung der **kombinierten Standardunsicherheit** der Messgröße ***y*** erfolgt nach ISO GUM (ISO ***G***uide on ***U***ncertainty of ***M***easurement (1995); vgl. auch EURACHEM / CITAC Guide "Quantifying Uncertainty in Analytical Measurement" (2000) und berücksichtigt auch Kovarianzen zwischen einzelnen Messgrößen.

Für die Standardunsicherheit einer Größe des Typs A, die aus einer statistischen Analyse einer Beobachtungsreihe (Messreihe) gewonnen wird, muss eine kleine statistische Auswertung (Mittelwert und Standardabweichung) außerhalb dieses Programms vorgenommen werden; nur der Mittelwert und die Standardabweichung werden hier verwendet. Üblicherweise wird bei der Messung von Radioaktivität der größere Anteil der Größen dem Typ B zugerechnet, u.a. die verschiedenen Zählraten bzw. Impulsanzahlen. Im Rahmen der **Bayes’schen Theorie der Messunsicherheit**, die hier zugrunde gelegt wird (Weise & Wöger, 1999; Weise et al., 2006), ist im Programm eine **ausdrückliche** **Unterscheidung** zwischen **Größen des Typs (Ermittlungsmethode) A und B** **überflüssig**. Aus demselben Grund werden auch keine Freiheitsgrade berücksichtigt.

Die analytische Ableitung von Formeln für die kombinierte Unsicherheit z.B. einer Masse- oder Volumen-bezogenen Aktivität nach dem Gesetz zur Fortpflanzung der Unsicherheiten kann bekanntlich leicht zu einem „Wust an einzelnen Formeln“ führen, deren Richtigkeit oft schwer zu kontrollieren ist. Hier wird daher ein numerisches Verfahren angewendet.

Der erste Schritt besteht darin, alle zur Berechnung erforderlichen Messgrößen/Parameter, dies können schnell mehr als z.B. 20 sein, in ein Feld ***Messwert(i) = p(i)*** zu transferieren. Dann wird in einem Unterprogramm **RESULT** mit Hilfe des function parsers aus den zuvor eingegebenen Gleichungen der Wert der Ergebnisgröße aus den Werten der Parameter *p(i)* berechnet.

Ebenso werden die bekannten Mess­unsicherheiten der einzelnen Parameter auf ein Feld ***StdUnc(i) = u(i)*** übertragen. Zur Be­rechnung der kombinierten Mess­un­sicherheit wird ein [**Unterprogramm UncPropa**](#URH_UNCPROPA_DE)verwendet, dem beide Fel­der, *p(i)* und *u(i)* übergeben werden. Kovarianzen werden hierbei einbezogen. Die Sensitivitäts­koeffi­zienten, d.h. die partiellen Ableitungen der mit RESULT berechneten Funktion nach den Parametern *p(i)*, werden numerisch durch Diffe­rentialquotienten approxi­miert.

Weiteres hierzu: siehe [**Unsicherheitsfortpflanzung**](#URH_UncPropMeth_DE)

### 2.2.4 Bestimmung von Erkennungsgrenze und Nachweisgrenze (iterativ)

Zur **Berechnung der Nachweisgrenzen** wird die mit der **Bayes-Statistik** begründete **Norm DIN EN ISO 11929-1:2021** (siehe auch Weise et al., 2006) herangezogen, die so gestaltet ist, dass die komplette Unsicherheiten-Fortpflanzung verwendet wird, unter ausdrücklicher Einbeziehung aller verfügbaren Einzelunsicherheiten; Kovarianzen werden ebenfalls einbezogen. Dazu wird auf die numerischen Berechnungen mit dem Funktionsunterprogramm **UncPropa** zurückgegriffen. Die Werte für Erkennungs- und Nachweisgrenze für die Ergebnisgröße *y* werden implizit durch ein numerisches Iterationsverfahren berechnet. Hierbei wird der Wert von *y* variiert, jetzt als angenommener Wert bezeichnet. Daraus errechnet sich über die Nettozählrate **der iterierte Wert der Bruttozählrate;** für jeden Schritt wird **aus die** **als „Unsicherheits-Funktion“**  **bezeichnete kombinierte Standardunsicherheit berechnet**.

Die **Erkennungsgrenze *y\**** für die Ergebnisgröße ***y*** wird nach DIN EN ISO 11929-1:2021 wie folgt berechnet:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

wobei das Quantil der standardisierten Normalverteilung zum Fehler 1. Art α ist. Zur Berechnung der kombinierten Standardunsicherheit der Ergebnisgröße wird **UncPropa** benutzt, und zwar mit der Randbedingung, dass die Netto-Zählrate des Analyten dabei gleich null gesetzt wird. Dies ist einfach zu erreichen.

Die **Nachweisgrenze** für die Ergebnisgröße wird wie folgt berechnet, wobei die zuvor berechnete Erkennungsgrenze und das Quantil der standardisierten Normalverteilung zum Fehler 2. Art *β* ist:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Dies stellt eine implizite Gleichung für dar, da auf der rechten Seite die Unsicherheit *u* für einen Wert von *Rn* zu berechnen ist, der dem Wert auf der linken Seite entspricht; letztere Beziehung erhält man mit der Umkehrfunktion , die wegen des einfachen linearen Zusammenhangs zwischen und *Rn (siehe Gleichung (3)* leicht zu berechnen ist.

Die Lösung der impliziten Gleichung (8) erfolgt durch ein einfaches Iterationsverfahren, am [**Beispiel für die Nachweisgrenze**](#URH_NWGiterat_DE) dargestellt. Der Wert für *Faktor* wird im Unterpro­gramm **RESULT** berechnet. Die Unsicherheit in Gl. (2) wird mit dem Funktions­unterprogramm **UncPropa** berechnet. Um **UncPropa** korrekt zu bedienen, wird bei je­dem Iterationsschritt aus dem Zwischenwert das dazugehörige berechnet, welches noch auf die Bruttozählrate des Analyten trans­formiert wird, die im Beispiel im Feldelement abgelegt wird. Deren Unsicherheit wird mit Hilfe der „Unsicherheits-Funktion (Standardunsicherheit) für die Bruttozählrate“, die vom Benutzer eingegeben wird, abgeschätzt. In der grafischen Darstellung wird die Bruttozählrate so dargestellt, als sei mit einer simplen Einkanal-Messung gemessen worden: , was in den allermeisten Fällen auch zutrifft.

Mit der Version 2.2.02 (2017/12) ist das oben beschriebene Iterations-Verfahren durch das Verfahren nach Ridders ersetzt worden (Subroutine Zriddr aus den Numerical Recipes, Press et al., 1992). Es arbeitet effektiver als das Sekantenverfahren. Ab Version 2.2.11 (2018/11) wird stattdessen das Verfahren nach Brent verwendet.

**Sonderfälle**

Im Falle einer linearen Entfaltung mithilfe der linearen Least squares-Analyse, z.B. bei der Auswertung von Abklingkurven, ist der Fitparameter für die gesuchte Nettozählrate diejenige Größe, die zur iterativen Bestimmung von Erkennungs- und Nachweisgrenze variiert wird (vgl. [**Hinweis zu Erkennungs- und Nachweisgrenzen bei linearer Entfaltung**](#URH_LSQ_NWG_DE)).

Bei der Aktivitätsbestimmung eines Radionuklids mit mehreren Gammalinien ist der Wert der Aktivität der Messgröße diejenige Größe, die variiert wird (vgl. [**Verfahren der Berechnung von Erkennungs- und Nachweisgrenze für Gamspk1**](#URH_GSPK1_NWG_DE)).

### 2.2.5 Vermeidung von „verdeckten“ Kovarianzen

Es wird im Folgenden angenommen, dass sich der arithmetische Ausdruck für die Ergebnisgröße aus mehreren Ausdrücken zusammensetzt, z.B., , die jeweils Funktionen der Eingangsgrößen xi sind. Oft werden zunächst die Unsicherheiten bestimmt und damit dann berechnet. Wenn es aber unter Eingangsgrößen bestimmte gibt, die in mehr als einem der Ausdrücke vorkommen, sind dadurch sogenannte „verdeckte“ oder „übersehene“ Kovarianzen zwischen den entstanden, die nachträglich berücksichtigt werden müssen.

Dieses Problem tritt bei den Berechnungen in UncertRadio dadurch nicht auf, dass sich die partiellen Ableitungen in dessen Unsicherheitsfortpflanzung immer auf die Ergebnisgröße beziehen. Warum das funktioniert, wird im Folgenden gezeigt.

Der Ansatz für die Unsicherheitsfortpflanzung mit partiellen Ableitungen, die auf die Ergebnisgröße bezogen werden, wird mit dem Vektor der Eingangsvariablen wie folgt formuliert:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |
|  |  | (2) |

Zunächst wird das Quadrat der Summe ausgeführt:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Nun wird die Summation über für jeden der sechs Terme ausgeführt, gleichzeitig werden die partiellen Ableitungen von nach jeweils vor die Summen gezogen:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Jede einzelne Summe über stellt entweder eine Varianz der oder eine Kovarianz der drei Ausdrücke dar:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Üblicherweise beinhaltet bei einer Zerlegung von in Ausdrücke oder Funktionen eine per Hand ausgeführte Unsicherheitsfortpflanzung nur die ersten drei Terme in Gl. (6), da Kovarianzen zwischen den meistens nicht vermutet werden; darauf geht die Bezeichnung als „verdeckte“ Kovarianzen zurück.

Der mit Gl. (6) entstandene Ausdruck ist genau derjenige, den man erwartet, wenn die verdeckten Kovarianzen zwischen den mitberücksichtigt werden. Dieses Ergebnis demonstriert, dass die Unsicherheitsfortpflanzung in UncertRadio die verdeckten Kovarianzen berücksichtigt, quasi automatisch, da es nach Gl. (1) mit partiellen Ableitungen der Ergebnisgröße selbst arbeitet.

### 2.2.6 Verwendung von Schaltvariablen in Gleichungen

Der in UncertRadio verwendete Funktions-Parser (fparser) erlaubt die Verwendung von sogenannten **Schaltvariablen**, deren Eigenschaft darin besteht, dass sie **nur die zwei Werte 0 und 1 annehmen** können. Damit kann man eine zweite Variable b **mit b^1 „einschalten“ bzw. aktivieren** oder **mit b^0 auf die Konstante 1 setzen** **(deaktivieren)**. In UncertRadio können die Schaltvariablen mit einem Zusatz „\_Trigger“ im Symbolnamen gekennzeichnet werden. Sie werden daher auch als **„Trigger-Variable“** bezeichnet. Beispiele: „min\_Trigger“, „kilo\_Trigger“; mit „60^min\_Trigger“ oder mit „1000^kilo\_Trigger“ kann ein Umrechnungsfaktor 60 (bei Minuten) oder 1000 geschaltet werden; siehe dazu auch Kapitel 2.2.7. Wenn Schaltvariable im Zusammenhang mit Zählraten vorkommen, müssen diese den Namensteil „\_Trigger“ enthalten, damit sie vom Programm als solche erkannt werden und die interne Auffindung von solchen Zählraten, die direkt zur Nettozählrate beitragen (siehe das Kapitel 2.3), nicht stören können.

### 2.2.7 Berechnung physikalischer Einheiten für abhängige Variable

Ab der Programmversion 2.4.13 enthält UncertRadio eine Option, mit der **die physikalischen Einheiten der abhängigen Variablen** testhalber mit Hilfe eines auf numerischen Rechnungen basierenden Algorithmus „berechnet“ werden können. Die Einheiten der Eingangsgrößen werden in sogenannte Basiseinheiten umgewandelt. Bei dieser Umwandlung anhand der primär eingegebene Einheiten (oft „abgeleitete Einheiten“) werden Umrechnungsfaktoren ermittelt, die in die numerischen Werte für die verschiedenen Größen eingehen. Wenn z.B. eine Messdauer in der Einheit min eingegeben wurde, wird bei der Umrechnung auf die Basiseinheit Sekunde ein Faktor 60 einbezogen. Zur Beschreibung der Basiseinheiten und dazugehöriger abgeleiteter Einheiten, sowie des Algorithmus zur „Berechnung“ von Einheiten wird auf das [Kapitel 7.21](#URH_Behandlung_Einheiten) verweisen.

Die mit dem Programm bereitgestellte CSV-Datei mit den möglichen Einheiten enthält nur eine kleine Anzahl von Einheiten, die primär auf den Bereich der Messung von Aktivität zugeschnitten ist. Die Basiseinheiten in der CSV-Datei können vom Anwender für den eigenen Bedarf angepasst werden.

Unter dem Menüpunkt „Bearbeiten – physik. Einheiten testen“ kann man die Umwandlung auf Basiseinheiten testen. Im Programm-Editor erfolgt für die Liste der Größen (Symbole) eine Gegenüberstellung der „alten“ und „neuen“ Einheiten sowie der dazugehörigen Werte und Unsicherheiten. Am Ende der Liste erfolgt ggf. die Ausgabe der ersten Fehlermeldung, die einen Hinweis auf einen möglichen Einheitenfehler geben kann.

Die Routine zu diesem Menüpunkt wurde auf alle zu UncertRadio gehörenden Beispielprojekte angewendet. Dabei wurden durchaus Fehler gefunden, oder es war erforderlich, für eine Detektor-Nachweiswahrscheinlichkeit nicht die Einheit „1“, sondern „1/Bq/s“ als Einheit anzugeben, um für die Ergebnisgröße Aktivität auch auf die Einheit „Bq“ zu kommen. Dazu wird auf das [Kapitel 7.21.3](#URH_Test_Einheiten) verwiesen.

## Gleichungen als Baum-Topologie

Bei den analog zu Abschnitt 2.2.1 aufgestellten Gleichungen zur Berechnung des Wertes der Ergebnisgröße handelt es sich um **hierarchische Gleichungen**. Dabei entsteht eine Liste mit abhängigen Größen (Anzahl *nab*), denen die Liste der unabhängigen Eingangsgrößen (Anzahl *nmu*) folgt. Den Größen lassen sich somit Nummern von 1 bis (*nab*+*nmu*) zuordnen, sie können also anhand der Symbolnummern identifiziert werden. Man kann sich diese Liste wie eine Leiter mit (*nab*+*nmu*) Stufen vorstellen, oder wie ein Atomkernniveauschema; auf jeder Stufe (Niveau) sitzt ein Größensymbol. Die abhängigen Größen werden mit der ihnen zugeordneten Hilfsgleichung mit anderen Größensymbolen weiter unten in dieser Leiter verknüpft. Verbindet man die derart verknüpften Symbole mit Linien, entsteht eine Baumstruktur, vergleichbar den bekannten Übergängen in einem Atomkernniveauschema. Die Folge der Verbindungslinien von der abhängigen Größe bis zu den Leiterstufen der unabhängigen Symbole entspricht einer Kaskade im Kernniveauschema.

Die folgende Betrachtung ist auf Anwendungen ohne lineare Entfaltung beschränkt.

Solche Symbolkaskaden lassen sich für alle vorkommenden Übergängen (*i*, *j*)= *i* 🡪 *j* zwischen den Symbolen *i* und *j* innerhalb einer Kaskade darstellen. Mit Hilfe eines rekursiven numerischen Verfahrens können diese Kaskaden ermittelt werden.

Dieses Verfahren wird dafür genutzt, um für die zur Ergebnisgröße proportionalen Nettozählrate *Rn* herauszufinden, aus welchen Zählratenbeiträgen *Ri* sie sich zusammensetzt. Die Zählraten *Ri* haben überdies zwei zusätzliche Eigenschaften:

1. ihnen kann jeweils eine Wurzelfunktion als Unsicherheit zugeordnet sein, z.B. sqrt(*Ri* / *ti*). oder,
2. sie lassen sich auf Impulsanzahlen *Ni* zurückführen, denen sich ebenfalls Formeln für die Unsicherheit (sqrt(*Ni*)) oder für die gar spezielle Verteilungstypen vereinbart sind (Gammaverteilung („x+1“)

Durch Ausnutzung dieser zusätzlichen Eigenschaften (weiter unten „rules“ genannt) können in den meisten Fällen diejenigen Symbole identifiziert werden, die eine Impulsanzahl repräsentieren und mit welchem Messdauersymbol sie verknüpft sind. Damit wird dann durch einen Schritt nach oben in der betreffenden Kaskade die dazugehörige Zählrate *Ri*=*Ni* / *ti* gefunden.

Kennt man den Zusammenhang zwischen der Bruttozählrate *Rb*, der Impulsanzahl *Nb* und der Messdauer *tb*, und deren Symbolnummern innerhalb ihrer Kaskade, ist es leicht möglich, die für Erkennungs- und Nachweisgrenze erforderliche Änderung von *Rb* in *Rb~* auf die dazugehörige Änderung von *Nb* in *Nb*~ zurückzuführen. Um dies zu erleichtern, werden in UncertRadio entsprechende Indexfelder angelegt, die von Zählrate auf Impulsanzahl und Messdauer und umgekehrt verweisen. Das setzt allerdings voraus, dass in den Gleichungen nicht nur die Zählratensymbole *Ri* allein definiert sind, sondern auch die Gleichungen *Ri* / *ti*; daraus resultiert die Empfehlung, dies in UncertRadio auch zu tun.

**Beispiel** Ra226\_U235-bei-186keV\_DE.txp.

Gleichungen (*nab*=8, *nmu*=10):

Formeltext=

1 : cRa = Phi \* RRa

2 : Phi = 1. / (eps \* pRA \* mp)

3 : RRa = RS - RU5

4 : RS = Rb - RT - RnNE

5 : RU5 = AU5 \* Ufakt

6 : Ufakt = eps \* pU5 \* mp

7 : Rb = Nb / tm

8 : RT = NT / tm

Tabelle der Übergänge *i* 🡪 *j*:

nd i j Symb(i) Symb(j)

---------------------------------------

1 3 4 RRa RS

2 3 5 RRa RU5

3 4 7 RS Rb

4 4 8 RS RT

5 4 12 RS RnNE

6 5 13 RU5 AU5

7 5 6 RU5 Ufakt

8 6 9 Ufakt eps

9 6 14 Ufakt pU5

10 6 11 Ufakt mp

11 7 15 Rb Nb

12 7 16 Rb tm

13 8 17 RT NT

14 8 16 RT tm

Tabelle der Kaskaden (chain) und drei identifiziete Zählraten, die zur Nettozählrate gehören

nc i j kcnt ktime krate rule Symbol chain

-----------------------------------------------------------------------

1 7 15 15 15 7 A5 Rb 3 4 7 15

2 7 16 0 0 0 3 4 7 16

3 8 17 17 17 8 A3 RT 3 4 8 17

4 8 16 0 0 0 3 4 8 16

5 4 12 0 0 12 A6 RnNE 3 4 12

6 5 13 0 0 0 3 5 13

7 6 9 0 0 0 3 5 6 9

8 6 14 0 0 0 3 5 6 14

9 6 11 0 0 0 3 5 6 11

Tabelle der Zuordnungen von Messdauer (iptr\_time) und Impulsanzahl (iptr\_cnt) zur Zählrate (iptr\_rate)

(*RnNE* ist nur als unabhängige Nettozählrate des Nulleffekts gegeben)

i iptr\_time iptr\_cnt iptr\_rate Symbol

--------------------------------------------

7 16 15 7 Rb

8 16 17 8 RT

12 0 0 0 RnNE

Unter den zu UncertRadio gehörenden Beispielprojekten sind zwei, bei denen mit dem oben skizzierten Algorithmus die Bruttozählrate und die Nulleffektzählrate je zweimal gefunden werden:

BSH\_Gesamt-Gamma\_var2\_DE.txp

DWD\_sr89\_sr90\_TDCR\_Verfahren\_DE.txp

Im ersten Beispiel führt dieser Befund zur Einsicht, dass die Gleichungen der Nettozählrate nicht weit genug zusammengefasst wurden. Tatsächlich kann man die Gleichungen dieses Beispiels algebraisch so weit umformen, dass daraus die Gleichungen des Beispiels BSH\_Gesamt-Gamma\_var1\_DE.txp werden.

Im zweiten genannten Beispiel ist die Gleichung für *Rn\_s* für die Berechnung der Sr-90-Aktivität so komplex, dass *R0\_s* und *R0\_c* jeweils zweimal darin vorkommen, und zwar auch in nicht-linearer Form.

**Hinweis**: Bei der Batch-Auswertung von Projekten wird zusätzlich eine Datei fort.64 angelegt, die für jedes Projekt (ohne lineare Entfaltung) in Kurzform die identifizierten Zählratenbeiträge zur Nettozählrate aufführt. Diese Option ist inzwischen deaktiviert.

**Hinweis**: Aus dem obigen Beispiel geht hervor, dass die Bruttozählrate Rb die erste in der Liste der an der Nettozählrate beteiligten Zählraten ist. Diese Eigenschaft kann dazu verwendet werden, intern zu überprüfen, ob im TAB „Gleichungen“ das richtige Bruttozählraten-Symbole selektiert wurde, **da die Bruttozählrate immer die erste Zählrate im Ausdruck für die Nettozählrate sein muss**.

## Verwendete Programmierhilfsmittel

Das Programm wurde unter Windows in Fortran 90/95/2008 entwickelt.

Für die Version 2.4.00 wurde der Fortrancode modifiziert um String-Felder variabler Längen („character arrays of deferred shape”) verwenden zu können, wofür ein User-Variablentyp definiert wurde. Dies bedeutet, dass die Elemente solcher String-Felder unterschiedlicher lang sein können. Es erfordert aber eine ausführlichere Belegung/Freigabe von Variablen-Speicher („memory allocation/deallocation“), auch für Felder des Typs integer und real. Es ermöglicht, dass in UncertRadio Symbole mit mehr als 20 Zeichen definiert werden können.

Der Fortran-Code umfasst derzeit etwa 430 Routinen/Funktionen mit ca. 47.000 Zeilen (ohne Kommentar- oder Leerzeilen, GTK-Fortran- und PLPLOT-Routinen nicht mitgezählt).

Damit das Programm als ein Windows-Programm betrieben werden kann, wurden folgende Programmiertools verwendet:

* ***Code::Blocks*** 20.03, plattformunabhängige Programmierumgebung; (<http://www.codeblocks.org>)
* ***GNUFortran***; 64 bit, Version 13.1.0
* ***GTK3+*** *Version 3.24.39****,*** das für C-Anwender entwickelte Graphical User Interface (GUI) (<https://www.gtk.org>);
* ***GTK-Fortran,*** Version 20.04 vom 07.05.2020, das Fortran-Interface zu den C-Funktionen von GTK+ (<https://github.com/jerryd/gtk-fortran/wiki>);
* ***Glade Interface Designer*** für Windows (Version 3.40.0) zum grafischen Aufbau des GUI (<http://glade.gnome.org>);
* ***PLplot*** *5.15.0* für grafische Darstellungen (<http://plplot.sourceforge.net/> ).

Die Tools GFortran-Compiler, GTK+3 und Glade wurden als Pakete einer **MSYS2-Installation** (<https://www.msys2.org/wiki/MSYS2-introduction/>) verwendet. Die CHM-Version der UR Windows-Hilfe wurde mit dem **NüHelp** Tool (Version v2018.04.23; <https://sourceforge.net/projects/nuhelp/> ) erstellt. Das Setup-Installationsprogramm wurde mit dem Programm **Inno Setup** 6.2.1 ( <http://www.innosetup.com/> ) erstellt.

Zur Umsetzung der vom Benutzer eingegebenen Gleichungen in während des Programmlaufs berechenbare Formeln wurde das von Roland Schmehl bei der Universität Karlsruhe bereitgestellte Public Domain ***Function Parser Modul*** verwendet, das als Fortran 90 Source auf einer – inzwischen nicht mehr existierenden – Internet-Seite zur Verfügung gestellt wurde.

Zur Erzeugung Gamma-verteilter Zufallszahlen wird ein Generator nach Marsaglia and Wang (2000) verwendet, in der Form, wie er von Alan Miller auf dessen bekannter Webseite für Fortran90-Routinen bereitgestellt wurde.

## Interne Rechengenauigkeit

Es wird programmintern bisher mit „double precision“-Genauigkeit (real(8)) gearbeitet. Ab der Version 2.1.4 wird die „extended precision“-Genauigkeit (real(10)) verwendet, bei der die Zahlen um etwa 3 Dezimalstellen genauer sind. Das bietet den Vorteil, dass speziell die numerischen partiellen Ableitungen um etwa 3 Stellen genauer berechnet werden können.

Dies bedeutet jedoch nicht, dass auch der Wert der Nachweisgrenze um drei Stellen genauer ist. Deren Genauigkeit ist durch das Abbruchkriterium (1.0E‑06 \* Nachweisgrenze) der dazugehörigen Iteration begrenzt, welches einen Kompromiss zwischen Genauigkeit und erforderlicher Rechenzeit darstellt.

In den Gleichungen vom Anwender ganzzahlig geschriebene Zahlen werden als real(10)-Werte interpretiert. Beispiel: Das bei der Darstellung von Zerfallskonstanten benötigte ln(2) wird sowohl in der Form „log(2)“ als auch in der Form „log(2.)“ korrekt als real(10)-Wert interpretiert.

## Hinweise zum Umsteigen auf UR2

Die Verwendung der neueren Programmier-Tools hat zu einigen Änderungen gegenüber UR1 geführt, die nachfolgend angedeutet werden.

***Installation***

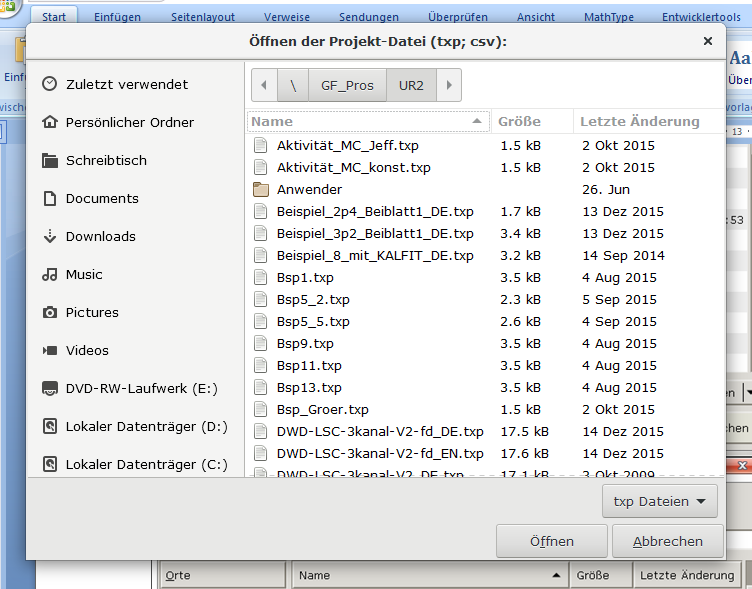
Für UR2 ist nicht nur eine allein unter Windows lauffähige EXE-Datei erforderlich, sondern auch diverse DLL-Dateien, die in einem zusätzlichen Ordner zu installieren sind. Somit ist die Installation von UR2 aufwändiger geworden. Ebenso sind bestimmte Umgebungsvariablen in Windows einzutragen. [Näheres dazu](#URH_WindowsEnvironment_DE)

***Laufzeiten***

Die Rechenzeit ist etwas größer geworden.

***Dialog für Datei-Selektion***

Das grafische Layout dieses Dialogs hat sich gegenüber dem aus Windows bekannten etwas geändert. Das ist darauf zurückzuführen, dass die jetzigen Programmierkomponenten Plattform-unabhängig sind.



Das Scrollen mit der Laufleiste am rechten Rand hat sich geändert. Die Laufleiste wird sichtbar, wenn sich der Mauszeiger dem rechten Rand des Dialogfelds nähert. Das Kontextmenü der rechten Maustaste erlaubt weitere Einstellungsoptionen.

Wird dieser Dialog in dem Modus „**Speichern unter**“ verwendet, *ist darauf zu achten, dass die gewünschte Extension des Dateinamens im Namen-Eingabefeld (ganz oben im Dialog) explizit anzugeben bzw. zu ändern ist.* In diesem Feld ist nur der Dateiname einzugeben, der gewünschte Ordnername wird separat in den Feldern darunter selektiert.

Durch Klicken auf „**Zuletzt verwendet**“ wird in der Mitte die Liste der zuletzt verwendeten Dateien angezeigt. Die Dateinamen werden mit dem *RecentManager* von GTK verwaltet; dieser arbeitet mit einer Datei „recently-used.xbel“, die z.B. bei WIN 7 im Ordner „c:\users\user\AppData\Local\“ zu finden ist („user“ im Ordnername ist durch den Windows-Namen des Benutzers zu ersetzen).

***Dialoge***

Dialoge, auch Teilelemente davon, sind jetzt in ihrer Größe veränderbar, mit der Maus steuerbar.

***Eingabe in Tabellen***

Die Eingabe eines Werts in eine Zelle einer Tabelle soll mit der Eingabe-Taste beendet werden.

***Spaltenblöcke in Tabellen***

In den UR-Tabellen können Spaltenblöcke nicht markiert werden, d.h. der Export eines Spaltenblocks ist nicht möglich. Wohl aber ist der umgekehrte Weg möglich: der Import eines Spaltenblocks, z.B. aus Excel oder aus dem Editor Notepad ++, in den entsprechenden Zielblock in einer UR-Tabelle; [siehe auch](#URH_TABTRICKS_DE).

Eine Tabellenzeile kann durch Klicken in den rechten Teil einer Zelle in dieser Zeile selektiert werden.

***Änderung bei Projekt-Dateien***

In der [Struktur der Projektdateien](#_Struktur_der_Projektdatei) sind zwei Parameter unter dem Abschnitt „Sonstige:“ ergänzt worden:

GamDistAdd=1.0000

GUM\_restricted=F

ModelType=PosLin

***Konfidenz-Ellipse***

Bei linearer Entfaltung mit mehr als einer Ergebnisgröße kann die Konfidenz-Ellipse für je ein Paar von Ergebnisgrößen grafisch dargestellt werden. Die Korrelations-Matrix wird ebenfalls dargestellt.

***Neue Funktion für einen Abklingfaktor***

Mit UR2 wird eine neue Funktion mit drei Parametern eingeführt, die das Schreiben von Formeln für Abklingfaktoren erleichtern soll:

fd(tA, tm, xlam) = exp(-xlam\*tA) \* (1.d0 - exp(-xlam\*tm)) / (xlam\*tm)

Aus einer bisherigen Formel

c\_89=Rn3\*q\*lamS89\*tm1\*exp(lamS89\*(tA+tE+tSr))/(V\*etaSr\*eps1\*(1.- exp(-lamS89\*tm1)))

wird die neue Formel

c\_89=Rn3\*q/fd(tA+tE+tSr, tm1, lamS89) /(V\*etaSr\*eps1 ).

Für den Fall von Sr-90 + Y-90 wird damit aus einer bisherigen Formel

X2 = eSr90A \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) + &

eY90A \* lamY90/(tmess\*(lamY90-lamSr90)) \* &

( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\*tmess)-1.) &

+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

die neue Formel

X2 = eSr90A \* fd(tAS+tstart,tmess,lamSr90) + &

eY90A \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

***Excel-VBA für Kontrolle von UR-Auswertungen***

In dem VBA-Modul ist eine Anpassung an UR2 erfolgt, die die „\“-Zeichen in dem Argument des Befehls-Strings, das den Pfadnamen der von UR2 auszuwertenden Projektdatei darstellt, in „/“-Zeichen umwandelt. Ab Version 2.1.1 ist dies jedoch überflüssig; [Näheres](#_UR_automatisiert_betreiben)

## 2.7 Verwendete Literatur

AKU, 2008*. Moderne Routine- und Schnellmethoden zur Bestimmung von Sr-89 und Sr-90 bei der Umweltüberwachung*. Bericht einer Ad-hoc-Arbeitsgruppe des Arbeitskreises Umweltüber­wachung (AKU). Bericht FS-08-147-AKU des Fachverbandes für Strahlenschutz. TÜV Media GmbH, Köln.

Barlow, R.J., 1999: *Statistics. A Guide to the Use of Statistical Methods in the Physical Sciences.* The Manchester Physics Series. John Wiley & Sons Ltd., Chichester, New York, 204 S.

Blobel, V., Lohrmann, E., 1998. *Statistische und numerische Methoden der Datenanalyse*. B.G. Teubner Stuttgart-Leipzig, 358 S.

Brandt, S., 1999: *Datenanalyse. Mit statistischen Methoden und Computerprogrammen*; 4. Auflage. Spektrum, Akademischer Verlag, Heidelberg-Berlin, 646 S.

Cox, M.G., Harris, P.M., 2001: *Measurement Uncertainty and the Propagation of distributions*. NPL, UK, Paper presented at the 10th International Metrology Congress, Saint-Louis, France, 22-25th October 2001.

Cox, M.G., Forbes, A.B., Harris, P.M., Smith, I.M., 2004: The classification and solution of regression problems for calibration*.* NPL Report CMSC 24/03, (chapter 6.3), National Physics Laboratory, Teddington, UK, 46

http://www.npl.co.uk/ssfm/download/nplreports.html.

Cox, M., Harris, P., Nam, G., Thomas, D., 2006: *The Use of a Monte Carlo Method for Uncertainty Calculation, with an Application to the Measurement of Neutron Ambient Dose Equivalent Rate*. Radiation Protection Dosimetry 121, pp. 12-23.

Cox, M.G., Eiø, C., Mana, G. and Pennecchi, F, 2006b. *The generalized weighted mean of correlated quantities*. Metrologia 43 S268-S275

DIN ISO 11929, Norm, 2010. *Bestimmung der charakteristischen Grenzen (Erkennungsgrenze, Nachweisgrenze und Grenzen des Vertrauensbereichs) bei Messungen ionisierender Strahlung – Grundlagen und Anwendungen. (ISO 11929:2019)*. 83 S.

DIN EN ISO 11929-1, Norm, 2021. *Bestimmung der charakteristischen Grenzen (Erkennungsgrenze, Nachweisgrenze und Grenzen des Überdeckungsintervalls) bei Messungen ionisierender Strahlung – Grundlagen und Anwendungen. Teil 1: Elementare Anwendungen (ISO 11929:2019)*. 53 S.

EURACHEM/CITAC, Guide CG 4, 2000. *Quantifying uncertainty in analytical measurement*. Second edn., 120 S. http://www.eurachem.ul.pt/guides/QUAM2000-1.pdf .

Gilmore, G.: Practical Gamma-Ray Spectrometry. 2nd Edition; J. Wiley & Sons Ltd; 2008.

Hauschild, T., Jentschel, M., 2001. *Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments*. Nucl. Instr. & Meth A 457 (1-2), S 384-401.

Hoover, W. E., 1984: *Algorithms For Confidence Circles and Ellipses*. NOAA Technical Report NOS 107 C&GS 3; Charting and Geodetic Services; Rockville, MD; September 1984

<http://www.ngs.noaa.gov/PUBS_LIB/AlgorithmsForConfidenceCirclesAndEllipses_TR_NOS107_CGS3.pdf>

International Organisation for Standardisation, 1993. *Guide to the Expression of Uncertainty in Measurement (GUM)*. (Geneva: ISO), corrected reprint (1995), also as ENV 13005 (1999).

International Safety Research, Safety Support Series, 2013. Radiation Counting Statistics. Volume 1. Canada.

Janßen, H., 2004. *Determination of Strontium-89 and Strontium-90 in soils and sediments.* In: Quantifying uncertainty in nuclear analytical measurements, IAEA-TECDOC-1401, pp. 149-166.

JCGM 101:2008. *Evaluation of measurement data — Supplement 1 to the “Guide to the expression of uncertainty in measurement” — Propagation of distributions using a Monte Carlo method* GUM, Joint Committee for Guides in Metrology, 2008. (GUM Supplement 1)

JCGM 102:2011. *Evaluation of measurement data – Supplement 2 to the "Guide to the expression of uncertainty in measurement" – Extension to any number of output quantities*. Joint Committee for Guides in Metrology; 2011; http://www.bipm.org/en/publications/guides/gum.htm

Kacker, R.N., Datla, R.U., Parr, A.C, 2002. *Combined result and associated uncertainty from interlaboratory evaluations based on the ISO Guide.* Metrologia 39, 279-293.

Kacker, R.N., Datla, R.U., Parr, A.C., 2004. *Statistical analysis of CIPM key comparisons based on the ISO Guide*. Metrologia 41, 340-352.

Kanisch, G., 2004. *Quantifying Uncertainties in the Alpha-spectrometric Analysis of Environmental Samples*. In: Quantifying uncertainty in nuclear analytical measurements, IAEA-TECDOC 1401, Vienna, pp. 127-139;

Kanisch, G., 2016. *Generalized evaluation of environmental radioactivity measurements with UncertRadio. Part I: Methods with linear unfolding*. Applied radiation and Isotopes 110, 28-41 <http://dx.doi.org/10.1016/j.apradiso.2015.12.003>

Kanisch, G., 2016. *Generalized evaluation of environmental radioactivity measurements with UncertRadio. Part II: Methods without linear unfolding*. Applied radiation and Isotopes 110, 74-86

<http://dx.doi.org/10.1016/j.apradiso.2015.12.046>

Kessel, R., Kacker, R., Berglund, M., 2006. *Coefficient of contribution to the combined standard uncertainty*. Metrologia 43, S189-S195.

Knoll, G.F.. *Radiation Detection and Measurement*, 2nd edition, John Wiley, NewYork,1989, pp. 96-99

Laurence, T. A., Chromy, B., 2009. *Efficient Levenberg-Marquardt Minimization of the Maximum Likelihood Estimator for Poisson Deviates*. Report LLNL-JRNL-420247, November 13, 2009.

Marsaglia, G., Tsang, W.W., 2000. *A Simple Method for Generating Gamma Variables*. ACM Transactions on Mathematical Software, Vol. 26, No. 3, 363–372.

Mathews, I.P., Kouris, K., Jones, M.C., Spyrou, N.M.: Theoretical and experimental investigations on the applicability of the Poisson and Ruark-DeVol statistical density functions in the theory of radioactive decay and counting. Nucl. Instr. Meth. 171 (1979), 369-375.

Michel, R., 2000: *Quality assurance of nuclear analytical techniques based on Bayesian characteristic limits*. J. Radioanal. and Nucl. Chem. 245: 137-144.

Michel, R., Kirchhoff, K., 1999. *Nachweis-, Erkennungs- und Vertrauensgrenzen bei Kernstrahlungsmessungen*. Fachverband für Strahlenschutz e.V., Köln: TÜV-Verlag, Publikation FS-99-108-AKSIGMA, ISSN 1013-4506, 157 S.

Miller, A. Alan Miller’s Fortran Software. <https://jblevins.org/mirror/amiller/>

Moreno, J., Vajda, N., Burns, K., Danesi, P.R., De Regge, P., A. Fajgelj, A., 2004. *Radiochemical determination of Strontium-90 in environmental samples by Liquid Scintillation Counting*. In: Quantifying uncertainty in nuclear analytical measurements, IAEA-TECDOC-1401, pp. 167-193.

Pengra, D., 2008: *Counting statistics of random events: A tutorial*. 9 S.

http://courses.washington.edu/phys433/muon\_counting/counting\_stats\_tutorial\_b.pdf

Pishro-Nik, H., Introduction to Probability:

<https://www.probabilitycourse.com/chapter11/11_1_2_basic_concepts_of_the_poisson_process.php>

Pommé, S., Keightley, J., 2007. *Countrate estimation of a Poisson process: unbiased fit versus central moment analysis of time interval spectra*. Applied Modeling and Computations in Nuclear Science. In: Semkow, T.M., Pommé, S., Jerome, S.M., Strom, D.J. (Eds.), ACS Symposium Series 945. American Chemical Society, Washington, DC, pp.316–334.2007.ISBN0-8412-3982-7.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., 1992. *Numerical Recipes in FORTRAN*, second edn. Cambridge: Cambridge Unversity Press.

Ratel G., Michotte, C., Bochud, F. O.: Uncertainty of combined activity estimations. Metrologia 52 (2015) S30–S41.

Rusconi, R., Forte, M., Caresana, M., Bellinzona, S., Cazzaniga, M.T., Sgorbati, G., 2006. *The evaluation of uncertainty in low-level LSC measurements of water samples.* Appl. Radiat. Isot. 64, 1124-1129.

Salma, I., Zemplén-Papp, É.: Experimental investigation of statistical models describing distribution of counts. Nucl. Instr. Meth. A 312 (1992), 591-597.

Semkow, T.M.: Bayesian Inference from the Binomial and Poisson Process for Multiple Sampling. In: T.M. Semkow, S. Pommé, S.M. Jerome, D.L. Strom (Ed.): Applied Modeling and Computations in Nuclear Science. ACS Symposium Series 945, ACS, Oxford University Press, 2007.

Spyrou, N.M., Foster, J., Jones, M.C., Kouris, K., Matthews, I.P.: Should the Poisson statistical density function be used in the measurement of short-lived isotopes? J. Radioanal. Chem. 61 (1981), 121-130.

Thompson, M.A., 2015: *Gaussian Statistics Lecture.*

<http://www.hep.phy.cam.ac.uk/~thomson/lectures/statistics/GaussianStatistics_Handout.pdf>

Weise, K., Hübel, K., Michel, R., Rose, E., Schläger, M., Schrammel, D., Täschner, M., 2004. *Nachweisgrenze und Erkennungsgrenze bei Kernstrahlungsmessungen: Spezielle Anwendungen. Vorschlag für eine Norm.* Fachverband für Strahlenschutz e.V., Köln: TÜV-Verlag, Publikation FS-04-127-AKSIGMA, ISSN 1013-4506, 31 S.

Weise, K., Wöger, W., 1999. *Meßunsicherheit und Meßdatenauswertung*. Verlag Wiley-VCH Weinheim, 345 S.

Weise, K., Hübel, K., Rose, E., Schläger, M., Schrammel, D. Täschner, M., Michel, R., 2006. *Bayesian decision threshold, detection limit and confidence limits in ionizing-radiation measurement*. Radiat. Prot. Dosimetry 121(1), 52 – 63.

Weise, K., Kanisch, G., Michel, R., Schläger, M., Schrammel, D., Täschner, M., 2009. *Monte Carlo determination of the characteristic limits in measurements of ionizing radiation – Fundamentals and numerics*. Radiation Protection Dosimetry 135 (3), 169–196.

Weise, K., Kanisch, G., Michel, R., Schläger, M., Schrammel, D., Täschner, M., 2013. *Characteristic values in measurements of ionizing radiation – Materials for a critical discussion on Fundamentals and alternatives.* Fachverband für Strahlenschutz e.V., Köln: TÜV-Verlag, Publikation FS-2013-167-AKSIGMA, ISSN 1013-4506, 51 pp.

Wübbeler, G., Krystek, M., Elster, C., 2008. *Evaluation of measurement uncertainty and its numerical calculation by a Monte Carlo method.* Meas. Sci. Technol. 19, 084009 (4pp)

# Erste Schritte

## 3.1 Programmstart, Hauptmenü und Toolbar

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Menü Datei**

Nach dem Start des Programms befindet sich das Programm in dem Zustand, dass ein neues Messproblem - im Programm als **Projekt** bezeichnet - definiert und bearbeitet werden kann. Es sind die TABs "Verfahren" und "Gleichungen" zugänglich.

Mit dem Menü-Punkt "**Datei – Projekt Laden**" oder dem Icon document-open.png kann ein bereits als Projektdatei (Extension.txp) vorliegendes Messproblem in das Programm geladen werden, wobei das Programm den gesamten Rechengang automatisch einmal „durchfährt“, bis zum TAB „Resultate“. Dies kann einige Sekunden in Anspruch nehmen, währenddessen ein zusätzlicher Windowsdialog, der auf diesen Vorgang hinweist, sich erst wieder schließt, wenn das Programm „stehen“ bleibt. Danach kann das Projekt weiterbearbeitet werden.

Falls während des Ladens bei dem automatischen „Durchfahren“ durch die verschiedenen TABs Probleme auftreten, kann das „Durchfahren“ mit dem Menü-Punkt „**Optionen – Project Load – ohne Berechnungen**“ abgeschaltet werden.

Mit dem Menü-Punkt "**Datei - Projekt speichern**" oder dem Icon document-save.png kann ein in Bearbeitung befindliches Messproblem als Projektdatei (Extension txp) unter demselben Dateinamen gesichert oder mit "**Datei - Projekt speichern unter**" oder mit document-save-as.png unter einem neuen Dateinamen gespeichert werden. Mit "**Datei - Projekt schließen**" oder mit  kann die Projektdatei geschlossen werden. Eine Projektdatei kann auch im CSV-Format selektiert und eingelesen oder gespeichert werden.

**Menü Bearbeiten**

Unter dem Menü-Punkt "**Bearbeiten - Report**" wird eine Reportdatei "Report.txt" erzeugt, die den augenblicklichen Stand der Bearbeitung des Projekts wiedergibt. Sie kann mit dem internen Texteditor unter dem TAB Text Editor angesehen bzw. damit auch unter einem anderen Dateinamen gesichert werden kann. In der Reportdatei werden die Ergebnisse für alle Ergebnisgrößen nacheinander ausgegeben, jeweils ab dem Unsicherheiten-Budget.

Unter dem Menü-Punkt "**Bearbeiten – Selektieren Ergebnisgröße**" kann, falls für das bearbeitete Projekt mehr als eine Ergebnisgröße definiert wurde, eine dieser Ergebnisgrößen ausgewählt werden. Auf diese beziehen sich dann die Berechnungen der Unsicherheiten, des Unsicherheiten-Budgets und von Erkennungs- und Nachweisgrenze.

Hinweis: Wenn die **Ergebnisgröße gewechselt** wird, bedeutet dies, dass auch die sich auf diese Größe beziehenden **Selektionen der Brutto-Zählrate und der Netto-Zählrate anzupassen sind**, solange es sich nicht um Auswerteverfahren der linearen Entfaltung handelt. Das Programm wechselt dann zum TAB „Gleichungen“ und gibt einen entsprechenden Hinweis im rechten Feld der Statusbar. Handelt es sich jedoch um das Verfahren der linearen Entfaltung, bei dem Brutto- und Nettozählrate nicht selektiert werden müssen, werden alle Schritte der Berechnungen bis zum TAB „Resultate“ automatisiert in einem Schritt durchgeführt.

Das Menü „**Bearbeiten – Zerfallskurve**“ erlaubt das Editieren einzelner Dialoge oder Zwischenergebnisse, sofern die lineare Entfaltung mit dem Aufruf **LINFIT(..)** in den Gleichungen aktiviert wurde.

* Untermenü „**Modell der Zerfallskurve**“ bzw. das Icon  in der Toolbar:

Hierüber können Parameter des Modells für die Auswertung geändert werden;

* Untermenü „**Dateneingabe**“ bzw. das Icon  in der Toolbar:

Damit wird der Dialog zum Editieren der Eingabedaten der Abklingkurve aufgerufen;

* Untermenü „**Kurvenfit-Tabelle**“ bzw. das Icon  in der Toolbar:

Hiermit wird das Editorfenster zum Einsehen der Fit-Resultate geöffnet.

Bei hierin erfolgten Änderungen erfolgt nach Schließen dieser Dialoge die erneute Auswertung bis hin zum TAB „Resultate“.

Das Menü „**Bearbeiten – Gammaspektrometrie**“ erlaubt das Editieren einzelner Dialoge oder Zwischenergebnisse, sofern die lineare Entfaltung mit dem Aufruf **Gamspk1(..)** in den Gleichungen aktiviert wurde.

* Untermenü „**Gammalinien editieren**“ bzw. das Icon  in der Toolbar:

Damit wird der Dialog zum Editieren der Eingabedaten der einzelnen Gammalinien aufgerufen.

* Untermenü „**Mittlere Linien-Aktivität**“ bzw. das Icon  in der Toolbar:

Hiermit wird das Editorfenster zum Einsehen der Resultate für den gewichteten Mittelwert geöffnet.

Bei hierin erfolgten Änderungen erfolgt nach Schließen dieser Dialoge die erneute Auswertung bis hin zum TAB „Resultate“.

Der Menü-Punkt „**Bearbeiten – Kalibrierkurve**“ ruft einen Dialog auf, der dazu dient die Daten einer Kalibrierkurve einzugeben, daran ein Polynom anzupassen und daraus für einen bestimmten Kalibierpunkt den dazugehörigen Wert mit Unsicherheit in UR weiter zu verwenden.

Der Menü-Punkt „**Bearbeiten – Ändern des Symbolnamens**“ erlaubt es mit Hilfe eines Dialogs den Namen eines darin selektierten Symbols zu ändern. Diese Änderung wird in alle betroffenen Dialoge und Programm-interne Felder übertragen. Falls man den Namen eines Symbols ändern möchte, sollte man diesen nicht in den Gleichungen direkt ändern, sondern diesen Menü-Punkt dafür verwenden.

Der Menü-Punkt „**Bearbeiten – Tausche 2 Symbole**“ erlaubt es in einem Dialog den Namen zweier darin selektierter Symbole auszutauschen. Dies kann dann genutzt werden, wenn man im Falle mehrerer Ergebnisgrößen die Reihenfolge zweier Ergebnisgrößen ändern möchte.

Der neue Menüpunkt „**Bearbeiten – Serielle Auswertung**“ wurde eingeführt, der die mehrfache Auswertung eines Projekts mit partiell geänderten Eingangsgrößenwerten/-unsicherheiten erlaubt. Zur Beschreibung wird auf den im Kapitel 5 neu erstellten Abschnitt 5.6 verwiesen.

**Menü Optionen**

Mit dem Menü-Punkt **„Optionen – Voreinstellungen“** können u. a. die beiden Werte der Quantile der Normalverteilung eingestellt werden, die den sog. Fehlern 1. und 2. Art,  und , entsprechen. Auch die **Sprache** kann selektiert werden, mit der UR betrieben werden soll (Deutsch, Englisch oder Französisch).

Unter dem Menü-Punkt „**Optionen – Projekt Laden**“ kann gewählt werden, ob beim Laden einer Projekt-Datei die Berechnungen beim automatischen „Durchfahren“ durch die TABs abgeschaltet werden; beim Programmstart ist „mit Berechnungen“ eingestellt.

Der ab Herbst 2013 neue Menü-Punkt „**Optionen – LSQ Export nach R?**“ erlaubt im Falle eines Fitverfahrens (Abklingkurven, mit Linfit) den Export bestimmter Eingangsdaten in eine Datei UR-Export-to-R.txt, von der aus man in Teilschritten diese Daten in das Statistik-Paket R importieren und von R auswerten lassen kann. Standardmäßig ist diese Option aktiviert.

Ab Version 2.1.10 wird mit dem **„Optionen – Modelltyp“** zwischen drei Typen von Modellen der Messung unterschieden:

* **positiv linear, mit Nachweisgrenze**:

der Ergebniswert nimmt mit der Brutto-Größe linear zu (dies ist das bisher schon für die Auswertung von Aktivitäts- und Dosimetrie-Messungen verwendete Modell);

* **nur GUM, ohne Nachweisgrenze**:

es sollen nur Wert und Unsicherheit der Ergebnisgröße ermittelt werden, z.B. die Bestimmung einer Masse durch Wägung. In diesem Fall braucht man weder eine Brutto- noch eine Nettozählrate noch eine Nachweisgrenze;

* **negativ linear, mit Nachweisgrenze (neu)**:

der Ergebniswert nimmt mit der Brutto-Größe linear ab, *neu ab Version 2.1.10*;

Ein Anwendungsfall ist die Ermittlung einer Nachweisgrenze für den Radon-222-Emanationskoeffizienten, bei dem in de Ra-226-Quelle die (nicht-emanierte) Rn-222-Aktivität kleiner als die Ra-226-Aktivität werden muss, um die Emanation sicher nachzuweisen (beide Aktivitäten werden mittels Gammaspektrometrie in der Ra-226-Quelle gemessen).

Für ein Projekt, in dem die lineare Entfaltung mit mehr als einer Ergebnisgröße verwendet wird, kann mit dem Menü-Punkt **„Optionen - Berechnen Konfidenz-Ellipse“** die Konfidenz-Ellipse für je zwei der Ergebnisgrößen grafisch dargestellt werden. Auch die Korrelationsmatrix wird in dem dafür vorgesehenen Dialog aufgeführt.

Für das **Menü – QC-Batch-Test** wird auf das Kapitel 3.9 “Programm-Test” verwiesen.

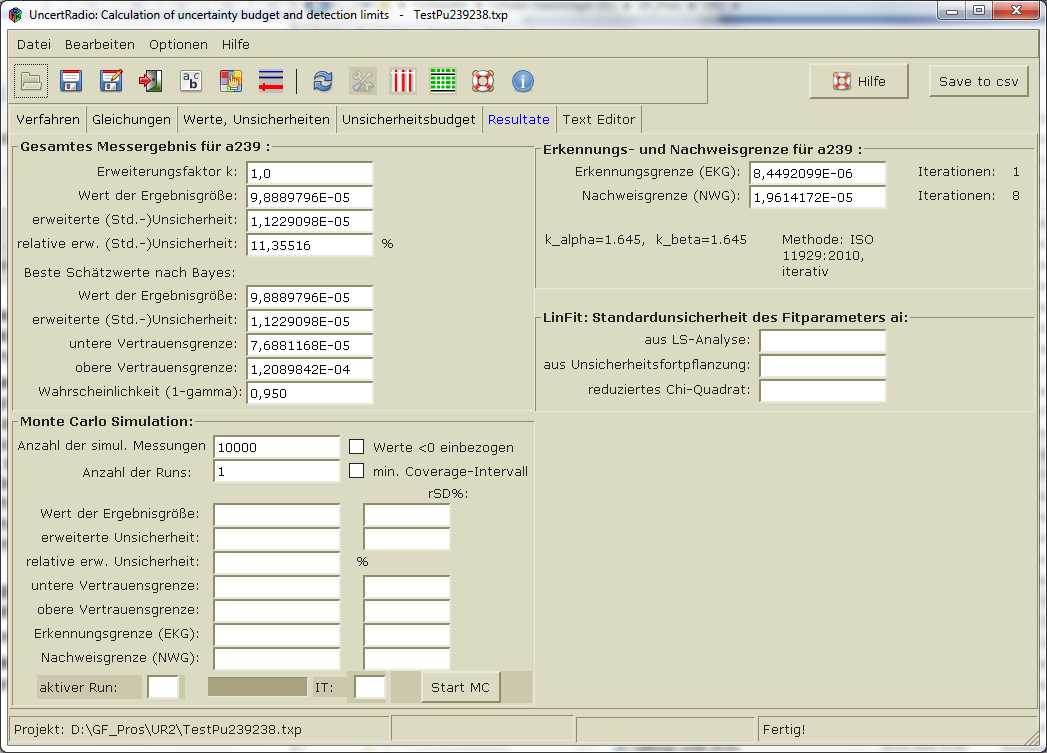
Mit **Menü – Monitor#** kann abgefragt werden, in welchem Monitor das UncertRadio-Fenster eingerichtet ist.

Unter den **restlichen Icons in der Toolbar** sind die wichtigsten:

* das **„Aktualisieren-Icon“ view-refresh.png** , mit dem z. B. nach Änderungen an Eingabedaten oder in anderen Dialogen die Berechnungen vom TAB „Werte, Unsicherheiten“ bis hin zum TAB „Resultate“ automatisch durchgeführt werden;
* das **„Zeilen löschen Icon“ **, das es gestattet, in den Grids „Tabelle der Symbole“ und, in anderen Dialogen, z.B. „Tabelle der Zerfallskurve“, solche Zeilen zu löschen, die zuvor mit der Maus markiert wurden;

dazu kann auch ein Zeilen-Block markiert werden:

mit der Maus in die obere Zeile klicken, dann Shift-Taste drücken und festhalten und dann mit der Maus in die unterste Zeile klicken;

* mit dem Icon help-contents.png kann die CHM-Hilfe aufgerufen werden;
* mit dem Icon dialog-information.png kann in der CHM-Hilfe eine Seite mit Ratschlägen bei Problemen aufgerufen werden;
* das „**Fontname-Icon**“  erlaubt die Änderung des Schrifttyps und der Schriftgröße;
* mit dem „**Farben-Icon**“  können Hintergrund-Farben geändert werden.

*--> Derzeit können die geänderten Farben allerdings NICHT angewendet werden*

* das „**Mittelwert-Icon**“  erlaubt die Eingabe von Einzelwerten einer Variablen und die Selektion solcher Variablen und des Mittelwerttyps.
* Mit dem Icon  kann ein Dialog zur **Ansicht der Parameter einer speziellen Verteilungsdichte** einer Eingangsgröße aufgerufen werden. Dazu muss die Zeile dieser Eingangsgröße in der „Werte, Unsicherheiten“-Tabelle markiert sein.
* Mit dem Icon Ein Bild, das Text enthält.

  Automatisch generierte Beschreibung können kurze Informationen über spezielle UR-Funktionen abgerufen werden.

**Eine Führung des Benutzers erfolgt in der Statusleiste am unteren Ende des UncertRadio-Fensters, in dem Feld ganz rechts. Wenn das Projekt in einigen Details geändert wurde, wird dies in dem Statusleisten-Feld links davon angezeigt („unsaved“).**

Zum Bearbeiten von Tabellen: siehe [**Menü-Bearbeiten-Tabelle**](#URH_TABTRICKS_DE)**.**

Dem Programm ist eine ganze Reihe von Rechenbeispielen als Projektdateien beigefügt. Zum Einstieg in das Programm wird empfohlen, eins davon mit dem Programm zu öffnen und „[**durch das Projekt zu blättern**](#_Existierendes_Projekt_einsehen)“.

## 3.2 Existierendes Projekt einsehen

Um die Arbeitsweise des Programms kennen lernen zu können, empfiehlt es sich, einige bereits fertige Projekt-Beispiele anzusehen. Wie man dazu vorgeht, wird hier kurz aufgezeigt; die Abfolge der Schritte und der zu klickenden Buttons und TABs (Register) entspricht im Wesentlichen auch der Vorgehensweise bei der Erstellung eines eigenen Projekts.

**Eine Führung des Benutzers erfolgt in der Statusleiste am unteren Ende des UncertRadio-Fensters, in dem Feld ganz rechts.**

 **Datei - Laden Project**" document-open.png : Laden eines bereits als Projektdatei (Extensions txp oder csv) vorliegenden Messproblems in das Programm, wobei alle Berechnungen durchgeführt werden und das TAB „Resultate“ sichtbar wird

 TAB „Verfahren“ einsehen

 TAB „Gleichungen“ auswählen

 Button „Lade Symbole aus Gleichungen“ betätigen

 Button „Laden der Symbole aus der ergänzten Symbol-Tabelle“ betätigen

* Button „Alles übernehmen“ betätigen -> damit wird das TAB „Unsicherheiten“ bedienbar

 Ansehen der Gleichungen und der Symbole-Tabelle

 TAB „Werte, Unsicherheiten“ auswählen

 Button „Berechnung der Unsicherheiten“ betätigen -> damit wird das TAB „Unsicherheiten-Budget“ bedienbar

 Ansehen der Unsicherheiten-Tabelle

 TAB „Unsicherheiten-Budget“ auswählen

 Ansehen der Ergebnisse für die Ergebnisgröße

 TAB „Resultate“ auswählen

 Ansehen der Werte von Erkennungs- und Nachweisgrenze

 eine Monte Carlo Simulation der Berechnungen als eine alternative Methode kann durch den Button “Start” gestartet werden

 Schließen der Projektdatei , Laden des nächsten Projekts (s.o.), oder Beenden des Programms: Datei - Exit.

**Sicherheitshinweis:**

Hat man während des Durchblätterns Parameter oder Werte geändert, sollte man bei Nachfrage des Programms das Projekt NICHT sichern (saven), um die Original-Projektdatei nicht zu verändern.

Durch Mausklick innerhalb von Tabellen bzw. im Gleichungsfenster ist auch das Scrollen in ihnen mit dem Mausrad möglich.

## Beispiele zum Ausprobieren

### 3.3.1 Liste der Beispiel-Projekte

**Projektdateien zum Kennenlernen**

Hinweis: In den meisten Fällen enthalten die Beispiel-Projekte die dazugehörige Literatur des Verfahrens; dies kann nach Laden eines Projekts unter dem TAB „Verfahren“ eingesehen werden. Siehe auch: Hilfe-Thema „verwendete Literatur“.

In der Mitte des Jahres 2021 wurden alle Projekte unter dem Aspekt überprüft, ob jeweils passende physikalische Einheiten verwendet wurden. Es wurden nur wenige echte Einheitenfehler gefunden und korrigiert. Zu den erfolgten Änderungen wird auf Abschnitt [2.2.7](#URH_Vorschau_phys_Einheiten) und [3.3.2](#URH_Ueberarbeitung_Einheiten_Projekte) verwiesen.

|  |
| --- |
| Alle farbig hinterlegten Beispiel-Projekte sind als evaluiert zu betrachten, da sie auf entsprechende Publikationen zurückgehen oder durch unabhängige Vergleichsrechnungen von Kollegen bestätigt wurden. |

|  |  |
| --- | --- |
| **Beispiele mit 1 Ergebnisgröße:** |  |
| ***ohne lineare Entfaltung:*** |  |
| ISO-Beispiel-1a\_DE.txp | Alpha-Aktivitätskonzentration in einem flüssigen Material: Messung mit Alpha-Detektor; entspricht Beispiel 1a, Abschnitt D.2.1, DIN ISO 11929:2010 |
| ISO-Beispiel-1b\_DE.txp | Alpha-Aktivitätskonzentration in einem flüssigen Material: Messung mit Ratemeter; entspricht Beispiel 1b, Abschnitt D.2.2, DIN ISO 11929:2010 |
| ISO-Beispiel-2a\_DE.txp | Sr-90 in Boden, Mehrfachbestimmungen; entspricht dem Beispiel 2, Abschnitt D.3.1, DIN ISO 11929:2010; NWG 5% zu klein |
| ISO-Beispiel-2a\_V2\_DE.txp | vereinfachte Version von ISO-Beispiel-2a\_DE.txp |
| ISO-Beispiel-2b\_DE.txp | Sr-90 in Boden, Mehrfachbestimmungen; entspricht dem Beispiel 2, Abschnitt D.3.2, DIN ISO 11929:2010; |
| ISO-Beispiel-2b\_V2\_DE.txp | vereinfachte Version von ISO-Beispiel-2b\_DE.txp |
| ISO-Beispiel-3a\_DE.txp | I-131-Anreicherung auf Luftfilter;  entspricht Beispiel 3(a) aus Abschnitt D.4, DIN ISO 11929:2010 |
| ISO-Beispiel-3b\_DE.txp | I-131-Anreicherung auf Luftfilter; entspricht Beispiel 3(b) aus Abschnitt D.4, DIN ISO 11929:2010 |
| ISO-Beispiel-4\_DE.txp | Aktivitätskonzentration mit Peakauswertung (Ge-Detektor); entspricht Beispiel 4 aus Abschnitt D.5.1, DIN ISO 11929:2010 |
| ISO-Beispiel-5\_DE.txp | Peak-Nettozählrate (NaI(Tl)-Detektor); entspricht Beispiel 5 aus Abschnitt D.5.2, DIN ISO 11929:2010 |
|  |  |
| Michel-2000-b\_DE.txp | I-129-Bestimmung in Boden mit AMS (ein Wert geändert) |
| Sterlinski-2008-NAA\_EN.txp | Messung von Cs in Tabak mit Neutronenaktivierung (**en**) (neues Beispiel) |
| ISO-Neutronen-Dosis\_DE.txp | Messung Neutronendosis in einem gemischten Strahl.-Feld;  ersetzt AKS-Neutronen-Dosis\_DE.txp |
| ISO-Photonen-Dosis\_DE.txp | Messung Photonendosis in einem gemischten Strahl.-Feld;  ersetzt AKS-Photonen-Dosis\_DE.txp |
|  |  |
| Moreno-Sr90\_IAEA-135\_V3\_EN.txp | Sr-90-Bestimmung mit LSC, an IAEA-135-Probe (**en**) |
| Alpha-IAEA-1401-Kanisch\_EN.txp | Alpha-spectrometry of Pu-238 in Fisch (**en**) |
| vTI-Alpha-Americium\_DE.txp | Alphaspektrometrie von Americium-241 in Fisch; recht komplex wegen zwei Interferenzen: a) aus einer Pu-241-Verunreinigung des Pu‑242-Tracers nachwachsendes Am-241 und b) Am-241 als Verunreinigung des Am-243-Tracers **Die entsprechende Excel-Variante liefert das gleich Ergebnis** |
| Ra226\_U235-bei-186keV\_DE.txp | gammaspektrometrische Ra-226-Bestim­mung (186 keV-Linie) mit Abzug eines U-235-Anteils (Interferenz); **Die entsprechende Excel-Variante liefert das gleiche Ergebnis** |
|  |  |
| DWD\_AB-Gesamt-Aeros-Alpha1\_DE.txp | Monitoring der künstl. Alpha-Akt.-Konzentration in Luft, Variante 1 |
| DWD\_AB-Gesamt-Aeros-Alpha3\_DE.txp | Monitoring der künstl. Alpha-Akt.-Konzentration in Luft, Variante 3 |
| DWD\_AB-Gesamt-Aeros-Beta1\_DE.txp | Monitoring der künstl. Beta-Akt.-Konzentration in Luft, Variante 1 |
| DWD\_AB-Gesamt-Aeros-Beta3\_DE.txp | Monitoring der künstl. Beta-Akt.-Konzentration in Luft, Variante 3 |
| Gamma-Dist\_DE.txp | Verwendung der (N+1)-Regel für den Fall sehr kleiner Impulsanzahlen bei der Brutto- und Nulleffektzählrate |
| Lira-GammaDist\_DE.txp | Verwendung der (N+1)-Regel für den Fall sehr kleiner Impulsanzahlen bei der Brutto- und Nulleffektzählrate; **Beispiel aus Lira & Grientschnig, Metrologia, 2010** |
| Fe-55-mit-LSC-und-Standardaddition\_DE.txp | LSC-Messung von Fe-55 nach dem Standardadditionsverfahren  Das Symbol der Bruttozählrate taucht zweimal in der Bestimmungsgleichung auf, sowohl im Zähler als auch im Nenner des Ausdrucks dafür. Weiteres siehe Beschreibung in der Projektdatei. (**Von D. Schrammel (KIT) und Prof. Michel gegen-gerechnet**) |
| NLWKN\_Fe-55\_mit\_KALFIT\_DE.txp | Fe-55-Betamessung für deren Auswertung eine komplette Kalibrierkurve für die Nachweiswahrscheinlichkeit verwendet wird. |
| Beispiel\_8\_mit\_KALFIT\_DE.txp | Beispiel 8 aus dem Beiblatt 1 zu DIN ISO 11929 (2014). |
| Mittelwert-theta\_DE.txp | Sr-90-Bestimmung mit Aufnahme eines Datensatzes mit Mehrfachmessungen einer Bezugsprobe in der Projektdatei, um daraus den Parameter theta abzuleiten, mit dem Mittelwert und Unsicherheit berechnet werden. |
| TemperaturKurve\_KALFIT\_V2\_DE.txp | Interpolation einer linearen Temperatur-Kalibrierkurve, **Beispiel aus JCGM 100:2008**: b(t) = y1 +y2\*(t - t0) |
| BSH\_Gesamt-Gamma\_var1\_DE.txp | Gesamt-Gamma-Messung in Meerwasser, Version 1 |
| BSH\_Gesamt-Gamma\_var2\_DE.txp | Gesamt-Gamma-Messung in Meerwasser, Version 2 |
| Ac228\_binomial\_V2\_DE.txp | Messung kurzlebiges Radionuklid: binomial-verteilter Probenbeitrag zur Bruttozählrate |
| Ra226\_U235-bei-186keV\_DE\_long.txp | Das gleiche Projekt wie weiter oben genannt, aber mit längeren Symbolnamen |
| sumEval\_summe\_V3\_DE.txp | Zusammenfassung von 4 Messungen durch Summation zur Bestimmung einer Ergebnisgröße |
| sumEval\_mitteln\_V3\_DE.txp | Zusammenfassung von 4 Messungen durch Mitteln zur Bestimmung einer Ergebnisgröße |
| ImpulsVorwahl\_DE.txp | Einfache Einkanalmessung zur Demonstration der Messung mit Impulsvorwahl (Impulsanzahl n festgelegt; Messdauer t variabel) |
| Ein Satz von 13 Projektdateien iso11929-4\_Example-6\_EN.txp  bis  iso11929-4\_Example-17\_EN.txp | Projekte, die für die in der **Norm ISO 11929-4:2022** behandelten Beispiele erstellt wurden. **(en)** |
|  |  |
| ***mit linearer Entfaltung:*** |  |
| vTI-Y90-16330\_Blw\_V2\_DE.txp | Y-90-Abklingkurve, mit Blindwertanteil (Fischprobe) |
| vTI-Y90-16671\_Blw\_V2\_DE.txp | Y-90-Abklingkurve, mit Blindwertanteil (Fischprobe) |
| vTI-Y90-16748\_Blw\_V2\_DE.txp | Y-90-Abklingkurve, mit Blindwertanteil (Fischprobe) |
| Mehr-Linien-Nuklid-Aktivitaet-V3\_DE.txp | Aktivität, bestimmt als gewichteter Mittelwert aus mehreren γ-Linien eines Radionuklids |
| La140\_REMSPEC-4Linien-V3\_DE.txp | Aktivität, bestimmt als gewichteter Mittelwert aus 4 γ-Linien des Radionuklids La-140 |
| Ratel\_Annex1\_Beispiel\_DE.txp | Messung der Abklingkurve von Fluor-18 (Halbwertszeit von 1,829 h), **Beispiel aus Ratel et al., Metrologia, 2015** |
|  |  |
| **Beispiele mit mehr als 1 Ergebnisgröße:** | |
| ***ohne lineare Entfaltung:*** |  |
| Janszen-Sr-89-Sr-90\_V4\_DE.txp | Sr-89/Sr-90-Bestimmung in Boden/Sediment 🡪 **IAEA-1401** |
| J-ALUFT-Sr-89-Sr-90\_V2\_DE.txp | Sr-89/Sr-90-Bestimmung in Abluft 🡪 **s. Messanleitung** |
| Galpha\_beta\_Rusconi\_2006\_V2\_DE.txp | Gesamtalpha- und Gesamtbeta-Bestimmung in Wasser, mit LSC-Messungen in zwei Fenstern, mit alpha/beta-Diskriminierung |
| dwd\_sr89\_sr90\_TDCR\_Verfahren\_V2\_DE.txp | Bestimmung der Betastrahler Sr-89 und Sr-90 mit einem TDCR-Verfahren, realisiert in einem HIDEX LSC Counter |
| ***mit linearer Entfaltung*:** |  |
| Sr89-Sr90\_Schrammel\_DE.txp | Sr-89/Sr-90-Bestimmung mit LSC, mit 1 Energiebereich; einfach |
| DWD-LSC-3Kanal-V2\_DE.txp | Sr-89/Sr-90-Bestimmung mit LSC, mit 3 Energiebereichen; komplex |
| DWD-LSC-3Kanal-V2-fd\_DE.txp | Sr-89/Sr-90-Bestimmung mit LSC, mit 3 Energiebereichen; komplex, mit Anwendung der Funktion fd() |
| J-ALUFT\_Sr-89\_Sr-90\_Linf\_DE.txp | Sr-89/Sr-90-Bestimmung in Abluft  (vgl. J-ALUFT-Sr-89-Sr-90\_V2\_DE.txp) |
| LUBW\_Sr-89\_Sr-90\_mit-Sr-85-fixed\_V2\_DE.txp | Sr-89/Sr-90-Bestimmungbei Anwesenheit eines Sr-85-Tracers, dessen Beta-Zählratenbeitrag jedoch NICHT gefittet wird. |
| Sr89-Sr90\_IAEA\_AQ-27\_2013\_V2\_EN.txp | Sr-89/Sr-90-Bestimmung mit LSC; Energiebereich und Nachweiswahrscheinlichkeit zwischen 1. und 2. Messung verschieden |
| Sr89\_Sr90\_LSC-ohne-Sr85\_DE.txp | Sr-89/Sr-90-Bestimmung mit LSC, mit 3 Energiebereichen; ohne Sr-85-Tracer; mit Kovarianzen zwischen einzelnen Fenster-Effizienzen |
| Tritium\_4Bubbler\_used\_1-3\_DE.txp | Messung von HT und HTO in Luft mit Hilfe eines 4fach-Bubblers (nach J.-M. Duda, JER 189 (2018) 235-249), Anwendung linearer Entfaltung, Verwendung der Bubbler 1,2 und 3 |
| Tritium\_4Bubbler\_used\_2-3\_DE.txp | Messung von HT und HTO in Luft mit Hilfe eines 4fach-Bubblers (nach J.-M. Duda, JER 189 (2018) 235-249), Anwendung linearer Entfaltung, Verwendung der Bubbler 2 und 3 |
|  |  |
|  |  |
| **Sonstige Beispiele, aus der Literatur, evaluiert, ohne Nachweisgrenzen:** | |
| Neutron-Dose-Cox-2006\_V2\_DE.txp | Bestimmung Neutronen-Äquivalentdosisleistung |
| Calibration-of-weight-Cox-2001\_V2\_EN.txp | Kalibrierung eines Gewichts (Literaturbeispiel) (**en**) |
| Kessel-1-2006\_DE.txp | Kalibrierung einer Masse von nominal 10 kg |
| Kessel-2a-2006\_DE.txp | Pb-Mol-Massenbestimmung, mit diversen Korrelationen |
| Kessel-2b-2006\_DE.txp | alternative Pb-Mol-Massenbestimmung, mit diversen Korrelationen |
| Wuebbeler-Ex1\_DE.txp | MC-Beispiel mit nicht-gaußförmiger Verteilung |
| Wuebbeler-Ex2\_DE.txp | MC-Beispiel mit nicht-gaußförmiger Verteilung |
| PearsonYork\_mit\_KALFIT\_DE.txp | Anwendung von „gewichtete totale LS“ (WTLS) auf den Datensatz von Pearson & York |
|  |  |
| **Beispiel für ein „negativ“ lineares Modell:** | |
| Rn-222-Emanation\_DE.txp | Ermittlung einer Nachweisgrenze für einen Rn-222-Emanationskoeffizienten |
|  |  |

### 3.3.2 Überarbeitung der physikalischen Einheiten in den Beispielen

Mit der Version 2.4.13 wurde in UncertRadio der Menüpunkt eingeführt, die physikalischen Einheiten der abhängigen Größen durch Berechnung zu ermitteln; siehe die Kapitel 2.27, 2.26 und 7.21. In dem Zusammenhang wurden mit der Anwendung der Routine zu diesem Menüpunkt die bisher in den Beispielprojekten aus Abschnitt 3.3.1 verwendeten Einheiten überprüft. Es war eine Reihe von Anpassungen erforderlich. Die Anpassungen bedeuten, dass die vom Autor des Programms früher gewählten Einheiten nicht immer zweckmäßig waren.

In einigen Fällen wurden die Anpassungen eines Projekts in dem TAB „Verfahren“ direkt festgehalten. Bei den allermeisten Projekten war es nötig, für Detektornachweiswahrscheinlichkeiten, die in der Regel Symbolnamen wie „eps“ tragen, die vorhandenen Einheiten „1“ (leer gelassenes Einheiten-Eingabefeld) durch die Einheit „1/Bq/s“ zu ersetzen. Damit wird erreicht, dass die für die Ergebnisgröße meistens zu erwartende Einheit die Teileinheit „Bq“ anstelle von „1/s“ bekommt. Diese Änderungen sind nur in wenigen Beispielen in ihrem TAB „Verfahren“ festgehalten worden. Bei dem Beispiel J-ALUFT-Sr-89-Sr-90\_V2\_DE.txp wurden zwei Parameter a und b, mit denen das eps2 berechnet wird, mit den Einheiten „1/Bq/s/mg“ bzw. „1/Bq/s“ versehen.

In einem Fall, Ra226\_U235-bei-186keV\_DE.txp, kam es mit der Gleichung RRa = RS – RU5 zur Differenz der Einheiten „1/s“ und „Bq“; hier wurde die erste der beiden Einheiten als Einheit für RRa gesetzt.

In einem anderen Fall, Ac228\_binomial\_V2\_DE.txp, wird die Nachweiswahrscheinlichkeit epsD zweimal verwendet, als Bestandteil des Parameters p, der als Parameter der Binomialverteilung dimensionslos sein soll, und als Teil des Kalibrierfaktors. Dies Dilemma wurde so gelöst, dass das epsD, wie es im Ausdruck für p benutzt wird, dimensionslos bleibt, während eine zusätzliche Variable, epsD\_U, die im Kalibrierfaktor als Detektornachweiswahrscheinlichkeit verwendet wird, die Einheit „1/Bq/h“ und den Wert 1 (ohne Unsicherheit) bekam.

In solchen Beispielen, die in den Gleichungen explizit Umrechnungsfaktoren von 60 für die Einheit „min“ oder 1/1000 für die Einheit „g“ enthalten, wurde diese Umrechnungsfaktoren mit zwei speziellen Schaltvariablen (oder Triggervariablen) versehen; siehe dazu Abschnitt [2.2.6](#URH_Verwendung_Schaltvariable).

Bei einer Reihe von Projekten wurde deren Versionsnummer (\_Vx\_) im Dateinamen erhöht:

Ac228\_binomial\_V2\_EN.txp

DWD\_sr89\_sr90\_TDCR\_Verfahren\_V2\_DE.txp

Galpha\_beta\_Rusconi\_2006\_V2\_DE.txp

J-ALUFT-Sr89-Sr-90\_V2\_DE.txp

Janszen-Sr-89-Sr-90\_V3\_DE.txp

Moreno-Sr90\_IAEA-135\_V2\_EN.txp

sumEval\_summe\_V2\_DE.txp

sumEval\_mitteln\_V2\_DE.txp

vTI-Y90-16330\_Blw\_V2\_DE.txp

vTI-Y90-16671\_Blw\_V2\_DE.txp

vTI-Y90-16748\_Blw\_V2\_DE.txp

Die Berechnung von Einheiten abhängiger Variable (Menüpunkt „Physikal. Einheit testen“) ist mit einer Umrechnung auf Basiseinheiten verknüpft. Die dazu gehörigen Skalierungsfaktoren führen in einigen Fällen dazu, dass die Werte der Ergebnisgrößen sich um diese Faktoren ändern:

Galpha\_beta\_Rusconi\_2006\_V2\_DE.txp: Faktor 1000 (1/g 🡪 1/kg) (permanent geändert)

Sterlinski-2008-NAA\_EN.txp: Faktor 1.0E-9 (Ausgangseinheit ng/g)

sumEval\_summe\_V2\_DE.txp: Faktor 1.0E+4 (1/cm2 🡪 1/m2)

sumEval\_mitteln\_V2\_DE.txp: Faktor 1.0E+4 (1/cm2 🡪 1/m2)

Die Änderungen der letzten drei Projekte kommen nicht zum Tragen, wenn das Menü des Tests physikalischer Einheiten nicht verwendet wird, wenn das Programm im normalen Modus verwendet wird.

Für die Datei mit den Referenzwerten der Beispielprojekte wurde eine neue Version erstellt:

BatListRef\_v04.txt

## 3.4 Dialog Optionen - Voreinstellungen

|  |
| --- |
|  |

Dieser Dialog erlaubt die Einstellung folgender Parameter:

Für die Berechnung von Erkennungs- und Nachweisgrenze:

* die **Quantile der Normalverteilung, und , sowie die ihnen zugeordneten Irrtumswahrscheinlichkeiten *α* und *β* („Fehler 1. und 2. Art“)**; das Programm benötigt nur die Werte von und , man kann aber auch zuerst die gewünschten Werte der Irrtumswahrscheinlichkeiten eingeben, woraus dann die Quantile berechnet werden.

**Voreinstellungen** bei Anlage eines neuen Projekts:  
k-alpha = 1.644854, alpha = 0.05  
k-beta = 1.644854, beta = 0.05.

Wenn der Wert eines Quantils geändert wird, muss der dazugehörige Wert der Wahrscheinlichkeit geändert werden; dazu wird letzterer zunächst als nicht editierbar eingestellt und der Button „anpassen“ (oben) wird aktivierbar. Wird stattdessen ein Wahrscheinlichkeitswert geändert, ist der Ablauf gerade andersherum. Mit Klicken des Buttons „anpassen“ wird der jeweils andere Wert angepasst; bevor das nicht geschehen ist, kann der Dialog nicht geschlossen werden.

* die **Nachweisgrenzenmethode**: hier kann nur die Methode nach DIN EN ISO 11929-1:2021) verwendet werden.
* Zur Berechnung des (zweiseitigen) Vertrauensbereichs (besser: coverage interval), benötigt man die dazugehörige **Wahrscheinlichkeit** , die in das entsprechende Feld eingegeben werden kann. Voreingestellt ist .
* Für die Angabe einer erweiterten Messunsicherheit wird der **Erweiterungsfaktor *k*** benötigt, der in einem dafür vorgesehenen Feld **Erweiterungsfaktor Ausgabe** eingetragen werden kann. Dieser ist standardmäßig mit dem Wert 1 vorbesetzt (interne Variable coverf).
* Stellen in einer Projektdatei die Unsicherheiten der gelisteten unabhängigen Eingangsgrößen bereits erweiterte Unsicherheiten dar, muss für das Einlesen der Erweiterungsfaktor wieder entfernt werden. Der für die Eingangsunsicherheiten der unabhängigen Größen verwendete Erweiterungsfaktor wird unter **Erweiterungsfaktor Eingabe** eingetragen. Dieser ist standardmäßig mit dem Wert 1 vorbesetzt (interne Variable coverin).

Wenn man mit UncertRadio ein Projekt im Dialog bearbeitet, ist normalerweise coverin=1. Der Wert coverin=2 wird eher der automatisierten Betriebsart vorbehalten sein, wenn das UncertRadio aus einer anderen Anwendung heraus automatisiert betrieben wird. Bei z.B. einer Excelanwendung übergibt diese eine in Excel mit Daten gefüllte Projektdatei an UncertRadio zur Auswertung. Sind hier die Eingangsunsicherheiten mit coverin=2 eingegeben worden, und ist coverin=2 in der übergebenen Projektdatei angegeben, wandelt UncertRadio diese Unsicherheiten schon beim Einlesen um, indem diese durch 2 geteilt werden. Die internen Rechnungen laufen dann alle mit dem Erweiterungsfaktor 1. Erst bei der Ausgabe in eine externe CSV-Datei, die danach von Excel wieder eingelesen wird, werden die auszugebenden Unsicherheitswerte mit dem Wert der Variablen coverf multipliziert, der im Projekt ebenfalls anzugeben ist.

Bei der automatisierten Auswertung mit Hilfe von UncertRadio ist zu beachten, in welcher Weise die Eingangsunsicherheiten in die an UncertRadio zu übergebende Projektdatei einzutragen sind: siehe dazu das [Kapitel 3.6.3](#URH_Hinweise_Eingabe_Unsicher_DE).

* Der Dialog enthält ein **Eingabefeld für die Variable GamDistAdd** ergänzt. Diese gibt das Argument x in der (N+x)-Regel für Impulsanzahlen oder Zählraten an, zwischen 0 und 1 liegend. Den Zahlen 0, ½ und 1 für GamDistAdd entsprechen in einer Bayes’schen Betrachtungsweise üblichen Prioren, die proportional zu sind, mit *c*=GamDistAdd. Für die entsprechende Variable, für welche die (N+x)-Regel ausgewählt wurde, wird eine entsprechende Gamma-Verteilung angenommen.

|  |  |  |
| --- | --- | --- |
| *c*=GamDistAdd | (1-c) | Mittelwert |
| 0 | 1 | N+0 |
| 1/2 | 1/2 | N+1/2 |
| 1 | 0 | N+1 |

* Die **Sprachdarstellung** in den Programmdialogen und das Listentrennzeichen für CSV-Dateien können während des Betriebs umgestellt werden (Wechsel zwischen DE und EN). Siehe dazu: [Länderspezifische Einstellungen](#URH_COUNTRYSPECS_DE).

Siehe auch: [**Hinweis zum nachträglichen Aufruf dieses Optionen-Dialogs**](#URH_AEND_OPTION_DE)

## [Ratschläge bei Problemen](#URH_FIRSTAID_DE)

* Wenn das Programm ohne Fehlermeldungen durchläuft, die Ergebnisse aber unplausibel sind, z.B., die Nachweisgrenzen-Iteration konvergiert nicht:

dann unbedingt prüfen, ob für die Netto- und Bruttozählraten die richtigen Symbole selektiert sind.

Daher muss für die Nettozählrate immer eine eigene Hilfsgleichung angelegt werden.

* Die Nachweisgrenzen-Iteration klappt auch dann nicht mehr, wenn die relative Unsicherheit des Kalibrierfaktors in die Nähe von 61% kommt oder diesen Wert übersteigt; die Ursache dafür ist im Verfahren der ISO 11929 selbst zu suchen.
* Auch Werte in der Tabelle des Dialogs „Werte, Unsicherheiten“, die noch den Wert null haben, können zu ähnlichen Problemen führen; dies kann bei nachträglichem Um-Editieren der Gleichungen geschehen, wobei es noch zu Verschiebungen zischen Symbol- und Werte-Listen kommen kann. Letzteres zu prüfen ist nach signifikanten Änderungen der Gleichungen ratsam.
* Durch Änderungen der Gleichungen *überflüssig gewordene Symbole*: zu deren Entfernung wird auf [Abschnitt 4.2](#URH_zu_loeschende_Symbole_DE) verwiesen.
* Aufrufe der Funktionen LINFIT bzw. KALFIT sollen jeweils in einer eigenen Hilfsgleichung definiert werden; hinter ihren schließenden Klammern dürfen in derselben Gleichung keine weiteren Operatoren oder Variablen auftauchen.
* Für Projekte mit linearer Entfaltung (z.B. für kombinierte Sr-89/Sr-90-Analysen):

Im Dialog für die Definition des Modells einer Abklingkurve sind seit der Version 1.08 Änderungen erfolgt. Daher ist zu empfehlen, bei Problemen auch diesen Dialog mal aufzurufen, um zu sehen, ob die dort eingestellten Selektion/Optionen noch den angenommenen entsprechen.

* Das Toolbar-Icon view-refresh.pnginitiiert eine Neuberechnung der Kennwerte nach Änderungen an Werten. Wenn das Programm danach im TAB „Werte, Unsicherheiten“ stehen bleibt, nochmals das Icon view-refresh.pngverwenden, um ans Ende zu gelangen.
* Editieren einer Tabellenzelle:

ein Klick in die Zeile markiert diese;

ein zweiter Klick, in eine Zelle dieser Zeile, erlaubt das Editieren der Zelle.

Eine Eingabe in die Zelle muss mit ENTER bestätigt werden, die TAB-Taste reicht dafür nicht.

* Einfügen aus der Windows-Zwischenablage in eine Tabellenzelle:

Das Einfügen mit Strg-V funktioniert nur beim ersten Mal, danach nicht mehr!

Danach funktioniert dies nach Markieren des Zellinhalts nur bei Verwendung des Punktes „Einfügen“ des Kontext-Menüs;

* Das Optimieren der Spaltenbreiten von Tabellen kann durch Doppelklicken auf den vertikalen Trennstrich zwischen zwei Spaltenköpfen erfolgen; zuvor ändert sich dabei die Form des Mauszeigers.
* Zum Einfügen von Spaltenblöcken aus Excel oder aus Notepad++ in Tabellen-Spalten in UR2 wird auf das Ende des Hilfe-Kapitels 7.7 verwiesen.

## 3.6 Struktur der Projektdatei

Die Kenntnis des Aufbaus einer Projektdatei kann für eine automatisierte Anwendung von UncertRadio nützlich sein. Z.B. bei regelmäßigen Wiederholungen einer Sr-89/Sr-90-Analyse mit linearer Entfaltung, die immer nach dem gleichen Schema erfolgt, wird man eine Grundform der UncertRadio-Projektdatei erstellen, in der sich bestimmte Teile von Messung zu Messung wenig oder gar nicht ändern. Ein anderer Teil jedoch, die Reihe der gemessenen Zählraten, wird sich von Messung zu Messung ändern. Diese „beweglichen“ Daten können beispielsweise mit Visual Basic in Excel in eine solche Projektdatei direkt hineingeschrieben werden.

Neben der bisher schon verwendeten Ausführung einer Projektdatei als Textdatei \*.TXP, deren Struktur hier ausführlicher dargestellt wird, kann ab Herbst 2013 auch die Variante als CSV-Datei verwendet werden; darauf wird am Ende dieses Hilfethemas eingegangen.

### 3.6.1 Projektdatei im \*.TXP-Format als Textdatei

Dies kann so geschehen, dass man zunächst die Grundform dieser Projektdatei Zeile für Zeile in eine Arbeitskopie – ebenfalls mit der Dateierweiterung .txp, z.B. „Test2.txp“ – kopiert. Nur bei bestimmten Schlüsselwörtern (mit @ eingeleitet, s.u.) werden an der Stelle die der aktuellen Messung entnommenen Daten in diese Kopie hineingeschrieben. Nach dem Schließen dieser Arbeitskopie kann UR z.B. direkt von Visual Basic aus so gestartet werden, dass es diese Arbeitskopie dabei auch lädt, z.B. mit einem „Start Uncertradio.exe Test2.txp“. Dieser externe Start kann auch über die Windows-Konsole erfolgen:

D:\UR > Uncertradio.exe Test2.txp

Die folgende Übersicht zeigt den einfachen Aufbau der Struktur einer Projektdatei (Extension TXP).

Note: numbers always with . as decimal point (no comma!)!

| Deutsche Beschreibung: | | | | | | | | English description: | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project:** Name der CSV-Projektdatei / name of CSV project file | | | | | | | | | | | |
| **@Titeltext:** | | | | | | | | | | | |
| Text zur Beschreibung des Verfahrens | | | | | | | | text describing the procedure | | | |
| **@Formeltext:** | | | | | | | | | | | |
| es folgen die Gleichungen zur Berechnung der Aktivität | | | | | | | | is followed by the equations for calculating the activity | | | |
| **@FormeltextFit:** | | | | | | | | | | | |
| es folgen die Gleichungen der energieabhängigen Funktionen im Verfahren *Gamspk1* oder der Abklingfunktionen im Verfahren *Linfit1*. | | | | | | | | is followed by equations describing the energy dependent curves in the method *Gamspk1* or the decay functions in the method *Linfit1*. | | | |
| **@Symbole-GRID:** | | | | | | | | | | | |
| folgende Variablen werden Werte zugeordnet: | | | | | | | | values are attributed to the following variables: | | | |
| nchs= 1  nEGr= 1  ngrs= 8  nab= 3  nmu= 5 | Anzahl der Messkanäle  Anzahl der Ergebnisgrößen  Anzahl der Variablen (Symbole)  Anzahl der abhängigen Symbole (a)  Anzahl der unabhängigen Symbole (u) | | | | | | |  | | number of counting channels  number of output quantities  number of variables (symbols)  number of dependent symbols (a)  number of independent symbols (u) | |
| Hiernach folgen zeilenweise die Informationen, die zum „Symbole-Grid“ im TAB „Gleichungen“ gehören, für jedes Symbol (in den Zeilen dient das Zeichen # als Trennzeichen):   * der Name, * der Typ (a oder u), * die Einheit und * die „Bedeutung“; | | | | | | | | For each symbol from the „symbol grid” of the TAB “Equations” following information is given (with # as separating character):   * name, * type (a or u), * unit and * the symbol‘s „meaning“; | | | |
| **@Menu1 und Menu2:** | | | | | | | | | | | |
| Nummern der Symbole der: | | | | | | | | numbers of the symbols of: | | | |
| knetto= 2 | | Nettozählrate | | | | | | net counting rate | | | |
| kbrutto= 0 | | Bruttozählrate (0: nicht verwendet bei Verfahren mit Entfaltung) | | | | | | gross counting rate (0: not used with procedures using linear unfolding) | | | |
| **@Unc-Grid:** | | | | | | | |  | | | |
| Es folgt für jede Variable eine Zeile, in der mit einer csv-Struktur (# als Trennzeichen) folgende Elemente (Zahlenwerte: double precision; -999.d0 bedeutet „nicht definiert“) stehen: | | | | | | | | for each variable the following elements are given in one row (with a csv structure; # as separating character; numbers: double precision, -999.d0 means “not defined”: | | | |
| * Symbol; * Wert der Größe; * Index des Verteilungstyps:   1 Normalverteilung  2 Rechteckverteilung  3 Dreiecksverteilung  4 (N+x)-Regel für Impulsanzahlen  5 Lognormalverteilung  6 Gammaverteilung  7 Binomia+Poisson-Verteilung  8..2-ParameterBetaverteilung  9.. t-Verteilung   * Text-Formel für die Standardunsicherheit; * Zahlenwert der Standardunsicherheit; * Halbreite der Verteilung (Dreieck oder Rechteck); * 1: absolute bzw. 2: relative Unsicherheitsangabe bei der Eingabe; * berechnete absolute Standardunsicherheit. | | | | | | | | * symbol; * value of the quantity; * Index of the distribution type   1 normal distribution  2 rectangular distribution  3 triangular distribution  4 (N+x) rule for counting numbers  5 lognormal distribution  6 gamma distribution  7 binomial+poisson distribution  8 2-parameter beta distribution  9..t-distribution   * text formula for standard uncertainty; * value of the standard uncertainty; * Half-width of a distribution (triangular or rectangular); * interpretation of the uncertainty:   1: absolute, 2: relative;   * calculated absolute standard uncertainty. | | | |
| **@Covar-Grid:** | | | | | | | | | | | |
| Falls Kovarianzen definiert sind, folgen diese hiernach folgt für jedes Variablenpaar:   * die zwei Symbol-Nummern des Variablenpaars; * 1 für Kovarianz oder 2 für Korrelation; * ggf. Textformel für die Kovarianz; * Wert der Kovarianz (1) bzw. Korrelation (2) | | | | | | | | If covariances are defined, they are characterized for each pair of variables:   * the two symbol numbers of the pair; * 1 for covariance, or 2 for correlation; * if necessary, covariance formula as text; * value of covariance (1) or correlation (2) | | | |
| **@Abkling-Grid:** | | | | | | | | | | | |
| ModPar= 1 1 1 1 1 0 0  8 Werte (8 values):  #1 -3: ifit(.)  #4:  #5:  #6:  #7:  #8:  01.01.2005 12:12:07  1 | | | 7 checkbox/radiobutton-Werte im Grid;  welche Parameter fitten? (1: fitten; 2: festhalten; 3: weglassen)  gewichteter Fit?  Benutzung Kovarianzen  Wahl der Fitmethode:  0: WLS, 1: PLSQ,  2: PMLE, 3: WTLS  WTLS: nein/ja (jetzt überflüssig)  Nachweiswahrscheinlichkeiten variabel zwischen Messungen:  0: nein; 1: ja  Datum/Uhrzeit Sr/Y-Separat.;  Zeiteinheit (1: s; 2: min). | | | | | 7 checkbox/radiobutton values within the grid defining the model of the decay curve;  which parameters to fit? (1: fit; 2: fix it; 3: omit it)  use weighted fit?  use covariances?  use of fitting method:  0: WLS, 1: PLSQ,  2: PMLE, 3: WTLS  WTLS: no/yes (now redundant)  counting efficiencies vary with measurements:  0: no; 1: yes  datetime of e.g. Sr/Y separation:  time base (1: s; 2: min). | | | |
| Hiernach folgen, für jede Messung der Abklingkurve eine Zeile, die folgende Daten enthält:   * Startdatum der Messung – oder Zeitdifferenz zum Abtrenn-Datum; * Messdauer der Bruttomessung; * Bruttoimpulse; * Bruttozählrate; * Unsicherheit der Bruttozählrate; * Messdauer der Nulleffektmessung; * Impulse der Nulleffektmessung; * Nulleffektzählrate; * Unsicherheit der Nulleffektzählrate; * Nettozählrate; * Unsicherheit der Nettozählrate; | | | | | | | | Hereafter, for each measurement of the decay curve one row is given, containing following data:   * start date of the measurement – or time difference to datetime of the chemical separation; * gross counting duration; * gross counts; * gross counting rate; * uncertainty of gross counting rate; * counting duration of background meas.; * counts of background measurement; * background counting rate; * uncertainty of background counting rate; * net counting rate; * uncertainty of net counting rate; | | | |
| **@Gamspk1-Grid:** | | | | | | | | | | | |
| CurveUse= 0 0 0 0 | | | | | für Checkbox-Werte im Dialog „Definition energieabhängiger Kurven“ (nur UR1-Projekte); wird nicht mehr verwendet; | | | for checkbox values within the dialog „Definition of energy dependent curves“ (only UR1 projects); no longer used; | | | |
| UnitRadio= ***1*** 1 ***1*** 2 ***1*** ***1*** ***1***  Reduced to, since V. 2.4.19:  UnitRadio= ***1*** ***1*** ***1*** ***1*** ***1***  *(see section 7.16)* | | | | | Werte der Radiobuttons (1 oder 2) im Dialog zur Eingabe der Gammazählraten | | | Radio button values (1 or 2) within the dialog for input of counting rates of gamma lines | | | |
| MeanTyp= 1 | | | | | Mittelwerttyp | | | type of mean used for averaging activities | | | |
| FBT= 1.1200 | | | | | (1+b/(2L))-äquivalenter Faktor | | | factor equivalent to (1+b/(2L)) | | | |
| EcorrUse=1 | | | | | sollen Effizienz-Korrelationen verwendet werden? | | | shall peak efficiency correlations be used? | | | |
| Hiernach folgen, für jede Gammalinie eine Zeile:   * 1 für Verwendung der Linie, sonst 0; * Energie der Linie (keV); * Wert der Nettozählrate, Rnet; * Wert der Untergrund-Kontinuum-Zählrate, RT; * Wert der Netto-Nulleffektzählrate, Rbg im separaten Spektrum; * Wert der Effizienz, effi; * Unsicherheit der Effizienz; * Gamma-Emissionsrate, pgamm; * Unsicherheit von pgamm; * Wert der Selbstschwächungskorrektion, f\_att; * Unsicherheit von f\_att; * Wert der Koinzidenzsummationskorrektion,  f\_coin; * Unsicherheit von f\_coin | | | | | | | | Hereafter, for each gamma line one row:   * 1 for using the line, 0 otherwise; * energy of the line (keV); * value of net counting rate; * value of the background continuum counting rate, RT; * value of net counting rate from separate background spectrum, Rbg; * value of the peak efficiency, effi; * uncertainty of effi; * Gamma-emission rate, pgamm; * uncertainty of pgamm; * value of self-attenuation correction, f\_att; * uncertainty of f\_att; * value of coincidence summing correction,  f\_coin; * uncertainty of f\_coin | | | |
| **@Kalfit-Grid:** | | | | | | | | | | | |
| KalPars= 5 2  #1: 5  #2: 2  Ctitel | | | | 2 Integer-Feld-Werte im Dialog:  Anzahl der Kalibrierpunkte  Grad des Fit-Polynoms (0-3)  (0: Mittelwert der Y-Werte)  Titel der Kalibrierkurve | | | | | 2 integer field values in the dialog:  number of calibration points  Degree of fit polynomial (0-3)  (0: Mean of the y-values)  Title of the calibration curve | | |
| Hiernach folgen zeilenweise 4 Werte der Kalibrierkurve:  X-Wert,  Std Unsicherheit des X-Werts (oder leer oder 1),  Y-Wert,  Std Unsicherheit des Y-Werts (oder leer oder 1)  (1 bedeutet identische Wichtung) | | | | | | | Hereafter, 4 values of the calibration curve follow, line by line:  X value,  Std uncertainty of X value (or empty or 1),  Y value,  Std uncertainty of Y value (or empty or 1)  (1 means identical weighting) | | | | |
| **@Sonstige:** | | | | | | | | | | | |
| kalpha=3.000000  kbeta=1.644854  coverf=1.000  coverin=1.0  1-gamma=0.9500  GamDistAdd=1.0  ModelType=PosLin  BinPoi=8 10 12 9  **Note:** NWGTyp is no  longer used | | | Quantil zum Fehler 1. Art α;  Quantil zum Fehler 2. Art β;  Erweiterungsfaktor für die Ergebnisunsicherheit;  Erweiterungsfaktor für die Eingabeunsicherheit unabh. Größen;  Wahrscheinlichkeit zum Vertrauensintervall  Parameter für Gammaverteilung;  ModellTyp  (one of: PosLin/GUMonly/NegLin);  Binom/Poisson-Fall: Symbol-Nummern von p, R0, tm, lambda | | | | | Quantile associated with type I error, α;  Quantile associated with type II error, β;  coverage factor for uncertainty of the output quantity;  coverage factor for uncertainties of the independent input quantity;  probability associated with the confidence interval  Parameter for Gamma distribution;  Type of model (one of PosLin/GUMonly/NegLin);  Binom/Poisson case: symbol numbers of p, R0, tm, lambda | | | |
| **@Means:** | | | | | | | | | | | |
| meantyp= 2 2  refmean= 2 (> 0)  gefolgt von / followed by:  n Zeilen der Art: / n lines like:  „name\_data 1.5 2.0 2.2…“ | | | | | | eine Folge von n Mittelwert-Typen (integer, Werte 1, 2 oder 3) der n Mittelwert-Variablen  Bedeutung der meantyp-Nummer:  („unbekannte zufällige Einflüsse“)  1: siehe Gl. (1) in 6.9.1  2: siehe Gl. (3) in 6.9.1  3: siehe Gl. (5) in 6.9.1  für Referenz-Zweck selektierter Datensatz  „bekannte Einflüsse“: siehe Gl. (6) u. (7) in 6.9.1  Ein Variablen-Bezeichner, gefolgt von den Einzelwerten der Variablen mit dem Namen „name“ aus der Symbolliste | | | | | a series of n types of means (integer, values 1, 2 or 3) of the n mean variables  Meaning of meantyp number:  (“unknown random influences”)  1: see Eq. (1) in 6.9.1  2: see Eq. (3) in 6.9.1  3: see Eq. (5) in 6.9.1  data record selected for reference purpose  “known influences”: see Eqs. (6) and (7) in 6.9.1  a variable-identifier, followed by the single values of the variable with the name “name” of the symbol  list |

### 3.6.2 Projektdatei im Excel-kompatiblen \*.CSV-Format

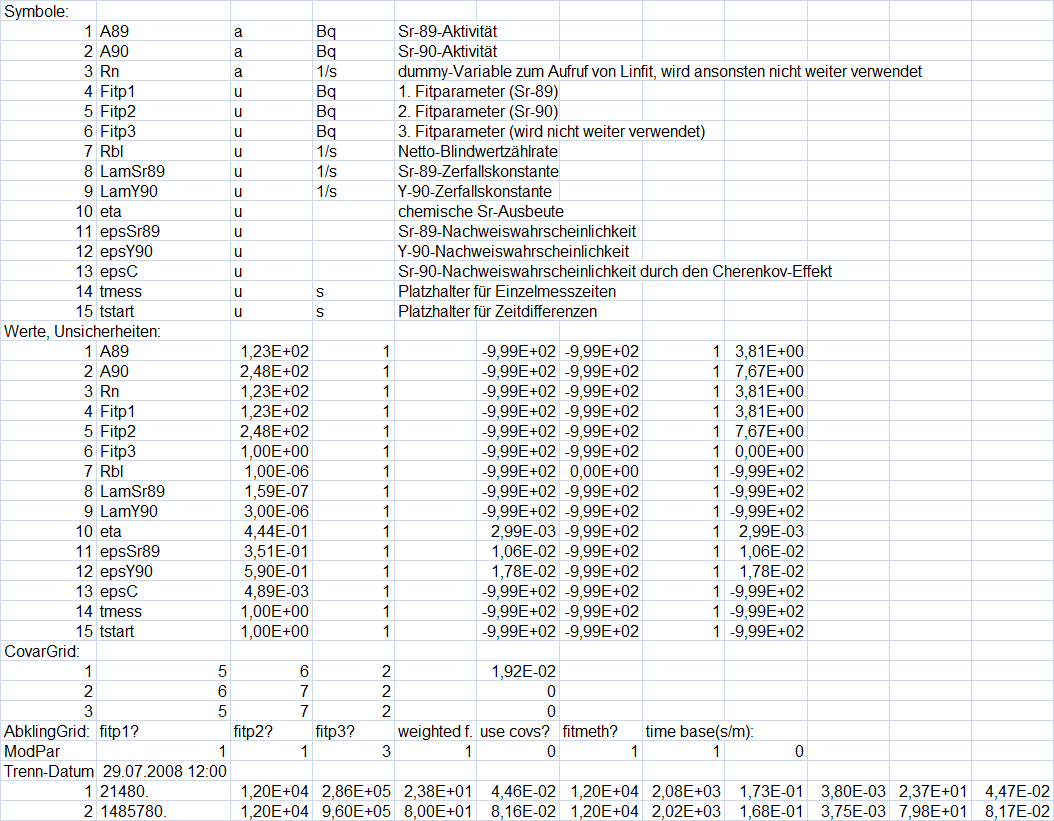
**Projektdatei im Excel-kompatiblen \*.CSV-Format**

Das CSV-Format ist ähnlich aufgebaut wie für das \*.TXP-Format beschrieben. Die für letztere gegebene Strukturbeschreibung findet sich auch im CSV-Format wieder. Die nachfolgende Abbildung zeigt ein Beispiel. Das Format ist unter Einbeziehung der Beschreibung des TXP-Formats weitgehend ohne umfangreiche Erklärungen verständlich. **@Sonstige** aus dem txp-Format taucht hier als **Optionen** auf.

**WARNHINWEIS**: Lädt man mit UR eine TXP-Projektdatei und speichert diese im CSV-Format ab, darf diese CSV-Datei nachträglich nicht mehr mit Excel bearbeitet und in Excel gesichert werden, da hierbei die Zahlenwerte in ihren Dezimalstellen deutlich verkürzt und damit verändert werden. Für eine solche Nachbearbeitung sollte besser ein einfacher Texteditor (Notepad) verwendet werden.

Da die Automatisierung von UR über eine Excelanwendung laufen kann, sei hier erwähnt, dass dieses Problem beim VBA-programmierten Exportieren des CSV-Projekts aus Excel heraus nicht auftritt.





### 3.6.3 Hinweise zur Eingabe von Eingangsunsicherheiten

Es ist zu beachten, dass Unsicherheiten von Eingangsgrößen in verschiedenen Feldern eingegeben werden können. Diese Felder sind in der folgenden Tabelle, die dem TAB „Werte, Unsicherheiten“ entspricht, nochmal aufgeführt.

Im \*.TXP-Format der Projektdatei sind dies in dem Abschnitt @Unc-Grid: die unter dem 6. bis 9. Bullet beschriebenen Felder; im CSV-Format sind dies im Abschnitt „Werte, Unsicherheiten“ die Spalten 5 bis 8 (E bis H).

Auf keinen Fall darf ein Unsicherheitswert in die Spalte 10 eingetragen werden, die Eingabe muss, je nach Anwendung, in den Spalten erfolgen, die nachfolgend mit 6 bis 9 benannt sind.

|  |  |  |  |
| --- | --- | --- | --- |
| 6: | **StdAbw-Formel** | Textfeld | Formel für die Standardabweichung der Messgröße  (der interne Erweiterungsfaktor ist immer 1); als Dezimalpunkt ist immer der „.“ zu verwenden! |
| 7: | **StdAbw-Wert** | Zahlenfeld | Wert der Unsicherheit für Normal-Verteilung;  Bei Selektion der (N+x)-Regel ist in dieser Zelle **nichts** einzu­tragen!  (der interne Erweiterungsfaktor ist immer 1) |
| 8: | **Halbbreite** | Zahlenfeld | Halbbreite von Rechteck- bzw. Dreieckverteilung  (der interne Erweiterungsfaktor ist immer 1) |
| 9: | **abs./rel.** | Auswahlfeld | Festlegung, ob der Unsicherheitswert in Sp. 6, 7 oder 8 als *absoluter* oder *relativer* Wert zu nehmen ist. |
| 10: | **abs. Std.Uns.** | Zahlenfeld | vom Programm aus den Werten der Spalten 6, 7, 8 und 9 berechneter absoluter Wert der (kombinierten) Standardunsicherheit  Achtung: Ein vom Benutzer hier eingegebener Wert wird immer vom Programm überschrieben! |

Wenn in die Spalten 6 bis 8 Werte mit einem Erweiterungsfaktor ungleich eins eingetragen werden, muss der Parameter coverin in derselben Projektdatei auf diesen Erweiterungsfaktor gesetzt sein. UncertRadio konvertiert dann alle Eingangsunsicherheiten auf den internen Erweiterungsfaktor 1; nach den Berechnungen gibt UncertRadio die Unsicherheiten mit dem im Projekt definierten Erweiterungsfaktor coverf multipliziert aus.

## Schrift und Farben

Die Einstellung von Schriften und Farben erfolgt üblicherweise mit sogenannten CSS-Dateien („cascaded style sheets“), die z.T. recht komplex sein können. Für UR wird dies auf die Verwendung einer stark verkürzten Datei **Settings.ini** (zu GTK+3 gehörend) reduziert. Diese ist auf Windows eingestellt und besitzt nur zwei Einträge.

Datei **Settings.ini**:

[Settings]

gtk-theme-name = win64

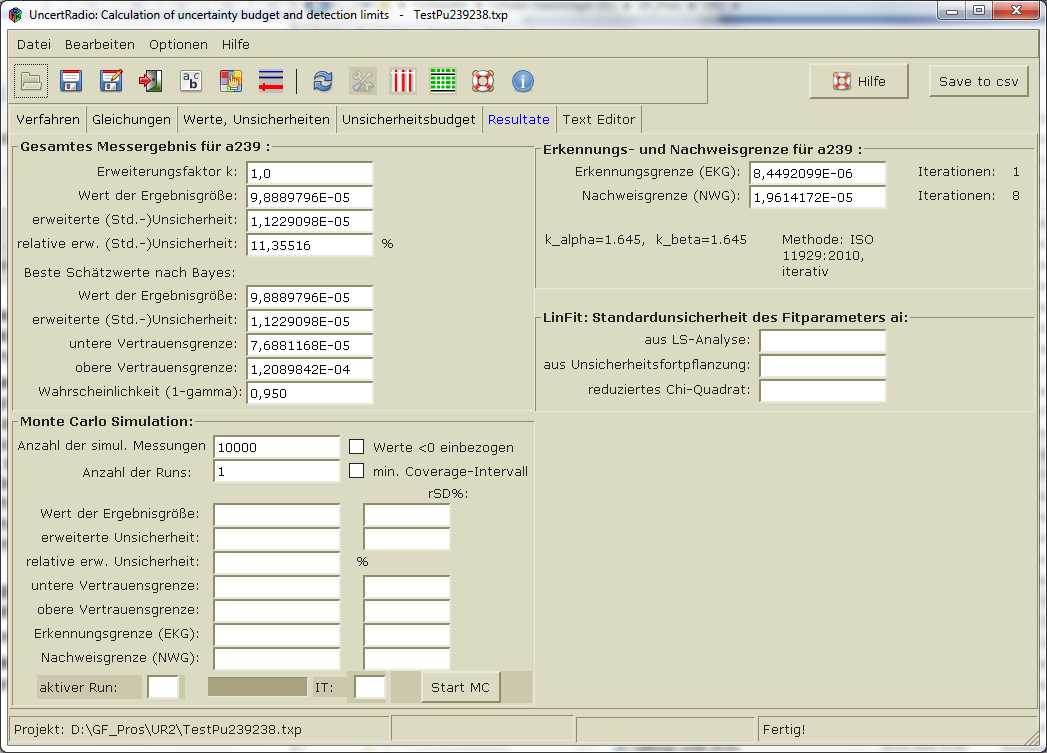
gtk-font-name = Sans Normal 10

Die jeweils links vom Gleichheitszeichen stehenden Strings dürfen nicht verändert werden. UR lädt beim Programmstart die Settings.ini-Datei. Dies Datei soll im selben Pfad stehen wie UncertRadio.exe auch.

Für die Einstellung von anderen Werten für *gtk-font-name* steht in der Toolbar der **Fontname**- **Button** zur Verfügung.

Mit dem **Fontname-Button ** können auf einfache Weise die Schriftart und/oder die dazugehörige Schriftgröße geändert werden. Dabei ist zu beachten, dass bei der Erhöhung der Schriftgröße allerdings auch das Programmfenster größer wird, ggf. zu groß.

Mit dem Dialog-Button „**Anwenden**“ wird der neue Fontname auf das Programm angewendet. Bei Gefallen kann dieser Fontname mit dem Dialog-Button „**Speichern**“ in der Datei Settings.ini gesichert werden; dies setzt voraus, dass der Fontname zuvor auch auf das Programm angewendet wurde. Der neue Fontname wird nicht gesichert, wenn man den Dialog dagegen mit „**Beenden**“ schließt.

*Hinweis: der Color-Button kann derzeit nicht verwendet werden.* Mit dem **Color-Button**  können die Farben für den *permanenten* Dialog-Hintergrund oder den *selektierten Hintergrund* (der die schwebende Selektion von Dialogobjekten andeutet) ausgewählt werden. Die eigentliche Farbauswahl erfolgt erst nach Klicken auf das rechteckige Feld, das den Colorbutton beinhaltet. Danach kann in einem weiteren Dialog eine Farbe aus einer vorgefertigten Palette ausgewählt werden. Nach Klicken des Plus-Zeichens eröffnet sich ein weiterer Dialog, der eine Maus-gestützte Farbauswahlunterstützt. In diesem Dialog wird oben ein Textfeld angezeigt, z.B. mit dem Inhalt „#E7E3BA“, das den RGB-Farbcode in hexadezimaler Darstellung enthält. Links neben diesem Textfeld wird die gerade selektierte Farbe dargestellt. Einen solchen Farbcode kann man in dieses Feld auch direkt eingeben, um danach die Maus-gestützte Selektion gezielter von dieser Startfarbe aus durchführen kann. Der hexadezimale Code ist derjenige, der bei „**Speichern**“ in die Settings.ini-Datei eingetragen wird. Die Anwendung der Buttons „**Speichern**“, „**Anwenden**“ und „**Beenden**“ erfolgt analog zum Fontname-Dialog.

## 3.8 Grafik-Fenster

Grafische Darstellungen werden in einem Fenster mit drei Reitern für verschiedene Anwendungen dargestellt:

* Monte Carlo-Simulation (Reiter MC),
* eine noch nicht freigegebene Anwendung (Reiter MCMC) und
* für Messdaten und Standardunsicherheiten sowie die Fitkurve aus der linearen Entfaltung (Reiter LFit).

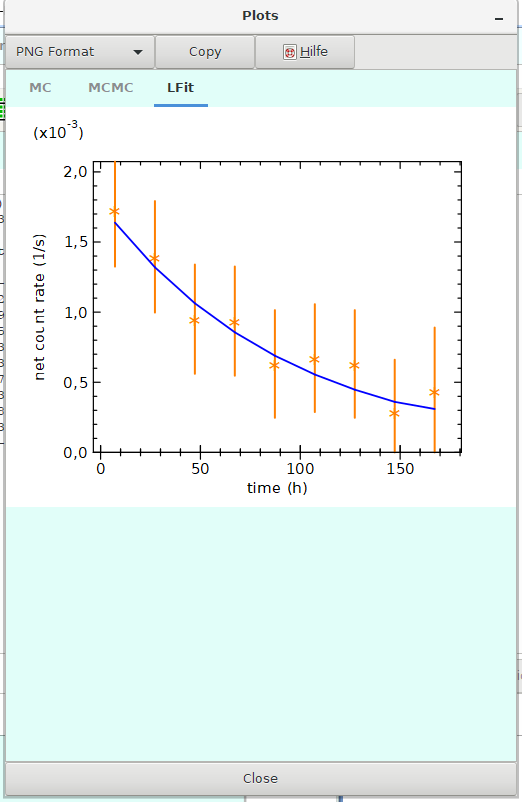
Die dazugehörigen Grafiken werden standardmäßig als PNG-Dateien abgelegt:

* MCplotfile.png
* -
* CurvePlot.png

Mit dem Button „Copy“ kann die Grafik auch in dem zuvor mit dem in der Combobox selektierten Grafikformat als Datei ausgegeben werden, mit dann explizit einzugebendem Dateinamen.

Mit dem Button „Close“ am unteren Ende des Fensters kann dieses geschlossen werden.

Die Grafik unter dem Reiter LFit kann mit dem Icon  in der Toolbar dargestellt werden.



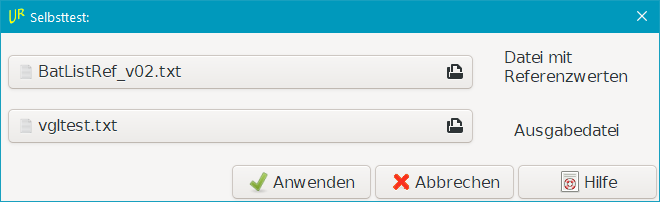
## 3.9 Programm-Test

Es besteht der Bedarf, nach Installation neuer Programm-Versionen selbst testen zu können, ob von den in UncertRadio verwendeten analytischen Verfahren weiterhin Auswertungsergebnisse zu erwarten sind, die mit den dokumentierten Ergebnissen (Referenzwerte) übereinstimmen. Ab der Version 2.3.03 ist dies in folgender Weise möglich. Eine schon länger vorhandene (interne) Testroutine zur Abarbeitung vieler Projekte im Pseudo-Batch-Betrieb diente bisher dem Zweck, für die Vielzahl der in Abschnitt 3.3 aufgeführten Projekte zu testen, ob die erhaltenen Ergebnisse mit früher dokumentierten Werten dafür übereinstimmen. Diese – bisher nicht verwendbare – Routine wurde so erweitert, dass sie vom Benutzer unter dem **Menü Optionen – QC-Batch-Test** selbst aufgerufen werden kann. Damit werden die analytischen Verfahren an zurzeit 106 Projekten getestet, was insgesamt ca. 90 Sekunden Rechenzeit in Anspruch nimmt.

Hierzu wird bei der Installation eine Datei mit dem Namen

BatListRef-v02.txt

mit den Referenzwerten bereitgestellt. Mit dem **Menü Optionen – QC-Batch-Test** wird ein Dialog aufgerufen, in dem die obige Eingabe-Datei mit den Referenzwerten und eine Datei für die Ausgabe des projekt-weisen Vergleichs der gerade berechneten Werte und der Referenzwerte festgelegt wird.



Nach einer Laufzeit von bis zu 70 Sekunden ist der Test beendet und man bekommt eine Information darüber, bei wieviel Projekten Abweichungen gefunden wurden. Die Details zu solchen Abweichungen können in der im obigen Dialog eingestellten Ausgabedatei eingesehen werden. Darin werden allerdings nur die Informationen zu den Projekten aufgeführt, für die Abweichungen gefunden wurden.

# Bedienung der einzelnen TABs

## 4.1 TAB "Verfahren"

Hier kann für das Messproblem in Kürze das "Prinzip des Verfahrens" textlich beschrieben und z.B. auf Besonderheiten darin hingewiesen werden. Literatur dazu kann hier ebenfalls aufgeführt werden. Solange man sich noch in der Phase der Aufstellung von Gleichungen für ein neues Projekt befindet, kann dieser Punkt auch übersprungen werden.

Wenn man die Eintragungen beendet hat, wechselt man per Mausklick zum TAB "Gleichungen".

## 4.2 TAB "Gleichungen"

1) Man beginnt mit dem zunächst bedienbaren ASCII-Textfenster ganz oben, in das zeilenweise die Gleichungen eingetragen werden können:

[**Textfenster für die Gleichungen**](#URH_GLEICHUNGEN_DE)

2) Wenn alle Gleichungen eingegeben wurden, wird mit der Maus der **Button "Laden Symbole(1) aus Gleichungen"** betätigt, der die Formelsymbole aus den Gleichungen extrahiert, intern einen Syntaxcheck der Gleichungen mit Hilfe des „Function parsers“ durchführt und dann für Ergänzungen von *Einheit* und *Bedeutung* in eine Tabelle lädt, die der Benutzer als nächstes bearbeiten kann.

Nach dem Bearbeiten der Gleichungen kann es hinsichtlich der (vorher schon vorhandenen) Liste von Symbolen zu folgenden Fällen kommen:

a. Es sind neue Symbole hinzugekommen. Die entsprechenden Zeilen in der Symbole-Tabelle werden grün hinterlegt.

b. Bisher vorhandene Symbole werden nicht mehr benötigt (**überflüssige Symbole**). Die entsprechenden Zeilen in der Symbole-Tabelle werden an das Ende der Tabelle verlegt und dort gelb hinterlegt.

Die gelb hinterlegten Zeilen müssen zunächst mit der Maus markiert werden; danach werden diese Zeilen mit dem Toolbar-Icon **„Zeilen löschen“ ** entfernt. Bestimmte Symbole, die nicht direkt in den Gleichungen vorkommen, aber noch benötigt werden (z.B. Messdauern), dürfen nicht gelöscht werden.

Zur weiteren Bearbeitung: siehe

[**Symbolliste bearbeiten**](#URH_SYMBOLLISTE_DE)

In der Symbolliste können solche Eingangsgrößen, für die Mittelwert und Standardabweichung aus einem noch einzugebenden Datensatz berechnet werden sollen, in der Spalte „Typ“ mit „m“ anstelle von „a“ (abhängige Größe) oder „u“ (unabhängige Größe) gekennzeichnet. Des Weiteren wird eine mit einer Gleichung zu berechnende Größe, die nur als Parameter ohne Unsicherheit verwendet werden soll, mit einem „p“ gekennzeichnet.

3) Wenn die Symbolliste vervollständigt worden ist (in der Testphase auch ohne Einheiten und Bedeutungen), wird mit der Maus der **Button "Laden Symbole(2) aus der ergänzten Symbol-Tabelle"** betätigt und die ab jetzt für die nachfolgenden Berechnungen gültige komplette Symbolliste geladen.

4) Inzwischen sind die beiden **Auswahl-Felder "Netto-Zählrate" und "Brutto-Zählrate"** bedienbar geworden. Hier sind aus der Symbolliste jeweils die Symbole für die Netto-Zählrate und die Brutto-Zählrate auszuwählen. Im weiteren Ablauf des Programms ist **diese Zuordnung ausschließlich von Bedeutung für die Berechnung der Erkennungs- und Nachweisgrenzen**. Hat man diese Auswahl getroffen, wird diese mit Klicken des Buttons "Netto- und Brutto-Zählrate übernehmen" bestätigt und übernommen. Wenn es in dem Projekt mehr als eine Ergebnisgröße gibt, müssen genau die zur aktuell gewählten Ergebnisgröße gehörenden Symbole für Netto- und Brutto-Zählrate ausgewählt werden.

Anmerkung: Die Identifikation der Netto-Zählrate (genauer: der [**verfahrensbezogenen Netto-Zählrate**](#URH_VERFAHREN_DE)**)** wird benötigt, um bei der Nachweisgrenzen-Iteration die Umkehrung der Gleichungen, vom variierten Ergebniswert (z.B. in Bq/kg Feuchtmasse) zurück zur Netto-Zählrate (in ), rechnerisch vollziehen zu können. Die Brutto-Zählrate muss identifiziert werden, da der Benutzer für diese die Unsicherheits-Funktion definieren muss; für die Ermittlung von Erkennungs- und Nachweisgrenze wird intern der Wert der Bruttozählrate variiert, wobei diese Unsicherheits-Funktion dazu dient, für jeden Iterationsschritt den geänderten Wert der Unsicherheit bestimmen zu können.

**Ausnahmen:**

a) bei der Analyse einer Abklingkurve mit Hilfe des linearen Least squares-Verfahrens (Aufruf von ***Linfit***) **entfällt die Angabe der Brutto-Zählrate**;

b) bei der Mittelung von Aktivitäten mehrerer Gammalinien eines Radionuklids (Aufruf von ***Gamspk1*)** **entfällt die Angabe der Brutto-Zählrate**; die **Netto-Zählrate ist in diesem Falle Platzhalter für die Aktivität (Bq) des Messpräparats,** dessen Symbol ausgewählt werden muss.

c) bei der Summation oder Mittelung mehrerer Aliquot-Messungen (Aufruf von **SumEval**)

Bei diesen Ausnahmen ist also für das Symbol der Netto-Zählrate (bzw. Probenaktivität bei Gamspk1) dasjenige auszuwählen, das in den Gleichungen durch den Aufruf von *Linfit* bzw. *Gamspk1* definiert wird.

5) Nach Schritt 4 ist das TAB "Gleichungen" zunächst erfolgreich abgearbeitet, **das nächste TAB "Werte, Unsicherheiten" wird bedienbar** und man wählt dieses per Mausklick aus.

## 4.3 TAB "Werte, Unsicherheiten"

Der wesentliche Teil dieses TABs besteht aus einer 10-spaltigen Tabelle für die Aufnahme der Messwerte und Unsicherheiten der einzelnen unabhängigen Messgrößen (Eingangsgrößen). Sie besteht aus den Spalten:

|  |  |  |  |
| --- | --- | --- | --- |
| 1: | **Symbol** | Textfeld | Formelsymbol |
| 2: | **Typ** | Textfeld | Variablentyp: **a**   für abhängig, **u**    für unabhängig,  **m**   für „Mittelwertgröße“, **p**   für zu berechnenden Parameter ohne Unsicherheit |
| 3: | **Einheit** | Textfeld | Eingabe der Einheit |
|  |  |  |  |
| 4: | **Messwert** | Zahlenfeld | Wert der Messgröße |
| 5: | **Verteilung** | Auswahlfeld | Festlegung der Verteilung der Messgröße (Normal / Rechteck / Dreieck / u.a.)  Es kann die [**(N+x)-Regel für sehr kleine Impulsanzahlen**](#URH_NP1REGEL_DE) selektiert werden, womit die damit verknüpften Zählraten Gamma-verteilt werden |
| 6: | **StdAbw-Formel** | Textfeld | Formel für die Standardabweichung der Messgröße  (der interne Erweiterungsfaktor ist immer 1) ; als Dezimalpunkt ist immer der „.“ zu verwenden! |
| 7: | **StdAbw-Wert** | Zahlenfeld | Wert der Unsicherheit für Normal-Verteilung;  Bei Selektion der (N+x)-Regel ist in dieser Zelle **nichts** einzu­tragen!  (der interne Erweiterungsfaktor ist immer 1) |
| 8: | **Halbbreite** | Zahlenfeld | Halbbreite von Rechteck- bzw. Dreieckverteilung  (der interne Erweiterungsfaktor ist immer 1) |
| 9: | **abs./rel.** | Auswahlfeld | Festlegung, ob der Unsicherheitswert in Sp. 6, 7 oder 8 als *absoluter* oder *relativer* Wert zu nehmen ist. |
| 10: | **abs. Std.Uns.** | Zahlenfeld | vom Programm aus den Werten der Spalten 6, 7, 8 und 9 berechneter absoluter Wert der (kombinierten) Standardunsicherheit  Achtung: Ein vom Benutzer hier eingegebener Wert wird immer vom Programm überschrieben! |

Die Spalten 1 bis 3 wurden aus der schon vorhandenen Symbolliste übernommen und können hier nicht mehr geändert werden.

Zur Unterstützung des Benutzers sind in der Tabelle **diejenigen Zellen mit rotem Hintergrund belegt, in die vom Benutzer keine Eintragungen zu machen sind**: Zellen der abhängigen Größen. Es sollen nur Zellen mit weißem Hintergrund vom Benutzer ausgefüllt werden. Die Zeilen für die abhängigen Größen stehen also im oberen Teil der Tabelle, **in Zeile 1 findet man die Information zur Ergebnisgröße, bei drei Ergebnisgrößen in den Zeilen 1-3.**

Besonderheit für nicht normalverteilte Impulsanzahlen und Zählraten: [siehe](#URH_Behandlung_Impulsanzahlen).

Für eine Datensatz-basierte Größe werden Mittelwert und Unsicherheit intern aus dem dazugehörigen Datensatz berechnet und mit dem Button „Berechnung der Unsicherheiten“ in die Zellen der Spalten „Messwert“ und „StdAbw-Wert“ eintragen, daher haben diese Zellen ebenfalls einen roten Hintergrund.

Die „weißen“ Zellen in den Spalten 4 bis 9 sind vom Benutzer auszufüllen. **In die Spalte „Messwert“ sind auf jeden Fall Zahlenwerte einzutragen**, während dies in den Spalten 6 bis 9 nicht für alle Messgrößen erforderlich ist. Auf die Unsicherheiten von Messzeiten z.B. wird man meistens verzichten, die entsprechenden Zellen in den Sp. 6 bis 9 lässt man dann leer.

Die Eintragung von Unsicherheitswerten kann in nur jeweils einer der Spalten 6, 7 oder 8 erfolgen. Das Auswahlfeld „abs./rel.“ hingegen muss bedient werden, falls in den Spalten 6 bis 8 ein Eintrag vorliegt.

Für Zählraten bzw. Impulsanzahlen kann man die **Standardabweichung als Formel** in Sp. 6 („StdAbw-Formel“) eingeben - ohne vorangestelltes Gleichheitszeichen. Für die als **normal-verteilt angenommenen Messgrößen** - in der Regel auch für Zählraten bzw. Impulsanzahlen möglich - erfolgt der Eintrag des Unsicherheitswertes in Spalte 7 („StdAbw-Wert“). Im Falle von **Rechteck- bzw. Dreiecks-verteilten Messgrößen** erfolgt die Eintragung des dann vorliegenden Unsicherheitswertes als Halbbreite in Spalte 8. Letztere wird programmintern nach GUM-Regeln in eine Standardabweichung (Normalverteilung) umgerechnet und in die Spalte 10 („abs. Std.Uns.“) übertragen.

Eine Zelle in der Spalte „StdAbw-Formel“ ist **grün markiert**. Hier muss die **Standardabweichung der Bruttozählrate als Formel** eingetragen werden. Diese Formel wird bei der späteren numerisch-iterativen Berechnung der Erkennungs- und Nachweisgrenzen als „**Unsicherheits-Funktion“ für die Variation der Standardabweichung der Brutto- (und damit implizit auch der (verfahrensbezogenen) Nettozählrate) als Funktion eines beliebigen Wertes der Bruttozählrate** verwendet. Sie hat dagegen keine Bedeutung für die Berechnung der kombinierten Unsicherheit der Ergebnisgröße.

**Mögliche Formeln für Standardabweichung der Brutto-Zählrate Rb** (nicht vollständig):

sqrt(Rb/tm) zählende Messgeräte (Messzeit tm), einmalige Messung

sqrt(Rb/tm/n) zählende Messgeräte (Messzeit tm), n-malige Messung

sqrt(Rb/2/tau) Ratemetermessungen (Dämpfungszeitkonstante tau)

Wird zusätzlich zur Bruttozählrate Rb auch die **Bruttoimpulsanzahl Nb** verwendet, z.B. als Gleichung Rb=Nb/tm, ist zu beachten, dass das Programm **bei der Variation von Rb auch die Impulsanzahl Nb entsprechend ändert**. Aufgrund dieser Gleichung, Rb=Nb/tm, kann Uncertradio die zu Rb gehörende Impulsanzahl identifizieren. Dadurch wird sichergestellt, dass das Programm beim Abarbeiten der Gleichungen von unten nach oben (Nb ist dann eine unabhängige Größe und Rb abhängig) ein zunächst geändertes Rb und seine Unsicherheit nicht von dem in der Hierarchie nachfolgenden Nb und dessen Unsicherheit wieder überschrieben würde, falls letzterer nicht geändert worden wäre.

Wenn der Wert der **Bruttozählrate Rb als Mittelwert mehrerer Einzelwerte** behandelt wird, muss eine lineare Interpolation zwischen dessen Varianz bei der Messung (u\_Rbm^2) und der für die Berechnung der Erkennungsgrenze anzusetzende Varianz (u\_R0^2) erfolgen. Die Formel für die Standardabweichung der Bruttozählrate lautet z.B. wie folgt:

sqrt( u\_R0^2 + (u\_Rbm^2 – u\_R0^2)\*(Rb - R0) / (Rbm – R0) ) (1)

Hierin sind Rb und Rbm der zu variierende und der gemessene Wert der Bruttozählrate. Wenn Rb den oberen Wert Rbm annimmt, kommt als Formel gerade sqrt(u\_Rbm^2) heraus, während für den unteren Wert Rb=R0 gerade sqrt(u\_R0^2) herauskommt. Rbm, u\_Rbm und u\_R0 sind hierfür in der Liste der unabhängigen Symbole zu ergänzen; ihnen wird keine Unsicherheit zugeordnet.

**Hinweis:**

Ab der Programmversion 2.3.05 2020/01 ist es nicht mehr nötig, dass die Gleichung (1), oder eine ähnliche Formel, vom Anwender in UncertRadio eingegeben werden muss. Auch die Eingabe zusätzlicher Hilfsgrößen entfällt dann. Das Kapitel 6.9 informiert darüber, wie generell die Eingabe von Datensätzen für Mittelwerte erfolgt. Darauf aufbauend wird im Kapitel 6.12 beschrieben, wie die zum Aufbau der (erweiterten) Gleichung (1) nötigen Berechnungen im Programm erfolgen.

Der Typ der Gleichung (1) für die Unsicherheit des Bruttozählraten-Mittelwerts hängt davon ab, ob die einem Mittelwert zugeordneten Einzelwerte solchen zufälligen Einflüssen unterliegen, die man als **nicht bekannt** oder als **bekannt** einstuft. Für diese beiden Fälle werden unterschiedliche Arten der Gleichung (1) verwendet. Dies wird im Kapitel 6.12ebenfalls erläutert. Dort werden auch Beispiel-Projekte genannt.

Es wird Anwendungen geben, in denen die Formel für die Standardabweichung der Bruttozählrate länger wird als der sichtbare Teil der entsprechenden Tabellen-Zelle. Um die Formel besser einzusehen zu können, kann man diese per „Kopieren und Einfügen“ („Copy and Paste“, im Kontext-Menü der rechten Maustaste zu finden) zwischen dieser Zelle sowie in der deutlich „längeren“ Textzelle **„Hilfszelle zum Formel-Entwurf“** oberhalb der Tabelle hin- und her kopieren. In der größeren Hilfszelle kann die Formel besser editiert werden.

Unterhalb der Unsicherheiten-Tabelle ist eine kleinere **Tabelle für die Eingabe von Kovarianzen** vorgesehen. Sie besteht aus den Spalten:

|  |  |  |
| --- | --- | --- |
| 1: | **Symbol A** | Symbole-Auswahlfeld |
| 2: | **Symbol B** | Symbole-Auswahlfeld |
| 3: | **Typ** | Auswahlfeld zur Eingabe einer Kovarianz oder eines Korrelationskoeffizienten |
| 4: | **Formel** | Textfeld zur Eingabe der Kovarianz als Formel als Funktion der bereits definierten Symbole. |
| 5: | **(oder) Wert** | Zahlenfeld zur Eingabe eines Wertes von Kovarianz / Korrelationskoeffizient |

In den Spalten 1 und 2 wählt man die Messgrößen-Symbole für die gewünschte Kovarianz aus. Hat man in Spalte 3 „Kovarianz“ als Typ ausgewählt, kann in der Spalte „Formel“ eine Formel dafür eingegeben werden. Ansonsten kann der Zahlenwert der Kovarianz oder des Korrelationskoeffizienten auch direkt in die 5. Spalte eingegeben werden.

Zusammenhang zwischen Korrelationskoeffizient ***r*** und Kovarianz ***cov***:

Nach Vervollständigung der Benutzereingaben in der Unsicherheiten-Tabelle und in der kleineren Kovarianz-Tabelle wird durch einen Mausklick auf den **Button „Berechnung der Unsicherheiten“** folgendes berechnet:

 Anmerkung: Alle Berechnungen in diesem TAB und in den darauffolgenden beziehen sich nur auf eine - die gegenwärtig eingestellte aktuelle – Ergebnisgröße, falls in dem Projekt mehr als eine definiert wurden. Im Menüpunkt „Bearbeiten – Wähle Ergebnisgröße“ kann eine andere Ergebnisgröße selektiert werden;

 **Unsicherheiten-Tabelle**: Messwerte der abhängigen Messgrößen (in roten Feldern) berechnen und in die entsprechenden Spalte eintragen; Formeln für Standardabweichungen in die Spalte „StdAbw-Wert“ umsetzen; für unabhängige Messgrößen die Standardunsicherheiten berechnen (Spalte „abs. Std.Uns.“);

 **Kovarianz-Tabelle:** Umsetzen der Kovarianz-Formeln in Werte der Kovarianz (Spalte „(oder) Wert“);

 **Unsicherheiten-Tabelle:** jetzt sind alle Varianzen/Kovarianzen für die komplette Fortpflanzung der Unsicherheiten bekannt; Berechnung der Standardunsicherheiten der abhängigen Größen (a) unterhalb der Ergebnisgröße (per definitionem nur Hilfsgrößen) einschließlich derjenigen der Ergebnisgröße in Zeile 1 der Tabelle; Die kombinierten Standardunsicherheiten der abhängigen Größen (rote Felder) werden dabei ausschließlich aus den Unsicherheiten/Kovarianzen der unabhängigen Größen (weiße Felder) berechnet.

**Erst nach Abschluss dieser Rechnungen wird das TAB „Unsicherheiten-Budget“ freigegeben.** Bei sehr komplexen Messproblemen und einem langsameren PC mag es sein, dass diese Berechnungen einen gewissen Moment dauern; dies wird in dem Statusbar-Segment am unteren rechten Rand des UncertRadio-Fensters mit dem Eintrag „Rechnet..“ angezeigt, aus dem nach Abschluss der Rechnung „Fertig!“ wird.

**Hinweis:** [**Auswirkung der Änderung von Werten im Menü Optionen**](#URH_AEND_OPTION_DE)

**Weiteres zur** [**Bearbeitung von Tabellen**](#URH_TABTRICKS_DE)

## 4.4 TAB "Unsicherheiten-Budget"

Unter diesem Tab - per Mausklick angezeigt - wird das sogenannte Unsicherheiten-Budget als eine Tabelle dargestellt. Es wird angezeigt, auf welche Ergebnisgröße sich das Unsicherheiten-Budget bezieht.

Die **Tabelle „Unsicherheiten-Budget“** enthält wiederum die drei Spalten „Symbol“, „Typ“ und „Einheit“.

Unter den Spalten „Werte“ und „Std.-Unsicherheit“ werden die eingegebenen Werte und Standardunsicherheiten der unabhängigen Messgrößen sowie die für die abhängigen Größen (Hilfsgrößen und Ergebnisgröße) berechneten Werte und Standardunsicherheiten wiedergegeben.

Die Spalte **„Sensitivitäts-Koeffizient“** enthält für die unabhängigen Größen ihre partiellen Ableitungen der Ergebnisfunktion (der über die Gleichungen bestimmten Funktion, mit der der Wert der Ergebnisgröße ***y*** berechnet wird).

Aus den zeilenweise berechneten **Produkten Unsicherheit *x* Sensitivitätskoeffizient**, deren Werte in der folgenden Spalte in zwei Varianten dargestellt werden können, ergibt sich formal das Unsicherheitsbudget.

Mit der Bezeichnung **„relat. Beitrag(%)“** in dieser Spalte werden für jede unabhängige Messgröße - soweit für sie Unsicherheiten eingegeben wurden - die prozentualen Anteile ihrer Varianz an der kombinierten Varianz der Ergebnisgröße dargestellt. An dieser Spalte lässt es sich leicht ablesen, welche der Größen am meisten zur kombinierten Standardunsicherheit der Ergebnisgröße beitragen. Die Angabe von 100% in dieser Spalte für die Ergebnisgröße ist lediglich die Kontrollsumme der einzelnen prozentualen Beiträge.

Mit Hilfe des **Buttons „Budget-Typ wechseln“** kann auf die Anzeige absoluter Unsicherheitsbeiträge, angegeben in der Einheit der Ergebnisgröße, umgeschaltet werden.

**Die Definition** des vorstehend erläuterten **„relat. Beitrag(%)“** zum Unsicherheiten-Budget **hat durch eine** **neuere Publikation von Kessel, Kacker und Berglund** (2006) mit dem Titel „Coefficient of contribution to the combined standard uncertainty“ **eine weitere plausible Deutung bekommen**:

Dieser relative Beitrag einer Eingangsgröße , dividiert durch 100, ist im Falle nicht-korrelierter Eingangsgrößen identisch mit dem Quadrat des Korrelationskoeffizienten zwischen dieser Eingangsvariable und der Ergebnisgröße ! Dieser Größe wird in der Publikation die Bezeichnung „**coefficient of contribution**“ (in Deutsch etwa: Beitrags-Koeffizient) gegeben. Ihr wird das Symbol zugewiesen.

Die verallgemeinerte Definition des „Beitrags-Koeffizienten“ lautet jetzt:

(1)

Da für nicht-korrelierte Eingangs-Größen gilt:

(2)

folgt damit aus Gleichung (1) die Gleichung (3), was bisher schon, d.h. für nicht-korrelierte Eingangsgrößen, als stets positiver „relativer Beitrag zur Varianz der Ergebnisgröße“ bekannt war:

(3)

Bei Vorliegen von Korrelationen zwischen den Eingangsgrößen werden diese in Gleichung (1) in dem nach folgender Gleichung zu berechnenden Faktor ,

, (4)

verwendet, so dass jetzt auch negative Werte annehmen kann.

**Hinweise zu Auswirkungen von Kovarianzen:**

Wenn Kovarianzen bei der Berechnung der Unsicherheiten berücksichtigt wurden, treten auch negative Werte in der Spalte „relat. Beitrag(%)“ auf; dies ist also kein Fehler des Programms!

Nach der oben angeführten Arbeit von Kessel et al. werden Korrelationen (Kovarianzen) zwischen den Eingangsgrößen nach obiger Gleichung (1) mit Gleichung (4) bei der Berechnung des „coefficient of contribution“ berücksichtigt und in der Spalte „relat. Beitrag(%)“ im Budget dargestellt. Gleichung (3) gilt dann nicht mehr.

Mit UncertRadio ab Version 0.05 (2007/11) wird dieses Verfahren nach Kessel et al. bereits verwendet, d.h. die Werte, die UncertRadio in der Spalte „relat. Beitrag(%)“ des Budgets ausgibt, entsprechen bereits dieser neuen Definition. Dies kann mit Hilfe der **Beispiel-Projekte Kessel-2a-2006.txp und Kessel-2b-2006.txp** demonstriert werden**,** die aus zwei Rechenbeispielen dieser Publikation editiert wurden.

## 4.5 TAB “Resultate“

Unter diesem Tab, per Mausklick angezeigt, werden für die Ergebnisgröße das gesamte Messergebnis einschließlich einer Reihe von weiteren Kenngrößen sowie die Werte der Erkennungs- und der Nachweisgrenze dargestellt. Es wird angezeigt, auf welche Ergebnisgröße sich die Resultate beziehen.

Dies sind im Einzelnen:

***das Messergebnis:***

* der Wert der Ergebnisgröße (die aktuell gewählte Ergebnisgröße, falls mehrere vorliegen)
* die erweiterte Unsicherheit, in der Einheit der Ergebnisgröße
* die relative erweiterte Unsicherheit (in %)
* der Erweiterungsfaktor (kann über das **Menü Optionen** geändert werden)

[***beste Schätzwerte nach Bayes und Vertrauensgrenzen***](#URH_BestBayes_DE) ***(vgl. hierzu DIN EN ISO 11929-1:2021):***

* der Wert der Ergebnisgröße
* die erweiterte Unsicherheit
* der Wert der unteren Vertrauensgrenze
* der Wert der oberen Vertrauensgrenze
* Wahrscheinlichkeit zum Vertrauensbereich:

Zur Darstellung der Grenzen des Überdeckungsintervalls im TAB “Resultate” kann mit den Toggle-Buttons „min. Coverage-Intervall“ zwischen der Anzeige der probabilistisch symmetrischen und der kürzesten Überdeckungsintervalle gewechselt werden, auch im Falle der MC-Simulation.

***Erkennungs- und Nachweisgrenze:***

* der Wert der Erkennungsgrenze mit der Anzahl der dafür benötigten Iterationen
* der Wert der Nachweisgrenze mit der Anzahl der dafür benötigten Iterationen
* die dafür verwendeten Quantile der Normalverteilung k\_alpha = und k\_beta =   
  zu den Fehlern 1. und 2. Art.

***WLS, PLSQ, PMLE bzw. WTLS: Standardunsicherheiten des der Ergebnisgröße entsprechenden Fitparameters bei Analyse einer Abklingkurve:***

* die Unsicherheit, die sich aus der Least Squares-Analyse ergibt; diese wird **NICHT** mit multipliziert, falls der Wert des reduzierten Chi-Quadrat größer als 1 ist; diese Variante der Unsicherheit der Nettozählrate wird für die Bestimmung der Unsicherheit der Ergebnisgröße verwendet;
* der Wert der Unsicherheit, der sich aus der Fortpflanzung der Unsicherheiten der Argumente der LinFit-Funktion (vor allem die Nulleffektzählrate, ggf. mit Blindwertanteil) und der Unsicherheiten der Bruttozählraten der Abklingkurve ergibt
* der Wert des reduzierten Chi-Quadrat

[***Monte Carlo Simulation***](#URH_MC_SIM_DE)***:***

* Eingabe der gewünschten Anzahl *N* von simulierten Berechnungen für die Ergebnisgröße (ein Run)
* Festlegung der Anzahl *r* der Runs
* Optional Wahl der Vertrauensgrenzen kürzester Länge (kleinstes **c**overage **i**nterval)

Die MC Simulation wird mit dem Button “Start” gestartet. Während der iterativen Ermittlung der Verteilung der Nachweisgrenze wird die Nummer der Iteration angezeigt.

Aus der *r*-fachen Wiederholung (Runs) werden jeweils ein Mittelwert und seine relative Streustandardabweichung (in %) ermittelt, und zwar für:

*Beste Schätzwerte nach Bayes:*

* die Ergebnisgröße
* die erweiterte Unsicherheit
* die relative erweiterten Unsicherheit (%)
* die untere Vertrauensgrenze
* die obere Vertrauensgrenze

*und:*

* die Erkennungsgrenze
* die Nachweisgrenze

**In einem Extradialog werden die drei erzeugten Monte-Carlo-Verteilungen grafisch dargestellt.**

Mit Hilfe des **Buttons "Save results"** werden die in diesem Dialog enthaltenen Werte, einschließlich derjenigen aus einer erfolgten MC-Simulation, mit Projektname, Datum und Uhrzeit in eine CSV-Datei geschrieben: UR‑Saved-Results.csv. Diese Datei wird entweder neu angelegt, oder die Datensätze werden hinten angehängt. Die Spalten sind ähnlich wie bei der Datei AutoReport-Result.csv aufgebaut; es werden noch Spalten für die LINFIT-Fitparameter, jeweils für Ergebnisgröße, Erkennungsgrenze und Nachweisgrenze angehängt.

In dem **Menü Optionen** können die beiden Werte der Quantile der Normalverteilung eingestellt werden, die den sog. Fehlern 1. und 2. Art, α und β, entsprechen. Dort kann auch das Verfahren für die Nachweisgrenzen-Berechnung ausgewählt werden - Standardeinstellung ist „ISO 11929:2010, iterativ“.

[**Auswirkung der Änderung von Werten im Menü Optionen**](#URH_AEND_OPTION_DE).

# UR automatisiert betreiben

## Benutzung von Kommandozeilen-Argumenten

Es ist möglich das Programm UncertRadio mit der Beigabe von sogenannten Kommandozeilen-Argumenten zu starten. Die Kommandozeile-Argumente, hier mit Arg\_n abgekürzt, sind Strings (ohne Leerzeichen!) und dienen dazu, dem Programm beim Start sowohl spezielle Dateinamen oder Schlüsselworte zur Steuerung des Ablaufs des Programms zu übergeben. Mit einem derart zusammengesetzten Kommando-String kann UR unter Windows-Start wie folgt aufgerufen werden:

>Start D:\Uncertradio\Uncertradio.exe Arg\_1 Arg\_2 Arg\_3

Zur Vorbereitung einer automatisierbaren Auswertung von mehreren UR-Projekten, hintereinander ausgeführt, sind in einem ersten Schritt folgende drei Kommandozeile-Argumente eingerichtet worden, die das Programm interpretiert:

>Start D:\Uncertradio\Uncertradio.exe AUTO Inputdatei.txp Sample\_ID

Hierbei initiiert das erste Argument AUTO (großgeschrieben) den Batchmode; das 2. Argument ist der Name der automatisch auszuwertenden UR-Projekt-Datei; das dritte Argument ist eine Proben-Identifikationsnummer (oder Versuchsnummer) und soll nur dabei helfen, solche Auswerteergebnisse, die zwar mit verschiedenen Probenummern, aber derselben UR-Projektdatei gerechnet wurden, voneinander zu unterscheiden.

Ab der Version 2.04.00 ist ein weiterer Batchmode integriert, mit dem ein vorgegebenes Projekt mehrfach seriell für spezifizierte Eingangswerte ausgewertet kann. Die Auswertung entspricht der in Kapitel 5.6 beschriebenen Auswertung. Das obige Argument AUTO wird hierbei durch BATSER ersetzt, gefolgt von der Projektdatei und der CSV-Datei mit den Datensätzen der darin spezifizierten Eingangswerte. Die Befehlszeile lautet in Anlehnung an Kapitel 5.6:

>Start D:\UR2\Uncertradio.exe BATSER J-ALUFT-Sr89-Sr-90\_DE.txp J\_Aluft\_serial\_DE.csv

**Hinweis**: **Um Konflikte mit Antivirussoftware zu vermeiden wird die Datei UR2\_start\_xls.bat, mit der UncertRadio bisher von Excel aus gestartet wurde, nicht mehr verwendet. Stattdessen ruft Excel-VBA das UncertRadio direkt auf, auch das temporäre Modifizieren der Windows-PATH-Variable erfolgt direkt**.

## Automatisierter Betrieb mit einer Excelanwendung

Die auf diese Weise automatisierbare Anwendung von UR macht erst dann Sinn, wenn diese UR-Aufrufe in ein anderes Programm eingebettet werden, welches auch den Ablauf der Auswertungen und welcher UR-Projektdateien steuert. Das kann z.B. ein eigenes Auswerteprogramm für ein bestimmtes Messverfahren sein, bei dem das Ziel darin besteht, eine feststehende UR-Projektdatei für jede separate Auswertung zu verwenden, allerdings erst nachdem ein vom Anwender zu erstellendes Programteil die von Versuch zu Versuch sich ändernden Eingangs-Daten in eine geeignete Arbeitskopie der UR-Projektdatei eingetragen hat.

Basierend auf dem neuen [CSV-Format eines UR-Projekts](#URH_CSV_FORMAT_DE) wurde mit VB für Excel ein einfacherer Vorschlag realisiert, der den automatisierten Betrieb von UncertRadio demonstriert. Die dafür erstellte Datei heißt:

UR2\_SingleAutoRun\_V11.xlsm.

In dieser Exceldatei werden vier Tabellen für UR reserviert, voreingestellt sind darin Tabelle4, Tabelle5, Tabelle6 und Tabelle7:

Tabelle4 : Ziel für den Import eines UR-Projekts im CSV Format

Tabelle5 : Ziel für den Import der von UR erzeugten Resultate

Tabelle6 : Liste von UR-Projekten für den Batch-artigen Betrieb von UR

Tabelle7 : enthält Buttons zum Ausführen verschiedener Makros

Die Datei enthält ein Modul mit dem Namen Modul\_Auto\_Single\_UR. Am Beginn des Moduls erfolgt eine Abfrage-Anweisung, die sicherstellt, dass sowohl 32 bit- als auch 64 bit-Excel verwendet werden kann. Danach wird die erste der zuvor genannten vier Tabellen definiert:

Public Const FirstTableNum As Integer = 4

und kann darin vom Anwender ggf. anders eingestellt werden.

Die anderen Tabellen, außer den vier genannten, können frei genutzt werden.

Das Modul Modul\_Auto\_Single\_UR beinhaltet mehrere Routinen (Makros), deren Funktion kurz erläutert wird.

|  |  |
| --- | --- |
| Init\_pathnames | In der kleinen Routine am Beginn des Moduls legt man fest, ob UR und die Exceldatei im selben Pfad oder in verschiedenen Pfaden stehen.  ***Seit Version 2.4.22 werden die wesentlichen Pfad-bezogenen Informationen direkt aus der Datei* UR2\_cfg.dat *entnommen:***  Sub Init\_pathnames()  ***' Further settings since version 2.4.22:***  ChDir (UR\_path) ‘ to prevent from Excel’s home!  ‘Read the filename for UR\_path, Excel\_path and UR\_output\_path from UR2’s configuraton file UR2\_cfg.dat:  **Fnum = FreeFile()**  **Open Trim(UR\_path) + "UR2\_cfg.dat" For Input As Fnum**  **For k = 1 To 100**  **If (EOF(Fnum)) Then Exit For**  **Input #Fnum, text**  **i1 = InStr(1, text, UCase("UR\_AUTO\_outputpath="),**  **vbTextCompare)**  **i2 = InStr(1, text, "=", vbTextCompare)**  **If (i2 > 0 And i1 > 0) Then**  **UR\_AUTO\_output\_path = Mid(text, i2 + 1, 300)**  **Exit For**  **End If**  **.**  **. repeat the same for the second and third information.**  **In the case of “UR\_path=”, add the following variable:**  **URGTK\_path = UR\_path & "GTKUser64\bin;" ‘ 26.6.2023**  **Next k**  **Close #Fnum**    **‘** Build the string for setting the environment variable  path:  **pathX = EnvironGetItem("path", "User")**  **Call SetEnvironmentVariableA("path", UR\_path & \_**  **";" & URGTK\_path & “;” & pathX)**    **Debug.Print "path=", EnvironGetItem("path", "User")**  **‘ Sprachabhängigkeiten:**  ‘Set Decimalpoint and ListSeparator characters :  sDecimalPoint = GetDecimalSeparator()  sListSeparator = \_  Application.International(xlListSeparator)  ‘Set language:  **Win\_langg = "EN"**  **Select Case Application.International(**  **XlApplicationInternational.xlCountryCode)**  **Case 1: Win\_langg = "EN"**  **Case 33: Win\_langg = "FR"**  **Case 49: Win\_langg = "DE"**  **End Select**    ...  End Sub  Es wird hier angenommen, dass die UR-Projekt-Dateien in einem Unterordner „pros\de\“ stehen. Ggf. muss dieser Substring angepasst werden. |
| Autorun\_UncertRadio | Ein einfaches Makro, das eine Batch-artige Auswertung von UR-Projekten erlaubt, die man zuvor in **Tabelle6** dafür selektiert hat. Es lässt sich aus Tabelle7 heraus mit einem dafür vorgesehenen Button ausführen (s. weiter unten). |
| Import\_UR\_CSV\_file | Damit kann ein außerhalb von Excel im CSV-Format vorliegendes UR-Projekt in die **Tabelle4** der Exceldatei importiert werden. Es lässt sich aus Tabelle7 heraus mit einem dafür vorgesehenen Button ausführen (s. weiter unten).  Ab UR2-Version 2.4.03 enthält diese Routine am Beginn ein If-Then Konstrukt, mit dessen Aktivierung man mittels „Run\_SheetName“ den Namen des Arbeitsblatts vorgeben kann. |
| SingleRun\_UR | Dieses Makro exportiert das zuvor importierte UR-Projekt nach Um-Editieren durch den Benutzer als UR-Projekt im CSV-Format, startet dessen Auswertung durch UR und importiert die Ergebnisse nach Tabelle5 der Exceldatei. Es lässt sich aus Tabelle7 heraus mit einem dafür vorgesehenen Button ausführen (s. weiter unten).  Im Einzelnen:  Export der um-editierten **Tabelle4**: Makro DoTheExport,  das Ausführen dieser externen CSV-Datei mit UR: Makro DoSingleRun\_UncertRadio,  Import der von UR zwischengespeicherten **Ergebnisse nach Tabelle5:** Makro doFileQuery. |
| Run\_UR\_AUTOSEP | Dieses Makro ruft ebenfalls SingleRun\_UR auf (neue public Variable UR\_AUTOSEP=True), benutzt aber zwei neue Tabellen, UR2\_data und UR2\_results, für das Projekt und die Ergebniswerte; UR2 lässt hierbei das Speichern in Auto\_Report-Dateien aus; nach Beendigung werden die hierbei angelegten CSV-Dateien wieder gelöscht (sie haben die Endungen \*\_xls.csv und \*\_xls\_res.csv). |

Genau zwischen dem Aufruf der beiden Makros Import\_UR\_CSV\_file und SingleRun\_UR können die jetzt in Tabelle4 enthaltenen Eingangsdaten vom Anwender editiert werden, indem hier z.B. die Eingangsdaten für die nächste nach demselben Messverfahren durchgeführte Messung eingetragen werden.

Nach dem Run der beiden Hauptmakros können die Ergebnisse vom Anwender weiter verwertet werden, d.h. z.B. in eigene Excel-Dateien transferiert werden.

Im VB-Code (Makro Autorun\_UncertRadio) liest sich der Aufbau der zum Starten der Auswertung eines UR-Projekts benötigte Kommando-Strings, dort in der Variablen UR\_string gespeichert, wie folgt:

ab Version 2.1.1:

UR\_string = Trim(UR\_path) & "UncertRadio.exe AUTO " & Chr(34) & \_

Trim(UR\_path) & "pros\" & Trim(fname) & Chr(34) & " " & Trim(sid)

ab Version 2.2.4 folgende Zeile zusätzlich:

' add the new language code LC=:

UR\_string = Trim(UR\_string) & " LC=" & Trim(Win\_langg) & Trim(sDecimalPoint) &\_

Trim(sListSeparator)

Ab Version 2.4.03:

UR\_string = Trim(UR\_path) & "UR2\_start\_xls.bat AUTO " & Chr(34) & Trim(fname) \_

& Chr(34) & " " & Trim(sid) ' 04.06.2020

UR\_string = Trim(UR\_string) & " " & Chr(34) & "LC=" & Trim(Win\_langg) & \_

Trim(sDecimalPoint) & Trim(sListSeparator) & Chr(34)

Die ab Version 2.4.03 eingeführte **Datei UR2\_start\_xls.bat wird ab Version 2.4.22 nicht mehr verwendet, um Konflikte mit Antivirensoftware zu vermeiden**. Stattdessen wird UncertRadio direkt aus Excel heraus gestartet, nachdem zuvor die Windows-PATH-Variable ebenfalls direkt mit Excel geändert wurde:

UR\_string = Trim(UR\_path) & "uncertradio.exe AUTO " & Chr(34) & Trim(UR\_path) & \_

Trim(fname) & Chr(34) & " " & Trim(sid)

' add the language code LC=: (since 13.1.2018)

UR\_string = Trim(UR\_string) & " " & Chr(34) & "LC=" & Trim(Win\_langg) & \_

Trim(sDecimalPoint) & Trim(sListSeparator) & Chr(34)

Ab Version 2.4.26 (~26.6.2023) wird die Environment variable path wie oben angedeutet mit

**pathX = EnvironGetItem("path", "User")**

**Call SetEnvironmentVariableA("path", UR\_path & ";" & URGTK\_path & “;” & pathX)**

gesetzt.

Beispiel für den UR\_string:

ab Version 2.1.1:

D:\UR2\UncertRadio.exe AUTO "D:\UR2\pros\zzURpr.csv" 556

ab Version 2.2.4:

D:\UR2\UncertRadio.exe AUTO "D:\UR2\zzURpr.csv" 556 LC=,;

ab Version 2.4.03:

d:\UR2\UR2\_start\_xls.bat AUTO "d:\UR2\zzURpr.csv" 556 "LC=DE,;"

ab Version 2.4.22:

d:\UR2\uncertradio.exe AUTO "d:\UR2\zzURpr.csv" 556 "LC=DE,;"

Die Variablen fname und sid enthalten den UR-Projektdateinamen und den Sample\_ID-String. Der komplette Pfadname UR\_Path muss vom Benutzer am Beginn der Routine Autorun\_UncertRadio festgelegt werden.

Im VBA-Code von SingleRun\_UR wird das exportierte CSV-Projekt in den Pfad geschrieben, der in der Variablen Excel-Path hinterlegt ist:

' write out the UR project CSV file:

Ab Version 2.1.1:

file\_csv = Trim(UR\_path\_unix) & "pros\" & "zzURpr.csv"

Ab Version 2.2.4:

file\_csv = Trim(UR\_path) & "zzURpr.csv"

Call DoTheExport(file\_csv, ifehl)

If (ifehl = 1) Then Exit Sub

' execute UR once with this input file:

Call DoSingleRun\_UncertRadio(file\_csv, ifehl)

If (ifehl = 1) Then Exit Sub

Mit einer Funktion bShellAndWait wird in Autorun\_UncertRadio mit dem übergebenen UR\_string das UncertRadio mit der in der in dem String enthaltenen UR-Projektdatei gestartet. Diese Funktion veranlasst Excel nach Absetzen des Start-Befehls so lange zu warten, bis UR die Berechnungen der aktuellen Projektdatei ausgeführt und sich selbst beendet hat. Danach wird die nächste Auswertung aufgerufen.

Der ab Version 2.4.00 in den oben angeführten Kommandozeilen anstelle des direkten Aufrufs von uncertradio.exe einzusetzende Batchdateiname UR2\_start\_xls.bat wird ab Version 2.4.26 nicht mehr verwendet.

Im Makro **DoSingleRun\_UncertRadio** wurde der String für die Ausgabe des CSV-Projekts geändert (an zwei Stellen):

Vorher: file\_csv = Trim(UR\_path) & "zzURpr.csv"

Ab V. 2.4.04.: file\_csv = Trim(UR\_AUTO\_output\_path) & "zzURpr.csv"

Vorher: file\_csv = Trim(UR\_path) & filename\_org

Ab V. 2.4.04.: file\_csv = Trim(UR\_AUTO\_output\_path) & filename\_org

Die vier Kommandozeilen-Argumente hierin sind:

AUTO (%1)

trim(fname) (%2)

sid (%3)

LC=.. (%4)

Die von UncertRadio für ein Projekt erhaltenen Ergebnisse werden in tabellarischer Form sowohl in eine ASCII-Datei als auch in eine CSV-Datei geschrieben. Hierfür sind folgende Dateinamen in UR festgelegt (derzeit nicht veränderbar):

ASCII-Datei: AutoReport-Results.txt

CSV-Datei: AutoReport-Results.csv

Die Ausgabe der Daten in diese beiden Dateien erfolgt kumulativ, d.h. vorherige darin befindliche Daten werden nicht gelöscht. Die Zahlenangaben werden mit dem in Windows eingestellten Dezimalpunkt-Zeichen ausgegeben, in der Regel ein Komma.

Diese beiden Dateien können gelegentlich gelöscht werden; sie werden dann von UR wieder neu angelegt.

Bedeutung der in der Ausgabedatei enthaltenen Spalten:

|  |  |  |
| --- | --- | --- |
| Spaltenbez. | Bedeutung | Meaning |
| # | Nummer der Ergebnisgröße | number of the output quantity |
| File | UR-Projekt-Dateiname | filename of UR project |
| Sample\_id | Proben/Analyse-Identifikation | identification of sample/analysis |
| Date | Datum + Uhrzeit | date and time of evaluation |
| quantity | Symbolname der Ergebnisgröße | name of the output quantity’s symbol |
| PE | Wert der Ergebnisgröße | value of the output quantity |
| uPE | erweiterte Unsicherheit, enthält den Faktor k, s. weiter unten | value of expanded uncertainty using the coverage factor k; see below |
| BE | bester Schätzwert | best estimate |
| uBQ | dem besten Schätzwert beigeordnete erweiterte Unsicherheit | uncertainty associated with best estimate |
| LQ | untere Grenze des Vertrauensbereichs | lower limit of the confidence interval |
| UQ | obere Grenze des Vertrauensbereichs | upper limit of the confidence interval |
| sLQ | untere Grenze des kürzesten Vertrauensbereichs | lower limit of the shortest confidence interval |
| sUQ | obere Grenze des kürzesten Vertrauensbereichs | upper limit of the shortest confidence interval |
| DT\* | Erkennungsgrenze | decision threshold |
| DL# | Nachweisgrenze | detection limit |
| NT | (Nachweisgrenzentyp; sollte nur noch 1 sein, d.h. ISO 11929) | type of detection limit calculation (can only be 1, according to ISO 11929) |
| k | Erweiterungsfaktor für die Unsicherheit | coverage factor k for the uncertainty |
| kalpha | Wert von *k*1-α | value of *k*1-α |
| kbeta | Wert von *k*1-β | value of *k*1-β |
| 1-gamma | Wahrscheinlichkeit 1-γ für das Vertrauensintervall | confidence interval related probability |
| Chisqr | reduziertes Chi-Quadrat, im Falle linearer Entfaltung | reduced Chi-square value, in the case of linear unfolding |

## Starten der einzelnen Makros

Für das Starten einzelner Makros sind in **Tabelle7** Start-Buttons eingerichtet worden:



## UR mit einem in Excel editierten Projekt ausführen

Nachdem man zunächst ein externes UR-Projekt, das im [CSV-Format](#URH_CSV_FORMAT_DE) vorliegt, importiert hat,



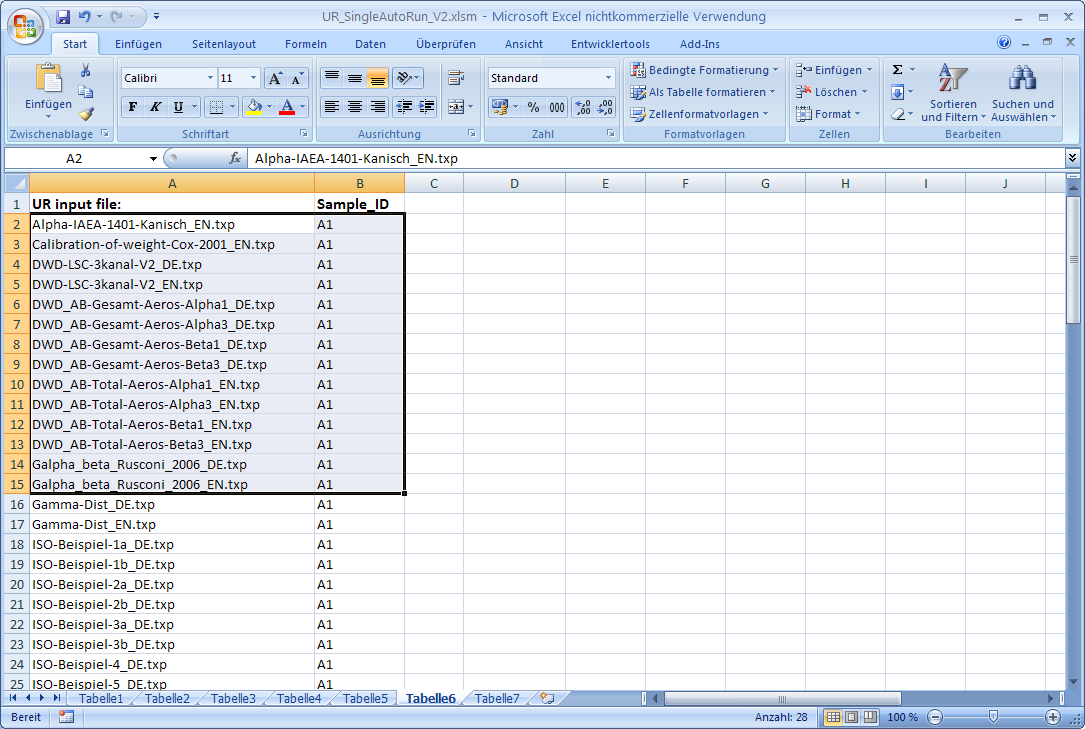
können die Daten darin angepasst/geändert werden.

Danach können die Schritte des Exports der CSV-Datei, des Startens von UR zur Auswertung und das abschließende Importieren der Ergebnisse daraus mit einem Button in Tabelle7 veranlasst werden:



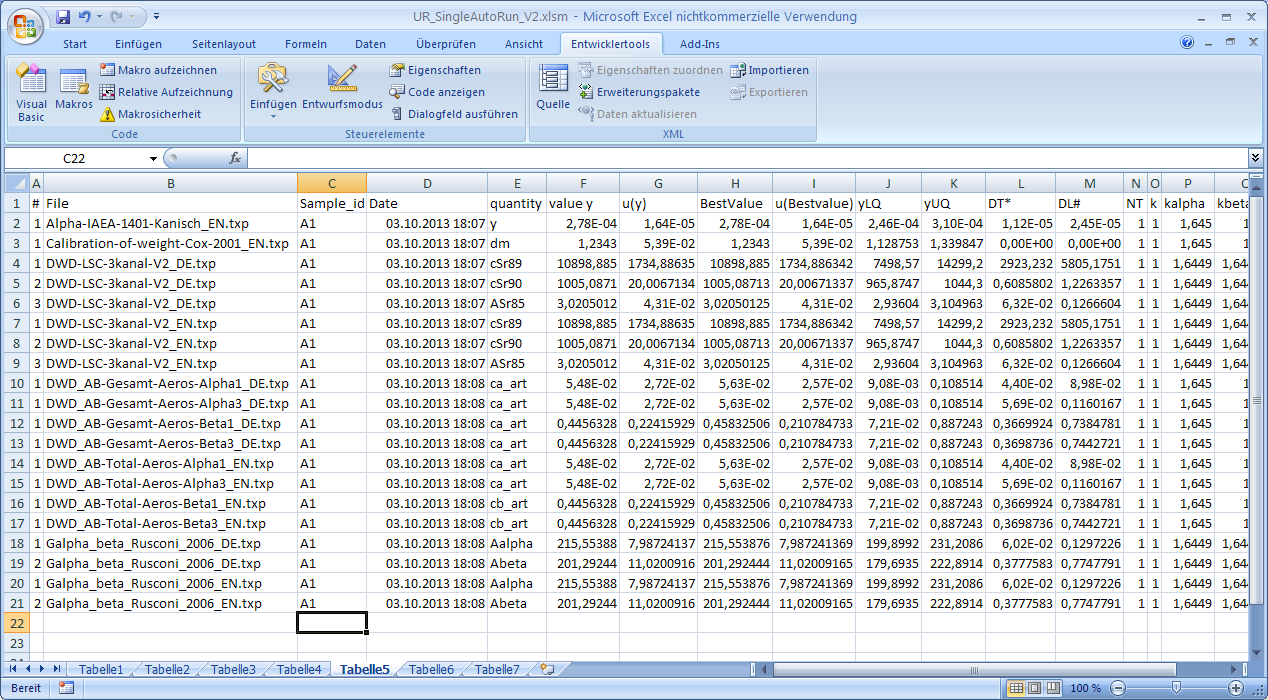
## Beispiel einer Excelanwendung für den Batch-artigen Betrieb

In der darin enthaltenen **Tabelle6** werden in Spalte A die Namen der automatisch zu bearbeitenden UR-Projektdateien und in Spalte B die Sample\_ID‘s zusammengestellt. In diesem Beispiel sind es alle UR-Projektdateien, die dem Programm beigegeben werden.



Bevor man mit dem Button Start Autorun\_UncertRadio in **Tabelle7** den Batchbetrieb startet, werden die gewünschten Dateinamen und Sample\_IDs als ein Spaltenblock wie in obiger Abbildung gezeigt markiert, die ausgewertet werden sollen.

Nach Beendigung der Batch-Auswertung wird die von UR erzeugte Ausgabedatei AutoReport-Result.csv am Ende der VB-Routine in die **Tabelle5** der Exceldatei eingelesen. Einen Ausschnitt daraus zeigt die folgende Abbildung.



## Serielle Auswertungen eines vorhandenen Projekts

Es kann der Bedarf entstehen, ein vorhandenes Projekt in der Weise mehrfach auszuwerten, dass Werte und/oder Unsicherheiten bestimmter Eingangsgrößen geändert werden. Einerseits können diese geänderten Eingangswerte zu verschiedenen Messungen gehören; anderseits möchte man u.U. erkunden, wie sich z. B. Erkennungs- oder Nachweisgrenze in Abhängigkeit der Werte bestimmter Eingangsgrößen ändern. Dieses Verfahren kann unter dem Menü-Punkt „**Bearbeiten – Serielle Auswertung**“ ausgeführt werden.

Die zu ändernden Werte und/oder Unsicherheiten bestimmter Eingangsgrößen werden vom Benutzer in Form einer CSV-Datei zusammengefasst. Ein Beispiel für eine solche Datei:

eps1; u(eps1); eps4; u(eps4)

0,338; 0,045; 0,390; 0,055

0,36; 0,045; 0,370; 0,049

0,35 0,047; 0,360; 0,052

Die erste Zeile enthält die Symbole der betreffenden Eingangsgrößen; darauf folgen Zeilen (=Sets) mit den geänderten Werten. Es können bis zu 60 Symbole/Werte verwendet werden. Die Symbole müssen im Projekt definiert sein.

Dabei bedeuten in der ersten Zeile:

Symbol der zu ändernde Wert der zum Symbol gehörenden Größe wird verwendet

u(Symbol) die zu ändernde Standardunsicherheit wird verwendet

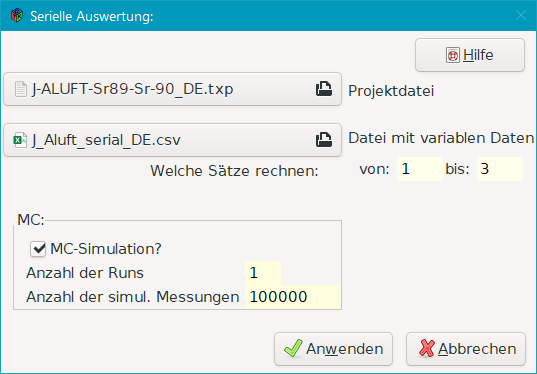
hw(Symbol) die zu ändernde Halbbreite einer rechteckverteilten Größe wird verwendet (aus

der intern die Standardunsicherheit ermittelt wird).

Zulässige Symbole sind in dieser Anwendungsform die als unabhängig deklarierten Symbole in der Tabelle „Werte, Unsicherheiten“.

Das obige Beispiel ist als Datei J\_Aluft\_serial\_DE.csv Bestandteil der UncertRadio-Installation; es ist für die Auswertung des Projekts J-ALUFT-Sr89-Sr-90\_DE.txp gedacht.

Nach Aktivieren des Menüs „**Bearbeiten – Serielle Auswertung**“ erscheint der folgende Dialog, mit dem die serielle Auswertung gestartet werden kann:



UncertRadio erzeugt daraufhin eine oder zwei Ausgabedateien mit den Ergebniswerten (ohne oder mit MC-Simulation). Deren Namen werden aus dem Namen der Eingabe-CSV-Datei erzeugt, im obigen Beispiel also

J\_Aluft\_serial\_DE\_res.csv

J\_Aluft\_serial\_DE\_mc.csv

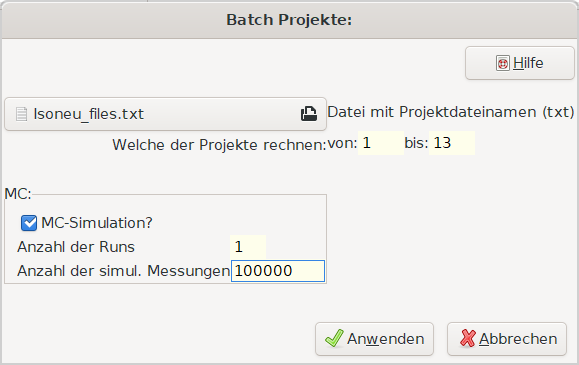
Diese Dateien enthalten Werte für:

File; #EG; PE; uPE; BE; uBE; LQ; UQ; sLQ; sUQ; DT; DL;

(Projektdateiname, Nr. der Ergebnisgröße, primärer Ergebniswert und Unsicherheit, Bester Schätzwert und Unsicherheit, unteres und oberes Quantil, Quantile mit kürzestem Abstand, Erkennungsgrenze, Nachweisgrenze)

## Projekte im Batch-Mode auswerten

UncertRadio erlaubt die Auswertung mehrerer Projekte im Batch-Modus. Dazu wird mit dem Menüpunkt **Bearbeiten – Batch-Auswertung Projekte** ein Dialog geöffnet, ähnlich dem schon in Abschnitt 5.6 vorgestellten Dialog. Darin kann eine einfache Textdatei selektiert werden, die zeilenweise die Dateinamen der auszuwertenden Projekte enthält.



UncertRadio startet dann die Berechnungen, ohne oder mit MC-Simulation, und schreibt die Ergebnisse in eine **Ausgabedatei batch\_out.csv** im CSV-Format. Für jedes ausgewertete Projekt und seinen bis zu drei Ergebnisgrößen erscheint darin der Dateiname gefolgt von einer Tabelle mit den Werten für

#EG, PE, uPE, BE, uBE, LQ, UQ, sLQ, sUQ, DT, DL.

Die Spaltenüberschriften in der Ausgabedatei sollen einen Bezug zu den Teilen 1 und 2 der ISO 11929:2019 andeuten.

Bedeutung der Kürzel:

Nr. der Ergebnisgröße, primärer Ergebniswert und Unsicherheit, Bester Schätzwert und Unsicherheit, unteres und oberes Quantil, Quantile mit kürzestem Abstand, Erkennungsgrenze, Nachweisgrenze.

Bei der MC-Simulation wird für jedes Projekt eine PNG-Datei mit der MC-Grafik erzeugt.

Bei Selektion der MC-Simulation kann es ratsam sein, für Projekte mit oder ohne Anwendung der linearen Entfaltung zwei verschiedene Batch-Listen zu verwenden, d.h. auch zwei separate Batch-Läufe, da die MC-Simulation bei linearer Entfaltung erheblich länger dauern kann.

## Testauswertung der Beispielprojekte im Batch-Mode

UncertRadio erlaubt im Batch-Modus die Auswertung aller Beispiel-Projekte. Die daraus erhaltenen Ergebnisse werden mit vorgegebenen Referenz-Ergebnissen verglichen. Dabei gefundene Abweichungen werden in einer Datei dokumentiert.

Dieser Test wird mit dem Menüpunkt **Optionen – QC-Batch – Test** aufgerufen. Es wird der folgende Dialog geöffnet:

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Hierin sind zwei Dateien voreingestellt, so dass dieser Test mit dem Button Anwenden sogleich gestartet werden kann. Die obere Datei enthält die Namen der Dateien und die Referenzwerte für die verschiedenen Beispielprojekte. Die untere Datei, deren Name frei wählbar ist, ist die Ausgabedatei. Bei festgestellten Abweichungen eines Projekts werden für die im Einzelnen verglichenen Kennwerte die Referenzwerte und die aktuell berechneten Werte aufgeführt und in Form von Verhältnissen miteinander verglichen.

Durch frühere Änderungen bzw. Korrekturen an 6 speziellen Projekten bedingt weichen hierfür die aktuellen Werte von den Referenzwerten ab. Es ist nur darauf zu achten, ob auch Abweichungen für andere als diese sechs Projekte gefunden werden. Die Abweichungen können in vgltest.txt nachgesehen werden.

Dieser Test dauert insgesamt etwa 47 Sekunden bei einer CPU-Rate von 4 GHz. Während des Testlaufs wird ein fester Bildschirminhalt gezeigt. Das jeweils in Bearbeitung befindliche Projekt wird nur in der der Kopfzeile des UncertRadio-Dialogs angezeigt. Nach Abschluss erscheint folgende Meldung auf dem Bildschirm:

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Von den drei Zeilen in dieser Meldung ist nur die mittlere zu beachten. Nur wenn darin mehr als null Abweichungen angezeigt werden, gibt es ein Problem.

Dieser Test wird immer ausgeführt, bevor eine neue Version der Software bereitgestellt wird.

# Einzelne Methoden

## Unsicherheitsfortpflanzung



### Verfahren ohne lineare Entfaltung

Eine Gleichung bzw. das System aus ggf. mehreren erforderlichen Gleichungen zur Beschreibung der Messgröße wird allgemein als **Modell der Auswertung** bezeichnet. Es wird in der Form

(1)

dargestellt. Durch Einsetzen der Schätzwerte der Eingangsgrößen erhält man den Schätzwert der Ergebnisgröße, das **primäre Messergebnis *y***:

(2)

Die dem primären Messergebnis beigeordnete Standardunsicherheit wird nach folgender Gleichung berechnet, sofern die Messwerte der Eingangsgrößen voneinander unabhängig gemessen wurden:

. (3)

Dies wird als **Unsicherheitsfortpflanzung** bezeichnet. Die partiellen Ableitungen werden auch **Sensitivitätskoeffizienten** genannt. Man schreibt es vereinfacht auch so, dass man *G* durch *Y* ersetzt:

Wurden die Eingangsgrößen nicht unabhängig voneinander gemessen, müssen in der letzten Gleichung noch Kovarianzen einbezogen werden. In allgemeiner Form lautet die erweiterte Gleichung dann:

(4)

Hierin ist die allgemeine Schreibweise für eine Kovarianz zwischen den Messwerten der beiden Eingangsgrößen, häufig auch als oder geschrieben. Hierfür gilt folgendes: und .

Gleichung (3) ist die Grundform der in UR für Verfahren ohne lineare Entfaltung angewendeten Unsicherheitsfortpflanzung, dazu werden die im TAB „Werte, Unsicherheiten“ definierten Werte und Standard-Unsicherheiten der Eingangsgrößen verwendet. Werden in dem TAB auch Kovarianzen eingegeben, wird Gleichung (4) verwendet.

### Erweiterung auf mehrere Ergebnisgrößen

Die obige Gleichung (4) für kann auch in der folgenden Form geschrieben werden:

. (5)

Als Überleitung zur linearen Entfaltung durch Erweiterung auf mehrere Ergebnisgrößen (y hat nun Indizes und ) lässt sich diese Gleichung wie folgt schreiben:

, (6)

Zu dieser letzten Gleichung gibt es eine gleichwertige Darstellung in Matrix-Schreibweise, die in UncertRadio auch angewendet wird. Dazu geht man von einem *n*-Vektor und einem *m*-Vektor aus, denen eine *n*×*n* Kovarianzmatrix und eine *m*×*m* Kovarianzmatrix zugeordnet sind. Führt man noch die *m*×*n* Matrix mit den Elementen , d.h. partiellen Ableitungen, ein,

, (7)

lässt sich Gleichung (6) wie folgt schreiben:

(8)

Diese Form ist diejenige, die dann bevorzugt angewendet wird, wenn sich die partiellen Ableitungen numerisch bestimmen lassen. Diese Form der Unsicherheitspflanzung lässt sich auch in Excel gut verwenden.

Wir betrachten den einfachen Fall, dass *m*=1 und *n*=2 ist. Dies entspricht z.B. dem Fall der linearen Regression mit 1 Gleichung und 2 Unbekannten, wobei die Kovarianzmatrix der beiden angepassten Parameter und . ist. Die Gleichung dazu ist . Dann lautet Gl. (8) ausgeschrieben (eine Unsicherheit von müsste zusätzlich in einem weiteren Term berücksichtigt werden):

Hierin erkennt man die in einzelne Terme zerlegte Gleichung (4) wieder.

### Anwendungen in UR

Im Falle der linearen Entfaltung liegen mehrere Gleichungen des Typs

vor, oder in Matrixschreibweise (**A** enthält die -Terme):

(9)

In diesem Falle stehen die Ergebnisgrößen **Y** auf der rechten Seite und die Eingangsgrößen auf der linken Seite der Gleichung. Das lineare Least squares-Verfahren liefert als Auflösung dieses Systems von Gleichungen bekanntlich:

; (10a,b)

Im nächsten Schritt wird davon ausgegangen, dass die Elemente der Matrix **A**, bei der Analyse von Abklingkurven z.B. zusammengesetzte Abklingfaktoren, nun weitere Parameter (als Vektor **p** mit Kovarianzmatrix bezeichnet) enthalten können, die auch mit Unsicherheiten behaftet sind. Dann kann mit einer Transformationsmatrix **Cp** der numerisch berechneten partiellen Ableitungen die erweiterte Kovarianzmatrix berechnet werden (der Vektor **y** erfährt dabei keine Modifikation; hier wird anstelle von **Q** die Bezeichnung **Cp** gewählt)**:**

(11)

Für jede einzelne Kombination (mit ) in der numerischen Approximation für die partielle Ableitung

(12)

müssen dazu die Matrix **A**, der Ergebnisvektor **y** und seine Kovarianzmatrix neu berechnet werden.

Alternativ kann man den zweiten Term in Gl. (11) durch Anwendung der Kettenregel für Ableitungen wie folgt ersetzen:

Dabei bedeuten: : Matrix der partiellen Ableitungen von *y*i nach den Elementen der Matrix **A**, **Dp** : Matrix der partiellen Ableitungen der Elemente von **A** nach den Parametern **p**; **UA** : Kovarianzmatrix der Elemente von **A**, die wiederum Funktionen von **p** sind. Die verwendete „Kettenregel“ für partielle Ableitungen hat die Form **Cp**= **CA·Dp**, mit der Eigenschaft  **.**

Speziell der Ausdruck wird im WTLS-Fitverfahren benötigt; er wird in einer Subroutine berechnet, die von den Fitverfahren WLS und WTLS gemeinsam genutzt werden kann. Die Matrix **UA** wird programmintern so berechnet wie im vorangehenden Unterkapitel beschrieben.

**Hinweis zur Kovarianzmatrix UA:**

Für die lineare Entfaltung nach dem Verfahren WTLS wurde mit Version 1.11 ein **Test** eingebaut, der mit Hilfe der Cholesky-Zerlegung prüft, **ob die Eingangs-Kovarianz-Matrix positiv-definit ist**. Dies wurde nochmals überarbeitet. Vor der Cholesky-Zerlegung kann man die Matrix-Elemente einem Test mit der Cauchy-Schwarz-Ungleichung unterziehen:

.

Für Paare (i, k), bei denen das = Zeichen zutrifft, wird das dazugehörige cov(i, k) mit dem Faktor (1-δ) multipliziert mit δ=1·10-09. Ist die Kovarianzmatrix danach immer noch nicht positiv-definit (die Matrix kann nicht invertiert werden), werden alle nicht-diagonalen Elemente mit dem gerade genannten Faktor multipliziert. Bleibt die Matrix damit weiterhin nicht positiv-definit, wird die Berechnung mit einem Hinweis abgebrochen.

Das primäre Ergebnis (d.h. die Fit-Parameter) der linearen Entfaltung nach den Berechnungen der vorangegangenen Gleichungen bedarf einer weiteren Ergänzung, da die Fit-Parameter in weiteren Gleichungen verwendet werden. Das ist der Fall, wenn etwa die bei der Messung erhaltene Aktivität auf Masse/Volumen bezogen und eine weitere Zerfallskorrektion angewendet werden muss, und die damit verknüpften neuen Parameter (Parameter-Vektor **q**) mit Unsicherheiten behaftet sind (Einbettung der Entfaltung). In diesem Fall kann man in der Regel ohne Matrixalgebra auskommen, da die Ergänzung der Fitparameter meistens mit Hilfe von Faktoren erfolgen kann, um die endgültige Ergebnisgröße zu erhalten:

(13)

Diese Funktionen werden in einem Vektor

(14)

zusammengefasst. Um die Kovarianzmatrix von zu berechnen, benötigt man in Analogie zu Gl. (11) eine (*n* x *n*)-Matrix der partiellen Ableitungen der Funktionen nach den angepassten Werten . Für die Berücksichtigung der Unsicherheiten der *n*q Werte ***q***benötigt man entsprechend die (*n* x *n*q)-Matrix der partiellen Ableitungen .

Unter der Voraussetzung, dass es keine Kovarianzen zwischen **y** und **q** gibt, ergibt sich damit in Verallgemeinerung von Gl. (11) als die den gesuchten (endgültigen) Ergebnisgrößen zugeordnete Kovarianzmatrix:

, (15)

wobei die Kovarianzmatrix ***U*y** diejenige aus Gl. (11) ist:

(16)

Hinweis: Werden die partiellen Ableitungen in der Matrix **Q** (Gleichung 11) auf den Vektor bezogen, d.h. durch ersetzt, wird der dazugehörige Term aus Gl. (16) entfernt und als dritter Term zu Gl. (15) hinzugefügt.

Weitere Literatur dazu:[*Cox et al., 2004*](#URH_LITERATUR_DE)*.*

## Beste Schätzwerte nach Bayes und Vertrauensgrenzen

Der bei der Auswertung für die Ergebnisgröße primär erhaltene Wert ***y*** und seine kombinierte Standardunsicherheit ***u(y)*** definieren Mittelwert und Standardabweichung einer Gaußverteilungsfunktion der möglichen Werte ,

,

die man bei der Bayes-Methode aus der Anwendung des Prinzips der maximalen Entropie erhält.

Abhängig vom Verhältnis ***u(y)/y*** kann es hierbei vorkommen, dass ein Teil dieser Kurve in den negativen Bereich fällt ( *< 0*). Nach der Bayes’schen Methode ergibt sich aus der Anwendung des „Modell-Priors“, dass diese Verteilungskurve nur für positive -Werte (Aktivitätswerte) größer als null sein soll. Die Verteilung wird also auf der linken Seite bei *= 0* abgeschnitten. Mit Hilfe dieser modifizierten (abgeschnittenen) Verteilungskurve lassen sich **Erwartungswerte für modifizierte Werte von Mittelwert und Standardabweichung** berechnen, welche als **„beste Schätzwerte nach Bayes“** bezeichnet werden. Das Integral über die modifizierte Gaußverteilungskurve von null bis unendlich beträgt ω, mit ω<1. Als Erwartungswerte ergeben sich:

Mittelwert:

mit

( ist die (kumulative) Verteilungsfunktion der Standard-Normalverteilung)

Standardabweichung:

Dies garantiert, dass immer einen positiven Wert hat.

**Das probabilistisch symmetrische Überdeckungsintervall**

Für den bestimmten Ergebniswert und der Standardunsicherheit ist dieses Intervall durch die folgenden Grenzen gegeben:

mit

mit

Zu dessen Herleitung wird die bei null abgeschnittene Normalverteilung zugrunde gelegt.

**Das kürzeste Überdeckungsintervall**

Dieses nicht mehr symmetrische Intervall ist wie folgt definiert:

mit

Im Falle von gilt stattdessen:

mit

**Numerische Bestimmung dieser Intervallgrenzen bei MC-Simulation**

Aus einer MC-Simulation einer Messgröße resultiert ein nach der Größe aufsteigendes Feld von z.B. 104 bis 106 simulierten Messwerten . Zwischen den Werten und dazugehörigen Werte der Wahrscheinlichkeit besteht ein einfacher Zusammenhang:

mit

Damit kann zu einer vorgegebenen Wahrscheinlichkeit der Index des Wertes aus dem Feld ermittelt werden:

.

Der dazugehörige Wert der als Quantil gesuchten Intervallgrenze liegt dann zwischen den Werten und .

Das kürzeste Überdeckungsintervall zur Wahrscheinlichkeit wird innerhalb einer einfachen Schleife im Feld y(i) schematisch wie folgt gesucht:

imax = \*N

min\_length = 1E+20

do i=1,imax

q\_left = y(i)

q\_right = y(int(N\*(1 – i/N)))

if(q\_right – q\_left < min\_length) then

min\_length = q\_right – q\_left

q\_left\_min = q\_left

q\_right\_min = q\_right

endif

enddo

## 6.3 Lineares Least squares-Verfahren

**Modell**

Zur Anwendung der (multi-) **linearen Least squares-Anpassung** wird von folgendem Modell für eine Abklingkurve ausgegangen:

(1)

An die von einer Zeitdauer abhängigen Messwerte der Größe *Y* kann die Summe aus bis zu drei Termen angepasst werden, die ebenfalls von der Zeitdauer abhängen; *k* zählt die Messwerte ab. Die sind die Koeffizienten (Fitparameter), deren Werte aus der Anpassung ermittelt werden. Die Größen sind als bekannt vorauszusetzende Funktionen (Abklingkurven), die nur von und anderen nicht explizit aufgeführten Parametern abhängen, die nicht Gegenstand der Anpassung sind. Sie dürfen insbesondere nicht von den anzupassenden Koeffizienten abhängen. wird als abhängige Variable, die als unabhängige Variable in der LSQ-Analyse verwendet. Es wird hierbei zunächst angenommen, dass die Messung der Werte von mit einem Einkanal-Zähler erfolgt.

Mit der Einführung von bis zu drei simultan zu behandelnden Ergebnisgrößen ist es auch möglich, dass z.B. die simultane Messung mehrerer Radionuklide in Zweikanal- oder Dreikanal-Zählern erfolgt. Mit einem LSC mit drei eingerichteten Zählkanälen können beispielsweise die Aktivitäten von Sr-90, Sr-89 und Sr-85 (Ausbeute-Tracer) simultan gemessen werden.

Sind für ein Projekt mehr als nur eine Ergebnisgröße definiert – z.B. zwei oder drei, die den Aktivitäten verschiedener Radionuklide entsprechen –, werden vom Programm drei Symbole, Fitp1, Fitp2 und Fitp3, in die Liste der Symbole eingefügt, die den Werten der entsprechen. Diese Namen dürfen nicht geändert werden. Der Benutzer kann aus ihnen dann mit Hilfe weiterer Gleichungen (im Fenster des TABs „Gleichungen“, oberhalb des Linfit-Aufrufs) z.B. die zerfallskorrigierten Aktivitätskonzentrationen berechnen lassen. Im TAB „Werte, Unsicherheiten“ werden für diese neuen Variablen vom Programm auch die aus der Anpassung erhaltenen Standard-Unsicherheiten eingetragen. Im selben TAB werden vom Programm auch die möglichen Korrelations-Paare zwischen den drei Fitparametern mit den jeweiligen Korrelationskoeffizienten eingetragen, so dass die gesamte Kovarianz-Matrix der drei Fitparamater bei der Fortpflanzung der Unsicherheiten korrekt berücksichtigt werden kann.

Es sei darauf hingewiesen, dass auch im Falle von zwei Ergebnisgrößen alle drei Symbole, Fitp1, Fitp2 und Fitp3, in die Symbolliste eingetragen werden. Lediglich dann, wenn **nur eine Ergebnisgröße** existiert, wird **keiner** dieser drei Parameter in die Symbolliste eingetragen. In diesem Fall wird der Wert dieses Fitparameters direkt der Variable *Rn* in dem Aufruf *Rn = Linfit(1,… )* zugeordnet (s weiter unten).

**In UncertRadio verwendete Routinen für Least squares- und andere Verfahren**

Im Programm kommen zwei verschiedene Routinen für die Least squares-Berechnungen zum Einsatz. Diese sind:

1. das üblicherweise angewendete **„einfache“ Least squares-Verfahren (LLSQ)**, **wenn die Werte keine Unsicherheiten haben**; die gemessene Werte jedoch haben Unsicherheiten, es werden auch Kovarianzen zwischen ihnen berücksichtigt. Falls die Werte doch Unsicherheiten haben, werden diese von UncertRadio ebenfalls berücksichtigt, indem sie in die übergeordnete Fortpflanzung der Unsicherheiten, d.h. außerhalb dieser Routine, einbezogen werden;
2. Im Herbst 2013 wurden drei verschiedene Optionen für das Anpassungsverfahren eingeführt, die sich in dem Ansatz für den Chi-Quadrat-Ausdruck unterscheiden, diese sind im Programm mit verschiedenen Kürzeln verknüpft:

**WLS**: Verwendung des **Neyman-Chi-Quadrat**; diese Verfahren ist identisch mit dem bisher als NLSQ abgekürzten Verfahren; linear, keine Iterationen;

**PLSQ**: Verwendung des **Pearson-Chi-Quadrat**; linear / iterativ;

**PMLE**: Poisson Maximum Likelihood Estimation (Poisson MLE); nicht-linear / iterativ.

1. der **„allgemeine Fall“ des Least squares-Verfahrens (WTLS, total least squares)**, welcher **darüber hinaus berücksichtigt, dass sowohl die Werte eigene Unsicherheiten haben können, als auch, dass es Kovarianzen zwischen diesen Werten geben kann.** Hierbei kommt intern ein iteratives, d.h. nichtlineares Matrix-Verfahren zum Einsatz, das wegen der Iterationen auch rechenaufwändiger ist. Die möglichen Kovarianzen zwischen den Werten ermittelt das Programm intern selbst mit Hilfe von numerisch berechneten partiellen Ableitungen nach den in den enthaltenen Symbolen, sie brauchen vom Anwender nicht selbst eingegeben werden.

Die Hintergründe zu den genannten Fitverfahren sind im [Kapitel 7.4.3](#URH_Chisq_Optionen_DE) zu finden.

Standardmäßig wird mit dem Aufruf *Rn = Linfit(1,… )* zunächst das „einfache“ LSQ-Verfahren aktiviert. Die Verwendung von WTLS kann im Dialog Festlegung des Modells der Abklingkurve aktiviert werden.

**Hinweise:**

**Untereinander korrelierte Messwerte** treten z.B. dann auf, wenn die Messwerte Nettozählraten darstellen, für deren Berechnung von ihren Bruttozählraten derselbe (identische) Wert einer Nulleffektzählrate als auch derselbe Wert für die Netto-Blindwert-Zählrate abgezogen wurde. Diese Kovarianzen - sie können in diesem Fall als Formel berechnet werden - müssen gerade bei sehr kleinen Nettozählraten berücksichtigt werden.

Im Falle der Nettozählraten () lautet die Formel für die Kovarianz zwischen zwei verschiedenen Werten:

Diese werden bereits programmintern in der entsprechenden Kovarianzmatrix zur Verfügung gestellt. Hierbei ist genau dann gleich null zu setzen, wenn und oder deren Unsicherheiten sich unterschieden.

Es sei hier darauf hingewiesen, dass in diesem Verfahren die Eingabe der Nulleffektzählrate in dem speziellen [**Eingabe-Dialog für Abkling-Kurven**](#URH_EINGABE_ABKLINGKURVE_DE) sehr individuell erfolgen kann. Es kann für jede einzelne Brutto-Zählrate eine individuelle Nulleffekt-Zählrate eingegeben werden – dies ist bei LSC-Messungen möglich –, oder es wird für die gesamte Kurve derselbe Wert () verwendet. enthält dabei ausschließlich den Detektor-bezogenen Nulleffekt.

Des Weiteren benötigt man einen Wert für die einer Blind-Analyse entsprechenden Zählrate, , von der der Nulleffektbeitrag bereits abgezogen ist. stellt also die „Netto-Blindwert-Zählrate“ dar und quantifiziert den Untergrundbeitrag aus den verwendeten Chemikalien bzw. Glasgeräten. Das Symbol *Rbl* wird mit Wert und Unsicherheit im TAB „Werte, Unsicherheiten“ geführt und mit dem Linfit-Aufruf (s. weiter unten in diesem Thema) an diese Routine übergeben.

Zur Mathematik siehe:

1. [**lineare Kurvenanpassung mit WLS**](#URH_LLSQ_MATHE_DE)**;**
2. [**lineare Kurvenanpassung mit WTLS**](#URH_GLSQ_MATHE_DE)**.**

Festlegung des Modells: siehe [***Dialog Festlegung des Modells der Abklingkurve***](#URH_MODELL_FESTL_ABKLG_DE)

Nachdem der Linfit-Aufruf in den Gleichungen erfolgte, kann in diesem Dialog die WTLS-Variante selektiert werden.

**Aktivierung der (multi-)linearen Least squares-Anpassung**

Dies wir anhand zweier unterschiedlich komplexer Beispiele gezeigt.

1. **Einfaches Beispiel:**

*Voraussetzung*:

Anzahl der Ergebnisgrößen: 1; aus der LSQ-Anpassung erhält man als Ergebnis eine Nettozählrate *Rn*.

In dem Textfenster für Gleichungen wird an der Stelle, wo man die sonst die Nettozählrate *Rn* z.B. wie folgt eingibt,

*Rn = Rb – R0*

die vorstehende Zeile durch folgende ersetzt:

*Rn = Linfit(1, Rbl, HwzY90, Hwzlong, HwzAc228, tmess, tstart)*

**Linfit** ist der Name der Prozedur, welche die gewünschte LSQ-Anpassung mit ihren dazugehörigen Dialogen aktiviert. Ihre Parameter sind:

*1* Nr. der Variante des Messproblems, für das diese Art der Berechnung der Nettozählrate erfolgen soll; z. Zt. gibt es noch nicht mehr als die oben beschriebene Variante der Y-90-Abklingkurven-Analyse;

*Rbl* von den gemessenen Y-90-Bruttozählraten abzuziehende Netto-Blindwert-Zählrate, in ;

*HwzY90* Halbwertszeit von Y-90, in

*Hwzlong* Halbwertszeit eines recht langlebigen störenden Radionuklids, in ; z.B. Th-234; für die Eingabe Hwzlong = 0 wird der dazugehörige Abklingfaktor konstant gesetzt: = 1

*HwzAc228* Halbwertszeit des möglichen Störnuklids Ac-228, in *;* dies kann jedoch auch einen Beitrag von kurzlebeigen Radon-Zerfallsprodukten simulieren

*tmess* Platzhalter für die Einzelmesszeiten

*tstart* Platzhalter für die Zeitspannen zwischen der Y-90/Sr-90-Separation und dem Startzeitpunkt der jeweiligen Einzelmessung

**Hinweis:** Ab Version 2.4.24 sollen in dem Linfit-Aufruf nur noch die drei Parameter *Rbl, tmess* und *tstart* enthalten sein: *Rn = Linfit(1, Rbl, tmess, tstart)*

Nach dem Laden der Symbole aus den Gleichungen stehen die Symbole der **Linfit**-Funktion in der Liste der Symbole zur Verfügung. Im TAB „Werte, Unsicherheiten“ müssen den Symbolen *Rbl, HwzY90, Hwzlong* und *HwzAc228* Werte und Unsicherheiten zugewiesen werden, nicht jedoch *tmess* und *tstart*.

Bis auf eine Ausnahme dürfen für die hier angeführten Symbole auch andere Namen verwendet werden, sie müssen nur in der gesamten Symbolliste des Projekts definiert sein; diese Symbole sind als „global“ gültig anzusehen.

Wichtig: Nur die Namen *Rbl*, *tmess* und *tstart* dürfen nicht verändert werden, auch nicht ihre Zuordnung zu der oben aufgeführten Bedeutung.

Nach dem Aufruf von Linfit hat *Rn* den Wert und die Unsicherheit des Fitparameters zugeordnet bekommen.

1. **Komplexeres Beispiel:**

*Voraussetzung*:

Anzahl der Ergebnisgrößen: 3; aus der LSQ-Anpassung erhält man als Ergebnis die Fitparameter Fitp1, Fitp2 und Fitp3, die den Fitparametern entsprechen, welche nun die Aktivitäten von Sr-89, Sr-90 und Sr-85 repräsentieren, in Bq. Dies ist dem Beispiel-Projekt DWD-LSC-3kanal-V2.txp entnommen.

In dem Textfenster für Gleichungen wird an der Stelle, wo man die sonst die Nettozählrate *Rn* z.B. wie folgt eingibt,

*Rn = Rb – R0*

die vorstehende Zeile durch folgende ersetzt:

*rd = Linfit(1, Rbl, eSr85A, eSr85B, eSr85C, eSr90A, eSr90B, eSr90C, eSr89A, eSr89B, &*

*eSr89C, eY90A, eY90B, eY90C, lamSr85, lamSr90, lamSr89, lamY90, tmess, tstart )*

***ab Version 2.4.24*** *wird dieser Aufruf verkürzt zu:* *Rn = Linfit(1, Rbl, tmess, tstart).*

Die Bedeutung der Symbole ist wie oben im „einfachen Beispiel“ erläutert. Die Symbol-Namen *Rbl*, *tmess* und *tstart* und ihre Bedeutung dürfen nicht verändert werden. Die Symbole *eNuklidX* (insgesamt 9 Stück) bezeichnen Nachweiswahrscheinlichkeiten der verschiedenen Radionuklide in den Zählkanälen A, B oder C. Die Symbole *lamNuklid* bezeichnen die Zerfallskonstanten der drei Radionuklide.

Anmerkung:

Neben den festen Symbolen *Rbl*, *tmess* und *tstart,* die in dem Aufruf von Linfit stehen müssen, sind die Namen weiterer Symbole nicht vorgegeben, auch nicht ihre Anzahl; sie müssen jedoch in der globalen Symbolliste vom Benutzer definiert werden. Sie müssen allerdings in den Definitionsgleichungen für die Funktionen verwendet werden; siehe dazu auch [***Dialog Festlegung des Modells der Abklingkurve***](#URH_MODELL_FESTL_ABKLG_DE).

Eingabe der Daten der Abklingkurve siehe: [**Dialog „Eingabe der Abklingkurve“**](#URH_EINGABE_ABKLINGKURVE_DE)

[**Ansicht des Ergebnisses der Least squares-Anpassung der Abklingkurve**](#URH_LSQ_ERGEBNIS_DE)

[**Hinweis zum Verfahren der Berechnung von Erkennungs- und Nachweisgrenze bei Least Squares-Anpassungen**](#URH_LSQ_NWG_DE)

## Verwendung einer Kalibrierkurve

Es bestand der Bedarf, den Wert einer bestimmten für die Auswertung benötigten Eingangsgröße (z.B. die Nachweiswahrscheinlichkeit) nicht direkt einzugeben, sondern in Abhängigkeit von einer weiteren Eingangsgröße (z.B. Massenbelegungsdichte, oder ein Externstandard-Kanalverhältnis) aus einer „Kalibrierkurve abzulesen“, d. h. in einer entsprechend als Polynom aufbereiteten Kurvendarstellung zu interpolieren.

Dazu wurde eine neue Funktion KALFIT geschaffen, die mittels eines Dialogs die Eingabe der Messwerte x (unabhängige Größe) und y (abhängige Größe) der Kalibrierkurve (max. 40 Werte) erlaubt. x und y repräsentieren (zusammen mit ihren Standardunsicherheiten) die Werte der abhängigen und unabhängigen Messgröße. Es wird ein Polynom bis maximal 3. Grades (maximal 4 anzupassende Koeffizienten) als Kurvenmodell zugrunde gelegt. Die Polynomkoeffizienten werden mit gewichteter oder ungewichteter (multi-) **linearer Least squares**-Anpassung berechnet.

(1)

Nicht besetzte Spalten der Unsicherheiten der x- oder y-Werte werden intern auf 1 gesetzt. Werden die Standardabweichungen der Werte jeweils gleich 1 gesetzt, entspricht dies einer ungewichteten LSQ-Anpassung.

Wählt man als Polynom-Grad 0, wird aus der Spalte der y-Werte formal ein **gewichteter Mittelwert (Unsicherheiten angegeben) oder ein arithmetischer Mittelwert** berechnet; dessen Standard-Unsicherheit bezieht sich auf den Mittelwert, ist also durch geteilt.

**Aktivierung des Kalibrierkurven-Tools:**

eps = KALFIT(1, eskv) (2)

In diesem Beispiel werden Wert und Unsicherheit der Nachweiswahrscheinlichkeit **eps** **als Funktion** des Externstandard-Kanalverhältnisses **eskv** durch Interpolation des damit verbundenen (angepassten) Polynoms berechnet.

Der zweite Parameter (hier eskv) stellt das Symbol für denjenigen Wert der **unabhängigen Größe X** dar, aus dem der Wert und die Standardunsicherheit der abhängigen Größe Y (hier: **eps**) berechnet werden soll.

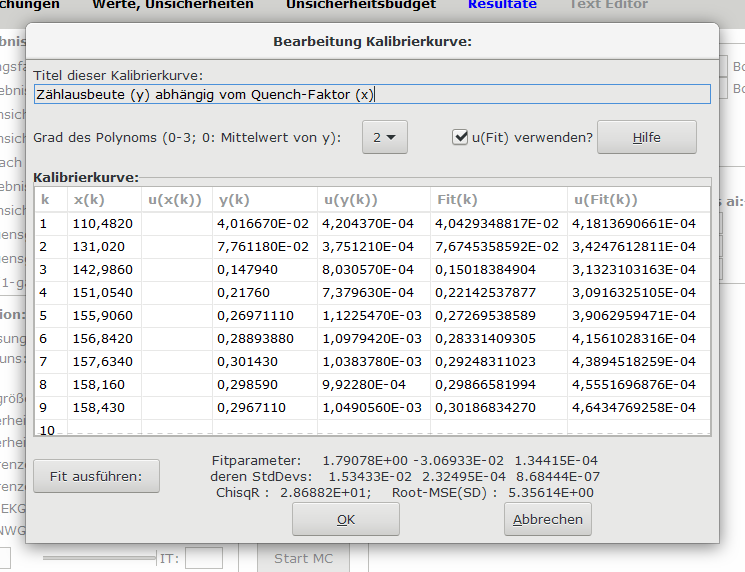
Der erste Parameter der Funktion (hier: 1) liefert die Information, wie der Wert der Größe, die auf der linken Seite der Gleichung mit KALFIT steht, berechnet wird. Der Wert 1 bedeutet, dass der Wert von Y so aus dem Polynom der Kurve berechnet wird („abgelesen“), wie oben gezeigt. Beim Wert 2 hingegen wird der Wert der Größe Y aus der Umkehrung der Polynomgleichung berechnet. Ein Beispiel für diesen zweiten Fall zeigt das UR-Projekt Beispiel\_8\_mit\_KALFIT\_DE.txp, indem die obige Gleichung durch die folgende Gleichung ersetzt wird:

Cx = KALFIT(2, Rnet)

Sie bedeutet, dass Rnet Zählraten darstellt, die zunächst in Abhängigkeit von (mit bekannter Aktivität verknüpften) bekannten Konzentrationen x kalibriert werden: Rnet = Polynom(x). Anschließend soll aus einer gemessenen Zählrate Rnetx einer Messprobe die unbekannte Konzentration Cx berechnet werden. Dazu würde man die Konzentration Cx bei einem Polynomgrad von 1 nach der Gleichung (Rnetx-a1)/a2 berechnen; für einen höheren Polynomgrad erfolgt die Invertierung numerisch durch Bisektion. In dem genannten UR-Projekt geht es um die Konzentration von Kalium (in g/L).

Der KALFIT-Aufruf hat die Bearbeitung eines neuen Dialogs zur Folge (s. auch neuer Punkt im Hauptmenü), in dem die x- und y-Werte einer Kalibrierkurve (inkl. ihrer Standardunsicherheiten) eingegeben werden und die Anpassung gerechnet wird.

Die Standardunsicherheit des gesuchten Y-Werts (eps) wird durch numerische Unsicherheitsfortpflanzung mit den angepassten Parametern und ihrer Kovarianzmatrix und mit der Unsicherheit des vorgegeben x-Werts (eskv) berechnet.

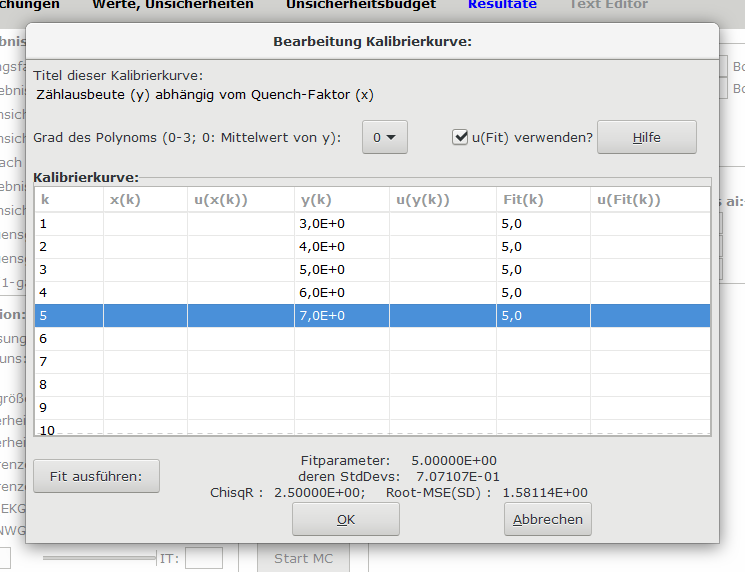


Obwohl das so wie oben definierte eps formal als eine „abhängige Größe“ zu betrachten ist, wird es programmintern (z.B. Unsicherheits-Budget) wie eine unabhängige Größe behandelt.

Derzeit wird davon ausgegangen, dass KALFIT nur für eine einzige Größe in einem UR-Projekt verwendet wird.

Die Unsicherheiten der x-Werte werden derzeit nicht berücksichtigt.

Ein Beispiel für eine Mittelwertberechnung (Polynomgrad gleich null), die zu mittelnden Werte stehen in der Spalte für y(i):



## 6.5 Aktivitätsbestimmung mit mehreren Gammalinien

Bei diesem Anwendungsbeispiel aus der **Gammaspektrometrie, vornehmlich mit hochauflösenden Germanium-Detektoren**, wird davon ausgegangen, dass zwar mehrere Radionuklide im Messpräparat vorliegen dürfen, dass aber die Gammalinien des Radionuklids (mehr als eine), für das man die Aktivität bestimmen möchte, sich nicht mit denen anderer Radionuklide überlagern. Der Fall, dass die Radionuklidaktivität mit nur einer Gammalinie bestimmt wird, gehört nicht hierher, er kann mit einer einzigen Formel (mit Hilfsgleichungen) standardmäßig mit UncertRadio behandelt werden.

Es seien die Nettozählraten von Gammalinien des Radionuklids gegeben. Aus diesen werden die einzelnen Aktivitätswerte , in Bq, nach der folgenden Gleichung berechnet:

(1)

Hierin bedeuten:

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbole:** | **Bedeutung:** |  | **Schreibweise in**  **Windows-Dialogen:** |
|  |  |  |  |
|  | Nettozählrate der Gammalinie bei der Energie , in |  | RnetRate oder  PeakNetRate |
|  | Linien-Nachweiswahrscheinlichkeit (full energy peak efficiency) bei der Energie |  | effi, eps oder  epsPeak |
|  | Emissionswahrscheinlichkeit der Linie *i* |  | pgamm |
|  | Selbstschwächungskorrektion für die Energie |  | fatt |
|  | Korrektion für Koinzidenzsummation der Linie bei der Energie |  | fcoinsu |

Die Standardunsicherheiten der nach obiger Gl. (1) berechneten Aktivitäten der einzelnen Gammalinien werden von UncertRadio intern durch Fortpflanzung der Unsicherheiten auf der Basis der Gl. (1) berechnet.

**Anmerkung**:

Es werden Kovarianzen zwischen den berechneten Aktivitätswerten berücksichtigt. Solche Kovarianzen entstehen z.B. durch gemeinsame Verwendung der Kurve der Nachweiswahrscheinlichkeit , da ihre zu verschiedenen Energien gehörenden Werte mit dem identischen Satz an Parametern berechnet werden, mit denen die Kurve dargestellt wird.

Dies kann die Einbeziehung der vollständigen Kovarianz-Matrix dieser Parameter erforderlich machen. Diese Matrix kann jedoch vom Programm nicht berücksichtigt werden; stattdessen können einzelne Werte der Kovarianzen bzw. Korrelationen im TAB „Werte, Unsicherheiten“ eingegeben werden.

**Beschreibung der Verfahren zur Mittelwertberechnung**

**a)** [**Gewichteter Mittelwert**](#URH_GSPK1_WMEAN_DE)

**b)** [**Gewichteter Least-squares Mittelwert**](#URH_GSPK1_LSQ_WMEAN_DE)

[**Verfahren der Berechnung von Erkennungs- und Nachweisgrenze für Gamspk1**](#URH_GSPK1_NWG_DE)

**Aktivierung der Auswertung mit mehreren Gammalinien**

In dem Textfenster für Gleichungen wird an der Stelle der Gleichungen, an der man sonst die Aktivität *A* des Messpräparats definiert, folgender Aufruf eingesetzt:

*A = Gamspk1(E, tlive)*

**Gamspk1** ist der Name der Prozedur, welche die gewünschte Mittelwertberechnung mit ihren dazugehörigen Dialogen aktiviert. Ihre Parameter sind:

*E* Platzhalter für die Energie einzelner Gammalinien, in keV; E werden im Programm automatisch Werte zugewiesen

*tlive* Messdauer (live-time) , in ;

Nach dem Laden der Symbole aus den Gleichungen stehen die Symbole der **Gamspk1**-Funktion in der Liste der Symbole zur Verfügung. Im TAB „Werte, Unsicherheiten“ muss dem Symbol *tlive* ein Wert und eine Unsicherheit zugewiesen werden, nicht jedoch *E*.

Natürlich dürfen für die Symbole *E* und *tlive* auch andere Namen verwendet werden, sie müssen nur in der gesamten Symbolliste des Projekts definiert sein; diese Symbole sind als „global“ gültig anzusehen.

Anmerkung: Für die hier definierte Aktivität *A* des Messpräparats muss die Zerfallskorrektion und z.B. eine Masse oder ein Volumen vom Benutzer in den Gleichungen, d.h. außerhalb von Gamspk1, eingesetzt werden.

**Eingabe der Werte:**

Weiteres zur Eingabe: siehe [**Dialog Werte der Spektrums-Auswertung**](#URH_GSPK1_WERTE_DE)

[**Ansicht des Ergebnisses der Mittelwertberechnung mit Gamspk1**](#URH_GSPK1_ANSICHT_DE)

## 6.6 Monte Carlo Simulation

Unter dem TAB „Resultate“ kann zur Überprüfung des für die Ergebnisgröße ***y*** erhaltenen Werts und seiner Unsicherheit eine **Monte Carlo Simulation** gestartet werden.

Es wird eine (große) wählbare Anzahl von Messungen der Ergebnisgröße simuliert. Dazu werden zunächst für jeden der in der im TAB „Gleichungen“ festgelegten Symbolliste als unabhängig (u) charakterisierten Messgrößen bzw. Parameter mit Hilfe eines Zufallszahlengenerators ihrer Verteilung (Normal-, Rechteck- oder Dreiecksverteilung) entsprechende Werte „gewürfelt“. Die jeweilige Verteilung einer unabhängigen Messgröße wurde zuvor im TAB „Werte, Unsicherheiten“ festgelegt. Wurde dort für Impulsanzahlen [**die (*N*+x)-Regel**](#URH_NP1REGEL_DE) ausgewählt, werden deren Ereignisanzahlen als Gamma-verteilt „gewürfelt“, auch die daraus hervorgehenden Werte der Zählraten sind dann Gamma-verteilt. Haben alle diese Größen einen Wert zugewiesen bekommen, wird damit nach den vorgegebenen Gleichungen der erste Wert der Ergebnisgröße - „die erste der simulierten Messungen“ - berechnet.

Aus der vielfachen Wiederholung dieses Vorgangs erhält man dann eine statistische Verteilung von Einzelwerten der Ergebnisgröße. Daraus wird ihr bester Schätzwert als arithmetischer Mittelwert und aus der Streuung die Standardabweichung der Einzelwerte der Ergebnisgröße berechnet. Derzeit werden bei der Methode nur Normal-, Rechteck- und Dreiecksverteilungen der einzelnen Größen berücksichtigt.

[**Ermittlung der MC-Verteilungen und daraus abgeleiteter Kenngrößen im Detail**](#URH_MCDETAIL_DE)

Der große Vorteil dieser Methode besteht darin, dass partielle Ableitungen nach den unabhängigen Größenwerten nicht benötigt werden!

**Hinweis**: der verwendete Zufallszahlengenerator hat eine Periode von etwa .

Das eben beschriebene Vorgehen gilt für den Fall, dass zwischen den als unabhängig deklarierten Größen keinerlei Korrelationen vorliegen. Hat man jedoch im TAB „Werte, Unsicherheiten“ für bestimmte **Paare von korrelierenden Größen** Kovarianzen festgelegt, werden den Paaren beim „Würfeln“ untereinander korrelierte Werte zugewiesen, entsprechend der Größe ihrer Kovarianz.

**Wichtiger Hinweis dazu**: Ab der UncertRadio-Version 0.05 (2007/11) wurde das Verfahren des „Würfelns von korrelierten Größen“ komplett auf das in Lehrbüchern dargestellte Verfahren umgestellt, welches hierzu eine Matrix-Methode verwendet (Stichwort: „Gauß-verteilte Zufallszahlen in n Dimensionen“; S. Brandt, Datenanalyse; V. Blobel und E. Lohrmann, Statistische und numerische Methoden der Datenanalyse; sowie im Entwurf des neuen Supplement 1 zum ISO GUM). Dies kann mit Hilfe der **Beispiel-Projekte Kessel-2a-2006.txp und Kessel-2b-2006.txp** aus der neueren Publikation von Kessel et al. (2006) demonstriert werden (siehe auch: [**Bedeutung des TABs "Unsicherheiten-Budget"**](#URH_TABBudget_DE)).

Zwischenergebnisse der MC-Simulation, zum Teil als Tabellen, werden ab Version 2.2.11 in der separaten Datei MC\_Tables.txt zusammengefasst.

Im Allgemeinen wird man feststellen, dass das mit der MC-Methode erhaltene Ergebnis (Wert und Unsicherheit der Ergebnisgröße ***y***) gut mit dem nach der analytischen Methode erhaltenen Ergebnis übereinstimmt. **Somit ist die MC-Methode eine relativ einfache und elegante Alternative zu dem sehr viel aufwendigeren analytischen Verfahren**.

**Was kann man aus Abweichungen zwischen beiden Methoden folgern?**

Kommt bei Anwendung der MC-Methode ein vom analytischen Verfahren abweichendes Ergebnis heraus, könnte man auf einen Fehler des Programmteils für die analytische Methode schließen. Dies muss aber nicht immer die richtige Folgerung sein!

**Was die geschilderte MC-Methode tatsächlich ausführt, kann man auch als eine „Fortpflanzung von Verteilungen“ interpretieren**. Dies bedeutet, dass damit im Prinzip für die Ergebnisgröße ***y*** der Erwartungswert, also das folgende *n*-fache Integral, abgeschätzt wird:

.

Hierin bezeichnet die zur analytischen Berechnung von ***y*** zusammengefassten Gleichungen (d.h. die Formel, mit der man ***y*** üblicherweise berechnet) und die Wahrscheinlichkeitsdichtefunktionen der *n* als unabhängigen deklarierten Messgrößen bzw. Parameter . Die Breiten der werden durch die dazugehörigen Messunsicherheiten charakterisiert. Eine wesentliche Annahme bei der herkömmlichen Fortpflanzung der Unsicherheiten besteht darin, dass die Unsicherheiten der „klein sein sollen“. In diesem Falle werden aus den Wahrscheinlichkeitsdichtefunktionen näherungsweise Delta-Funktionen, so dass sich das o. a. *n*-fache Integral zum herkömmlich berechneten Wert der Ergebnisgröße ***y*** selbst vereinfacht. In diesem Sinne können Abweichungen zwischen beiden Verfahren im Wert und der Unsicherheit der Ergebnisgröße auftreten, wenn einige der beteiligten Messunsicherheiten „nicht klein sind“ und diese zudem noch im Nenner einer Gleichung stehen (Nicht-Linearität).

Die Sammlung der Projektdateien enthält **Beispiele, bei denen die diskutierte Abweichung zwischen beiden Verfahren signifikant ist**:

|  |  |
| --- | --- |
| **Projektdatei** | **Besonderheit** |
| ISO-Beispiel-1a\_DE.txp,  ISO-Beispiel-1b\_DE.txp | Darin verursacht der Selbstabsorptionsfaktor *f* der Alphastrahlung*,* dem eine vergleichsweise breite Rechteck-Verteilung zugeordnet wurde und der im Nenner steht, diesen Effekt. |
| Neutron-Dose-Cox-2006\_DE.txp | Hierin verursacht die im Nenner stehende feldspezifische Korrektion *K* mit einer signifikant breiten Rechteckverteilung praktisch den gleichen Effekt wie im vorstehenden Beispiel: die erhaltene Verteilung ist deutlich asymmetrisch. |
| Calibration-of-weight-Cox-2001\_DE.txp | In diesem Beispiel erhält man mit der MC-Methode eine signifikant größere Messunsicherheit. Drei der beteiligten Messgrößen ist eine Rechteckverteilung zugeordnet. |
| Wuebbeler-Ex1\_DE.txp | Eine nichtlineare Modellfunktion in Verbindung mit großen Unsicherheiten der normalverteilten Eingangswerte führen zu einer schiefsymmetrischen Verteilung der Ergebnisgröße |
| Wuebbeler-Ex2\_DE.txp | Rechteck-verteilte Eingangsgrößen führen zu einer trapezförmigen Verteilung der Ergebnisgröße |

## 6.7 Low-Level-Anwendungen, (*N*+x)-Regel

Sehr kleine Impulsanzahlen sind nicht mehr normal-, sondern Poisson-verteilt. Mit Bayes’schen Methoden kann man zeigen, dass die daraus abgeleiteten Zählraten Gamma-verteilt sind. Unter Verwendung eines als gleichförmig angenommenen Priors (Weise et al., 2009) ergibt sich für den Erwartungswert der Zählrate der Wert und für die dazugehörige Standardabweichung , mit . Daraus folgt für den Fall sehr kleiner Impulsanzahlen die Empfehlung, durch zu ersetzen, **hier als -Regel bezeichnet**, die in der Literatur schon lange bekannt ist.

Man kann die Zahl x in (N+x) auch als Variable betrachten, als **GamDistAdd** bezeichnet. Sie entspricht dann einer Gammaverteilung der Größe, die in der Bayes’schen Betrachtung auf einem Prior beruht, der proportional zu 1/ρ (GamDistAdd=0.0), zu ρ-1/2 (GamDistAdd=0.5) oder konstant ist (GamDistAdd=1).

Es wird hier davon ausgegangen, dass von dieser Regel im Allgemeinen nur die Bruttozählrate *R*b und die Nulleffektzählrate *R*0 betroffen sind, weil die anderen Zählraten anderweitig bzw. weniger direkt gemessen werden. Im UncertRadio-Dialog brauchen die Werte dieser beiden Zählraten nicht modifiziert werden, wenn die (*N*+x)-Regel direkt auf die Brutto- und Nulleffektimpulsanzahlen, *N*b und *N*0, angewendet werden soll. Programm-intern wird folgende Ersetzungsregel angewendet:

Vor einem Programmteil, das Berechnungen damit ausführt:

*N*b wird ersetzt durch den Ausdruck (*N*b+ GamDistAdd),

*N*0 wird ersetzt durch den Ausdruck (*N*0+ GamDistAdd).

Nach diesem Programmteil: Diese Ersetzungen werden wieder rückgängig gemacht.

GamDistAdd ist auf den Wert 1 voreingestellt.

Die Anwendung dieser (*N*+x)-Regel auf die genannten Impulsanzahlen kann im TAB „Werte, Unsicherheiten“ „eingeschaltet“ werden, indem man als Verteilungstyp dieser Größen „(*N*+x)-Regel“ auswählt.

**Hinweis: Die (*N*+x)-Regel darf nur für Impulsanzahlen selektiert werden, nicht aber für die aus ihnen abgeleiteten Zählraten**; letztere werden dabei programmintern auf die Impulsanzahlen zurückgeführt und sind somit ebenfalls gamma-verteilt. Das bedeutet, dass diese Regel nur angewendet werden kann, wenn die Zählrate R mit einer Gleichung R=N/t auf N zurückgeführt wird. Für N wird vom Benutzer in diesem Fall nur der Wert eingegeben, das Feld für dessen Unsicherheit lässt man leer; die sich aus der Gammaverteilung ergebende Unsicherheit wird nur programmintern berechnet.

Zwei Beispiele (Gamma-Dist\_DE.txp und Lira-Gammadist\_DE.txp) zeigen die Anwendung der (*N*+x)-Regel für den Fall sehr niedriger Impulsanzahlen.

**Wichtig: Im Vorgriff auf eine neue Ausgabe der ISO 11929 wird die Anwendung der (*N*+x)-Regel in UncertRadio wie folgt abgeändert. Bis auf eine Ausnahme wird x=0 für beliebiges N verwendet. Die Ausnahme ist N=0: dann wird x=1 verwendet. Dafür muss die Variable GamDistAdd gleich null gesetzt werden (🡪 Optionen – Voreinstellungen). Unter dieser Voraussetzung, x=GamDistAdd=0, addiert UncertRadio intern nur dann den Wert 1 zu N, wenn N=0 ist (also „0+1“). Das bedeutet, dass für N>0 eine Gammaverteilung der dazugehörigen Zählrate R mit einem Prior(R) ~ 1/R verwendet wird. Bei N=0 hingegen wird ein gleichförmiger Prior(R) unterstellt.**

**Wenn dagegen x=GamDistAdd > 0 gesetzt wurde, was bei bereits existierenden UR-Projekten der Fall sein kann, wird das generell zu N addiert, auch wenn N>0 ist.**

Man kann davon ausgehen, dass die (*N*+x)-Regel nur bei den Bestimmungsverfahren ohne lineare Entfaltung zum Einsatz kommt, auch nur dann, wenn die betreffenden Impulsanzahlen sehr klein sind. **Bei Verfahren mit linearer Entfaltung** dagegen würde man bei solch kleinen Impulsanzahlen die Entfaltung vermutlich gar nicht erst versuchen; leider würde dann auch der Fall vorliegen, dass die Methode der kleinsten Quadrate bei sehr kleinen – Poisson-verteilten – Impulsanzahlen zu einem Bias führen kann und man auf Poisson-Maximum-Likelihood-Verfahren ausweichen müsste.

Hier soll mit folgendem einfachen Beispiel darauf aufmerksam gemacht werden, dass man bei Anwendung der MC-Simulation mit ggf. sehr asymmetrischen Verteilungen rechnen kann, die deutlich von der Normalverteilung abweichen.

In diesem Beispiel lauten die Gleichungen wie folgt:

A = phi \* Rn

Rn = Rg - R0

Rg = ng / t

R0 = n0 / t0

Der Ausgangsfall sei gegeben durch: Phi = 1, urel(Phi) = 0,1, t =100 s und t0 = 500 s sowie ng = 8 Impulse und n0 = 6 Impulse, d.h., Rg = (8+1)/100=0,09 s-1 und R0 = (6+1)/500=0,014 s-1. Hierfür ergibt sich bei der MC-Simulation folgendes Bild, mit nur leichter Asymmetrie der simulierten (grünen) Verteilungen:

|  |  |
| --- | --- |
| MCplotfile_a_DE.png | Ausgangsfall |

Für den nächsten Fall, mit t0 = 500 s, aber n0 = 0 Impulsen, ergibt sich eine Verteilung für die Erkennungsgrenze (nachfolgende Grafik), die eine noch größere Asymmetrie zeigt. Letztere ist die Ursache dafür, dass der MC-Wert der Erkennungsgrenze (rote vertikale Linie) um etwa das 2fache über dem entsprechenden Wert aus dem analytischen Verfahren (blaue Kurve) liegt.

|  |  |
| --- | --- |
| MCplotfile-b_DE.png | n0 auf null  heruntergesetzt |

Jetzt wird *gegenüber dem Ausgangsfall* als Einziges die Messdauer t0 der Nulleffektzählrate auf t0=100\*tm erhöht (ein in der Praxis unrealistischer Fall), auf z.B. 10 000 s, d.h. man macht R0 sehr klein, erhält man speziell für die Verteilung der Erkennungsgrenze eine extrem asymmetrische Verteilung, die nur ganz wenig in den negativen Bereich hineinragt, also ganz anders als z.B. [**bei MC-Details gezeigt**](#URH_MCDETAIL_DE).

|  |  |
| --- | --- |
| MCplotfile-c_DE.png | t0 auf t0=100\*tm erhöht |

Die Ursache ist darin zu sehen, dass bei der jetzt extrem kleinen Nulleffektzählrate die Verteilung für die Erkennungsgrenze hauptsächlich durch die Bruttozählrate verursacht wird, für die die Gammaverteilung bei kleinen Impulsanzahlen immer im positiven Bereich bleibt und stark asymmetrisch ist.

## Konfidenz-Ellipsen

Die Darstellung einer Konfidenz-Ellipse kann über das Hauptmenü „Optionen – Berechne Konfidenz-Ellipse“ aufgerufen werden.

**Konstruktion der Ellipse**

Die Konstruktion einer Konfidenz-Ellipse eines Paares von Ergebnisgrößen kann nach dem GUM Supplement 2 wie folgt skizziert werden.

Zunächst wird die Kovarianzmatrix **Uy** zweier Ergebnisgrößen ermittelt. Für letztere, hier als *y*1 und *y*2 bezeichnet, besteht ihre Kovarianzmatrix **Uy** aus den Diagonalelementen und sowie den identischen nicht-diagonalen Elementen , mit dem Korrelations-Koeffizienten .

Für die untere Dreiecks-Matrix **L** der Cholesky-Zerlegung von **Uy**, angedeutet durch **Uy**=**L** **L**T, werden mit der Jacobi-Methode die Eigenwerte **d** berechnet.

Die Längen **a** der Halbachsen der Ellipse und der Winkel *θ* , den die Hauptachsen der Ellipse bezüglich der Koordinatenachsen haben, ergeben sich aus den Gleichungen

, *j* =1,2

wobei das (1-γ)-Quantil der Chi-Quadrat-Verteilung mit 2 Freiheitsgraden ist.

**Grafische Umsetzung**

Die nachfolgende Grafik zeigt im linken Bild eine solche Konfidenz-Ellipse. Diese entspricht noch nicht dem gewohnten Bild einer Ellipse, da die Hauptachsen der Ellipse nicht senkrecht aufeinander stehen. Dies ist darauf zurückzuführen, dass die beiden Koordinaten-Achsen unterschiedlich skaliert sind. So sind 5 Skaleneinheiten der beiden Koordinatenachsen im Bild unterschiedlich lang.

|  |  |
| --- | --- |
|  |  |
| beide Achsen haben unterschiedliche Skalen | *re-scale*: beide Achsen haben gleiche Skalen (5 Skalenteile haben auf beiden Achsen dieselbe Länge) |

Diesen Nachteil kann man beseitigen, indem diese gleichartig skaliert werden. Dies erreicht man, durch folgende Re-Skalierung (im GUM Supplement 2 ist die Verwendung unterschiedlicher Skalierungen vermieden worden):

: ,

oder

: .

Zur grafischen Darstellung werden zunächst die Punkte der Ellipsenkurve unter der Voraussetzung berechnet, dass der Ursprung der Ellipse im Koordinatenursprung liegt und der Winkel der Ellipsenachsen bezüglich der Koordinatenachsen null ist. Diese Punkte werden dann mit einer Koordinatentransformation, die sich aus einer Translation und einer Drehung zusammensetzt, auf die endgültige, aber re-skalierte, Form gebracht und geplottet wird.

Die obige Abbildung zeigt im rechten Bild die re-skalierte Form der Ellipse; ihre Achsen stehen jetzt senkrecht aufeinander. In beiden Bildern sind die Intervalle

als gestrichelte Geraden eingezeichnet.

**Literatur:**

JCGM 102:2011 (GUM Supplement 2, 2011)

Brandt, S., 1999, Kapitel 5.10 und A.12

Press et al., 1992, chapter 11.1

M. A. Thompson, 2015: *Gaussian Statistics Lecture*

W. E. Hoover, 1984

## Verwendung von Datensätzen für Mittelwert und Varianz

### Mathematischer Hintergrund

Die hier verwendeten Formeln für Mittelwert und Varianz basieren auf der Bayesstatistik. Ihre Herleitung wurde im Kapitel 5.8 und im Anhang C der Publikation von Weise et al. (2013) erläutert. Es wird zwischen zwei Fällen a) und b) unterschieden (siehe auch Verfahren A und B in Tabelle 2 des [Kapitels 6.12.1](#URH_Definitionen_MW_DE)):

**Unbekannte zufällige Einflüsse:**

1. *Mittelwert-Typ 1*. Für eine beliebige Eingangsgröße *,* **die keine Impulsanzahl darstellt**, wird die Varianz wird aus der experimentellen Streuung der Einzelwerte berechnet:

(1)

Hierin bedeuten:

, und (2)

1. *Mittelwert-Typ 2.* Eine Eingangsgröße stellt eine Impulsanzahl dar, und es wird ein zusätzlicher, z.B. proben- oder verfahrensbedingter Einfluss unterstellt, der die Poisson-bedingte Streuung vergrößert. Für diesen Einfluss wird eine Normalverteilung mit den Parametern *µ* und angenommen. Dann gilt für die Varianz des Mittelwerts:

(3)

und werden entsprechend und berechnet. Der Varianzanteil

(4)

wird als bester Schätzwert des Parameters  der beteiligten Normalverteilung betrachtet. Der erste Term in der Klammer von Gl. (2), , entspricht dem Poisson-Anteil der Varianz.

Die Anwendung dieser Formeln führt zu dem überraschenden Resultat, dass eine Varianz nur berechnet werden kann, wenn mehr als 3 Einzelwerte vorliegen.

*Mittelwert-Typ 3.* Mit Version 2.3.01 kann für Vergleichszwecke die klassische Formel für die Standardabweichung des Mittelwerts verwendet werden,

, (5)

und zwar dann, wenn als Mittelwerttyp „klassisch“ verwendet wird.

**Bekannte zufällige Einflüsse**

Wenn der Anteil von (4) an (3) relativ klein ist, wird ein Parameter definiert:

Dann wird aus Gl. (3):

(6)

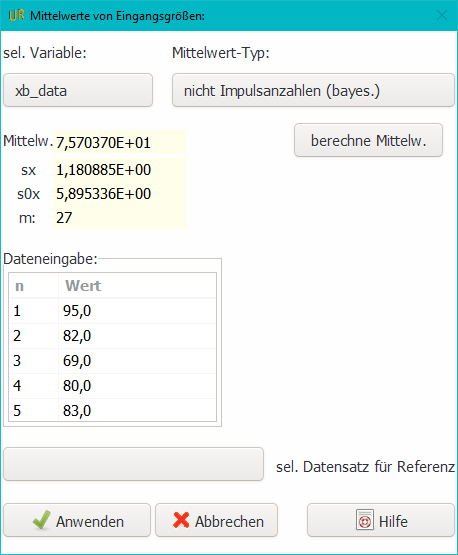
Durch Auflösen der Gl. (4) nach erhält man eine Gleichung, mit der aus einer Reihe von Messungen an einer Referenzprobe ermittelt werden kann:

(7)

Der Parameter soll kleiner als etwa 0,2 sein.

### Anwendung von Mittelwerten in UncertRadio

Wird in der Symbolliste im TAB „Gleichungen“ der Symboltyp einer Größe von „a“ oder „u“ in „m“ geändert, erwartet das Programm, dass Wert und Unsicherheit dieser Größe aus einem noch einzugebenden Datensatz ermittelt werden sollen. Diesem Zweck dient der folgende Dialog, der mit dem Icon  aus der Toolbar aufgerufen wird (es verlangt zuvor die Zeile der „m“-Variable im TAB „Werte, Unsicherheiten“ zu selektieren):



Hierin sind die Bezeichner für die Mittelwert-Datensätze schon voreingestellt, im Bild ist xb\_data für die Dateneingabe ausgewählt. Daneben kann der Typ des Mittelwerts und seiner Varianz nach den Gleichungen (1) und (3) selektiert werden. Für den „Notfall“, dass nur 3 Einzelwerte vorliegen, oder im „klassischen“ Sinne ausgewertet werden soll, steht als dritte Option die Varianz nach Gl. (5) zur Verfügung. Letztere kann also auch für mehr als 3 Einzelwerte verwendet werden. In dem gezeigten Dialog entsprechen die Standardabweichungen sx und s0x den Gleichungen (1) und (2) in 6.9.1.

Mit der unten im Dialog mit dem Hinweis „sel. Datensatz für Referenz“ angeführten Combobox kann einer der Mittelwert-Datensätze selektiert werden, der im Falle von „bekannten zufälligen Einflüssen“ als Referenz herangezogen werden soll. Ein Beispiel dafür ist das Projekt ISO-Beispiel-2b\_V2\_DE.txp. Ist kein solcher Datensatz selektiert, erfolgt die Auswertung mit der Option „unbekannte zufälligen Einflüsse“. Die Einzelheiten zu diesen beiden Optionen sind im Kapitel 6.12 erläutert.

Die daraus erhaltenen Werte für Mittelwert und Unsicherheit werden durch das Programm im TAB „Werte, Unsicherheiten“ durch den Button „Berechnung der Unsicherheiten“ in die Tabelle eingetragen.

Die Einzelwerte einer Größe mit dem Namen Symbol werden in der Projektdatei (\*.txp) als Datensatz unter dem dazugehörigen Bezeichner (symbol\_data) gesichert.

Für den **Ablauf der Dateneingabe** wird empfohlen, zunächst die Tabelle im TAB „Werte, Unsicherheiten“ auszufüllen. Für die mit „m“ markierten Variablen ist die „t-Verteilung“ als Verteilungstyp zu selektieren, damit die mit dem obigen Dialog durchzuführende statistische Behandlung korrekt verläuft. Erst dann sollte der Mittelwertdialog aufgerufen werden. In diesem wird zunächst die gewünschte Mittelwertvariable selektiert; nach Eingabe ihrer Einzelwerte wird der Mittelwerttyp selektiert und dieser dann berechnet. Nach Beendigung dieses Dialogs ist die Auswertung zu aktualisieren.

Die Eingabe der Einzelwerte wurde in diesem Dialog so geändert, dass nach Eingabe eines Wertes die Zelle darunter schon zur Eingabe aktiviert wird. Die aktivierte Zelle kann in dem Grid auch „etwas verschoben“ auftreten, der darin eingegebene Wert (mit Enter oder Cursor-down abgeschlossen) wird jedoch in die vorgesehene Zelle eingetragen. Das Eingeben von Werten wird mit Enter in der gerade aktivierten Zelle beendet, die dazu noch leer sein muss.

## 6.10 Messung eines kurzlebigen Radionuklids mit vergleichsweise langer Messdauer

### 6.10.1 Grundlagen

Wird bei der Messung der Aktivität eines Nuklids mit kurzer Halbwertszeit das Produkt deutlich größer als 0,1 oder gar , stellt die Poissonverteilung der Bruttozählrate nur noch eine Näherung dar. In diesem Falle tritt die Besonderheit auf, dass die Verteilung der Bruttoim­puls­anzahl eine Überlagerung von Binomialverteilung (Probenbeitrag) und Poissonver­teilung (Nulleffekt) ist. Charakteristisch für ein Verfahren mit binomialverteilten Impulsen des Probenbeitrags ist, dass die Varianz der Bruttoimpulsanzahl kleiner ist als die Bruttoimpulsanzahl, d.h. kleiner als die Varianz einer rein Poisson-verteilten Bruttoimpulsanzahl ist. Zur Verwendung der Binomialverteilung für die nachgewiesenen Zerfälle unter dieser Randbedingung wird auf Literatur verwiesen (Mathews et al, 1979; Spyrou et al., 1981; Salma and Zemplén-Papp, 1992; Gilmore, 2008; Semkow, 2007).

Es sei die zu Beginn der Messung vorliegende Anzahl von Atomen. Das Produkt aus der Wahr­schein­lichkeit für den Zerfall eines Atoms während der Dauer und der Wahrscheinlichkeit , diesen Zerfall in einem Detektor nachzuweisen, stellt den einen Parameter der Binomialverteilung dar, den anderen. ist mit der Aktivität verknüpft, .

Die Literatur beschränkt sich im Wesentlichen auf die Betrachtung allein der Binomialverteilung des Probenbeitrags. Die Verteilung der Bruttoimpulsanzahl (Probenbeitrag plus Nulleffekt) wird jedoch ebenfalls benötigt. Diese kann durch Faltung zweier diskreter Verteilungen der Binomial- und Poisson-Variablen X und Y hergeleitet werden:

|  |  |  |
| --- | --- | --- |
| X: |  |  |
| Y: |  |  |
| Z = X + Y: |  |  |

Mit , und der Bruttoimpulsanzahl folgt:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

**Anmerkung:**

Streng genommen ist die Form der so definierten -Verteilung nur für ganzzahlige Werte des zur Binomialverteilung gehörenden Parameters definiert. Für nicht ganzzahlige ist die darin enthaltene Binomialverteilung für sich genommen nicht auf 1 normiert. Zur Vermeidung dieses Problems kann Gl. (1) allerdings durch eine spezielle numerische Form ersetzt werden, die mit Hilfe einer hypergeometrischen Verteilung unter Verwendung der sogenannten **Kummer confluent hypergeometric function** dargestellt werden kann. Diese Form der Verteilung ist auch für ungeradzahlige Werte von geeignet und kommt, insbesondere bei größeren Werten von , mit deutlich kürzerer Rechenzeit aus. Diese Version der -Verteilung wird tatsächlich in UncertRadio verwendet.

Bei der **Monte Carlo-Simulation** werden für einen zu erzeugenden Wert der Bruttoimpulsanzahl ein binomial-verteilter Wert für den Probenbeitrag und ein Poisson-verteilter Wert für den Nulleffektbeitrag erzeugt. Die beiden Werte werden zur Bruttoimpulsanzahl addiert. Dabei kann jedoch nur die zu Gl. (1) äquivalente Form der Verteilung (diskrete Werte) erzeugt werden, da der Zufallszahlengenerator für binomialverteilte Werte nur ganzzahlige Werte erzeugt, während der Poisson-verteilte Nulleffektbeitrag kontinuierliche Werte liefert. Für eine kleine Impulsanzahl erhält man daher, für jede ganze Zahl der Binomialverteilung eine Art von Peak, eine Reihe überlappender Peaks.

### 6.10.2 Hinweise zu Unsicherheiten und zur Auswertung

Erwartungswert und Varianz der Bruttoimpulsanzahl werden wie folgt aus der -Verteilungs­dichte berech­net:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Für die Binomialverteilung allein gilt:

und (4)

(5)

so dass in den Gl. (2) und (3) das Produkt durch ersetzt werden kann.

Setzt man , das als Netto-Impulsanzahl zu interpretieren ist, in Gl. (3) ein, folgt:

(6)

Zwei für die Unsicherheitsfortpflanzung wichtige Größen sind die Brutto- und die Nettozählrate. Sie müssen auf direkt gemessene Größen zurückgeführt werden:

**Bruttozählrate**:

(7)

(8)

**Nettozählrate:**

Es gilt für :

**Zusammenhang zwischen Aktivität und**

Für die bei vorliegenden Aktivität wird in der Zeitdauer die allein auf den Beitrag des kurzlebigen Radionuklids verursachte Impulsanzahl registriert/nachgewiesen. Dafür gilt:

(9)

Mit folgt:

bzw. auch (10)

Aus dem Ausdruck für die Zählrate

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

folgt ( ist die Korrektion für den Zerfall während der Messung):

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

**Aktivitätskonzentration :**

Für die zu berechnende Aktivitätskonzentration wird die Beziehung verwendet:

(13)

Auf der Basis dieser Gleichung kann in UncertRadio die Unsicherheit durch Unsicherheitsfort­pflan­zung berechnet werden.

### 6.10.3 Ein Beispielfall

Es wird radiochemisch abgetrenntes Ac-228 (Halbwertszeit (6,15 0,03) h) während der Messdauer von 8 h gemessen. Für diese Messbedingung gilt 0,9017. Da dies Produkt deutlich oberhalb von 0,1 liegt, fast 1 ist, ist der Ac-228-Beitrag zur gemessenen Bruttoimpulsanzahl als binomialverteilt anzusetzen. Die Bruttoimpulsanzahl folgt also der Summe aus Binomial- und Poisson-verteilter Beiträge.

Verwendete Symbole und Eingangswerte:

*(entnommen aus dem UR2-Projekt Ac228\_binomial\_DE.txp)*

|  |  |
| --- | --- |
|  | Anzahl der Atome zum Beginn der Messung ( ) |
|  | : Parameter der Binomialverteilung:  ; () (1)  0,23764104; 0,004982491  Hinweis: beginnt die Messung der Dauer nicht bei 0, sondern bei , kann der Parameter erweitert werden: |
|  | Nachweiswahrscheinlichkeit: 0,4 0,0083; |
|  | Zerfallskonstante des Ac-228, Halbwertszeit 6,15 h 0,03 h;  0,1127069 h-1; 5,497896E-04 h-1 |
|  | Dauer der Ac-228-Messung, die nicht klein ist gegenüber der Halbwertszeit (z. B. 8 h); |
|  | Nulleffektzählrate, währen der Dauer = 20 h gemessen: 50 Imp./20 h = 2,50 h-1; |
|  | Bruttoimpulsanzahl: 50 Impulse in 8 h; |
|  | Faktor, der die Aktivität (Bq) in eine bezogene Aktivität (z.B. Aktivitätskonzentration) konvertiert |

Aus diesen Daten folgt:

Impulse

Impulse

Impulse

Bq

weitere Rechnungen dazu:

w = 3,79418826 u(w)= 7,90495202E-02 (w0=1)

a = 14,2282066 u(a)= 3,39566064

Summe(Produkt(Bi x Po)):

mean(BiPo)= 50.0000000 var(BiPo)= 42.8707695

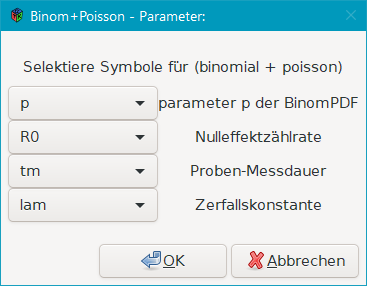
expected Var: Ng\*(1-p) + N0/t0\*tm\*p= 42.8707695

Anmerkung: Die Varianz der Bruttoimpulsanzahl (42,871) ist kleiner ist als die Bruttoimpulsanzahl (50,0), d.h. kleiner als die Varianz (50,0) einer rein Poisson-verteilten Bruttoimpulsanzahl. Für eine 10mal kleinere Nachweiswahrscheinlichkeit wird 10mal kleiner, d.h. . Bei dieser Annahme kann die Binomialverteilung in guter Näherung durch eine Poissonverteilung ersetzt werden. Dann würde man näherungsweise „Varianz der Bruttoimpulsanzahl = Bruttoimpulsanzahl“ erwarten. Dies wird durch Gl. (6) bestätigt, nach der der Wert der Varianz beträgt, also schon nahezu dem für den „Poisson plus Poisson“-Fall erwartbaren Wert von 50 entspricht.

### 6.10.4 Implementierung in UncertRadio

Der erste Schritt zum Aufruf des speziellen Verfahrens für kurze Halbwertszeiten besteht darin, in dem UR-Projekt im TAB „Werte, Unsicherheiten“ für die Variable der Bruttoimpulsanzahl den Typ „Binom+Poiss“ der Verteilungsform zu selektieren.

Danach sind 4 weitere Parameter zu selektieren: . Dazu kann über das Menü **Bearbeiten – Binomial/Poisson-Fall setzen** ein Dialog aufgerufen werden:



Beim Aufbau eines solchen UR-Projekts wird dieser Dialog auch programm-intern aufgerufen.

Die Symbolnummern der vier Parameter *p*, *R0*, *tm*, *lambda*, werden in der txp-Datei gespeichert, z.B. „BinPoi=8 10 12 9“.

## 6.11 Spezielle Verteilungen und ihre Eigenschaften

In UncertRadio sind folgende Wahrscheinlichkeitsverteilungen implementiert:

|  |  |  |
| --- | --- | --- |
| Verteilung | Kürzel | Hinweise |
| Normalverteilung | Normal |  |
| Rechteckverteilung | Rechteck |  |
| Dreiecksverteilung | Dreieck |  |
| (N+x)-Regel | (N+x)-Regel | siehe Abschnitt 6.7 |
| Lognormalverteilung | LogNormal |  |
| Gammaverteilung | GammaDist |  |
| Binomial+Poisson-Verteilung | Binom+Poiss | siehe Abschnitt 6.10.1 |
| Betaverteilung, 2 Parameter | Beta2Dist |  |
| T-Verteilung | T-Vertlg |  |
| Betaverteilung, 4 Parameter | Beta4Dist |  |
| Erlangverteilung der Messdauer für Impulsvorwahl | Npreset | ist Gammaverteilung für ganzzahlige Impulse |

Neben den bekannten Verteilungen, wie z.B. Normal-, Rechteck- oder Dreiecksverteilungen, kommen noch spezielle Verteilungen zum Einsatz,

* die Gammaverteilung,
* die Betaverteilung und
* die t-Verteilung.

deren Eigenschaften nachfolgend beschrieben werden. Diese können zusätzlich zwei bis vier Parameter verwenden. Im Folgenden werden die Wahrscheinlichkeitsdichte-Funktionen (probability density function, pdf) und der Zusammenhang ihrer Parameter mit den Messwerten erläutert.

Mit dem Icon  kann ein Dialog zur **Ansicht der Parameter einer speziellen Verteilungsdichte** einer Eingangsgröße aufgerufen werden. Dazu muss die Zeile dieser Eingangsgröße in der „Werte, Unsicherheiten“-Tabelle markiert sein.

### 6.11.1 Die Gammaverteilung

Wahrscheinlichkeitsdichte als Funktion von , mit Parametern und :

ist die Gammafunktion, es gilt

Mittelwert und Varianz der Wahrscheinlichkeitsdichte sind wie folgt definiert:

;

Ordnet man diesen den Mittelwert und die Varianz von Messwerten zu, können in Umkehrung der Gleichungen die beiden Parameter berechnet werden:

Die Gammaverteilung wird u.a. für Zählraten oder für eine Nachweiswahrscheinlichkeit mit größerer relativer Unsicherheit verwendet.

### 6.11.2 Die Betaverteilung

Wahrscheinlichkeitsdichte als Funktion von , mit Parametern und :

ist die Betafunktion.

Mittelwert und Varianz der Wahrscheinlichkeitsdichte sind wie folgt definiert:

;

Ordnet man diesen den Mittelwert und die Varianz von Messwerten zu, können in Umkehrung der Gleichungen die beiden Parameter berechnet werden:

oder (nach NIST):

Die Betaverteilung ist im Gegensatz zur Gammaverteilung nur im Bereich definiert. Sie ist damit z.B. für eine Nachweiswahrscheinlichkeit mit dem gleichen möglichen Wertebereich gut geeignet.

### 6.11.3 Die *t*-Verteilung

Es wird nicht die einfachere Student-*t*-Verteilung sondern die „nicht-standard“ *t*-Verteilung benutzt, die im englischen auch als „Scaled-and-shifted *t*-distribution bezeichnet wird. Sie besitzt als Parameter die Anzahl der Freiheitsgrade und zwei Parameter („Verschiebung“) und („Skalierung“).

Die Wahrscheinlichkeitsdichte:

Üblich ist dafür auch folgende Schreibweise .

Mittelwert und Varianz der Wahrscheinlichkeitsdichte sind hier wie folgt definiert:

;

;

Zur Ermittlung der Werte der beiden Parameter wird eine Messreihe von Wiederholungsmessungen normalverteilter Werte herangezogen, für die und als unbekannt angenommen werden. Für die Eingangsgröße ergibt sich die oben angeführte Wahrscheinlichkeitsdichte der -Verteilung in der Form

mit folgenden Parameterwerten (Mittelwert, Varianz, ):

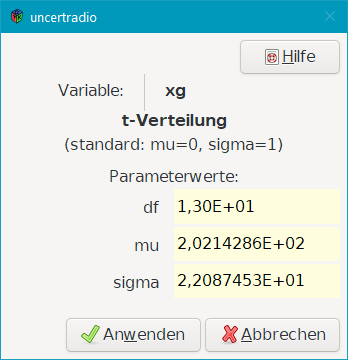
|  |  |
| --- | --- |
|  |  |

Die oben angeführten Erwartungswerte führen zu folgenden Werten:

;

Hierbei sind und als die aus den Messungen erhaltenen Eingangswerte der *t*-Verteilung anzusehen, während der Faktor sich aus der Eigenschaft der *t*-Verteilung ergibt.

Da sich die *t*-Verteilung auf eine Messreihe bezieht, muss die betreffende Eingangsgröße in UncertRadio als eine Mittelwertegröße deklariert werden ([siehe Anwendung von Mittelwerten](#URH_Datensatz_Mittelwert_DE)). Damit sind dann die Werte bzw. und programm-intern bekannt und werden über das Toolbar Icon  in dem folgenden Dialog zusammengefasst:



Zufallswerte dieser Verteilungsdichte werden mittels gewonnen, wobei die Zufallswerte der Standard-*t*-Verteilung sind, die mit einem Zufallszahlengenerator erzeugt werden. In der genannten Formel für darf der Faktor nicht enthalten sein; er ergibt sich allein aus der Verwendung der -Werte.

### 6.11.4 Erzeugung von Zufallszahlen

Zur Erzeugung von gamma-verteilten Zufallswerten wird ein Generator von Marsaglia and Wang (2000) und ein Generator aus Alan Millers Sammlung von Fortran-90-Routinen ( <https://jblevins.org/mirror/amiller/> ) verwendet. Für die beiden anderen Verteilungen werden ebenfalls Routinen aus Alan Millers Sammlung verwendet.

## 6.12 Bruttogröße: Varianzinterpolation für einen Mittelwert

### 6.12.1 Definitionen

In Anlehnung an Kapitel 6.9.1 werden folgende Definitionen für die Mittelwerte und deren Varianzen verwendet:

Tabelle 1:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Den Eingangsgrößen, denen Mittelwerte zugeordnet sind, wird dabei pauschal die *t*-Verteilung als Verteilungstyp zugeordnet. Die möglichen Werte, die den drei Parametern (Anzahl der Freiheitsgrade , Mittelwert , Standardabweichung (Skalierung)) der *t*-Verteilung zuzuordnen sind, sind in der folgenden Tabelle am Beispiel der Bruttogröße (Index g) gegeben; die Tabelle für die Untergrundgröße (Index (b) sähe entsprechend aus.

**Tabelle 2:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Verfahren A**  **„nicht Impulsanzahl“** | **Verfahren B**  **„Impulsanzahl, mit Einfluss“** | **Verfahren C**  **„klassisch“** |
|  | ***t*-Verteilung** | ***t*-Verteilung / Normalvert.** | **Normalverteilung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

( und werden entsprechend und berechnet.)

**Hinweise:**

Die Verteilung zum Mittelwerttyp „Impulsanzahl“ ist eine Überlagerung einer „shifted“ *t-*Verteilung (Mittelwert ) und einer Normalverteilung (Mittelwert 0); siehe Abschnitt 6.12.2. Der Fall in der dritten Spalte wird im Programm intern im Modus „normalverteilt“ berechnet, auch dann, wenn in UR der betreffenden Variablen die *t-*Verteilung zugeordnet wurde.

Die Varianz einer Summe einer *t-*verteilten und einer normal-verteilten Größe ist nur dann durch die Summe ihrer Varianzen gegeben, wenn mehr als etwa 5 Werte der *t-*verteilten Größe vorliegen.

Wenn die Untergrundgröße nicht als Mittelwert behandelt wird, dann wird das dazugehörige *.*

### 6.12.2 Prinzip der MC-Simulation

Zur theoretischen Behandlung des **Verfahrens B** aus Tabelle 2 wird auf den Anhang C, insbesondere C.2, bei Weise et. al. (2013) verwiesen. Dort wird gezeigt, wie der Ausdruck für die Varianz des Mittelwerts der Impulsanzahlen

(1)

herzuleiten ist. Hierin ist der erste Term, als ein Beitrag der Zählunsicherheit zu verstehen. Der ganze zweite Term in der Gleichung ist als ein *t*-verteilter Beitrag zusätzlicher zufälliger Einflüsse zur Varianz zu interpretieren.

Für eine Normalverteilung genügt die Größe

einer Student-*t*-Verteilung mit Freiheitsgraden, Erwartungswert null und Varianz ; ist der Erwartungswert von . Durch Auflösen dieser Gleichung nach entsteht die Gleichung

(2)

Diese wird als Rezept zur Erzeugung von *t*-verteilten Zufallszahlen verwendet. Mit standard-*t*-verteilten Zufallswerten werden zur Simulation der Verteilung von nach Gleichung (1) die MC-Werte wie folgt erzeugt.

Mit

(3)

werden Zufallswerte mit Mittelwert und Varianz gewonnen; im zweiten Schritt wird jeweils noch der normalverteilte Zufallswert hinzuaddiert, wobei standard-normalverteilte Zufallswerte sind.

(4)

Dieser Schritt trägt nur zur Verbreiterung der Verteilung bei.

Im Falle des einfacheren **Verfahrens A** wird allein die Gleichung (2; wird durch ersetzt) zur Erzeugung von Zufallszahlen verwendet.

**Hinweise**:

Beim Würfeln *t*-verteilter Werte entsteht der multiplikative Faktor , er darf also nicht in (2) bis (3) vorgegeben werden.

In der Tabelle „Werte, Unsicherheiten“ in UncertRadio werden diejenigen Unsicherheiten geführt bzw. angezeigt, die der Zeile für in Tabelle 2 entsprechen. Um für einen angenommenen Wert die MC-Werte nach den Gleichungen (2) oder (3) ermitteln zu können, wird programmintern zuvor aus das dazugehörige durch Umkehrung der Gleichungen (2) und (3) berechnet.

Aus der Unsicherheit berechnet man:

Gl. (2):

Gl. (3):

**Besonderheit bei der MC-Simulation von Erkennungs- und Nachweisgrenze:**

In diesem Fall ist der Faktor für die Bruttozählrate bereits in dem Ausdruck für die nach Gl. (11, unten) variierte Unsicherheit der Bruttozählrate enthalten. Wie im obigen Hinweis schon angedeutet, entsteht der Faktor zusätzlich durch das Würfeln mit : das ist die Standardunsicherheit der Standard--Verteilung. Um die doppelte Berücksichtigung des Faktors zu verhindern, wird in den Gleichungen (2) bis (4) einfach durch ersetzt.

### 6.12.3 Verfahren mit unbekannten zufälligen Einflüssen

Es wird unterstellt, dass bei Wiederholungen von Messungen unbekannte zufällige Einflüsse vorliegen, die nicht klein sind und zu einer größeren Streuung führen. Dann werden Messreihen zur Bestimmung der Brutto- und der Nulleffektzählrate (bzw. der Brutto- und Untergrundgrößen) durchgeführt.

#### Interpolation mit Hilfe der Bruttozählrate

Wird **der Wert der Bruttozählrate Rg bzw. der Bruttogröße xg** mit Hilfe eines **Mittelwerts einer Messreihe ermittelt**, kann die dazugehörige Unsicherheit nicht mehr mit z.B. u(Rg)=sqrt(Rg/t) berechnet werden. Stattdessen ist eine **Interpolation zwischen zwei bekannten Werten der Varianz** erforderlich. Nach ISO 11929 erfolgt dies für einen angenommenen Wert der Ergebnisgröße so, dass zwischen den Varianzen des primären Messergebnisses und der Varianz interpoliert wird (***in diesem Kapitel stehen die Indices g und b für „brutto“ und „Untergrund“***):

(5)

In UncertRadio wird jedoch eine solche Interpolation zwischen entsprechenden zwei Varianzen der Bruttogröße benötigt. Dieser Fall wird wie folgt auf Gl. (5) zurückgeführt. Es wird ein **Modell der Messung mit Größen (Brutto, Untergrund, Interferenz)** angenommen, in dem sowohl als auch als Mittelwerte behandelt werden:

(6)

Dafür gilt

(7)

Setzt man die rechten Seiten von (1) und (3) gleich, folgt:

(8)

Mit der Ersetzung

, (9)

folgt weiter:

(10)

Mit einem Ausdruck für :

(11)

kann nun der Ausdruck für die Varianz hergeleitet werden.

Setze jetzt noch :

(12)

Für den programm-internen Zweck werden in dieser Gleichung hierin noch und ersetzt.

Für die letzte runde Klammer in (12) folgt damit:

(13)

Einsetzen in (12) ergibt:

(14)

Die Gleichung (12) bzw. (14) stellt im Prinzip die Gleichung dar, die vom Anwender in UncertRadio in abgewandelter Form in der Tabelle „Werte, Unsicherheiten“ in die „grüne Zelle“ einzutragen wäre. Das würde die zusätzliche Definition von diversen Hilfsgrößen in UncertRadio erfordern. Das schon vorhandene Tool für Mittelwerte nach Kapitel 6.9 (siehe auch 6.12.1) eröffnet jedoch die Möglichkeit, diese Werte für diese Hilfsgrößen programmintern zu ermitteln.

In Gleichung (14) werden allein der Variablen vom Programm im Rahmen der Iterationen zur Erkennungs- und Nachweisgrenzen-Berechnung Werte direkt zugeordnet. Die fixierten Werte werden aus den beiden Tabellen in 6.12.1 bzw. den intern für die Mittelwertbehandlung angelegten Datenfeldern entnommen. Die ebenfalls fixen Werte für werden intern aus der UncertRadio-Tabelle „Werte, Unsicherheiten“ gelesen.

Somit entfällt das Eintragen einer Formel in die „grüne Zelle“ für die Standardabweichung der Bruttogröße, wenn der Größenwert als Mittelwert bestimmt wird. Dazu muss für die Größensymbole und die *t*-Verteilung als Verteilungstyp selektiert werden.

**Beispiel-Projekte dazu:** ISO-Beispiel-2a\_DE.txp (nach altem UR-Verfahren)

ISO-Beispiel-2a\_V2\_DE.txp (nach neuem UR-Verfahren)

**Äquivalenz der linearen Interpolationsalternativen**

Die nach ISO 11929 auszuführende Interpolation von Varianzen der Ergebnisgröße soll linear sein, wie in Gleichung (1) ausgeführt. Da in diesem Kapitel die Interpolation jedoch zwischen den Varianzen der Bruttozählrate erfolgt, bleibt zu testen, ob letztere Interpolation mit derjenigen nach Gl. (1) übereinstimmt. Dies wird in den beiden folgenden Abschnitten mit zwei kleinen Programmen in R getestet, getrennt nach Verfahren A und B.

#### Anwendung auf Verfahren A („nicht Impulsanzahl“)

Wenn man die Varianzen und nach 6.12.1 als Produkte und führt, kommt man für im Ergebnis zu folgender Gleichung für die interpolierte Varianz der Bruttogröße :

(15)

**Testen der Varianzinterpolation:**

(Programm Var\_intpol\_Ex13.R, zu Beispiel 13 aus ISO 11929-4)

Dazu wurden folgende Formeln für herangezogen, um nach Einsetzen in die Gl. (7) mit den Werten aus Gl. (5) verglichen zu werden ( wurde null gesetzt):

var\_Rg\_tilde\_a = fg\*(sb^2) + uxint^2 +

q\_tilde \* (fg\*(sg^2-sb^2) – uxint^2 + xn^2\*(uw/w)^2\*(1- q\_tilde) )

var\_Rg\_tilde\_b = ((fg+fb)\*sb^2)\*(1. - q\_tilde) +

q\_tilde\*( (uym/w)^2) - fb\*sb^2 - uxint^2 - q\_tilde^2\*(ym/w)^2\*(uw/w)^2

q\_tilde = (xg\_tilde – xb - xint) / (xg – xb - xint)

Mit den hierin verwendeten Werten:

sg= 71.71839 sb= 5.895336 fg= 0.03857143 fb= 0.04012346

xgtilde= 75.704 q\_tilde= 2.542306e-06

xg= 192.25 uxg= 14.08521 xb= 75.7037 uxb= 1.180885

ym= 116.5463 uym= 37.71288 uy0= 1.653796

und der Gl. (6) zur Berechnung von xg\_tilde aus y\_tilde wurden für 11 Werte von ytilde, von 0 bis 116,54, folgende Werte berechnet:

y\_tilde xg\_tilde var\_xg\_tilde\_a var\_xg\_tilde\_b vary\_tilde\_lin varytilde2

[1,] 0.00000 75.70370 1.340549 1.340549 2.73504 2.73504

[2,] 11.65463 87.35833 131.068433 131.068433 144.68766 144.68766

[3,] 23.30926 99.01296 236.346847 236.346847 286.64028 286.64028

[4,] 34.96389 110.66759 317.175789 317.175789 428.59290 428.59290

[5,] 46.61852 122.32222 373.555262 373.555262 570.54552 570.54552

[6,] 58.27315 133.97685 405.485263 405.485263 712.49814 712.49814

[7,] 69.92778 145.63148 412.965795 412.965795 854.45075 854.45075

[8,] 81.58241 157.28611 395.996855 395.996855 996.40337 996.40337

[9,] 93.23704 168.94074 354.578446 354.578446 1138.35599 1138.35599

[10,] 104.89167 180.59537 288.710565 288.710565 1280.30861 1280.30861

[11,] 116.54630 192.25000 198.393214 198.393214 1422.26123 1422.26123

Es sind keine Unterschiede zwischen den zwei Varianz-Werten der Bruttozählrate (Spalten 3, 4) festzustellen. Gleiches gilt auch für die nach den Gleichungen (5) und (7) berechneten Varianzen der Ergebnisgröße (Spalten 5, 6).

Damit ist die Äquivalenz der beiden verglichenen Interpolationsverfahren verifiziert.

#### Anwendung auf Verfahren B („Impulsanzahl, mit Einfluss“)

Für das Modell werden nach Gl. (1) die Varianzen und durch folgende Formeln ersetzt:

(16)

With

;

**Testen der Varianzinterpolation:**

(Programm Var\_intpol\_Ex14.R, zu Beispiel 14 aus ISO 11929-4)

Dazu wurden folgende Formeln für herangezogen, um nach Einsetzen in Gl. (7) mit den Werten aus Gl. (5) verglichen zu werden ( wurde null gesetzt):

var\_Rg\_tilde\_a = (un0\_mean/t0)^2 + uxint^2 +

q\_tilde \* ((ung\_mean/tg)^2-(un0\_mean/t0)^2 - uxint^2 + Rn^2\*(uw/w)^2\*(1- q\_tilde) )

var\_Rg\_tilde\_b = ((un0\_mean/t0)^2 + uxint^2) \* (1. - 2.\*q\_tilde) +

q\_tilde\*( (uym/w)^2 - q\_tilde\*(ym/w)^2\*(uw/w)^2 )

var\_Rg\_green = urbt^2 + (uRg^2 - urbt^2)\*(Rg\_tilde-rbt)/(Rg-rbt) +

(uw/w)^2\*(Rg\_tilde-rbt)\*(Rg-Rg\_tilde)

mit

q\_tilde = (Rg\_tilde - R0 - xint) / (Rg - R0 - xint)

Rg\_tilde = ytilde/w + un0\_mean/t0 + xint

rbt = R0 + xint; urbt = sqrt(uR0^2 + xint^2)

Mit var\_Rg\_green ist eine früher in der „grünen“ Zelle in UncertRadio verwendete Formel bezeichnet.

Mit den hierin verwendeten Werten:

R0= 0.02723333 u(R0)= 0.002929202 Rg= 0.06798667 u(Rg)= 0.006185528

Rn= 0.04075333 w= 34.39972 uw= 2.786688 ym= 1.401903 uym= 0.261393

uy0= 0.1425014

wurden für 11 Werte von ytilde, von 0 bis 1,402, folgende Werte berechnet:

y\_tilde var\_Rg\_tilde\_a var\_Rg\_tilde\_b var\_Rg\_tilde\_green

[1,] 0.0000000 8.580222e-06 8.580222e-06 8.580222e-06

[2,] 0.1401903 1.252920e-05 1.252920e-05 1.252920e-05

[3,] 0.2803807 1.626019e-05 1.626019e-05 1.626019e-05

[4,] 0.4205710 1.977321e-05 1.977321e-05 1.977321e-05

[5,] 0.5607614 2.306823e-05 2.306823e-05 2.306823e-05

[6,] 0.7009517 2.614528e-05 2.614528e-05 2.614528e-05

[7,] 0.8411421 2.900434e-05 2.900434e-05 2.900434e-05

[8,] 0.9813324 3.164542e-05 3.164542e-05 3.164542e-05

[9,] 1.1215228 3.406851e-05 3.406851e-05 3.406851e-05

[10,] 1.2617131 3.627363e-05 3.627363e-05 3.627363e-05

[11,] 1.4019035 3.826076e-05 3.826076e-05 3.826076e-05

Es sind keine Unterschiede zwischen den jeweils drei Varianz-Werte festzustellen.

In der folgenden Tabelle werden die für jeden Wert ytilde berechneten Werte

var\_y\_tilde\_lin aus Gl. (5);

Rg\_tilde (aus der Umkehrung von Gl. (6),

var\_Rg\_tilde (das obige var\_rg\_tilde\_b) ,

varytilde2 (aus Einsetzen von var\_Rg\_tilde in Gl. (7),

ratio (das Verhältnis varytilde2 / vary\_tilde\_lin)

präsentiert.

y\_tilde vary\_tilde\_lin Rg\_tilde var\_Rg\_tilde varytilde2 ratio

[1,] 0.0000000 0.02030666 0.02723333 8.580222e-06 0.02030666 1

[2,] 0.1401903 0.02510862 0.03130867 1.252920e-05 0.02510862 1

[3,] 0.2803807 0.02991058 0.03538400 1.626019e-05 0.02991058 1

[4,] 0.4205710 0.03471254 0.03945933 1.977321e-05 0.03471254 1

[5,] 0.5607614 0.03951451 0.04353467 2.306823e-05 0.03951451 1

[6,] 0.7009517 0.04431647 0.04761000 2.614528e-05 0.04431647 1

[7,] 0.8411421 0.04911843 0.05168533 2.900434e-05 0.04911843 1

[8,] 0.9813324 0.05392039 0.05576067 3.164542e-05 0.05392039 1

[9,] 1.1215228 0.05872235 0.05983600 3.406851e-05 0.05872235 1

[10,] 1.2617131 0.06352432 0.06391133 3.627363e-05 0.06352432 1

[11,] 1.4019035 0.06832628 0.06798667 3.826076e-05 0.06832628 1

Damit ist die Äquivalenz der beiden verglichenen Interpolationsverfahren verifiziert.

### 6.12.4 Verfahren mit bekannten zufälligen Einflüssen

Hier wird unterstellt, dass die bei Wiederholungen von Messungen von Impulsanzahlen auftretenden unbekannten zufällige Einflüsse klein sind. Es wird angenommen, dass die Einflüsse zu ähnlichen Streuungen der Brutto- und Nulleffektimpulsanzahlen führen, auch bei verschiedenen unter gleichen Bedingungen durchgeführten Messungen. Mit Hilfe einer Referenz-Analyse, d.h. einer größeren Anzahl von Messungen einer Probe (bekommt Index r), kann der unbekannte Einfluss in Form eines Parameters quantifiziert werden und auf Nulleffekt- und Bruttoimpulsanzahlen übertragen werden.

Der Paramater wurde schon in Kapitel 6.9.1 eingeführt. Er wird aus den Referenz-Messungen ermittelt und verwendet, um die Unsicherheiten der anderen Impulsanzahlen (Index x) zu schätzen.

bzw. für die Brutto- und Nulleffektimpulsanzahlen bzw. eine im Rahmen der Nachweisgrenzen-Iteration angenommene Impulsanzahl :

(1)

(2)

(3)

Die zur Berechnung von nötigen sowie alle auf die Mittelwert-Datensätze bezogenen Werte sind im Programm verfügbar. Damit sind die zur Auswertung erforderlichen Gleichungen (1), (2) und (3) zur Berechnung der einzelnen Unsicherheiten programmintern leicht berechenbar. Es ist daher, im Gegensatz zu früheren UncertRadio-Versionen, nicht erforderlich, diese selbst als Unsicherheitsformeln im TAB „Werte, Unsicherheiten“ einzutragen; auch die früher erforderliche Definition von Hilfsgrößen entfällt.

Von den in UR aufgenommenen Datensätzen muss zusätzlich der Datensatz der Referenzmessungen gekennzeichnet werden. Das erfolgt mit einem Combobox-Feld in dem in 6.9.2 gezeigten Dialog.

**Beispiel-Projekte:** ISO-Beispiel-2b\_DE.txp, Mittelwert-theta\_DE.txp (nach altem UR-Verfahren)

ISO-Beispiel-2b\_V2\_DE.txp (nach neuem UR-Verfahren)

## 6.13 Zusammenfassung der Aktivitäten mehrerer Aliquots

In bestimmten Fällen muss die Messung einer Aktivität so erfolgen, dass die Ergebnisgröße z.B. aus Messungen verschiedener Kompartimente einer Probe zu bestimmen ist, oder dass für die Bestimmung der Oberflächenkontamination einer größeren Fläche mehrere Einzelmessungen erforderlich sind, um die gesamte Fläche zu erfassen.

Dieses Kapitel beschreibt eine Möglichkeit, wie man vorgehen kann, um mehrere, auch verschiedenartige Messungen von Aktivitäten zu einem Ergebniswert zusammenzufassen. Der Ergebniswert kann entweder als Summe oder als Mittelwert der Einzelwerte berechnet werden. Das nachfolgend vorgestellte Verfahren der Zusammenfassung von Einzelmessungen berechnet auch die Werte der Erkennungs- und Nachweisgrenze, die dem Ergebniswert zugeordnet sind.

Wenn es sich bei diesen Messungen jedoch um reine Wiederholungsmessungen handelt, also um eine Messwertereihe für eine Eingangsgröße, wird auf die Kapitel [6.9](#_Anwendung_von_Mittelwerten) und [6.12](#_6.12.1_Definitionen) verwiesen.

**Aktivierung der Auswertung mehrerer Aliquotmessungen**

In dem Textfenster für Gleichungen wird an der Stelle der Gleichungen, an der man sonst die Aktivität *A* eines Messpräparats definiert, folgender Aufruf eingesetzt:

*Asum = SumEval(mode, np, A1, A2, …)*

Der Name des links vom Gleichheitszeichen stehenden Symbols kann frei gewählt werden. Der Name **SumEval** steht für eine interne Prozedur, mit der die gewünschte Art der Zusammenfassung der Messwerte einzelner Aliquots berechnet wird. Ihre Parameter sind:

*mode* ganze Zahl, mögliche Werte:

1: Berechnung des Mittelwerts der Einzelergebnisse,

2: Berechnung der Summe der Einzelergebnisse,

*np* ganze Zahl, Anzahl der Aliquotmessungen

*A1, A2, …* Die frei wählbaren Symbolnamen der Aktivitäten oder Aktivitätskonzentration der ersten bis *np*-ten Messung

Direkt im Anschluss an den SumEval-Aufruf werden zunächst die zur Berechnung der Aktivitäten Ai erforderlichen durch Formeln aufgeführt,

Ai = wi \* Rneti , nacheinander für i=1 bis np

Darauf folgt eine Liste von Formeln zum Aufbau der Kalibrierfaktoren wi und der Nettozählraten Rneti:

Rneti = Rbi – R0i

Es empfiehlt sich, durch weitere Gleichungen auch die Zählraten durch Gleichungen auf Impulsanzahlen zurückzuführen.

Ein komplettes Beispiel kann für zwei Aliquotmessungen z.B. wie folgt aussehen:

*a = 1/F \* Asum*

*Asum = SumEval(1, 2, A1, A2)*

*A1 = w1 \* Rnet1*

*A2 = w2 \* Rnet2*

*W1 = 1/eps1*

*W2 = 1/eps2*

*Rnet1 = Rb1 – R0*

*Rnet2 = Rb2 – R0*

*Rb1 = Nb1 / tb*

*Rb2 = Nb2 / tb*

*R0 = N0 / t0*

Faktoren, die in allen Ausdrücken wi vorkommen, können aus *SumEva*l herausgezogen werden, wie z.B. das 1/F (1/Fläche) in der Gleichung oberhalb der Gleichung für Asum. Damit werden Kovarianzen vermieden. Eine Eingangsgröße mit einer Unsicherheit, die in mehreren Gleichungen vorkommt, verursacht Kovarianzen zwischen den Gleichungsgrößen. Dies ist im obigen Beispiel für die Zählrate R0 gegeben, wodurch eine Kovarianz zwischen Rnet1 und Rnet2 verursacht wird.

Solche Kovarianzen müssen vom Anwender nicht extra benannt werden. Sie werden im Rahmen der Unsicherheitsfortpflanzung innerhalb der *SumEval*-Prozedur dadurch erfasst, dass intern Kovarianzterme der Form

berücksichtigt werden, die von unabhängigen Eingangsgrößen zwischen abhängigen abhängigen Größen erzeugt werden; siehe dazu [Kap. 6.1.](#_Verfahren_ohne_lineare)

Für dieses Verfahren werden keine weiteren speziellen Programmdialoge benötigt.

**Hinweise zur Berechnung von Erkennungs- und Nachweisgrenze**

Die für die Berechnung von Erkennungsgrenze und Nachweisgrenze erforderliche Iteration erfolgt durch Variation des Werts der Ergebnisgröße; ein geänderter Wert sei mit bezeichnet. Dieser muss auf die neuen Werte der in *SumEval* enthaltenen Einzelwerte übertragen werden. Dazu sind zwei Möglichkeiten denkbar, wobei zum Verständnis der Symbole das obige Beispiel verwendet wird.

Wenn aus den ein Mittelwert bestimmt werden soll, macht die Option Sinn, alle auf den gleichen Wert zu setzen. Für die Berechnung des Mittelwerts wird das in [Kap. 7.14](#_7.14_Least-squares-Berechnung_des) angedeutete Least-squares Verfahren für einen gewichteten Mittelwert verwendet.

Wenn stattdessen eine Summe der Aliquotwerte berechnet wird, kann es sinnvoll sein, die Werte so zu ändern, dass das bei der Messung erhaltene Werteverhältnis der untereinander beibehalten wird. Dies kann man durch Einführung relativer „Formfaktoren“

erreichen, so dass gilt

Hiermit lassen sich intern die zu ändernden Werte – und damit auch die Bruttozählraten – wie folgt aus berechnen:

Die Formeln für die zugeordneten Unsicherheiten lauten, wiederum auf das oben angeführte komplette Beispiel bezogen:

Mit Hilfe der Unsicherheiten der modifizierten Bruttozählraten kann die zu dem vorgegebenen modifizierten Aktivitätswert gehörende Unsicherheit berechnet werden. Mit solchen Paaren wird die für die Berechnung der Nachweisgrenze erforderliche Iteration ausgeführt.

**Beispiel-Projekte:** sumEval\_summe\_DE.txp, sumEval\_mitteln\_DE.txp

# Verschiedenes



## Auswirkung der Änderung von Optionen

Erst der **Wechsel vom TAB „Messwerte, Unsicherheiten“ zum TAB „Unsicherheiten-Budget“** beinhaltet die Ausführung der Berechnung von Erkennungsgrenze, Nachweisgrenze, Ergebnis sowie weiterer (später im TAB „Resultate“ angezeigten) statistischen Kenngrößen. Diese Rechnungen sind abhängig von den Einstellungen im **Menü Optionen - Vorgaben**. Wenn dort nach Berechnung dieser Kenngrößen Werte geändert werden, müssen die eben genannten Rechnungen erneut durchgeführt werden. Dies erreicht man durch den eingangs in diesem Hinweis genannten TAB-Wechsel. Beachten Sie den dann im letzten Feld in der Statusbar gegebenen Hinweis, wie fortzufahren ist.

## Algorithmus für die numerische Berechnung der kombinierten Unsicherheit:

**Funktionsunterprogramm UncPropa(p,u):**

! Berechnungen in Double Precision

= 0

! Berechnungen in einer „Do-Schleife“:

! Beginn:

do i=1 bis n

! Sensitivitätsfaktor = partielle Ableitung nach dem Parameter :

! Varianzbeitrag von :

end do

! Ende der Schleife:

**Unsicherheiten-Budget:** Alle Werte durch den Wert dividieren: dies sind dann die relativen Anteile der Parameter *i* an der Varianz!

: dies ist der Wert der **kombinierten Standardunsicherheit**

## Algorithmus für die iterative numerische Berechnung der Nachweisgrenze :

Startwert:

i = 0

! Beginn Iterationsschleife:

do

i = i + 1

! Die Bruttozählrate sei hier im 8ten Parameter abgelegt;

! Bruttozählrate und ihre Unsicherheit , dabei

! ist die Nulleffektzählrate des Analyten und

! die Blindzählrate aus weiteren Quellen (z.B. aus Tracer-Verunreinigungen)

! *z*1 und *z*2 sind die von UncertRadio anfangs berechneten Parameter aus *y* = *z*1 \* *R*n + *z*2.

; und

EXIT ! Iteration ist beendet.

end do

! Ende Iterationsschleife: das zuletzt erhaltene ist der

! gesuchte Wert für die Nachweisgrenze.

## Mathematik der linearen LSQ-Kurvenanpassung mit Messwerten, die miteinander korreliert sind

### Literatur

DIN ISO 11929-1 :2010, Anhang C.5.4

Klaus Weise u. Wolfgang Wöger, 1999: Meßunsicherheit und Meßdatenauswertung.

Verlag Wiley-VCH Weinheim,

S. 200 oben (Abschnitt 5.4.2 Lineare Kurvenanpassung)

Roger J. Barlow: Statistics, 1999. A Guide to the Use of Statistical Methods in the Physical Sciences. The Manchester Physics Series. John Wiley & Sons Ltd., Chichester, New York.

Section 6.6, pp. 111-113.

Zur Durchführung der Berechnung werden Matrix-Routinen aus der Datan-Library verwendet (in Fortran 90 konvertiert):

Datan-Library aus:

Siegmund Brandt: Datenanalyse, 1999. Mit statistischen Methoden und Computerprogrammen; 4. Auflage. Spektrum, Akademischer Verlag, Heidelberg-Berlin.

Weitere:

T. Hauschild, M. Jentschel, 2001: Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments. Nucl. Instr. & Meth A 457 (1-2), S 384-401.

S. Pommé & J. Keightley, 2007: Countrate estimation of a Poisson process: unbiased fit versus central moment analysis of time interval spectra. Applied Modeling and Computations in Nuclear Science. In: Semkow, T.M., Pommé, S., Jerome, S.M., Strom, D.J. (Eds.), ACS Symposium Series 945. American Chemical Society, Washington, DC, pp.316–334. 2007. ISBN0-8412-3982-7.

T.A. Laurence and B. Chromy, 2009: Efficient Levenberg-Marquardt Minimization of the Maximum Likelihood Estimator for Poisson Deviates. Report LLNL-JRNL-420247, November 13, 2009.

### Grundlagen

Das Auswertemodell wird angesetzt in der Form (Matrix-Schreibweise, nach Weise & Wöger; *Hinweis: und haben hier gegenüber Weise & Wöger vertauschte Bedeutungen*):

Der nr-Vektor charakterisiert die zu bestimmenden Parameter, der n-Vektor die eingehenden Messwerte, welche die Fitkurve repräsentieren. Die (*n* x *nr*)-Matrix enthält die partiellen Ableitungen der Fitfunktion, nach den zu bestimmenden Parametern, das sind die Funktionen . *Die Varianzen sowie Kovarianzen zwischen den gemessenen x-Werten seien in der nicht-diagonalen Unsicherheitsmatrix zusammengestellt*.

Nach Weise & Wöger erhält man für die Lösungen folgende Gleichungen (siehe auch DIN ISO 11929:2010, Anhang C.5.4, Gl. (C.32); gleiche Notation wie hier!):

; (5.63)

Hierin ist der Ergebnisvektor der angepassten Fitparameter und die dazugehörige Kovarianzmatrix.

Für den Wert der Minimumfunktion erhält man unter Verwendung von :

(5.65)

Die Kovarianzmatrix kann mit multipliziert werden, falls dessen Wert größer als eins ist, *obwohl dies bei strikter Befolgung der Bayes-Statistik nicht erlaubt ist und daher in UncertRadio nicht gemacht wird*.

### Chi-Quadrat-Optionen

Hierzu kann folgende Literatur empfohlen werden:

S. Baker, R.D. Cousins, 1983: Clarifications of the use of Ch-square and likelihood functions. Nucl. Instr. & Meth 221, S 437-442.

T. Hauschild, M. Jentschel, 2001: Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments. Nucl. Instr. & Meth A 457 (1-2), S 384-401.

Die bis hierher durchgeführte Rechnung arbeitet mit dem sogenannten ***„Neyman Chi-square“-Ausdruck***, das wie folgt definiert ist (Verfahrens-Kürzel: **WLS**):

,

d.h. die Diagonalelemente von enthalten Varianzen der gemessenen Werte . Dies kann bei niedrigen Zählraten zu einem Bias im Ergebnis für führen. Zur Reduktion des Bias kann das sogenannte ***„Pearson Chi-square“-***Ausdruck herangezogen werden, das definiert ist als (Verfahrens-Kürzel: **PLSQ**):

Die im Nenner stehende Varianz wird mit Hilfe der anzupassenden Funktion berechnet.

,

worin und die Nulleffekt- und die Netto-Blindwert-Zählrate darstellen.

Da dieser Funktionswert nur mit Hilfe der dabei anzupassenden Parameter ***y*** berechnet werden kann, ist zur Minimierung dieses Chi-Quadrat-Ausdrucks ein iteratives Vorgehen erforderlich. Die Iteration lässt sich in der zu Gl. (5.63) analogen Schreibweise, mit (m) und (m‑1) als aufeinander folgender Iterationsschritte, wie folgt ausführen (m=3):

;

Hierzu werden die „gemessenen“ Varianzen vor der Wiederholung der Rechnung durch die Varianzen der zuvor angepassten Funktion ersetzt, indem die diagonalen Elemente der Unsicherheitsmatrix der Messwerte durch die mit Hilfe der aus dem ersten Auswertelauf (Neyman-Chi-Quadrat) erhaltenen angepassten Werte neu berechnet werden (die nicht-diagonalen Elemente bleiben unverändert):

**Anmerkung**: Sind die Kovarianzen zwischen den Eingangswerten gleich null, d.h. die Matrix diagonal mit den Varianzen als Elementen, entspricht das Verfahren mit dem Neyman-Chiquadrat-Ausdruck vollständig der bekannteren Methode der *gewichteten linearen Least-squares-Anpassung*.

Alternativ zum Pearson-Chi-Quadrat-Ausdruck kann zur Verringerung des Bias auch das sogenannte „**Poisson Maximum Likelihood Estimation**“-Verfahren (oder Poisson MLE) verwendet werden. Der dazugehörige Chi-Quadrat-Ausdruck lautet (Verfahrens-Kürzel: **PMLE**):

.

Zur numerischen Durchführung der Parameter-Anpassung muss hierzu jedoch ein nicht-lineares Fitverfahren verwendet werden: in UncertRadio wird dazu eine modifizierte Version der „Levenberg-Marquardt“-Fitroutine LSQmar aus dem Buch „Datenanalyse“ von S. Brandt verwendet. Die für Poisson MLE erforderliche Modifikation von LSQmar ist dem Report LLNL-JRNL-420247 (2009) zu entnehmen.

Das PMLE-Verfahren kann in dem Beispiel-Projekt vTI-Y90-16748\_BLW\_V2\_DE.txp aktiviert werden.

Werden mit den anderen Fitverfahren Nettozählraten angepasst, erfordert das PMLE-Verfahren die Anpassung an Brutto-Impulsanzahlen. Dies wird intern in UncertRadio berücksichtigt. Dies hat den Vorteil, dass es zwischen Bruttoimpulsanzahlen keine Kovarianzen gibt, was bei der Verwendung von Nettozählraten jedoch der Fall ist.

Nach Baker & Cousins führt die Verwendung des **PMLE-Verfahrens zur Erhaltung der Fläche unter der anzupassenden Kurve**, während PLSQ zur Überschätzung und WLS zur Unterschätzung führt.

[**Hinweise zum Verfahren PMLE**](#URH_PMLE_Hinweise_DE)

**Statistische Tests der Chi-Quadrat-Optionen**

Es wird die Summe zweier gleichhoher Gauß-Peaks mit identischer Breite (sigma = 5,1 Kanäle, Nettopeakfläche je 150 Impulse, bei Kanal 30 und Kanal 90) auf einem konstanten Untergrund von 4 Impulsen/Kanal als die anzupassende Funktion verwendet. Die Fitfunktion hat also 6 Parameter: 4.0, 150, 30, 5.1, 150, 90. Diese Parameter werden als wahre Werte vorgegeben, um danach eine große Anzahl Nsp=400 entsprechender Spektren zu erzeugen deren „wahre“ Kanalinhalte als Poisson-verteilte Werte statistisch verrauscht werden. Jedes dieser Spektren wird mit den verschiedenen oben angeführten Optionen angepasst. Es wird sowohl ein nicht-lineares als auch ein lineares Fitverfahren verwendet. Die nachfolgende Abbildung zeigt ein Spektrum mit der „wahren“ Fitfunktion und den mit nicht-linearem Fitten erhaltenen Kurven für die Verfahren WLS und PMLE.

|  |
| --- |
|  |

Die Auswertung von je 400 Spektren, die nach den Verfahren WLS, PLSQ und PMLE angepasst wurden, ist nachfolgend tabellarisch zusammengestellt. Von den je 400 gefitteten Werten der 6 Parameter wird ein Mittelwert (pa=), eine Standardabweichung der Einzelwerte (sd=) sowie der Mittelwert der einzelnen von der Fitroutine berechneten Standardunsicherheiten (mean(sds..)) ermittelt. Zusätzlich werden für die jeweils 6 Parameter die Summen (über die Parameter) der absoluten Abweichungen (Fitwert minus wahrer Wert) und der Differenzen (sd minus sds) berechnet, die in je zwei Zeilen unter jeder Einzel-Tabelle aufgeführt sind. Diese Werte dienen als gute Indikatoren der Konsistenz der Anpassungen.

Nicht-lineares Fitten:

mean values: for method WLS

i=1 pa= 2.75534E+00 sd(pa(i))= 2.34472E-01 mean(sds(i))= 1.73935E-01

i=2 pa= 1.57094E+02 sd(pa(i))= 1.91809E+01 mean(sds(i))= 1.59540E+01

i=3 pa= 2.99839E+01 sd(pa(i))= 7.59367E-01 mean(sds(i))= 5.81371E-01

i=4 pa= 5.18531E+00 sd(pa(i))= 5.09876E-01 mean(sds(i))= 3.88957E-01

i=5 pa= 1.58079E+02 sd(pa(i))= 1.84934E+01 mean(sds(i))= 1.59980E+01

i=6 pa= 8.99873E+01 sd(pa(i))= 7.84781E-01 mean(sds(i))= 5.79122E-01

sum of absolute deviations from true(pa) : 16.5316238

sum of absolute deviations of two sd types : 6.28735828

mean values: for method PLSQ

i=1 pa= 3.99617E+00 sd(pa(i))= 2.11079E-01 mean(sds(i))= 1.73482E-01

i=2 pa= 1.49851E+02 sd(pa(i))= 1.75574E+01 mean(sds(i))= 1.57870E+01

i=3 pa= 3.00262E+01 sd(pa(i))= 6.38806E-01 mean(sds(i))= 5.91121E-01

i=4 pa= 5.07812E+00 sd(pa(i))= 4.50239E-01 mean(sds(i))= 3.95102E-01

i=5 pa= 1.50346E+02 sd(pa(i))= 1.63419E+01 mean(sds(i))= 1.58062E+01

i=6 pa= 8.99958E+01 sd(pa(i))= 6.39568E-01 mean(sds(i))= 5.90819E-01

sum of absolute deviations from true(pa) : 0.551150203

sum of absolute deviations of two sd types : 2.49521804

mean values: for method PMLE

i=1 pa= 4.00893E+00 sd(pa(i))= 2.10194E-01 mean(sds(i))= 2.07077E-01

i=2 pa= 1.51928E+02 sd(pa(i))= 1.69198E+01 mean(sds(i))= 1.68943E+01

i=3 pa= 2.99583E+01 sd(pa(i))= 6.46945E-01 mean(sds(i))= 6.26552E-01

i=4 pa= 5.08396E+00 sd(pa(i))= 4.22725E-01 mean(sds(i))= 4.34723E-01

i=5 pa= 1.49174E+02 sd(pa(i))= 1.73843E+01 mean(sds(i))= 1.67870E+01

i=6 pa= 8.99848E+01 sd(pa(i))= 6.22015E-01 mean(sds(i))= 6.35672E-01

sum of absolute deviations from true(pa) : 2.83508420

sum of absolute deviations of two sd types : 0.671966136

Die beste absolute Abweichung (Fitwert minus wahrer Wert) wird für das Verfahren PLSQ erhalten; die beste Konsistenz der Unsicherheiten (abs. Abweichung (sd minus sds) ) hat sich für das PMLE-Verfahren ergeben.

Dass in diesem Beispiel die Verfahren PLSQ und PMLE deutlich besser als der klassische Fit (WLS) abschneiden, ist damit begründet, dass der konstante Untergrund mit 4 Impulsen/Kanal sehr niedrig ist. Wenn der Untergrund deutlich zunimmt, fallen die Unterschiede zwischen den Fitverfahren geringer aus.

Dieselben Spektren können auch linear angepasst werden, wenn der Breiten- und der Lageparameter bei ihren wahren Werten festgehalten werden. In dem Falle werden nur der Untergrund-Parameter und die beiden Peakflächen gefittet. Eine entsprechende Auswertung ist nachfolgend aufgeführt.

Lineares Fitten:

mean values: method = WLS

i=1 pa= 2.77091E+00 sd(pa(i))= 2.19480E-01 mean(sds(i))= 1.61749E-01

i=2 pa= 1.53462E+02 sd(pa(i))= 1.78632E+01 mean(sds(i))= 1.50700E+01

i=5 pa= 1.54138E+02 sd(pa(i))= 1.74788E+01 mean(sds(i))= 1.51000E+01

sum of absolute deviations from true(pa) : 8.82905006

sum of absolute deviations of two sd types : 5.22971344

mean values: method = PLSQ

i=1 pa= 4.00339E+00 sd(pa(i))= 1.95047E-01 mean(sds(i))= 1.92895E-01

i=2 pa= 1.49325E+02 sd(pa(i))= 1.63338E+01 mean(sds(i))= 1.58374E+01

i=5 pa= 1.49789E+02 sd(pa(i))= 1.53364E+01 mean(sds(i))= 1.58544E+01

sum of absolute deviations from true(pa) : 0.889460266

sum of absolute deviations of two sd types : 1.01658702

Auch hierbei ist das Ergebnis für den PLSQ-Fit deutlich besser als für den Fit mit WLS.

### Export von Eingangsdaten nach R

Ab dieser Version von UncertRadio besteht die Möglichkeit, die Eingangsdaten in der für die verschiedenen Anpassungsverfahren (außer für WTLS: **R** hat keine entsprechende Routine!) erforderlichen Form in eine Textdatei (URExport-to-R.txt) zu exportieren, von der aus man die Daten in Teilen in das **Statistik-Paket R** importieren kann. Das erlaubt die aus R erhaltenen Anpassungs-Ergebnisse mit denen aus UncertRadio zu vergleichen. Diese Möglichkeit, sie ist über das Menu „Optionen – LSQ-Export nach R“ aktivierbar, besteht für die Berechnungen der Ergebnisgröße und der Erkennungsgrenze. Die dazu erforderlichen Daten beziehen sich nur auf Zählraten, nicht auf Aktivitäten. Man erhält z.B. aus UR für den Fall der Ergebnisgröße (s. Datei URExport-to-R.txt):

(Hinweis: die Kovarianz-Matrix ist hier rechts abgeschnitten worden)

**Case: output quantity**

Blank count rate= 4.66670009E-08 background rate= 1.88333332E-03

Input data: variance-covariance matrix: (rank= 18 )

2.87780E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.46788E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 1.96152E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 1.64805E-07 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 1.41898E-07 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 1.35870E-07 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 1.43104E-07

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

Arrays y, X1, x2, X3:

y X1 X3 (Eingangsdaten-Matrix)

1 5.65134E-03 8.80220E-01 2.73093E-01

2 4.47079E-03 8.07210E-01 1.10866E-01

3 3.01245E-03 7.40257E-01 4.50075E-02

4 2.10968E-03 6.78857E-01 1.82714E-02

5 1.44995E-03 6.22550E-01 7.41751E-03

6 1.27634E-03 5.70913E-01 3.01124E-03

7 1.48468E-03 5.23559E-01 1.22245E-03

8 9.98564E-04 4.80133E-01 4.96272E-04

9 8.47116E-04 4.40700E-01 2.03089E-04

10 8.24953E-04 4.03787E-01 8.17887E-05

11 1.24162E-03 3.70295E-01 3.32032E-05

12 5.12453E-04 3.39582E-01 1.34793E-05

13 9.63842E-04 3.11415E-01 5.47210E-06

14 3.38842E-04 2.85585E-01 2.22147E-06

15 1.65231E-04 2.61897E-01 9.01838E-07

16 -7.78244E-05 2.40175E-01 3.66113E-07

17 2.69398E-04 2.20253E-01 1.48629E-07

18 1.99953E-04 2.01985E-01 6.03378E-08

Parameter values and std uncertatinties obtained by UR:

1 2.83190E-03 3.55440E-04

3 1.45234E-02 2.01819E-03

Chisqr= 1.23143363

Für den Import in R werden zusätzlich jeweils die Kovarianz-Matrix und die Eingabedaten in separate Dateien geschrieben:

Ergebnisgröße: covmat1.txt und data1.txt

Erkennungsgrenze: covmat2.txt und data2.txt

Mit Hilfe dieser Dateien kann eine statistische Auswertung mit R erfolgen:

(load package MASS) (R)

Für die Ergebnisgröße (obige Daten) erhält man mit R:

> covmat <- read.table("covmat1.txt")

> data <- read.table("data1.txt")

> res <- lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

> summary.lm(res)

Call:

lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

Residuals:

Min 1Q Median 3Q Max

-8.076e-04 -4.422e-04 -3.702e-04 -3.134e-05 5.747e-04

Coefficients:

Estimate Std. Error t value Pr(>|t|)

X1 2.832e-03 1.643e-07 17231 <2e-16 \*\*\*

X3 1.452e-02 9.332e-07 15564 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0004624 on 16 degrees of freedom

Multiple R-squared: 0.9625, Adjusted R-squared: 0.9578

F-statistic: 205.1 on 2 and 16 DF, p-value: 3.95e-12

Warning message:

In summary.lm(res) : calling summary.lm(<fake-lm-object>) ...

Um die Unsicherheit mit derjenigen aus UR vergleichen zu können, teilt man die Unsicherheit aus R (Std. error) durch den Wert Residual standard error.

Man erhält aus R für die Unsicherheit des zu X1 gehörenden Fitparameters:

1.643E-07 / 0.0004624 = 3.5532E-04

**Fall der Erkennungsgrenze:**

Case: Decision threshold

Blank count rate= 4.66670009E-08 background rate= 1.88333332E-03

Input data: variance-covariance matrix: (rank= 18 )

2.29269E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 1.47460E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 1.14249E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 1.00767E-07 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 9.52931E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 9.30711E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 9.21690E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

Arrays y, X1, x2, X3:

y X1 X3

1 3.96624E-03 8.80220E-01 2.73093E-01

2 1.61015E-03 8.07210E-01 1.10866E-01

3 6.53662E-04 7.40257E-01 4.50075E-02

4 2.65363E-04 6.78857E-01 1.82714E-02

5 1.07728E-04 6.22550E-01 7.41751E-03

6 4.37335E-05 5.70913E-01 3.01124E-03

7 1.77542E-05 5.23559E-01 1.22245E-03

8 7.20756E-06 4.80133E-01 4.96272E-04

9 2.94955E-06 4.40700E-01 2.03089E-04

10 1.18786E-06 4.03787E-01 8.17887E-05

11 4.82232E-07 3.70295E-01 3.32032E-05

12 1.95772E-07 3.39582E-01 1.34793E-05

13 7.94798E-08 3.11415E-01 5.47210E-06

14 3.22691E-08 2.85585E-01 2.22147E-06

15 1.31030E-08 2.61897E-01 9.01838E-07

16 5.32199E-09 2.40175E-01 3.66113E-07

17 2.16298E-09 2.20253E-01 1.48629E-07

18 8.80329E-10 2.01985E-01 6.03378E-08

Parameter values and std uncertatinties obtained by UR:

1 1.98944E-11 3.10543E-04

3 1.45234E-02 1.73864E-03

Chisqr= 8.88178420E-16

Für den Fall der Erkennungsgrenze erhält man mit R

(hier kommt es auf die Unsicherheit des zu X1 gehörenden Parameters an, dessen Wert nahe bei null sein sollte):

> covmat <- read.table("covmat2.txt")

> data <- read.table("data2.txt")

> res <- lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

> summary.lm(res)

Call:

lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

Residuals:

Min 1Q Median 3Q Max

-1.370e-09 -5.200e-13 -4.300e-13 3.721e-11 8.881e-10

Coefficients:

Estimate Std. Error t value Pr(>|t|)

X1 2.193e-11 1.334e-13 1.644e+02 <2e-16 \*\*\*

X3 1.452e-02 7.467e-13 1.945e+10 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.295e-10 on 16 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: 1

F-statistic: 5.105e+13 on 2 and 16 DF, p-value: < 2.2e-16

Warning message:

In summary.lm(res) : calling summary.lm(<fake-lm-object>) ...

Man erhält aus R für die Unsicherheit des zu X1 gehörende Fitparameters

1.334E-13 / 4.295E-10 = 3.10594E-04

### Hinweis zur Berechnung von Erkennungs- und Nachweisgrenze:

Diese Kenngrößen beziehen sich auf die Komponente des Ergebnisvektors und ihre Unsicherheit , die solange iteriert wird, bis und die Iterationsbedingung erfüllen. wird durch einen aus der Iterationsschleife vorgegeben neuen Wert ersetzt. Damit erhält man gemäß dem Auswertemodell , wonach sich eine geänderte Kovarianzmatrix bestimmen lässt. Dazu werden zunächst den **Bruttozählraten** der Abklingkurve folgende neue Werte zugewiesen:

Daraus folgt die **Unsicherheits-Funktion (Standardunsicherheit) der Bruttozählrate:**

**.**

Die Nettozählraten der modifizierten Abklingkurve lauten:

,

woraus sich die Diagonalelemente der geänderten Kovarianzmatrix (Varianzen der Nettozählraten) ergeben:

,

während die nicht-diagonalen Elemente unverändert bleiben.

Aus der rechten Formel bei obiger Gl. (5.63) wird die Unsicherheit bestimmt. Mit dem Wertepaar und kann der nächste Iterationsschritt ausgeführt und dieser Rechengang wiederholt werden, falls die Bedingung für die Konvergenz noch nicht erfüllt ist.

## 7.5 Hinweis zu Erkennungs- und Nachweisgrenzen bei linearer Entfaltung

Für die in der Least Squares-Anpassung verwendete Gleichung

ist zunächst festzulegen, welcher der drei Fitparamater der aktuell eingestellten Ergebnisgröße entspricht. Für die folgenden Erläuterungen sei angenommen, dass der Parameter der aktuellen Ergebnisgröße entspricht.

Das Vorgehen zur Berechnung der Erkennungs- und Nachweisgrenze richtet sich nach DIN EN ISO 11929-1:2021. Dazu wird iteriert. Zu diesem Zweck wird in der obigen Gleichung zunächst (der Parameter für die gesuchte Nettozählrate, z.B. im Falle einer Y‑90-Abklingkurve) durch einen iterierten Wert ersetzt, alle anderen Werte/Parameter bleiben unverändert. Damit werden neue Messwerte berechnet sowie auch deren Unsicherheiten, die eingangs „gemessene“ Unsicherheiten waren, durch berechnete Unsicherheiten (Unsicherheits-Funktion) ersetzt. Mit diesen geänderten Werten, d.h. , wird die Rechnung der Least squares-Analyse wiederholt, und man erhält für den iterierten Wert des Parameters seine Unsicherheit . Mit diesem Wertepaar wird geprüft, ob die Konvergenzbedingung für die Iteration schon erfüllt ist; falls nicht, wird der nächste iterierte Wert bestimmt und die obige Rechnung wiederholt, um dessen Unsicherheit zu erhalten; usw.

Eine detailliertere Beschreibung der Rechnungen während der Iteration findet sich am Ende der folgenden Hilfe-Themen:

1. [**lineare Kurvenanpassung mit WLS**](#URH_LLSQ_MATHE_DE),
2. [**lineare Kurvenanpassung mit WTLS**](#URH_GLSQ_MATHE_DE)

## 7.6 Hinweise zur linearen Kurvenanpassung mit dem „allgemeinen Fall“ der Least squares-Anpassung (WTLS)

Der sogenannte „allgemeine Fall“ der Methode der kleinsten Quadrate (WTLS) wird in UncertRadio dann verwendet, wenn die -Werte des Modells eigene Unsicherheiten und ggf. auch Kovarianzen aufweisen. Es ist das einzige dem Autor bekannte und zugängliche Verfahren, das die Einbeziehung von Unsicherheiten und Kovarianzen der -Werte erlaubt.

Zur Durchführung der Berechnung werden Matrix-Routinen aus der Datan-Library verwendet (in Fortran 90 konvertiert), die Haupt-Subroutine des Verfahrens ist LSQGEN:

Datan-Library aus:

Siegmund Brandt: Datenanalyse. Mit statistischen Methoden und Computerprogrammen; 4. Auflage. Spektrum, Akademischer Verlag, Heidelberg-Berlin, 1999.

Obwohl das Modell weiterhin linear in den anzupassenden Parametern ist, erfordert das WTLS-Verfahren ein iteratives Vorgehen zur Lösung. Daher benötigt es Startwerte der anzupassenden Parameter, die aus einem vorgeschalteten Aufruf der einfacheren LSQ-Routinen erhalten werden.

Leider ist die in der Routine LSQGEN verwendete Mathematik sehr kompliziert, so dass auf eine Wiedergabe an dieser Stelle gegenwärtig verzichtet wird. Dieser Routine wird in UncertRadio eine weitere Routine vorgeschaltet, die die Eingangsdaten in einer speziellen Form zusammenstellt, wie sie von LSQGEN erwartet wird. Dabei werden die gemessen Nettozählraten und die verschiedenen -Werte zu einem gemeinsamen Vektor zusammengefasst, ebenso die Unsicherheiten und Kovarianzen, sowohl der - als auch der -Werte, zu einer gemeinsamen Kovarianzmatrix cy zusammengefasst. Der Rang des Vektors und der Matrix cy kann schnell sehr groß werden. Bearbeitet man ein Problem aus der LSC-Messung mit 3 Zählkanälen und macht insgesamt 10 Messungen, erhält man:

3x10 = 30 Werte (je 10 Werte für jeden Kanal A, B und C)

3x3x10 = 90 Werte (zu jedem der 30 Werte gehören 3 Werte )

Somit beträgt der gesuchte Rang 30+90=120. Dies bedeutet, dass die Eingangs-Kovarianzmatrix eine 120x120-Matrix ist, wodurch sofort plausibel wird, dass die iterativ arbeitende Routine LSQGEN durchaus zeitaufwändig werden kann.

**Hinweis zur Berechnung von Erkennungs- und Nachweisgrenze**:

Wie bei der Beschreibung der LSQ-Mathematik soll der Ergebnisvektor der anzupassenden Parameter mit bezeichnet werden.

Die gesuchten Kenngrößen beziehen sich auf eine Komponente L des Ergebnisvektors der anzupassenden Parameter, , und ihre Unsicherheit , die so lange iteriert wird, bis und die Iterationsbedingung erfüllen. wird dazu zunächst durch einen aus der Iterationsschleife vorgegeben neuen Wert ersetzt. Damit werden zunächst den **Bruttozählraten** der Abklingkurve folgende neue Werte zugewiesen, wobei ohne Beschränkung der Allgemeinheit für das folgende der Index L=1 ausgewählt wurde:

.

Daraus folgt die **Unsicherheits-Funktion (Standardunsicherheit) der Bruttozählrate:**

**.**

Die Nettozählraten der modifizierten Abklingkurve, d.h. die , lauten:

,

woraus sich die Diagonalelemente der geänderten Kovarianzmatrix ergeben:

,

während nicht-diagonale Elemente unverändert bleiben.

Mit diesen modifizierten Werten und Unsicherheiten der Nettozählraten wird die Auswertung wiederholt, woraus sich dann die gesuchte Unsicherheit ergibt. Mit dem Wertepaar und wird das Konvergenz-Kriterium geprüft; ist dieses noch nicht erfüllt, kann der nächste Iterationsschritt, d.h. der hier angedeutete Rechengang, wiederholt werden, wozu zunächst der nächste Wert aus dem Wertepaar und ermittelt wird.

## 7.7 In Tabellen Zeilen entfernen, Arbeiten mit Spaltenblöcken

Hat man mit dem Programm nach der Erstellung der Gleichungen schon eine (nahezu) fertige Symbollisten-Tabelle erstellt, kann ein **nachträgliches Verändern der Gleichungen, d.h. Symbole kommen neu hinzu oder können ganz wegfallen,** zu Verschiebungen innerhalb dieser beiden Tabellen führen.

Nachträglich **neu hinzugekommene Symbole** tauchen in der Symbollisten-Tabelle als **grün** hinterlegte Zeilen auf. Für diese sind die Felder Einheit und Bedeutung auszufüllen.

Nach Bearbeiten der Gleichungen können bestimmte **Symbole wegfallen**, die offenbar nicht mehr verwendet werden; die entsprechenden Symbole-Zeilen werden dazu an das Ende der Tabelle verlagert und **gelb** hinterlegt. Hier muss der Benutzer entscheiden, ob diese Zeile(n) gelöscht werden sollen; wenn ja, könne diese mit dem Toolbar Icon DeleteRow.bmp entfernt werden.

**Spaltenblöcke** lassen sich derzeit in den Tabellen nicht definieren, es können also keine Datenblöcke aus UncertRadio heraus exportiert werden. Wohl aber ist es möglich, solche Spaltenblöcke, die z.B. in Excel oder Notepad++ markiert und in die Windows-Zwischenablage kopiert wurden, **in eine UR-Tabelle importieren**. Dazu sind folgende Schritte nötig:

* öffnen der linken oberen Zelle im Ziel-Block der UR-Tabelle mit Maus-Doppel-Klick;
* einfügen des Spaltenblocks mit dem Maus-Kontextmenü „Einfügen“ in diese Zelle; der gesamte Inhalt des Blocks befindet sich noch in dieser Zelle.
* einmaliges Drücken der Eingabetaste verteilt den Inhalt des eingefügten Blocks auf die einzelnen Zellen des Zielblocks.

## 7.8 Textfenster für die Gleichungen

Die Gleichungen können hier zeilenweise eingetragen werden können. Ein besonderes Zeilenende-Symbol ist nicht erforderlich; lediglich im Falle, dass eine Gleichung über mehrere **Zeilen fortgesetzt** werden soll, ist am Ende einer fortzusetzenden Zeile das „&“-Zeichen zu schreiben.

Der **Aufbau** der Gleichungen hat **hierarchisch** zu erfolgen.

**Man beginnt mit der Grundgleichung,** welche die **Ergebnisgröße** ***y*** definiert. Diese kann z.B. lauten:

*y = w \* Rn - Ai*

Man kann hierbei selbstverständlich ein anderes Symbol für die Ergebnisgröße verwenden. In den darauffolgenden Zeilen werden die auf der rechten Seite verwendeten Symbole - soweit sie nicht schon primäre Eingangsgrößen darstellen - durch weitere Gleichungen (**Hilfsgleichungen**) definiert, z.B.:

*Rn = Rb - R0* Nettozählrate

*w = 1. / (eps \* eta \* m) \* f1* verfahrensbezogener Kalibrierfaktor

*f1 = exp(+log(2.) \* t1 / tr)* inverser Abklingfaktor

Hierbei ist *Ai* ein abzuziehender Interferenzbeitrag zur Aktivität, d.h. *Ai* ist gleich der programmintern ermittelten Konstanten *FC*. Der Faktor *w* entspricht dem programmintern ermittelten Faktor *FL.*

Anmerkungen: a) Falls in dem Projekt mehr als eine Ergebnisgröße definiert wurde, z.B. drei, gibt es für jede Ergebnisgröße eine Grundgleichung; dies müssen dann die ersten drei Gleichungen sein. b) Unter *Rn* ist in den Fällen, bei denen eine Störung durch ein anderes Radionuklid vorliegt, die „**verfahrensbezogene Nettozählrate**“ zu verstehen, deren Gleichung einen dafür berechneten Interferenzterm enthält. *Bei vorliegendem Interferenzbeitrag darf für die Nettozählrate Rn der einfache Ausdruck Rn = Rb - R0 verwendet werden; der in der programmintern ermittelten Hilfsgröße FC enthaltene Interferenzbeitrag wird ab jetzt automatisch berücksichtigt (*[*siehe auch*](#URH_VERFAHREN_DE)*).*

Wegen der hierarchischen Struktur werden die **Gleichungen** vom Programm **zur Berechnung von Werten von unten nach oben abgearbeitet**. Dies bedeutet, dass in einer bestimmten Gleichung nur Symbole für Hilfsgleichungen verwendet werden dürfen, die unterhalb dieser definiert werden. Das Programm testet intern, ob diese Regel eingehalten wird; falls nicht, erhält der Benutzer eine entsprechende Warnung.

**Es ist erforderlich**, wie oben gezeigt eine **eigene Gleichung für die Definition der Netto-Zählrate** *Rn* zu verwenden. In der wichtigen Gleichung für die Nettozählrate *Rn* darf die Bruttozählrate auch mit einem Faktor multipliziert auftreten, was in ganz seltenen Anwendungen vorkommen kann. Dieser Faktor, wenn er denn ungleich eins ist, wird programmintern ermittelt; er spielt lediglich bei der Erkennungs- und Nachweisgrenzen-Berechnung eine Rolle.

Anmerkung: Die Bruttozählrate muss als Symbol in der Gleichung für die Nettozählrate direkt auftauchen, oder eine andere darin auftauchende Größe verweist auf eine weitere Hilfsgleichung, in der sie dann direkt aufgeführt sein muss.

Beispiel: *Rn = Rn1 - Rblank*; *Rn1 = Rb - R0*.

Der **Faktor *w*** enthält auch den inversen Abklingfaktor *f1* für die Korrektion des radioaktiven Zerfalls des Nuklids r mit der Halbwertszeit *tr* in der Zeitdauer *t1* zwischen Probeentnahme und Messbeginn. Mit *eps*, *eta* und *m* z.B. werden hier die Zählausbeute, die chemische Ausbeute und die Masse bezeichnet.

Die auf den linken Seiten der Gleichungen auftretenden Symbole werden als "**abhängige (a)**", diejenigen der rechten Seiten, soweit sie nicht auch auf einer linken Gleichungsseite auftreten, als "**unabhängige (u)**" **Messgrößen (Eingangsgrößen)** eingestuft.

Es können ausgiebig Hilfsgleichungen benutzt werden. **Es kann jedoch nicht passieren,** wie es sonst bei dem herkömmlichen Vorgehen leicht möglich wäre, **dass Korrelationen zwischen den eingeführten Hilfsgrößen auftreten**, wenn die den Hilfsgrößen zugeordneten Unsicherheiten auch zur Unsicherheitsfortpflanzung benutzt würden.

Die **Syntax für die Schreibweise der Formelsymbole** soll sich an die in Programmiersprachen üblichen Schreibweisen für Variable halten. Hier wird programmintern mit FORTRAN 90 gearbeitet. Zwischen Groß- und Kleinschreibung wird intern nicht unterschieden, es wird aber - der besseren Lesbarkeit wegen - empfohlen, Groß- und Kleinschreibung zu verwenden. Der Unterstrich (underscore) darf in den Symbolen verwendet werden, nur nicht als erstes Zeichen. Ein Formelsymbol soll immer mit einem Buchstaben beginnen.

Für Zahlen in den Gleichungen (auch in den Tabellen) ist **immer** der **Dezimalpunkt** zu verwenden. In den Gleichungen verwendete Zahlen, z.B. 1. bzw. 2. in den obigen Gleichungen für *Fakt* bzw. *f1* werden vom Function parser immer als doppelt genaue Zahlen interpretiert.

**Interne Funktionen und Operatoren:**

Die programminternen Berechnungen werden in "double precision" Arithmetik ausgeführt.

Folgende *intrinsische* arithmetische Funktionen können, ähnlich - aber nicht identisch - wie z.B. in MS Excel, benutzt werden:

sqrt(x) Wurzelfunktion

exp(x) natürliche Exponentialfunktion

log(x), ln(x) natürlicher Logarithmus

log10(x) dekadischer Logarithmus

Es kann auch die neue **Funktion fd**() mit drei Parametern verwendet werden, die einen über die Messdauer tm gemittelten Abklingfaktor darstellt:

fd(tA,tm,xlam) = exp(-xlam\*tA) \* (1 - exp(-xlam\*tm)) / (xlam\*tm)

Diese Funktion gab es in UR1 noch nicht.

Bei bestimmten Projekten ist es erforderlich, auch die Unsicherheit u(x) einer Eingangsgröße x als Wert zu verwenden. Dazu wurde die **Funktion uval(x)** durch Erweiterung des Function parsers eingeführt. Will man z.B. die relative Unsicherheit verwenden, kann dies durch folgende Gleichung erfolgen:

urelw = uval(w) / w

Die Funktion uval() darf nur das Symbol einer in der Symboltabelle definierten Größe enthalten, jedoch keinen aus mehr als einem Symbol bestehenden arithmetischen Ausdruck; im letzteren Fall würde sonst uval(a+b) automatisch eine Unsicherheitsfortpflanzung implizieren, was diese Funktion uval jedoch nicht leisten kann. Soll der mit uval(x) verwendete Wert konstant sein, darf für x nicht die Bruttozählrate oder die Bruttoimpulsanzahl verwendet werden, da die Werte und Unsicherheiten der letzteren bei der Berechnung von Erkennungs- und Nachweisgrenze variieren.

Neben den üblichen Operatoren +, -, \* und / wird für das Potenzieren noch \*\* verwendet: a\*\*b heißt a hoch b, wofür man hier auch a^b schreiben darf, wie in Excel oder R.

**Hinweise:**

Es ist bereits ein Verfahren in das Programm integriert, das es gestattet, **die Nettozählrate als Ergebnis der Ausgleichsrechnung mit Hilfe der gewichteten multilinearen Least squares-Anpassung einer gemessenen Abklingkurve** zu erhalten. Dies wurde bisher für die komplexere Auswertung von Y-90-Abklingkurven erstellt, lässt sich aber z.B. auch auf die Auswertung von Y-90-Aufbaukurven in Sr-89/Sr-90-Präparaten bei der Sr-89/Sr-90-Bestimmung verwenden.

*Dieses Tool hat noch nicht seinen endgültigen Stand erreicht. Hierzu ist es nötig, noch weitere Fallanwendungen dafür zu sammeln; für entsprechende Vorschläge ist der Autor dankbar!*

Weiteres dazu: [**Verwendung der Least squares-Anpassung**](#URH_LSQ_DE)

Für den **Bereich der Gammaspektrometrie** ist ein Verfahren integriert, das es gestattet, **für ein Radionuklid mit mehreren Gammalinien die Aktivität des Radionuklids als ein Mittelwert der Aktivitäten der einzelnen Gammalinien zu bestimmen.** Für die Mittelwertberechnung werden zwei Methoden angeboten.

Die erste Methode ist der **gewichtete Mittelwert**, verbunden mit seinen sogenannten „internen“ und „externen“ Standardabweichungen. Wenn die Werte der beiden Standardabweichungen etwa gleich groß sind, folgert man, dass die Aktivitätswerte der einzelnen Linien sich in statistischer Kontrolle befinden. Dies ist ein eingeführtes Verfahren, es wird aber darauf hingewiesen, dass die Verwendung der „externen“ Standardabweichung nicht Bayes-konform ist.

Die zweite Methode verwendet anstelle von Formeln für diesen Mittelwert direkt das Matrix-basierte **least squares-Verfahren**. Für die Einbeziehung von Kovarianzen ist dieses besser geeignet.

Voraussetzung für dieses Verfahren ist, dass die für die Bestimmung der Aktivität des Radionuklids verwendeten Gammalinien nicht von Linien anderer Radionuklide gestört werden.

Weiteres dazu: [**Aktivitätsbestimmung mit mehreren Gammalinien**](#URH_GSPK1_DE)

## 7.9 Bearbeiten der Symbolliste

Nach Betätigung des Buttons "Symbole aus Gleichungen laden" wird die Tabelle mit der Symbolliste bedienbar.

Danach wird intern zunächst mit dem Function Parser die **korrekte Syntax der Gleichungen** geprüft. Im Fehlerfalle gibt es entsprechende Windows-„MessageBox“-Meldungen und man muss danach die Gleichungen selbst überprüfen.

Die Symbolliste-Tabelle besteht aus den 4 Text-Spalten "**Symbol**", "**Typ**", "**Einheit**" und "**Bedeutung**".

Die automatisch aus den Gleichungen extrahierten Symbole werden in der Spalte "Typ" eingeteilt in die abhängigen (a) und unabhängigen Größen (u), die abhängigen werden zuerst aufgeführt. **Nur die unabhängigen Messgrößen sind diejenigen, deren Messunsicherheiten im Programm berücksichtigt werden. Damit werden von vornherein mögliche Kovarianzen zwischen solchen als abhängig eingestuften Größen vermieden, die gemeinsame unabhängige Größen enthalten.**

In der Symbolliste können solche Eingangsgrößen, für die Mittelwert und Standardabweichung aus einem noch einzugebenden Datensatz berechnet werden sollen, in der Spalte „Typ“ mit „m“ anstelle von „a“ (abhängige Größe) oder „u“ (unabhängige Größe) gekennzeichnet. Des Weiteren wird eine mit einer Gleichung zu berechnende Größe, die nur als Parameter ohne Unsicherheit verwendet werden soll, mit einem „p“ gekennzeichnet.

Die Spalten "Einheit" und "Bedeutung" sollen vom Benutzer ausgefüllt werden. Man kann dies jedoch auf einen späteren Zeitpunkt verschieben. Dies empfiehlt sich besonders dann, wenn man sich noch in der Probierphase für die zweckmäßigste Aufstellung der Gleichungen befindet, sich die Reihenfolge von Symbolen in der Symbolliste also noch ändern kann.

Bei einer späteren Überarbeitung der Gleichungen kann es dazu kommen, dass vorher bestehende Symbolnamen nachher nicht mehr in den Gleichungen vorkommen; solche Symbole werden automatisch an das Ende der Symbolliste gesetzt, wo sie nach manueller Prüfung zeilenweise gelöscht werden können (siehe unten: Weiteres zur Bearbeitung von Tabellen).

Ein Mausklick innerhalb der Tabelle erlaubt das **Scrollen mit dem Mausrad**.

Es kann vorkommen, dass man den Namen eines bestimmten in den Gleichungen verwendeten Symbols ändern möchte. Wenn man diesen Namen in den Gleichungen selbst austauscht, geht die Beziehungen zu bereits für diese Variable eingegebenen Werte sowie zu Einträgen bei Einheit und Bedeutung verloren. Mit dem neuen [Menüpunkt „Bearbeiten – Symbolname ändern“](#_Programmstart,_Hauptmenü_und) wird dies erheblich vereinfacht und sicherer.

**Ganz wichtig:**

Es ist meistens erforderlich, zusätzlich zu den aus Gleichungen automatisch extrahierten Symbolen noch weitere in der Symbolliste-Tabelle am Ende zu ergänzen, die dort nicht explizit vorkommen. Dazu gehören oft Messzeiten für die Brutto- und die Nulleffektmessung, z.B. tm und t0. Diese werden später zur formelmäßigen Berechnung von Unsicherheiten von Brutto- und Nulleffektzählrate benötigt, z.B.: sqrt(Rb / tm) oder sqrt(R0 / t0).

**Weiteres zur** [**Bearbeitung von Tabellen**](#URH_TABTRICKS_DE)

## 7.10 Dialog "Eingabe der Abklingkurve"

Der Aufbau des Dialogs ist im folgenden Bild gezeigt.

|  |
| --- |
|  |

In diesem Dialog ist zunächst **Datum und Uhrzeit der Y-90/Sr-90-Separation** einzugeben. Das Datumsfeld wird wie folgt benutzt: „12.06.2006 09:05:00“. Das Jahr kann zwar auch 2-stellig eingeben werden, es wird aber die Benutzung der 4-stelligen Jahreszahl empfohlen. Das Programm ersetzt intern im Datumsfeld alle Interpunktionszeichen durch ein Leerzeichen und liest dann die 6 Datumselemente formatfrei ein.

Für die **Basiseinheit für Zeitangaben von Messdauern und Zählraten** kann zwischen s und min gewählt werden. Intern erfolgt bei der Wahl von min eine Umrechnung auf die Basiseinheit s!

Der in diesem Dialog nur angezeigte (nicht editierbare) Wert der **Netto-Blindwert-Zählrate** ist der Wert des Symbols *Rbl*, der vom Benutzer in der Symbolliste im TAB „Werte, Unsicherheiten“ eingeben wurde. Der für die Messung von *Rbl* gültige Wert der Nulleffektzählrate darf darin nicht enthalten sein.

Für die **Eingabe der Einzelmesswerte der Abklingkurve** steht eine 11-spaltige Tabelle zur Verfügung, deren Spalten folgende Bedeutung haben; dabei sind nur die hier in schwarzer Schrift gezeigten Spalten vom Benutzer direkt mit Daten zu versehen, die in roter Schrift werden danach vom Programm berechnet (s. unten).

|  |  |  |
| --- | --- | --- |
| Spalte-# | Name | Bedeutung |
| *Messungen der Bruttozählrate:* | | |
| 1 | Start-Datum | Datum und Uhrzeit des Starts der k-ten Messung (Eingabeformat wie oben angegeben), oder Zeitdifferenz zwischen Y-90/Sr-90-Separation und dem Start der k-ten Messung (als Zahl mit Dezimalpunkt); das Programm erkennt selbst, welche der beiden Eingabemöglichkeiten verwendet wird; |
| 2 | Messdauer | Messdauer der k-ten Messung |
| 3 | Impulse | Brutto-Impulsanzahl der k-ten Messung |
| 4 | ZählRate | berechnete Brutto-Zählrate |
| 5 | u(ZählRate) | Unsicherheit der Brutto-Zählrate |
| *Messungen der Nulleffektzählrate:* | | |
| 6 | Messdauer | Messdauer der k-ten Nulleffekt-Messung |
| 7 | Impulse | Impulsanzahl der k-ten Nulleffekt-Messung |
| 8 | ZählRate | berechnete Nulleffekt-Zählrate |
| 9 | u(ZählRate) | Unsicherheit der Nulleffekt-Zählrate |

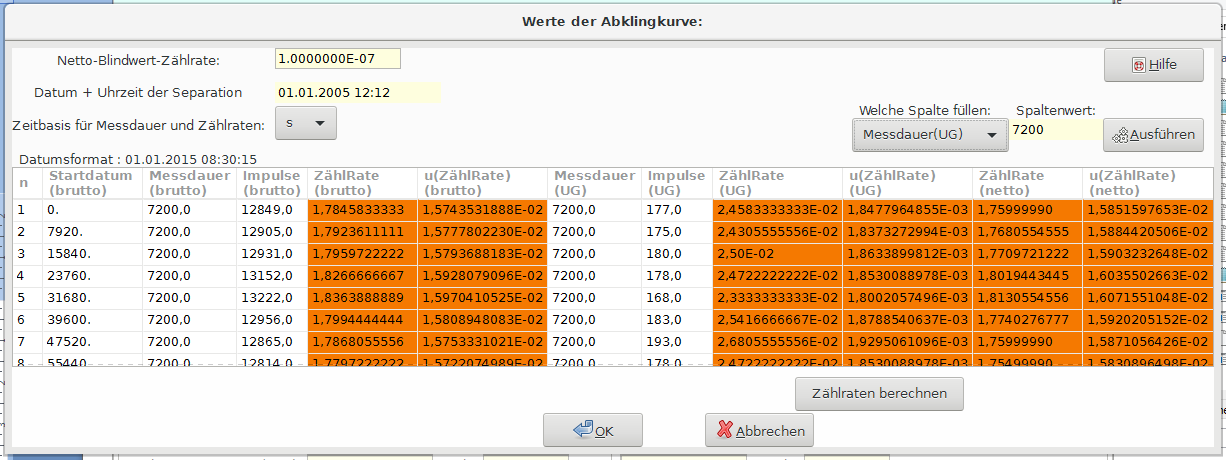
|  |
| --- |
| *Ergebniswerte der Nettozählrate:* |

|  |  |  |
| --- | --- | --- |
| 10 | NettoZählRate | berechnete Netto-Zählrate |
| 11 | u(Netto-ZR) | Unsicherheit der Netto-Zählrate |

Mit Hilfe des **Buttons „Zählraten berechnen“** werden in den Spalten 4/5, 8/9 und 10/11 (im rot gekennzeichneten Bereich) die Werte der Brutto-Zählrate, Nulleffekt-Zählrate und Nettozählrate berechnet und eingetragen. Dabei wird der Wert der Netto-Blindwert-Zählrate *Rbl* berücksichtigt (siehe oben).

**Vereinfachte Eingaben in die Spalten 2, 6 und 7:**

Hierfür steht jetzt eine Option innerhalb dieses Dialogs zur Verfügung, mit dem innerhalb einer dieser Spalten ein fester Wert in alle betreffenden Zellen dieser Spalte übertragen werden kann.



**Tipp zur Erleichterung der Eingabe von Werten in die ersten drei Spalten:**

**Import aus ASCII-Datei:** Liegen die in den ersten drei Spalten einzutragenden Werte außerhalb des Programms in einer ASCII-Datei vor, können die entsprechenden Spalten dort markiert, in die Windowsablage kopiert und dann im UncertRadio-Dialog per Einfügen (rechte Maustaste: „paste“) in (nur) eine Spalte eingefügt werden. Dabei braucht der Mauszeiger im UncertRadio-Dialog zuvor nur in die oberste zu füllende Zelle des Spaltenbereichs gesetzt zu werden.

**Import aus Exceldatei:** Liegen die gewünschten Daten bereits in einer Exceldatei vor, können sie von dort - wie schon für Import aus ASCII-Datei beschrieben - per Maus „copy and paste“ über die Windowszwischenablage in den Zellenbereich der Tabelle im UncertRadio-Dialog übertragen werden. Hier können auch die drei Spalten gleichzeitig übertragen werden.

Weiteres zur [**Bearbeitung von Tabellen**](#URH_TABTRICKS_DE)

## 7.11 Dialog Festlegung des Modells der Abklingkurve

### 7.11.1 Eine einzige Ergebnisgröße vorhanden

Der Aufbau des Dialogs ist der folgenden Abbildung zu entnehmen.

|  |
| --- |
|  |

In diesem Dialog wird das [**beschriebene multilineare Least squares-Verfahren**](#URH_LSQ_DE) für den Fall der Analyse von komplexeren Abklingkurven (Nettozählraten) dargestellt.

Die drei Terme beschreiben Beiträge von Radionuklid-Abklingfaktoren zu den Nettozählraten und hängen von der Zeit für die *k*-te Messung und den Messdauern ab.

Für den Fall einer Y-90-Abklingkurve ist die Bedeutung der Terme wie folgt:

|  |  |
| --- | --- |
|  | beschreibt das Abklingen der reinen Y-90-Komponente; (in ) ist die anzupassende Y-90-Nettozählrate zum Zeitpunkt der Y/Sr-Abtrennung; |
|  | beschreibt den (praktisch) konstanten Zählratenbeitrag einer Verunreinigung mit einem langlebigen unbekannten Radionuklid (Halbwertszeit gleich unendlich angenommen, wird dann gleich 1 gesetzt). Als Beispiel kann hier Th-234 genannt werden; wenn dieses vorliegt, kann auch dessen Halbwertszeit Hwzlong (im TAB „Werte, Unsicherheiten“) spezifiziert werden; (in ) ist die anzupassende Amplitude dieser Komponente; |
| *:* | beschreibt das Abklingen des relativ kurzlebigen Ac-228, das als Verunreinigung im Y-Oxalatpräparat auftreten kann; dieser Term kann auch verwendet werden, wenn der Verdacht auf kürzerlebige Verunreinigungen durch Radon-Folgeprodukte besteht; (in ) ist die anzupassende Amplitude dieser Komponente; |

Der Benutzer kann auswählen, ob der 2. und/oder 3. Term bei der Anpassung verwendet werden soll.

Weiterhin kann er wählen, ob eine mit den inversen Varianzen der Nettozählraten gewichtete Anpassung (Fit) durchgeführt werden soll - oder ungewichtet. Die interne Verwendung der Kovarianzen der gemessenen Nettozählraten kann (zum Testen) auch abgewählt werden.

Eine Auswahlliste in diesem Dialog erlaubt es, das Anpassungsverfahren aus vier Varianten auszuwählen: WLS, PLSQ, PMLE und WTLS. Das komplexere und rechenintensivere Total least squares-Verfahren (WTLS) ist in der Lage, Unsicherheiten der -Werte und Kovarianzen zwischen ihnen direkt zu berücksichtigen. [Siehe auch: Chi-Quadrat-Optionen](#_Chi-Quadrat-Optionen)

In den meisten Fällen unterscheiden sich die Term-Funktionen Xi(t) zwischen einzelnen Messungen nur in der verwendeten Zeitdifferenz zur Sr-/Y-Abtrennung (Parameter t: tstart). Daher brauchen die Funktionen Xi(t) nur für die 1. Messung (bei mehreren Messkanälen: für die 1. Messung jeden Messkanals) aufgeführt werden.

Wenn jedoch Nachweiswahrscheinlichkeiten in den Xi(t) enthalten sind, die sich von Messung zu Messung in ihren Werten unterscheiden, kann mit Hilfe einer Checkbox im Dialog der Modus ausgewählt werden, dass die Term-Funktionen für alle Messungen explizit vorgegeben werden. Dadurch verlängert sich die Argumentliste des Linfit-Calls um weitere Werte von Nachweiswahrscheinlichkeiten; **seit Version 2.4.24** kann die Eingabe dieser Nachweiswahrscheinlichkeits-Parameter im Linfit-Aufruf entfallen. Siehe dazu das Beispiel-Projekt Sr89-Sr90\_IAEA\_AQ-27\_2013\_V2\_EN.txp.

In einem Textfeld des Dialogs können die Gleichungen für die drei Funktionen angegeben werden. Für das oben verwendete Beispiel der Analyse einer Y-90-Abklingkurve wird folgendes eingegeben:

X1 = (1. - exp(-log(2.)\*tmess/HwzY90)) / (log(2.)\*tmess/HwzY90) \* exp(-log(2.)\*tstart/HwzY90)

X2 = (1. - exp(-log(2.)\*tmess/Hwzlong)) / (log(2.)\*tmess/Hwzlong) \* exp(-log(2.)\*tstart/Hwzlong)

X3 = (1. - exp(-log(2.)\*tmess/HwzAc228)) / (log(2.)\*tmess/HwzAc228) \* exp(-log(2.)\*tstart/HwzAc228)

Hierin sind tmess die Messdauer(n) und tstart die Zeitdauer(n) zwischen Y-90/Sr-90-Trennung und dem Start der Messung(en). Beide Variablen stellen tatsächlich Felder dar, so dass Werte dafür in einem dafür ausgelegten Windows-Dialog für jede Einzelmessung eingegeben werden können. Hwzxxx sind die Radionuklid-bezogenen Halbwertszeiten. **Die Formeln berücksichtigen auch den radioaktiven Zerfall während der Messungen.**

Unter Verwendung der neuen Abklingfunktion fd() verkürzen sich die Gleichungen:

X1 = fd(tstart, tmess, log(2)/HwzY90)

X2 = fd(tstart, tmess, log(2)/Hwzlong)

X3 = fd(tstart, tmess, log(2)/HwzAc228)

### 7.11.2 Erweiterung auf zwei oder drei Ergebnisgrößen

Bei der simultanen Messung von z.B. Sr-90, Sr-89 und ggf. auch Sr-85, mittels eines LSC werden die Beiträge dieser Radionuklide zu den Zählraten in zwei oder drei Zählkanälen, hier als A, B und C bezeichnet, bestimmt. Nähere Einzelheiten können dem Report AKU (2008) entnommen werden, der sich mit modernen Methoden der Sr-89/Sr-90-Bestimmung befasst.

Die Anzahl der Zählkanäle (nchs) kann im Dialog gewählt werden. In diesem Fall liegen bis zu drei gemessene Abklingkurven , und vor. Damit erhält man bis zu drei Modell-Gleichungen anstatt nur einer:

In diesem Falle stellen die Fitparameter Aktivitäten dar, nicht Zählraten wie in dem einfachen Beispiel ganz oben. Diese Parameter werden vom Programm selbständig als neue Symbole FITP1, FITP2 und FITP3 in die Liste der Symbole eingefügt.

Programmtechnisch wird dieser Fall auf den einer einzigen Abklingkurve reduziert, indem man die drei Zählratenfelder aneinanderhängt (Reihenfolge A, B und C), desgleichen die unabhängigen Abklingfunktionen:

Zählraten:

Abklingfunktionen: Eingabe in das Dialogfeld für die Terme:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Zum besseren Verständnis der Anwendung wird auf das Beispiel-Projekt *DWD-LSC-3kanal-V2.txp* verwiesen, das einer sehr ausführlichen Darstellung der Bestimmungsgleichungen des Reports AKU (2008; Seite 160) entspricht. Für diesen Fall lauten die neun Abklingfunktionen:

X1 = eSr89A \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X2 = eSr90A \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) +eY90A \* &

lamY90/(tmess\*(lamY90-lamSr90)) \*( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\* &

tmess)-1.)+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X3 = eSr85A \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

X4 = eSr89B \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X5 = eSr90B \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) +eY90B \* &

lamY90/(tmess\*(lamY90-lamSr90)) \*( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\* &

tmess)-1.)+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X6 = eSr85B \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

X7 = eSr89C \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X8 = eSr90C \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) +eY90C \* &

lamY90/(tmess\*(lamY90-lamSr90)) \*( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\* &

tmess)-1.)+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X9 = eSr85C \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

Hierin werden anstelle der Halbwertszeiten die Zerfallskonstanten lamNuklid verwendet. Die mit einem kleinen e beginnenden Symbole bezeichnen Nachweiswahrscheinlichkeiten der verschiedenen Radionuklide in den Zählkanälen A, B und C.

Unter Verwendung der neuen Abklingfunktion fd() verkürzen sich die Gleichungen:

X1 = eSr89A \* fd(tAS+tstart,tmess,lamSr89)

X2 = eSr90A \* fd(tAS+tstart,tmess,lamSr90) + &

eY90A \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

X3 = eSr85A \* fd(tAS+tstart,tmess,lamSr85)

X4 = eSr89B \* fd(tAS+tstart,tmess,lamSr89)

X5 = eSr90B \* fd(tAS+tstart,tmess,lamSr90) + &

eY90B \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

X6 = eSr85B \* fd(tAS+tstart,tmess,lamSr85)

X7 = eSr89C \* fd(tAS+tstart,tmess,lamSr89)

X8 = eSr90C \* fd(tAS+tstart,tmess,lamSr90) + &

eY90C \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

X9 = eSr85C \* fd(tAS+tstart,tmess,lamSr85)

Der Beitrag des vierten anwesenden Radionuklids, Y-90, welches aus dem Zerfall von Sr-90 nachwächst, wird hier durch zusätzliche Terme mit eY90X in den Ausdrücken für X2, X5 und X8 berücksichtigt.

Wurde dieselbe Kalibrier-Aktivität *A*kaI eines Radionuklids verwendet, um die Nachweiswahrscheinlichkeiten der zwei bis drei Energiefenster zu ermitteln, sind diese Nachweiswahrscheinlichkeiten korreliert. Ihre Kovarianzen, paarweise durch

gegeben, müssen für jedes Paar von Fenstern N und M, separat für jedes Radionuklid, in die Kovarianzen-Tabelle im TAB "Werte, Unsicherheiten" eingegeben werden.

### Organisation der Abklingfunktionen Xi

1. Anzahl der Xi-Formeln =

(Anzahl der Zählkanäle) x (Anzahl der verwendeten Ergebnisgrößen)

(verwendete Ergebnisgrößen: Fitparameter, für die „fitten“ oder „fixieren“ selektiert wurde)

oder

1. Anzahl der Xi-Formeln =

(Anzahl der Messungen) x (Anzahl der verwendeten Ergebnisgrößen) x

x (Anzahl der Zählkanäle)

(wenn alle Term-Funktionen (Xi) für jede der Messungen explizit vorgegeben werden)

Die **einzuhaltende Reihenfolge** der Xi-Funktionen ist in den beiden folgenden Beispielen angedeutet. Sie entspricht formal der Reihenfolge, die sich aus dem SQL-Statement

ORDER BY Zählkanal, Nr. der Messung, Nummer der Ergebnisgröße

ergeben würde.

Beispiel 1: Fall a): 2 Messkanäle, 4 Messungen, 3 Ergebnisgrößen,

die Xi(t) unterscheiden sich zwischen den Messungen

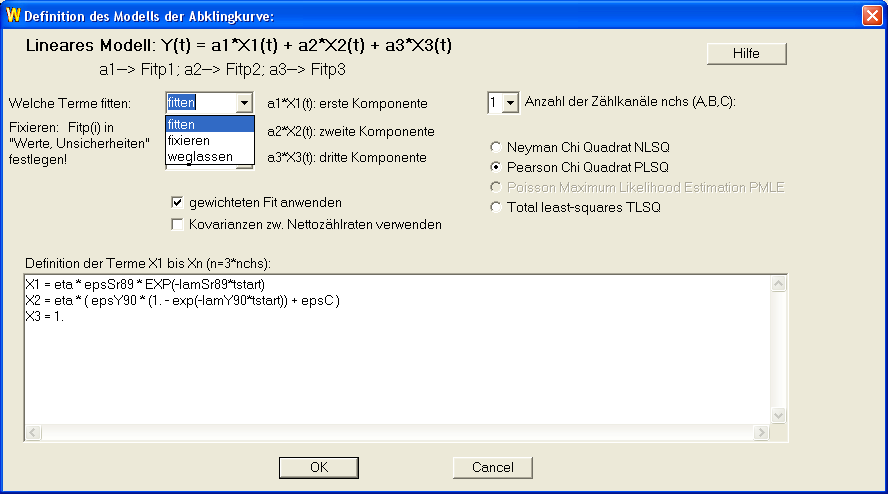
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mess-kanal | Nr. der Messung | Index von xi(t) |  | laufende Nr. |
| 1 | 1 | 1 |  | 1 |
| 1 | 1 | 2 |  | 2 |
| 1 | 1 | 3 |  | 3 |
| 1 | 2 | 1 |  | 4 |
| 1 | 2 | 2 |  | 5 |
| 1 | 2 | 3 |  | 6 |
| 1 | 3 | 1 |  | 7 |
| 1 | 3 | 2 |  | 8 |
| 1 | 3 | 3 |  | 9 |
| 1 | 4 | 1 |  | 10 |
| 1 | 4 | 2 |  | 11 |
| 1 | 4 | 3 |  | 12 |
| 2 | 1 | 1 |  | 13 |
| 2 | 1 | 2 |  | 14 |
| 2 | 1 | 3 |  | 15 |
| 2 | 2 | 1 |  | 16 |
| 2 | 2 | 2 |  | 17 |
| 2 | 2 | 3 |  | 18 |
| 2 | 3 | 1 |  | 19 |
| 2 | 3 | 2 |  | 20 |
| 2 | 3 | 3 |  | 21 |
| 2 | 4 | 1 |  | 22 |
| 2 | 4 | 2 |  | 23 |
| 2 | 4 | 3 |  | 24 |

Beispiel 2: Fall b): wie Beispiel 1, jedoch Xi(t) unterscheiden sich NICHT zwischen den Messungen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mess-kanal | Nr. der Messung | Index von xi(t) |  | laufende Nr. |
| 1 | 1 | 1 |  | 1 |
| 1 | 1 | 2 |  | 2 |
| 1 | 1 | 3 |  | 3 |
| 2 | 1 | 1 |  | 4 |
| 2 | 1 | 2 |  | 5 |
| 2 | 1 | 3 |  | 6 |

### 7.11.4 Einen Fitparameter fixieren

Für die Terme gibt es drei Möglichkeiten der Berücksichtigung:



Die Option „fixieren“ wurde nötig, um auch solche Projekte bearbeiten zu können, bei denen Sr-85 als Tracer verwendet wird, aber in der Weise, dass die Sr-85-Aktivität nicht Fitparameter ist. In solch einem Fall wird die Sr-Ausbeute durch eine separate Gamma-Messung bestimmt, so dass die bei der Betamessung auftretende Sr-85-Zählrate unabhängig vom Fitten berechnet werden kann. Dazu stellt man für die Sr-85-Komponente die Auswahl auf „fixieren“ ein.

Da die Fitroutine in diesem Fall die Unsicherheit der Sr-85-Aktivität bzw. -zählrate nicht einbeziehen kann, wurde eine Sonderbehandlung eingeführt. Zunächst wird für jede der Bruttomessungen die Sr-85-Beta-Zählrate (und ihre Unsicherheit) berechnet und von den schon erhaltenen Nettozählraten abgezogen.

Hierin bedeuten: die gammaspektrometrisch bestimmte Sr-85-Aktivität; die Sr-85-Beta-Zählausbeute und die Sr-85-Zerfallskonstante.

Die Funktion wird vom Anwender selbst durch diejenige Gleichung definiert, die im Dialog für das Modell der Abklingkurve als Gleichung für X3 eingegeben wird. Ein Beispiel dafür:

X3 = ASr85\_Gam \* eSr85 \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \*

exp(-lamSr85\*(tAS+tstart))

Der dazugehörige festzuhaltende Fitparameter wird dabei intern auf den Wert 1 gesetzt. Fasst man hierin die Eingangsgrößen , und in einem Vektor zusammen, , ist die für die Unsicherheitsfortpflanzung benötigte Kovarianzmatrix wie folgt zu berechnen:

diagonale Werte:

nicht-diagonale Werte:

Der erste Term in der letzten Gleichung tritt nur auf, wenn für die Berechnung von immer derselbe Wert der Nulleffektzählrate abgezogen wird (hier als abgekürzt).

An die derart nachträglich **korrigierten Nettozählraten** (einschließlich ihrer Kovarianzmatrix) werden die verbleibenden Komponenten für Sr-90, Sr-89 angepasst.

Die oben im Vektor zusammengefassten Symbole müssen direkt in der Argumentliste des Linfit-Aufrufs erscheinen, z.B. (die Gleichung für cSr85 ist ein „dummy“, d.h. nur ein Platzhalter):

cSr90 = Fitp1 \* PhiSr90

cSr89 = Fitp2 \* PhiSr89

cSr85 = Fitp3 \* 1

rd = Linfit(1, Rbl, **ASr85\_Gam**, **eSr85**, eSr90, eSr89, eY90, **lamSr85**, lamSr90,

lamSr89, lamY90, tmess, tstart )

phiSr90 = 1 / (etaSr\*Vol) \* exp(lamSr90 \* (tBS - tAS))

phiSr89 = 1 / (etaSr\*Vol) \* exp(lamSr89 \* (tBS - tAS))

X1 = eSr90 \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) + &

eY90 \* lamY90/(tmess\*(lamY90-lamSr90)) \* &

( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\*tmess)-1.) &

+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X2 = eSr89 \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X3 = ASr85\_Gam \* eSr85 \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

**Ab Version 2.4.24** soll die Gleichung für rd verkürzt eingegeben werden:

rd = Linfit(1, Rbl, tmess, tstart )

## 7.12 Ansicht des Ergebnisses der LSQ-Anpassung der Abklingkurve

Man erhält als Ergebnis eine Tabelle mit folgendem Aufbau, dargestellt mit dem internen Editor, der in diesem Fall keine Bearbeitung des Textes zulässt.

Ergebnis der Abklingkurven-Anlyse (mit Kovarianzen): Verfahren: PLSQ LinFit(t) = a1\*X1(t) + a2\*X2(t) + a3\*X3(t)

i t X1(t) X2(t) X3(t) NetRate rUnc. LinFit relDev uTest (m) (cps) (%) (cps) (%)---------------------------------------------------------------------------------------

1 433.00 0.83176 0.00000 0.00000 0.0017278 23.33 | 0.0016381 5.5 0.2

2 1633.00 0.67000 0.00000 0.00000 0.0013944 28.49 | 0.0013195 5.7 0.2

3 2833.00 0.53970 0.00000 0.00000 0.0009500 40.99 | 0.0010629 -10.6 -0.3

4 4033.00 0.43474 0.00000 0.00000 0.0009361 41.57 | 0.0008562 9.3 0.2

5 5237.00 0.34994 0.00000 0.00000 0.0006306 60.85 | 0.0006892 -8.5 -0.1

6 6437.00 0.28189 0.00000 0.00000 0.0006722 57.19 | 0.0005552 21.1 0.3

7 7637.00 0.22707 0.00000 0.00000 0.0006306 60.85 | 0.0004472 41.0 0.5

8 8837.00 0.18291 0.00000 0.00000 0.0002833 133.18 | 0.0003602 -21.3 -0.2

9 10037.00 0.15710 0.00000 0.00000 0.0004382 103.13 | 0.0003094 41.6 0.3---------------------------------------------------------------------------------------

LinFit: a1= 0.0019694 a2= 0.0000000 a3= 0.0000000 (in cps angegeben !) ra1= 16.240 ra2= 0.000 ra3= 0.000 (in % angegeben !)

CHi2R= 8.481E-02 Prob= 0.000272 Prob= 0.000000 Prob= 0.000000 (t-Test-Signifik. !)

Die Spalten der Tabelle sind:

* Nr. der Messung i;
* Zeitdifferenz zw. Y-90/Sr-90-Separation und dem Start der i-ten Messung, wird hier in Stunden ausgegeben;
* die Abklingfaktoren der möglichen Komponenten; im Beispiel nur von Y-90; 0.000 bedeutet, dass die Komponente im Modell nicht definiert wurde;
* , Nettozählraten, in 1/s;
* relat. Standardunsicherheit, in %;
* , der mit dem gewählten Modell der Nettozählrate angepasste Wert, in 1/s;
* relAbw., relative Abweichung , in %;
* eine u-Test-Größe, definiert als . Werte, die betragsmäßig größer als 2 sind, deuten Abweichungen vom Modell an.

Unterhalb der Tabelle folgen in zwei Zeilen die aus der LSQ-Anpassung erhaltenen Fitparameter und deren relative Standardunsicherheiten . Der erste Parameter, , ist die gesuchte Nettozählrate von Y-90, die auf den Zeitpunkt der Y-90/Sr-90-Separation bezogen ist. Chi2R ist der Wert des dabei erhaltenen reduzierten Chi-Quadrats.

Nach dem Schließen des Editor-Fensters kann, falls die darin gezeigte Anpassung nicht zu den gemessenen Nettozählraten passte, das Modell mit dem Menüpunkt „Bearbeiten - Zerfallskurve - Modell editieren“ bzw. mit dem Icon  geändert werden.

## 7.13 Berechnung des gewichteten Mittelwerts und seiner Standardunsicherheit

Der gewichtete Mittelwert der Aktivitäten der einzelnen Gammalinien wird nach folgender Gleichung berechnet:

(1)

Die Standardunsicherheit des gewichteten Mittelwerts, zunächst ohne Einbeziehung von Kovarianzen, wird nach der folgenden Gleichung berechnet:

(2)

Diese wird auch als „interne Standardabweichung“ bezeichnet. Sie berücksichtigt ausschließlich die Unsicherheiten der Einzelaktivitäten. Sie ist Bayes-konform.

Es kann aber vorkommen, dass die Einzelwerte stärker voneinander abweichen, als es die Unsicherheiten erwarten lassen würden. Um diese auf „externe Einflüsse“ zurückgehende zusätzliche Streukomponente zu berücksichtigen, wird vielfach auch die so genannte „externe Standardabweichung“ verwendet, die wie folgt definiert ist:

(3)

Es wird darauf hingewiesen, dass diese Standardunsicherheit nicht mehr Bayes-konform ist.

Die Faktoren in den jeweiligen Summen in den Gleichungen (1-3) stellen statistische Gewichte dar. Diese müssen als konstant betrachtet werden. Natürlich werden sie vor Anwendung obiger Gleichungen aus anderen Größenwerten berechnet, die auch in den vorkommen. Wenn aber nach der Berechnung von Gl. (1) auch eine formale Unsicherheitsfortpflanzung für bezüglich der mittels Differenzenquotienten erfolgt, dürfen die Werte dabei nicht geändert werden. Unter dieser Bedingung ergibt die für Gl. (1) durchgeführte Unsicherheitsfortpflanzung direkt die in Gl. (2) angegebene Unsicherheit. Diese Bedingung wird ab der Version 2.4.05 berücksichtigt. Die Werte werden nur im Verlaufe der Iterationen für Erkennungs- und Nachweisgrenze neu berechnet, einmal für jeden Iterationsschritt.

**UncertRadio berechnet beide, interne und externe Standardunsicherheit, verwendet aber für die weiteren Rechnungen nur die interne Standardunsicherheit.** Die externe Standardabweichung wird zur Information mit angezeigt. Der Hintergrund ist der, dass es eine wertvolle Information für den Benutzer ist, wenn die externe deutlich größer als die interne Standardabweichung ist; dies kann ein Hinweis auf mögliche Fehlerquellen sein, dem man nachgehen sollte.

Kovarianzen zwischen den Werten der Nachweiswahrscheinlichkeiten der einzelnen Linien, die derselben Effizienzkurve entnommen wurden, gehen in die Kovarianzmatrix **Ux** der Eingangswerte **x** ein:

**Wichtiger Hinweis**: Nach Cox et al. (2006b) stellen die hier gezeigten Gleichungen für den gewichteten Mittelwert und dessen Unsicherheit nur noch eine Näherung dar, wenn zusätzlich noch Kovarianzen zwischen den Unsicherheiten auftreten. Stattdessen ist in solch einem Falle ein gewichtetes Least-squares-Verfahren für den Mittelwert anzuwenden. Die Beschreibung dieses Verfahrens erfolgt weiter unten.

## 7.14 Least-squares-Berechnung des gewichteten Mittelwerts und seiner Standardunsicherheit

Für dieses Verfahren benutzt UncertRadio das im Abschnitt „[Mathematik der linearen Kurvenanpassung..](#URH_LLSQ_MATHE_DE)“ Matrix-basierte Verfahren ***x*** = **A** ***y.***

Dazu werden die einzelnen Aktivitätswerte als Komponenten des Vektors ***x*** verwendet. Die Design-Matrix **A** = (1,1,…,1)T hat in diesem Falle nur eine Spalte, deren Elemente alle aus dem Wert 1 bestehen; ***y*** reduziert sich zu einem Vektor mit nur einem einzigen Element, dem gesuchten Mittelwert. Für die Anwendung des Verfahrens wird die quadratische Kovarianzmatrix **Ux** benötigt. UncertRadio stellt intern alle benötigten Elemente einschließlich der Kovarianzen, wie schon [beim gewichteten Mittelwert beschrieben](#URH_GSPK1_WMEAN_DE), zusammen. Die Matrix **Uy** besteht in diesem Falle aus nur einem Element, nämlich der dem gewichteten Mittelwert beigeordneten Varianz.

## 7.15 Vorgehen zur Berechnung der Erkennungs- und Nachweisgrenze für Gamspk1

Die iterative Berechnung von Erkennungsgrenze und Nachweisgrenze erfolgt, nachdem der Mittelwert aus den Einzelwerten berechnet worden ist ([siehe Dialog der Spektrums­auswertung](#URH_GSPK1_WERTE_DE) für die Erklärung der verwendeten Symbole). Die Iteration erfolgt durch Variation (d.h. Änderung) des Mittelwerts, ein geänderter Wert sei mit bezeichnet. Danach werden alle Einzelwerte der Aktivität durch den neuen (fiktiven) Wert von ersetzt. Aus der Bestimmungsgleichung für die :

(1)

erhält man durch Umkehrung der Gleichung

(2a)

und daraus mit der Ersetzung eine Gleichung für die zum variierten Wert passenden (fiktiven) Nettozählraten:

(2b)

Ziel ist es jetzt, zunächst die Unsicherheiten der , dann über die Fortpflanzung der Unsicherheiten gemäß Gleichung (1) die Unsicherheiten der Einzelaktivitäten und abschließend mit Hilfe des gewählten Mittelwertverfahrens die Unsicherheit des (iterierten) Mittelwerts zu bestimmen.

Für die Unsicherheiten wird der folgende Ansatz gemacht (Separation; [siehe auch](#URH_GSPK1_WERTE_DE)):

(3)

Hierin ist allein der erste Summand auf den Anteil der Probenaktivität an der Zählrate bezogen. Die restlichen Terme darin beschreiben die Unsicherheitsbeiträge von Parametern, die den Untergrund der i-ten Gammalinie definieren, d.h. Compton-Untergrund inklusive Nettozählrate eines „Peaks im Nulleffekt“.

Unter Verwendung der Gleichung (2) können nun, wie weiter oben bereits angedeutet, mit den Gleichungen (2b) und (1) die (fiktiven, variierten) Aktivitäten der einzelnen Gammalinien und ihre Unsicherheiten ermittelt werden. Anschließend stehen nach der Berechnung des sich daraus ergebenden (fiktiven) Mittelwerts und seiner Standardunsicherheit diejenigen Werte zur Verfügung, die benötigt werden, um den nächsten Iterationsschritt auszuführen und das Erreichen von Konvergenz testen zu können.

**Wichtige Anmerkung:**

Es kann „externe“ Einflüsse geben, die dazu führen, dass die berechneten Werte der einzelnen Gammalinien stärker untereinander streuen als es von der Größe der Einzelunsicherheiten her zu erwarten wäre.

Bei dem **gewichteten Mittelwert** kann man auf diesen Effekt stoßen, wenn die „externe“ Standardunsicherheit deutlich größer als die „interne“ Standardunsicherheit ist bzw. der für das „reduzierte Chi-Quadrat“ angegebene Wert deutlich größer als eins ist.

Bei dem **arithmetischen Mittelwert mit additiver Korrektion** lässt sich darauf schließen, wenn die Korrektion und insbesondere deren Unsicherheit den zuvor berechneten einfachen arithmetischen Mittelwert oder dessen Standardunsicherheit deutlich verschieben oder vergrößern.

Diese Einflüsse werden in der Regel nicht in dem Modell der Gleichung (1) berücksichtigt. Für die Bestimmung von Erkennungs- und Nachweisgrenze müssen Werte der Aktivität iteriert werden. Die dazu erforderliche Umkehrung der Gleichung, d.h., indem aus einem vorgegebenen (iterierten) Wert der Aktivität, , die dazu zu erwartenden Nettozählraten nach Gl. (1b) zu berechnen sind, führt dazu, dass der externe Einfluss eliminiert wird. Aus den jetzt erhaltenen Nettozählraten folgen wiederum Einzelwerte der Aktivität, die alle denselben Wert nach Gl. (1) haben, deren Streuung also gleich null ist!

Diese Eigenschaft bedeutet für die Berechnung von Erkennungs- und Nachweisgrenze, dass der bei der Messung der Ergebnisgröße möglicherweise feststellbare externe Einfluss nicht zum Tragen kommen kann. Insofern wird die Verwendung der externen Standardabweichung beim gewichteten Mittelwert bzw. des NIST-2004-Verfahrens überhaupt stark eingeschränkt, auf jeden Fall im Hinblick auf Erkennungs- und Nachweisgrenze.

## 7.16 Dialog Werte der Spektrums-Auswertung

In einer Tabelle eines Windows-Dialogs können zeilenweise für jede der Gammalinien die Messwerte und Unsicherheiten der in der obigen Gleichung verwendeten Messgrößen (Symbole) eingegeben werden.

Es sei noch auf die Besonderheit hingewiesen, dass z.B. im Falle eines natürlich vorkommenden Radionuklids von den zunächst bestimmten Nettozählraten auch die entsprechenden Nulleffektzählraten („Peak im Peak“) abgezogen worden sein müssen. **Dementsprechend muss die Unsicherheit einer Nettozählrate auch den Unsicherheitsbeitrag der Nulleffektzählrate beinhalten**. Unter dieser Voraussetzung ist die zusätzliche Eingabe der einzelnen Nulleffektzählraten nicht erforderlich.

Es wird angenommen, dass alle erforderliche Werte und Unsicherheiten aus dem mit einem Gammaspektrometrie-Programm erhaltenen Auswerteprotokoll eines Gammaspektrums entnommen werden.

Der Aufbau des Dialogs ist dem folgenden Bild zu entnehmen.

|  |
| --- |
|  |

Die einzugebenden Messwerte sind:

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbolnamen im Dialog:** | **Bedeutung:** |  | **Symbole:** |
| Rnet | Nettozählrate der Gammalinie *i* bei der Energie , in |  |  |
| RT | Zählrate des integrierten Compton-Untergrunds im Bereich 1,7×Fwhm der Gammalinie *i* bei der Energie , in |  |  |
| Rbg | Nettozählrate einer Gammaline bei im separat gemessenen Nulleffektspektrum, in |  |  |
| effi | Linien-Nachweiswahrscheinlichkeit (full energy peak efficiency) bei der Energie |  |  |
| pgamm | Emissionswahrscheinlichkeit der Linie *i* |  |  |
| f\_att | Selbstschwächungskorrektion für die Energie ; wird in der multiplikativen Form verwendet; |  |  |
| f\_coin | Korrektion für Koinzidenzsummation der Linie bei der Energie ; wird in der multiplikativen Form verwendet; |  |  |

Die Einheit der Nettozählraten können in cps () oder in cpm () eingegeben werden.

Die gemessenen Werte müssen als absolute Werte, nicht als relative Werte eingegeben werden. Für deren Unsicherheiten kann die Eingabe mit Hilfe der Radiobuttons wahlweise in % oder als absolute Unsicherheitswerte erfolgen.

**Anmerkung:** Bis zur Version 2.4.18 waren 7 Radiobuttons dafür vorgesehen, seit der Version 2.4.19 sind es nur noch 5. Entfallen sind die Radiobuttons für die Werte effi und pgamm, die jetzt nur noch als absolute Werte einzugeben sind. Wenn ältere Projekte eingelesen werden, noch mit 7 Radiobuttons versehen, erfolgt intern eine Umstellung auf die 5 Radiobuttons, indem die Werte von effi und pgamm jeweils durch 100 geteilt werden, falls die Buttons nicht auf „abs“ eingestellt waren.

Die Aktivitäten *A*i der einzelnen Linien werden nach der folgenden Gleichung berechnet:

Die Unsicherheiten der Nettozählraten werden programmintern nach der folgenden Gleichung berechnet:

Hierbei ist ein Faktor, der davon abhängt, wie die Nettozählrate *R*ni bestimmt wurde. Im Falle der klassischen TPA-Methode (total peak area) ist:

Hierin ist *b* die Breite des Peaks, z.B. *b*=1,7xFwhm, und *L* die Anzahl der Kanäle, über die jeweils links und rechts vom Peak der mittlere Peakuntergrund ermittelt wird.

Wird *R*ni dagegen durch Peak-Anpassung bestimmt, kann für näherungsweise ein fester Wert verwendet werden, der „etwas größer“ als 1 ist; dieser Faktor ist abhängig vom verwendeten Anpassungs-Verfahren und kann z.B. durch eine Art von „Kalibrierung“ bestimmt werden.

Im oberen Bereich des Dialogs gestattet ein Listenfeld die **Auswahl des Mittelwert-Typs**:

* [gewichteter Mittelwert](#URH_GSPK1_WMEAN_DE),
* [Mittelwert mit gewichtetem Least-squares-Verfahren](#URH_GSPK1_LSQ_WMEAN_DE).

Desweiteren wird in einem Feld rechts oben im Dialog der Wert des Faktors eingegeben. Auch kann die Verwendung von Kovarianzen zwischen den Werten der Nachweiswahrscheinlichkeiten der Gamm-Linien an- oder abgewählt werden.

In der ersten Spalte der Tabelle lassen sich einzelne Linien zu- oder abwählen.

## 7.17 Ansicht des Ergebnisses der Mittelwertberechnung mit Gamspk1

**Für den gewichteten Mittelwert der Aktivitäten** einzelner Gammalinien erhält man folgenden Protokollausdruck, in diesem Falle handelt es sich um die Messung von Co-60 auf einem HPGe-Detektor:

----------------------------------------------------------------------

(1 + b/2L)-äquivalenter Faktor für Compton-UG-Rate: 1.120

Einzelne Peakdaten:

(pgamm\*fcoin ist ein Maß für Wichtigkeit der Linie!)

i E PNRate epsPeak pgamm fatt fcoin (pgamm\*fcoin)

keV cps %

----------------------------------------------------------------------------------

1 1173.20 5.699E-03 0.7790 0.99850 1.0000 1.0615 1.0599 Werte

2.71 1.6789 0.03000 1.0000 1.3810 u\_rels in %

2 1332.50 5.360E-03 0.7030 0.99986 1.0000 1.0641 1.0640 Werte

2.76 1.6245 0.00060 1.0000 1.3890 u\_rels in %

Ergebnisse einzelner Peak-Aktivitäten:

A(i) = PeakNetRate(i) \* (fatt(i) \* fcoin(i)) / (epsPeak(i) \* pgamm(i))

i E(keV) Aktivität (Bq) rel.StdAbw (%)

--------------------------------------------------

1 1173.20 7.7771E-01 3.61

2 1332.50 8.1147E-01 3.63

Auswertung des gewichteten Mittelwerts:

gewichteter Mittelwert = 0.79379

Std.Abw. d. Mittelwerts = 2.23372E-02 ( 2.81 %) (Bayes-konform)

ext. Std.Abw. d. Mittelwerts = 1.85227E-02 ( 2.33 %) (nicht Bayes-komform)

Chi-Quadrat = Testwert T = 0.68763

reduziertes Chi-Quadrat = 0.68763

Signifikanz (Chi-Quadrat > T) = 4.06973 %

Hinweis: nur die interne Standardabweichung wird verwendet!

----------------------------------------------------------------------

In der ersten Tabelle dieses Ausdrucks werden in verkürzter Form die Eingangsdaten, ohne die Angabe ihrer Unsicherheiten, wiedergegeben. Bei diesem Beispiel treten aufgrund der Bohrloch-Messgeometrie signifikante Korrektionen für Koinzidenzsummation (fcoinsu) auf. Das in der letzten Spalte angegebene Produkt (pgamm \* fcoinsu) ist dabei ein Maß für die Gewichtung der einzelnen Gammalinien.

Die zweite Tabelle des Ausdrucks gibt die für die einzelnen Linien berechneten Aktivitätswerte (in Bq) und deren relative Standardunsicherheiten (in %) wieder. Darauf folgen die Kenndaten zum gewichteten Mittelwert.

Für das **gewichtete Least-squares-Verfahren (LSQ Mittelwert)** erhält man entsprechend des letzten Teils des oben beschriebenen gewichteten Mittelwerts:

(Hinweis: In diesem Falle sind die Varianzen der Aktivitätswerte der zwei Gammalinien praktisch gleich, daher ergeben sich praktisch identische Ergebnisse für dieses Verfahren und das des gewichteten Mittelwerts.

Auswertung des gewichteten Mittelwerts mit least-squares:

gewichteter Mittelwert = 0.79358

Std.Abw. d. Mittelwerts = 2.23360E-02 ( 2.81 %)

reduziertes Chi-Quadrat = 0.86694

----------------------------------------------------------------------

## 7.18 Ermittlung der MC-Verteilungen und daraus abgeleiteter Kenngrößen im Detail

Bei der Monte Carlo Simulation werden nacheinander drei verschiedene Verteilungen erzeugt, aus denen jeweils bestimmte Kennwerte abgeleitet werden. Bei den abgeleiteten Kennwerten handelt es sich neben arithmetischem Mittelwert und Standardabweichung der Verteilung vor allem um bestimmte Quantile.

Am Beginn wird die (große) Anzahl N von simulierten Messwerten der Ergebnisgröße festgelegt. Dies definiert einen „Run“. Es können mehrere „Runs“ (Anzahl r; maximal 50) gerechnet werden, so dass mit den jeweils erhaltenen Kenngrößen statistische Auswertungen gemacht werden können; man erhält aus der r-fachen Wiederholung der Simulation z.B. mit der Standardabweichung eine statistische Angabe über die Unsicherheit der erhaltenen Kennwerte.

**Ermittlung eines Quantils**

Das bisher in UR2 benutzte MC-Verfahren war primär für das Vorliegen nahezu symmetrischer Verteilungen der Ergebnisgröße ausgelegt. Mit der Anwendung des GUM Supplements 1 können nicht-symmetrische Verteilungen bestimmter Eingangsgrößen zu einer asymmetrischen Verteilung der Ergebnisgröße führen. Hierfür erwies sich das bisher benutzte Verfahren, für das vor der Simulation die zu erwartenden Extremwerte im Programm festzulegen waren, als nachteilig.

Das MC-Verfahren wurde daher ab Version 2.1.9 derart umgestellt, dass alle MC-Werte der Ergebnisgröße in einem Array gesammelt werden. Damit ergibt die MC-Simulation für jede der drei o.a. Verteilungen ein Array von maximal 2 000 000 MC-Werten der Ergebnisgröße. Aus diesem Array können Mittelwert, Standardabweichung sehr einfach berechnet werden. Die Bestimmung von Quantilen erfolgt hierbei nach einem verteilungsfreien Verfahren, wofür die Einzelwerte der Verteilung zunächst sortiert werden müssen. Lediglich für die grafische Darstellung der Verteilung als Histogramm werden die MC-Werte dafür in 20 000 Bins („Vielkanal“) einsortiert; die Grenzen des Vielkanals werden erst nach Ende des Simulationslaufs festgelegt. Durch diese Umstellung ist das MC-Verfahren programmtechnisch einfacher geworden.

1. **Schätzwerte nach Bayes**

Für die Bestimmung der Werte für Mittelwert und Standardabweichung sowie untere und obere Vertrauensgrenze nach Bayes wird nur die bei null abgeschnittene Verteilung ermittelt, Werte < null werden nicht berücksichtigt, auch nicht bei der Gesamtanzahl der Werte. Die Vertrauensgrenzen werden hierbei als unteres und oberes Quantil ermittelt. Die Gesamtanzahl der Werte in einem Run beträgt hierbei nur ca. N/2.

Mit der Version 1.03 wurde eine Option eingeführt, die es gestattet, anstelle der symmetrischen Konfidenzgrenzen ein solches Paar von Grenzen zu berechnen, das den kürzesten Abstand aufweist (Bayes‘sches Coverage-Intervall kürzester Länge). Test-Beispiel: Neutron-Dose-Cox-2006\_EN.txp. Die Aktivierung dieser Option führt nur zu unmerklicher Erhöhung der Rechenzeit.

1. **Erkennungsgrenze**Für die Erkennungsgrenze erfolgt eine Modifikation dergestalt, dass der „wahre Wert“ der Nettozählrate (und damit auch der Aktivität) gleich null gesetzt wird. Es resultiert eine Verteilung der damit simulierten Einzelwerte, die zu etwa 50% im negativen Bereich liegt. Es erfolgt keine Abschneidung bei null. Das hieran ermittelte obere (1-α)-Quantil stellt den erhaltenen Wert der Erkennungsgrenze dar.

Bei der MC-Simulation können Eingangsparamater, die eine asymmetrische Verteilung oder eine Rechteckverteilung besitzen, zu einer deutlich **asymmetrischen Verteilung der Ergebnisgröße** führen. Dies kann im Falle der Erkennungsgrenze dazu führen, dass der Mittelwert (Schwerpunkt) der dazugehörigen Verteilung vom Wert null abweicht.

Es wird zunächst geprüft, ob eine Abweichung der MC-Verteilung von der erwarteten, zu null symmetrischen Normalverteilung vorliegt. Die Prüfung besteht darin, den Betrag des von null abweichenden Mittelwerts *y*1 mit der MC-Unsicherheit des Mittelwerts, *u*(*y*1)/√*N*, zu vergleichen (s. weiter unten). Ist das Verhältnis

**wird mit einem Sekantenverfahren auch für die Erkennungsgrenze eine Iteration durchgeführt**, die dafür sorgen soll, dass der Schwerpunkt dieser Verteilung, *y*1, näher an den Wert null gebracht wird. Es werden bis zu 12 Schritte ausgeführt, in denen jeweils eine volle MC-Simulation erfolgt, was die Rechendauer erheblich erhöhen kann. Wird das obige Verhältnis von 0,075 dabei nicht erreicht, wird aus den 12 Schritten diejenige MC-Simulation mit der dem geringsten Verhältniswert für die Berechnung der Erkennungsgrenze zugrunde gelegt. Üblicherweise kann der Wert *y*1 dadurch um ein bis zwei Größenordnungen verringert werden; es kann aber vorkommen, dass *y*1 um kaum mehr als den Faktor 10 erniedrigt wird.

1. **Nachweisgrenze**Die Ermittlung der Nachweisgrenze ist der zeitaufwändigste Teilder Simulation, da hierbei die Verteilung in einem iterativen Vorgang mehrere Male, jeweils nach Erreichen der Gesamtanzahl von Einzelwerten, durch Änderung des Mittelwerts der Verteilung verschoben wird. Ziel der schrittweisen Verschiebung ist, dass das (untere) *ß*-Quantil dieser Verteilung dem zuvor bestimmten Wert der Erkennungsgrenze immer näherkommt. Das Iterationsverfahren basiert auf dem Bisektionsverfahren, wobei jedoch nicht eine simple Halbierung eines Intervalls erfolgt, sondern die Teilung durch lineare Interpolation zwischen seinen beiden Grenzen erfolgt.

Auch hierbei werden negative Einzelwerte ausdrücklich berücksichtigt.

Die Motivation, bei Erkennungsgrenze und Nachweisgrenze deren negative Anteile ihrer Verteilungen nicht zu verwerfen, besteht darin, dass der primär erhaltene Messwert der Ergebnisgröße direkt mit der Erkennungsgrenze verglichen werden soll. Würde man die negativen Anteile verwerfen, müsste der primäre Messwert, der ja negativ sein kann, erst eine zum Abschneiden analoge Transformation erfahren, bevor der Vergleich wieder möglich wäre.

Die Verteilung bei den Schätzwerten nach Bayes nicht-negativ zu machen, ist möglich, nachdem der Vergleich zwischen primärem Messergebnis und Erkennungsgrenze bereits erfolgt ist und zu dem Schluss geführt hat, es liegt eine aus der Probe herrührende Aktivität vor.

**Schätzung der MC-Unsicherheiten der Kennwerte**

Bei einer MC-Simulation mit nur einem einzigen Run fehlen Schätzungen der Unsicherheiten der Kennwerte. Auch die aus der Streuung nur weniger Runs erhaltenen Schätzwerte sind noch wenig verlässlich.

Da die erhaltenen MC-Verteilungen in der Regel normal-verteilt sind, können jedoch Unsicherheiten der Kennwerte wie folgt geschätzt werden (siehe z.B. Barlow, 1999):

|  |  |  |
| --- | --- | --- |
| Kennwert: | Breitenparameter: | Formel für die (absolute) Unsicherheit: |
| Wert der Ergebnisgröße |  |  |
| Unsicherheit |  |  |
| untere Vertrauensgrenze |  |  |
| obere Vertrauensgrenze |  | dito |
| Erkennungsgrenze |  |  |
| Nachweisgrenze |  | ,  mit |

(N: Anzahl der MC-simulierten Einzelmessungen; ϕ(.) und Φ(.): Dichtefunktion und Verteilungsfunktion der Standardnormalverteilung).

Die nach dieser Tabelle berechneten MC-Unsicherheiten werden im Dialog – im Falle nur eines Runs – als relative Werte in Prozent angegeben.

**Grafische Darstellungen**

Die bei den drei vorstehenden Punkten erzeugten Verteilungen werden während des Ablaufs der Simulation in einem separaten Fenster auch grafisch als Histogramm dargestellt. Die Verteilungen zeigen während der r Runs die aus den bereits erfolgten Runs akkumulierte Verteilung, die sich schon nach wenigen Runs „stabilisiert“. Bei den Runs zur Nachweisgrenze wird erst nach Beendigung aller Runs die akkumulierte Verteilung dargestellt. Die X-Achse (Abszisse) entspricht den Werten der ausgewerteten Größe, die im Titel des jeweiligen Graphs bezeichnet ist, die Y-Achse (Ordinate) stellt den Wertebereich der Wahrscheinlichkeit dar.

Beispiel zum separaten Fenster mit den MC-Grafiken:

|  |
| --- |
| MCplotfile-d_DE.png |

**Vertikale grüne Linien** in den Grafiken markieren, jeweils von links nach rechts in den Verteilungen, folgende Werte:

**Verteilung Werte**

Ergebnisgröße untere Vertrauensgrenze, Mittelwert, obere Vertrauensgrenze

Erkennungsgrenze Wert derselben

Nachweisgrenze Erkennungsgrenze, Nachweisgrenze

Nach jedem einzelnen Run wird **in blauer Farbe jeweils diejenige Gaußkurve** eingezeichnet, die dem Ergebnis des analytischen Verfahrens entspricht.

Hinweis: Nach Abschluss der Rechnungen in der MC-Simulation bleibt das separate Fenster mit den MC-Grafiken zunächst stehen. Dies erlaubt, im Anschluss z.B. die Daten in den verschiedenen TABs nachträglich einsehen zu können, oder auch einen Ergebnisreport zu erzeugen und dann wieder auf das TAB „Resultate“ mit diesem Fenster zurückzugehen. Es wird erst dann entfernt, wenn Änderungen bei bestimmten Eingangsdaten oder Optionen dazu führen, dass die ursprünglichen Voraussetzungen für die erfolgte MC-Simulation nicht mehr gegeben sind. Dies erfolgt auch dann, wenn nochmals Rechnungen, z.B. im TAB „Werte, Unsicherheiten“, oder durch den Wechsel vom TAB „Werte, Unsicherheiten“ zum TAB „Unsicherheiten-Budget“, initiiert werden.

## 7.19 Hinweise zum PMLE-Verfahren für lineare Entfaltung

Das Verfahren der Poisson MLE (PMLE) erfordert, dass die abhängigen Größen in der Entfaltung, z.B. die Impulsanzahlen einer Abklingkurve, Poisson-verteilt sind. Da letztere bei den anderen Methoden der Entfaltung bisher als Nettozählraten verwendet worden sind, werden die Nettozählraten programmintern lediglich in Bruttoimpulsanzahlen zu konvertieren. Die Bruttoimpulsanzahlen einer Abklingkurve

;

werden als nicht-korreliert betrachtet. muss für alle Zählraten denselben Wert haben, andernfalls wäre die Form der Abklingkurve der Bruttoimpulsanzahlen gestört.

Soll zum Beispiel eine Y-90-Abklingkurve angepasst werden, wird aus der dazugehörigen Modellgleichung in der Nettozählraten-Darstellung

das Modell für die PMLE-Anpassung:

Dazu wurde eine zweite Fitfunktion eingeführt, mit dem Fitparameter und . Der Parameter repräsentiert die Summe aus Nulleffekt und Blindwert, er wird mit dem PMLE-Verfahren angepasst und kann dabei einen Wert bekommen, der von dem bekannten Wert abweichen kann.

**Besonderheiten:**

Dieses Verfahren benötigt also einen Fitparameter mehr als die anderen Verfahren, bei denen die Nettozählraten-Kurven angepasst werden. Es kann also nur verwendet werden, wenn maximal 2 physikalische Komponenten bestimmt werden sollen. Beide physikalischen Komponenten sollten die Eigenschaft haben, dass ihre Kurvenverläufe nicht konstant oder quasi-konstant sind. Falls eine dieser beiden Komponenten den Beitrag eines Radionuklids mit sehr langer Halbwertszeit darstellt, deren Beitrag also im gesamten Zeitbereich der Abklingkurve kaum abnimmt, kann das Fitverfahren nicht zwischen dieser Komponente und der zusätzlichen Komponente für unterscheiden. In diesem Falle wäre das PMLE-Verfahren nicht anwendbar.

Das PMLE-Verfahren kann in dem Programm derzeit auch dann nicht angewählt werden, wenn, wie im Falle von Sr-Messungen mit zugesetztem Sr-85-Tracer, der Fitparamater für das Sr-85 auf den Status „fixieren“ gesetzt wurde.

Das PMLE-Verfahren ist auch dann nicht auswählbar, wenn zu wenige Messpunkte im Vergleich zur Anzahl der anzupassenden Parameter vorliegen.

**Anwendung des Verfahrens:**

Die Daten im Dialog „Eingabe der Abklingkurve“ werden ebenso aufgenommen wie bei den anderen Verfahren, daran ändert sich nichts.

Wurde das Projekt bereits für z.B. die Verwendung eines WLS-Verfahrens erstellt, hat das Programm bereits die nötigen Informationen über die anzupassenden physikalischen Komponenten und kann entscheiden, ob die oben genannten Kriterien für die Anwendung des PMLE zutreffen. Falls die Kriterien nicht erfüllt sind, wird in dem Modell-Dialog „Modell der Abklingkurve“ die Auswahl des PMLE-Verfahrens verhindert.

Ist die Anwahl des PMLE-Verfahrens im Modell-Dialog erlaubt, wird nach dessen Selektion das Verfahren wie üblich ausgewertet. Lediglich in der Ansicht der Anpassung () werden Bruttozählraten anstelle von Nettozählraten dargestellt.

## 7.20 Behandlung von Impulsanzahlen und Zählraten

Die nachfolgend beschriebene Besonderheit für nicht normalverteilte Impulsanzahlen und Zählraten bezieht sich nur auf die Monte Carlo-Simulation nach ISO 11929-2019, Teil 2. Nach Teil 1 von ISO 11929 werden die Eingangsgrößen ohnehin als normalverteilt angenommen oder lassen sich mit dem Prinzip der maximalen Entropie dieser Verteilung zuordnen.

Nach GUM Supplement 1 (JCGM 101:2008), Abschnitt 6.4.11, wird für gezählte Ereignisse, z.B. gezählte Photo­nen, die Poisson-verteilt sind und eine Eingangsgröße repräsentieren, das folgende Verfahren zur Bestimmung der Verteilung der Eingangsgröße empfohlen. Mit Hilfe des Bayes-Theorems **wird der Größe bei gezählten Ereignissen** unter Verwendung eines konstanten Priors **eine Gammaverteilung als Posteriorverteilung zugeordnet**, die für die Unsicherheits­ermittlung als Eingangsverteilung von zu verwenden ist:

für (1)

Mittelwert und Varianz sind dann und . Dies bezieht sich auf Impulsanzahlen.

In ISO 11929-2019 wird aus einer Poisson-verteilten Impulsanzahl auf die Gammaverteilung der Eingangsgröße „Zählrate“ geschlossen, wobei nicht der konstante Prior verwendet wird, sondern der Prior , so dass Mittelwert und Varianz gleich und sind.

In UncertRadio wird dies so gelöst, dass, wie oben nach GUM Supplement 1 angegeben ist, schon die Impulsanzahl als gamma-verteilt angesetzt wird (Verteilungstyp „(N+x)-Regel“; dies entspricht dem Prior ). Durch die dann erforderliche Berechnung der Zählrate , mit einer Gleichung wie z.B. , ist diese Zählrate ebenfalls gamma-verteilt. Das bedeutet aber auch, dass eine als gamma-verteilt anzusehende Zählrate (in UncertRadio) immer eine Gleichung wie erfordert.

Bei der Aktivitätsmessung kann noch zwischen zwei Varianten unterschieden werden,

* die Messung mit Zeitvorwahl (die Impulsanzahl ist zufällig verteilt, sie folgt einer **Poissonverteilung**) und
* der Messung mit Impulsvorvorwahl (die Messdauer ist zufällig verteilt, sie folgt einer **Erlang-Verteilung**).

Die Erlangverteilung wird im Lehrbuch von Knoll angeführt (Knoll, G.F., Radiation Detection and Measurement, 2nd edition, (JohnWiley, NewYork,1989), pp. 96-99);

siehe auch:

International Safety Research, Safety Support Series, 2013. Radiation Counting Statistics. Volume 1. Canada.

Pengra, D., 2008:

<http://courses.washington.edu/phys433/muon_counting/counting_stats_tutorial_b.pdf>

Pishro-Nik, H., Introduction to Probability:

<https://www.probabilitycourse.com/chapter11/11_1_2_basic_concepts_of_the_poisson_process.php>

.

**Vergleich der Erlang- und der Poissonverteilung**

Die beiden Verteilungen sind wie folgt definiert, dabei ist der Zählratenparameter:

Poisson-Verteilung (2)

Erlangverteilung (3)

Die Erlangverteilung ist also eine Gammaverteilung für ganzzahlige . Man kommt mit Hilfe dieser beiden Formeln auf einen einfachen Zusammenhang:

(4)

Wendet man auf beide Verteilungen das Bayes-theorem mit einem Prior von an, ergeben sich wie folgt gleichartige Posterior-Verteilungen für die Zählrate , nämlich eine Gammaverteilung:

**Messung mit Zeitvorwahl:**

(5)

**Messung mit Impulsvorwahl:**

(6)

Aus den Gleichungen (5) und (6) kann auf folgende Gleichsetzung geschlossen werden, die wiederum zum einfachen Zusammenhang zwischen Erlang- und Poissonverteilung nach Gl. (4) führt:

oder

(7)

Verwendet man im Falle der Poissonverteilung einen anderen Prior, , ergibt sich ebenfalls eine Gammaverteilung, aber nicht dieselbe: .

**Im Falle der Impulsvorwahl** (Messdauer variabel) muss in UncertRadio **der Messdauer** mit der Wahl des Verteilungstyps „Npreset“ **die Erlang-Verteilung zugeordnet** werden; mit der dann erforderlichen Gleichung wie , wird der Zählrate intern die Gammaverteilung zugeordnet.

Beispielprojekt: **ImpulsVorwahl\_DE.txp**

## 7.21 Behandlung physikalischer Einheiten

In den Gleichungen zur Berechnung des Werts einer Ergebnisgröße bestehen die als Eingangsgrößen verwendeten Variablen aus einem Zahlenwert und einer physikalischen Einheit. Gelegentlich wird nicht beachtet, dass anstelle der sogenannte Basiseinheiten (wie kg, m, s) daraus abgeleitete Einheiten (wie g, cm, min) verwendet werden. Um den Wert der Ergebnisgröße mit seiner aus Basiseinheiten zusammengesetzten Einheit korrekt zu berechnen, müssen zusätzlich die Skalierungsfaktoren, mit denen abgeleitete Einheiten auf ihre Basiseinheiten bezogen werden, in die Gleichungen eingesetzt werden. Es gibt zwei Konzepte, die hierbei helfen können und in UncertRadio verwendet werden können:

* Einführung sogenannter Schaltvariable oder Trigger, die zur Verwendung von Skalierungsfaktoren in die Gleichungen explizit eingesetzt werden; dazu wird auf Abschnitt 7.21.5 verwiesen. Der Anwendungsbereich von Schaltvariablen geht im Allgemeinen über die Verwendung für Skalierungsfaktoren hinaus.
* Man kann versuchen, die Einheit einer abhängigen Größe aus den Einheiten ihrer Eingangsgrößen auf rechnerischem Wege zu ermitteln. Dieser Weg kann in UncertRadio zumindest als Test auf das jeweilige Projekt angewendet werden. Diese Möglichkeit näher zu beschreiben ist der wesentlich Zweck dieses Kapitels.

Das Ziel ist demnach, für ein Projekt die Zahlenwerte und Einheiten der Eingangsgrößen korrekt festzulegen, die Verwendung von abgeleiteten Einheiten ist möglich. Dann sollte es dem Programm möglich sein, daraus die Einheit der Ergebnisgröße zu berechnen, wobei dafür dann die Konvertierung auf Basiseinheiten erfolgt.

Wenn dieser Weg genutzt werden soll, ist es nötig, Wert auf eine mehr systematisierte Verwendung von Einheiten und ihrer Schreibweise in Texteditoren zu legen. Dazu **ist es wichtig, die Einheiten von Eingangsgrößen möglichst vollständig zu benutzen**. Um zum Beispiel für eine Aktivitätsmessung zur Einheit „Bq“ als Bestandteil der Einheit der Ergebnisgröße zu kommen, ist es z. B. ganz wichtig:

* das Einheiten-Feld einer Detektornachweiswahrscheinlichkeit nicht leer lassen, sondern dafür die Einheit „1/Bq/s“ ansetzen;
* für einen Kalibrierfaktor eine Einheit wie z. B. „Bq\*s/kg“ verwenden;
* Für chemische Ausbeuten können, wenn sie durch Wägung bestimmt wurden, Einheiten wie „g/g“ oder „g/kg“ angegeben werden.

Fehlende Einheiten bei Eingangsgrößen können verhindern, dass die Einheit der Ergebnisgröße richtig „berechnet“ werden kann.

**Wichtig:** Leider ist es derzeit noch so, dass die automatisierte Form der Berechnung von Einheiten nicht kompatibel mit der Verwendung von Schaltvariablen für Skalierungsfaktoren ist.

### 7.21.1 Zusammenstellung von Basiseinheiten und abgeleiteter Einheiten

Es wird zwischen Basiseinheiten und abgeleiteten Einheiten unterschieden. Ziel dabei ist es, bei der Auswertung eines UncertRadio-Projekts für die Größen, zu denen eine abgeleitete Einheit eingegeben wird, eine Umwandlung zur Basiseinheit vorzunehmen. Ein solcher Wechsel ist mit einem Skalierungsfaktor verbunden, der bei der Umwandlung zu berücksichtigen ist.

Zur UncertRadio-Installation gehören zwei CSV-Dateien, **unitsTable\_DE.csv** und **units\_other\_DE.csv**. Diese werden hier wiedergegeben.

**unitsTable\_DE.csv:**

In dieser Tabelle bedeuten:

Spalte A: Basiseinheiten;

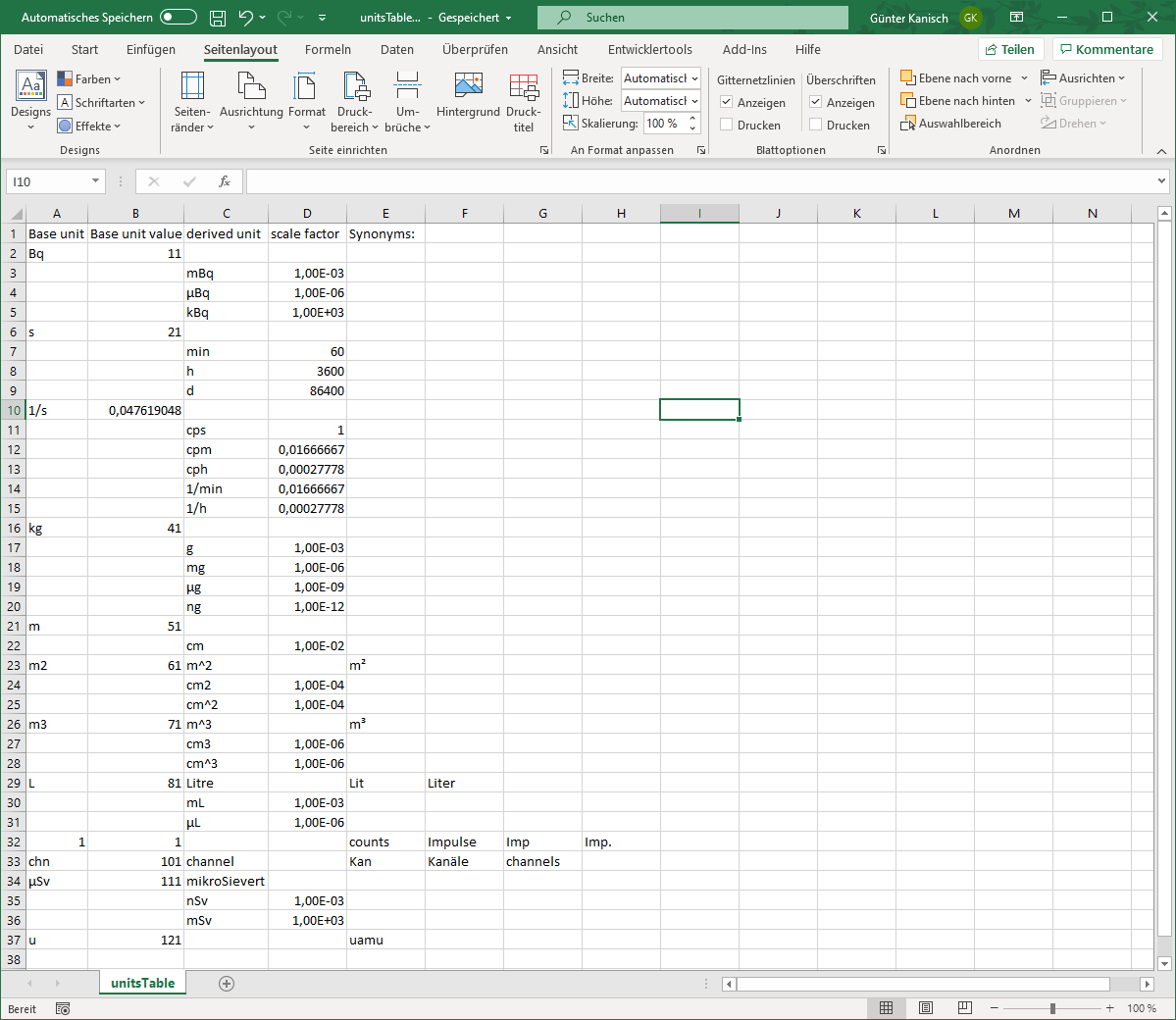
Spalte B: ein der Basiseinheit zugeordneter numerischer Wert;

Spalte C: abgeleitete Einheiten;

Spalte D: mit der Einheit in Spalte C verbundener Skalierungsfaktor;

Spalte E und folgende: in der Zeile der Basiseinheit können hier zur Basiseinheit synonyme Namen oder Schreibweisen davon eingegeben werden.

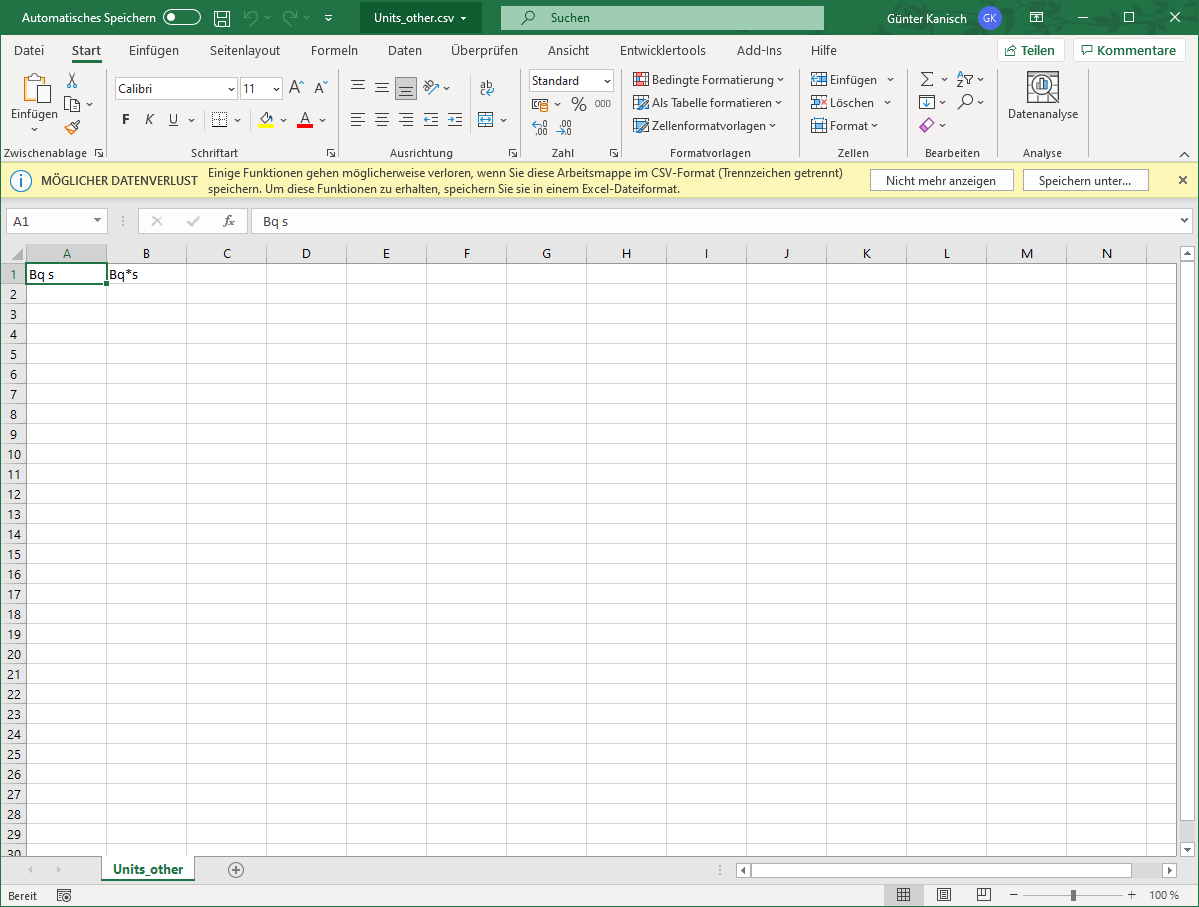
**Hinweis:** Da CSV Dateien sprachabhängig sind, werden die beiden CSV-Dateien durch Textdateien ersetzt, die nach den beiden CSV-Dateien gezeigt werden.



*Hinweis: die Einträge für m2 und m3 wurden inzwischen entfernt und in Units\_other.txt transferiert.*

**Units\_other\_DE.csv:**

Diese Datei enthält nur zwei Spalten, A und B. In Spalte A wird ein „laborgängiger“ Einheitenname eingegeben, in Spalte B wird dazu die korrekte Schreibweise dieser Einheit angegeben.



**Die** **folgenden zwei Textdateien lösen die beiden CSV-Dateien ab**:

**unitsTable.txt**  (nur der erste Teil)**:**

base:Base unit;base#: base unit value; syn:synonym;derv:derived unit;conv: scale factor

base=Bq

base#=11.

derv=mBq

conv=1.00E-03

derv=µBq

conv=1.00E-06

derv=kBq

conv=1.00E+03

base=s

base#=21.

derv=min

conv=60.

derv=h

conv=3600.

derv=d

conv=86400.

base=1/s

base#=0.047619047619

derv=cps

conv=1.0

derv=cpm

conv=0.01666666666667

derv=cph

conv=0.000277777777778

derv=1/min

conv=0.01666666666667

derv=1/h

conv=0.000277777777778

base=kg

base#=41.

derv=g

conv=1.00E-03

derv=mg

conv=1.00E-06

**…**

**units\_other.txt**  (komplett):

unit=Bq s

ubase=Bq\*s

unit=m2

ubase=m^2

unit=m³

ubase=m^3

unit=cm2

ubase=cm^2

unit=cm3

ubase=cm^3

In den meisten Fällen ist der mit dem Wechsel der Einheit der Messdauer verknüpfte Skalierungsfaktor für eine Zählrate *R* derselbe wie für *u*(*R*), solange dabei allein die Poisson-Statistik anwendbar ist. Eine Ausnahme davon bildet die Bruttozählrate aus Kapitel 6.10, die auf eine Summe von Binomial- und Poisson-verteilten Größen zurückgeht. Für einen Kalibrierfaktor *w* bzw. *phi*, der sich als verallgemeinertes Produkt darstellen lässt, ist der Skalierungsfaktor für *w* bzw. *phi* derselbe wie für *u*(*w*) bzw. *u*(*phi*).

### 7.21.2 Erläuterung zur Berechnung von Einheiten abhängiger Größen

Das Verfahren für eine abhängige Größe basiert auf der dazugehörigen in UncertRadio eingegeben Gleichung. Die rechte Seite der Gleichung ist allgemein ein arithmetischer Ausdruck von darin enthaltenen Variablennamen. Um Einheiten zu berechnen, werden im ersten Schritt alle Variablennamen durch die Namen der Einheiten ersetzt; ein Variablenname „eps“ für eine Detektor-Nachweiswahrscheinlichkeit wird durch „(1/Bq/s)“ ersetzt; die Klammer darum soll sicherstellen, dass damit richtig gerechnet werden kann. Später werden die darin enthaltenen Teileinheiten, „Bq“ und „s“ durch die in Spalte B der Datei unitsTable.csv angeführten numerischen Basiseinheiten-Werte „11“ und „21“ ersetzt.

Um mit Basiseinheiten-Werten rechnen zu können, müssen noch folgende Modifikationen an der inzwischen erhaltenen Formel ausgeführt werden:

* die rechte Seite einer Gleichung mit der Nummer besteht aus einzelnen Größen, mit nummeriert; die Einheit jeder Größe kann sich aus einer oder mehreren Einheitenteilen zusammensetzen. Die -te Größe hat den Index oder Adresse in der vollständigen Symbol-Liste von UR.
* Außerhalb von Funktionsargumenten werden alle Minus-Zeichen durch Plus-Zeichen ersetzt; damit soll vermieden werden, dass z.B. bei einer einfachen Nettozählrate die Differenz zweier gleicher Einheiten-Werte Null wird;
* Für in der Formel auftretende Funktionen, das sind Log(), Exp() und Sqrt(), werden in deren Argumenten ebenfalls die Variablenamen durch Einheitennamen und auch Basiseinheitenwerte ersetzt. Damit wird das Argument einer solchen Funktion numerisch ausrechenbar. Für dessen Berechnung wird ein **zweiter einfacher Funktions-parser in UncertRadio benutzt, seval genannt**, der nicht erst Variablen mit Zahlen ersetzen muss, sondern mit Zahlen direkt rechnet, d.h. mit Zahlen als Bestandteil eines Formel-Strings.
* Wenn das Argument von Log() (meistens 2) keine Einheit enthält, d.h. eine Zahl darstellt, wird der Ausdruck Log(Argument) gleich 1 gesetzt.
* Die Einheit einer Eingangsgröße der Nummer kann aus mehreren Teileinheiten bestehen, die eine einfache Formel bilden. Für eine Detektornachweiswahrscheinlichkeit , kann die Einheit zum Beispiel 1/mBq/min sein. Die Teileinheiten werden in Basiseinheiten gewandelt und aus den dabei zu verwendenden einzelnen Skalierungsfaktoren wird der Skalierungsfaktor für die zusammengesetzte Einheit berechnet: im Beispiel ist , wenn die Zieleinheit 1/Bq/s sein soll.
* Um rechnen zu können, wird für das Beispiel von ein Einheitenstring gebildet, der sich aus den beteiligten Basiseinheitenwerten zusammensetzt (vgl. unitsTable.csv): „(1/11.0/21.0)“.
* In einer Gleichung (für eine abhängige Größe) wird jeder einzelne darin enthaltene Variablenname (Nummer ) durch einen solchen String ersetzt. Der Skalierungsfaktor für die Größe wird wie folgt bestimmt:

(1)

Hierin ist der String der Formelstring der Gleichung , in dem die Namen der Symbole durch das Produkt als String ersetzt wurden; ist der Formelstring der Gleichung , in dem die Namen der Symbole nur durch als String ersetzt wurden.

Beispiel für (LAMSR \* TS \* 60^0) / (1. - EXP(-LAMSR \* TS \* 60^0)):

strgv1=(( 7.63000000E-10) \* ( 2.40000000E+04) \* 60^0) / (1. - EXP(-( 7.63000000E-10) \* ( 2.40000000E+04) \* 60^0))

strgv3=(( 7.63000000E-10) \* ( 4.00000000E+02) \* 60^0) / (1. - EXP(-( 7.63000000E-10) \* ( 4.00000000E+02) \* 60^0))

uconv(i) = 1.00000906

* Es wird angenommen, dass im Argument einer Exp-Funktion Größen wie eine Zerfallskonstante *lambda* (1/s) und eine Messdauer *t* (s) stehen. Daher wird für das Argument zugelassen, dass neben den darin eingesetzten Basiseinheitenwerte nur noch Skalierungsfaktoren von , oder und gleichzeitig mit Hilfe von Triggervariablen gebildete dazugehörige Faktoren wie usw. auftreten. Wenn also der Wert des Arguments nicht ganzzahlig und ungleich , bzw. ist, wird ein Einheitenfehler angenommen, andernfalls wird der gesamte Exp(Argument)-Ausdruck gleich 1.0 gesetzt (Exp(Argument)=1).

Oft kommt ein Exp()-Ausdruck auch in der Form (1.0 – Exp()) vor. Das Minuszeichen darin wird durch ein Pluszeichen ersetzt. Wenn die Analyse des Exp()-Arguments zu dem Ergebnis geführt hat, dass der Exp()-Term gleich 1 gesetzt werden kann, wird der gesamte Ausdruck (1.0 – Exp()) gleich 1 gesetzt. Dazu muss nur noch, vom Teilstring EXP aus nach links gehend, nach der öffnenden Klammer gesucht werden.

* Bei einer Summe im Argument von Sqrt(), beispielsweise für eine Varianz einer Nettozählrate mit z.B. drei Termen der Einheit (1/s^2), wird sich der Wert 3.0 \* (1/s^2) als Argument ergeben. Wenn sich dabei nicht die 3.0 als Faktor ergibt, sondern eine krumme Zahl, hat einer von drei Termen eine andere, vermutlich eine falsche Einheit. Wenn aber eine glatte ganze Zahl als Faktor herauskommt, kann der gesamte Wurzelausdruck durch den Einheitenwert 1 ersetzt werden.

Nach der wie eben beschriebenen Ersetzung von Funktionsausdrücken durch Einheitenwerte-Strings sollte sich ein mit direkt ausrechenbarer Formelausdruck (als String RSeiteG2(i)) für eine Gleichung ergeben. Der sich ergebende Zahlenwert sei

(2)

Dabei können sich einige Einheiten quasi herausgekürzt haben.

Man steht also vor der Frage, wieviel – und welche – Basiseinheiten sind an dieser Formel letztlich noch beteiligt? Dazu werden in diesem Formelstring, in dem Basiseinheitenwerte wie z.B. „21.0“ für „s“ stehen, die einzelnen Zahlenwertstrings wieder durch die dazugehörigen Basiseinheiten-Namen (Strings) ersetzt. Mit den aus der Datei unitsTable.csv aus den ersten beiden Spalten zu entnehmenden Paaren „Einheitenname, Einheitenwert“ kann dann der komplexere Funktions-parser *parsef* dazu benutzt werden, partielle Ableitungen des Formelstrings nach den Einheiten zu berechnen. Nur einzelne Einheiten, deren partielle Ableitungen nicht (praktisch) Null sind, tragen zur gesuchten Einheit bei.

Nachdem auf diese Weise die beteiligten Teileinheiten bekannt sind, z.B. „Bq“, „s“ und „kg“, ist noch die Frage zu klären, welche davon – als Teil eines verallgemeinerten Produkts – im Zähler und im Nenner stehen. Bei diesen drei Einheitenwerten, jetzt mit a, b und c abgekürzt, bestehen folgende Möglichkeiten:

(3)

Diese 8 Möglichkeiten werden gerechnet; wenn eine daraus den oben erwähnte Wert *Evalue* ergibt, ist die richtige Kombination hinsichtlich Zähler oder Nenner gefunden: z.B., „Bq\*s/kg“, wenn der Wert *Evalue* gleich 11.0\*21.0/41.0 ist.

### 7.21.3 Anpassungen im Verfahren

Zuerst werden die Einheiten aller (unabhängigen) Eingangsgrößen durch die Basiseinheiten ersetzt. Wenn bei einer Eingangsgröße ein Skalierungsfaktor ungleich 1 entsteht, wird der dazugehörige Wert der Größe (temporär) mit diesem Faktor multipliziert.

Im zweiten Schritt werden die Einheiten der abhängigen Größen aus den Einheiten der Eingangsgrößen wie oben beschrieben berechnet.

Die Skalierungsfaktoren für die abhängigen Größen werden nicht aus der Einheitenbehandlung abgeleitet. Nachdem die Eingangsgrößenwerte durch Modifikation ihrer Einheiten geändert wurden, werden stattdessen die Werte der abhängigen Größen intern mit der Funktion Resulta neu berechnet, ohne mögliche Einheitenanpassungen dabei zu berücksichtigen. Die dazugehörigen Einheiten-Konvertierungsfaktoren unit\_conv\_factor() werden erst danach berechnet: ganz einfach als Quotienten der neu berechneten Größenwerte und ihrer vorherigen Werte. Die Werte der Unsicherheiten werden anschließend mit diesen Fakoren unit\_conv\_factor() skaliert.

### 7.21.4 Ausführung des Tests der Einheitenberechnung

Das Berechnen von Einheiten der abhängigen Größen kann unter dem Menüpunkt „Bearbeiten – physik. Einheiten testen“ aufgerufen werden. Dazu muss das Projekt so weit entwickelt sein, dass unter dem TAB „Resultate“ Werte verfügbar sind.

**Wichtig**:

* Die Berechnung von Einheiten führt zur Umrechnung auf die Basiseinheiten. Wenn andere Basiseinheiten gewünscht sind, müssen letztere in der Datei unitsTable.csv umdefiniert werden. Man kann z.B. das kg durch das g als Basiseinheit ersetzen und kg zur abgeleiteten Einheit machen.
* Wenn Trigger-Variable im Projekt vorhanden sind, wird das Aufrufen des Testmodus verhindert. Der Grund ist, dass ein aus diesem Modus heraus als neue Datei gespeichertes modifiziertes Projekt nicht ordentlich funktioniert.
* Wenn der Test noch Fehler anzeigt, muss der Testmodus mit dem expliziten Schließen des Editor-Tabs beendet werden! Damit werden die ursprünglichen Werte wiederhergestellt.
* Wenn keine Einheiten-bezogene Fehler angezeigt werden, erscheint ein zusätzlicher Button. Dieser erlaubt, das ganze Projekt als neue Datei zu speichern. Das ist aber nur dann nötig, wenn Abweichungen beim Wert der Ergebnisgröße angezeigt werden.

Das Programm führt intern die Berechnungen nach Abschnitt 7.21.2 durch und stellt danach im programm-internen Editor eine Gegenüberstellung für die Liste der Symbole dar.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Für abhängige Größen werden die anfangs vorhandenen Einheitennamen durch die „berechneten“ Einheitennamen ersetzt, d.h. das Projekt wird dadurch modifiziert, was zu beachten ist

Für die Ausgabe des Tests berechnet UncertRadio die skalierten Messwerte und Standardunsicherheiten der *abhängigen* Variablen (*MVals\_scd* und *StdUnc\_scd*) wie folgt.

* Die Spalte „MVal\_scd/MVals\_org“ zeigt die entstandenen Skalierungsfaktoren unit\_conv\_factor().
* Die Werte *Mvals\_scd* (abhängig) allein durch Berechnung mit der Funktion Resulta aus den Werten der unabhängigen Eingangsgrößen.
* Alle Werte *StdUnc\_scd* durch Skalierung der schon bekannten Werte (*StdUnc\_org*) mit den Faktoren unit\_conv\_factor().

Die Ausgabe der aus den Berechnungen resultierenden Gegenüberstellung im Editor beginnt jedoch mit einer Fehlermeldung, wenn z.B. für die Einheit einer Größe der Vergleich zwischen dem Wert *Evalue* (Gl. (2) und Werten nach Gl. (3) (voriger Abschnitt) zu keiner Übereinstimmung führt.

### 7.21.5 Einführung Trigger-Variablen

Das Programm kann keine Hilfe geben, wie zu verfahren ist, wenn der genannte Vergleich misslingt. Ursache kann ein in der Gleichung vorhandener Umrechnungsfaktor sein, der bei Umrechnung auf eine Basiseinheit intern zusätzlich berücksichtigt würde. Solche schon vorhandenen Umrechnungsfaktoren können mit Schaltvariablen (siehe Kapitel 2.2.6) wie folgt „unschädlich“ gemacht werden: Für „Minuten (min)“ oder für „Gramm (g)“ können vorhandene Umrechnungsfaktoren 60 oder 1/1000 wie folgt ersetzt werden (oder man streicht sie gleich ganz und ändert die Einheit der betreffenden Größe in s):

60 🡪 60^min\_Trigger

1/1000 🡪 1/1000^kilo\_Trigger

Diese beiden speziellen Triggernamen werden von UncerRadio erkannt, sie erhalten beide temporär den Wert Null, wenn die Option „physik. Einheiten testen“ ausgeführt wird, oder den Wert 1 im anderen Falle.

Falls in der erwarteten Einheit der Ergebnisgröße, im Falle einer Aktivität, kein „Bq“ vorkommt, sondern nur „1/s“, ist das Problem meistens dadurch zu lösen, dass man der oder den Detektornachweiswahrscheinlichkeiten die Einheit „1/Bq/s“ zuordnet.

**Übungsbeispiel: Janszen-Sr-89-Sr-90\_V2\_DE.txp**

Der Test endet zunächst mit einer Fehlermeldung. Dann nimmt man folgende Änderungen vor:

Die ersten beiden Gleichungen:

a89 = As89 / (ms/1000.)

a90 = AS90 / (ms/1000.)

werden geändert in:

a89 = As89 / (ms/1000.^kilo\_Trigger)

a90 = AS90 / (ms/1000.^kilo\_Trigger)

Um jetzt noch die gewünschte Einheit „Bq/kg“ zu erhalten, müssen im TAB „Gleichungen“ in der Tabelle der Symbole die vier Symbole für Detektor-Nachweiswahrscheinlichkeiten noch die Einheit 1/Bq/s bekommen.

Die derart geänderte Projektdatei steht als **Janszen-Sr-89-Sr-90\_V3\_DE.txp** zur Verfügung.

Wenn man einen Einheitenfehler einsetzt, z.B. in der Symboltabelle im TAB „Gleichungen“ die Einheit von t2m0 von „s“ in „min“ ändert, danach das Projekt bis zum TAB „Resultate“ rechnen lässt und dann den Test aufruft, erhält man eine (zu erwartende) Fehlermeldung, in diesem Fall:

Error messages:

Eq. #=8 Error CLCU: Units in EXP argument do not match: seval=-60.0000000 arg(EXP)=-(1.0/21.0) \* ( 6.000000E+01\*(21.0))

Eq. #=4 RD89 = RSr - w\*RY\*f7 - Abl\*(etaSr \* epsSrSr89 \* f1): no unit found! Einvor=1 RSide=RSR-W\*RY\*F7-ABL\*(ETASR\*EPSSRSR89\*F1)

Ähnliche Änderungen waren auch im Projekt **Moreno-Sr90\_IAEA-135\_EN.txp** erforderlich.

Die Einheitenangaben „u (uamu)“ wurden durch „u“ ersetzt. In den Gleichungen für die Größen a, w, f2 und epsSr wurden die Schaltvariablen kilo\_Trigger bzw. min\_Trigger eingesetzt. Eine Einheit „mg Sr“ wurde durch „mg“ ersetzt.

Die derart geänderte Projektdatei steht als **Moreno-Sr90\_IAEA-135\_V2\_EN.txp** zur Verfügung.

### 7.21.6 Erfahrungen mit der Option zum Berechnen von Einheiten

Der Umgang mit der Option zum Testen bzw. Berechnen von physikalischen Einheiten konnte mit der Version 2.4.21 deutlich verbessert werden. So konnte diese Option das erste Mal in den „QC-Batch-Test“, d.h. in den Lauf über die Beispielprojekte integriert werden. Dabei wurden einige Projekte identifiziert, die sich durch eine spezielle Wahl von Einheiten auszeichneten, z.B.,

Wuebbeler\_Ex2\*.txp (Ohm, Vol, Ampere)

Sterlinski\*.txp (Aktivierung, Einheit ng/g).

Bei den Projekten Kessel\*.txp und Calibration-of-weight-Cox\*.txp (calibration of masses) wurden Abweichungsfaktoren von 1000 gefunden. Bei den Projekten Calibration-of-weight-Cox\*.txp ist die Einheitenermittlung noch zu schwierig. Bei den Projekten Neutron-dose-Cox-2006\*.txp ergab sich durch Änderungen der Einheiten eine Änderung des Ergebniswerts von 100 durch 36.

Diese Projekte wurden so belassen.

Ein echter Einheitenfehler wurde bei den insgesamt 4 Projekten sumEval\*V2\*.txp (DE+EN) gefunden. Bei der Version V2 wurde seinerzeit zwar die Einheit der Ergebnisgröße auf Bq/m² gesetzt, die dazugehörige Fläche von 400 cm² wurde jedoch nicht in 0,04 m² geändert. Dieser Fehler wurde bei den V2-Versionen dieser Projekte sofort angezeigt. Die vom Programm vorgeschlagenen korrekten Versionen wurde nun als Version V3-Projekte gespeichert.

Bei den Projekten Moreno-Sr90\_IAEA-135\_V2\*.txp wurden die Trigger und die expliziten in den Gleichungen enthaltenen Umrechnungsfaktoren wie 60 oder 1000 entfernt und dann erfolgreich auf die Einheiten getestet. Mit der Einheitenberechnung konnte das Fehlen der expliziten Faktoren wieder ausgeglichen werden. Wert und Unsicherheit der Ergebnisgröße änderten sich am Ende nicht. Die mit dieser Programmoption erzeugten Projekte werden als Version V3 zur Verfügung gestellt.

Ebenso wurde bei dem Projekt Janszen-Sr-89-Sr-90\_V3\*.txp verfahren. Der kilo\_trigger und der Faktor bei der Masse in mg wurden entfernt. Das Fehlen des Faktors 1000 in den Gleichungen wurde durch die Berechnung der Einheiten wieder ausgeglichen. Damit wurde die Version V4 dieser Projekte erzeugt, Wert und Unsicherheit sind gleich denen von der Version V3.