Version 2.3.0

Installation Guide
Source Distribution

April 20, 2023

Wolfgang Sturhahn

https://www.nrixs.com support@nrixs.com

Contents

1 License information

CONUSS-2.3.0 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

CONUSS-2.3.0 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is distributed with CONUSS-2.3.0 and can also be seen at `http://www.gnu.org/licenses/gpl.html`.

Any use of results obtained using CONUSS-2.3.0 in related or unrelated publications have to be properly acknowledged by reference to the name of the package, to the name of the developer(s), and to the *NRIXS software* site `https://www.nrixs.com`.

2 What is CONUSS ?

The CONUSS software is a scientific application to simulate experimental data obtained using the techniques of Synchrotron Mössbauer Spectroscopy (SMS), Nuclear Forward Scattering (NFS), grazing-incidence nuclear resonant scattering (GINS), nuclear resonant Bragg/Laue diffraction (NBS), and conventional Mössbauer spectroscopy. The software calculates time and energy spectra and supports automated parameter optimization (fitting).

The first version of the program was created from 1983 to 1986 by E. Gerdau and W. Sturhahn at the University of Hamburg shortly after the discovery of nuclear resonant scattering of synchrotron radiation using ^{57}Fe-enriched single crystals of Yttrium-iron-garnet. It was improved since then by W. Sturhahn to handle various data input formats and to provide comprehensive diagnostic tools for high-quality data evaluation. A detailed treatment of nuclear forward scattering was added in 1991, automated parameter optimization was added in 1996, and a graphical display option was provided in 2010. A new approach to parameter space exploration using randomized trials in combination with a Beam Search method was added in 2010. Nuclear resonant scattering by thin-film systems in grazing incidence geometry is supported since 2015.

CONUSS-2.3.0 supports all Mössbauer isotopes and offers great flexibility in defining sample properties, such as, arbitrary number of sites, combined hyperfine interactions, distributions of hyperfine interactions, partial alignment, and Blume-Tjon-type magnetic relaxation. The program fully implements polarization, directional, and thickness dependences for time as well as energy spectra. Further it supports sample combinations, averaging in time, energy, angle, and thickness distributions. In 2023, various energy scan modes as well as the TDI mode have been added to support evaluation of energy-resolved Synchrotron Mössbauer Spectroscopy and time-domain-interferometry to investigate non-resonant glassy scatterers. The flexible assignment and grouping of fit parameters permits efficient evaluation of experimental data.

The software was created by W. Sturhahn to offer traceable evaluation codes for publications using coherent elastic NRS techniques. The CONUSS core programs are written in Fortran77 and Fortran90 with wrappers using bash scripts. The CONUSS software has been used for data evaluation in numerous publications and has been extensively tested over the last three decades. CONUSS and the other software packages of *NRIXS software* install on UNIX-like operating systems: Sun's Solaris, Apple's MacOS, Ubuntu, and various other Linux versions. For installation on Microsoft's Windows operating systems, a virtual machine, e.g.,VirtualBox (free) or VMware (paid), hosting a Linux-type guest-system is needed.

3 Requirements

Before installation please verify the following list of requirements. You must install these before installation of CONUSS.

3.1 Linux and Unix

- bash shell, should be installed already.
- Fortran90 compiler. The recommended compiler is gfortran 4.8 or later, available in native software installer.
- Optional: xmgrace plotting tool, available in native software installer.

3.2 MacOS

The order of installation is most important here.

- `Xcode`, Apple's developer package freely available from the Appstore.
  Run `Xcode` to accept the license and install the 'linecommand tools': open Terminal app and type `sudo xcode-select --install` on the command line. You must have administrator privileges for this.
- bash shell, should be installed by Xcode.
- macports or homebrew package installer.
- Fortran90 compiler. The recommended compiler is gfortran, version 4.8 or later, available via macports or homebrew package installer. Binary packages are available here.
- Optional: Xquartz for run-time graphics.
- Optional: xmgrace plotting tool, available via macports or homebrew package installer.

4 How to install

CONUSS is distributed as a compressed tar-ball named `CONUSS-2.3.0_source.tar.gz`. Several steps are needed to install the CONUSS software. If this is an update from an earlier version of CONUSS CONUSS it is recommended that you uninstall the earlier version as described in section 6.

4.1 MacOS: remove restrictions

Before you extract the files, you must defeat the Gate-keeper mechanism and you need administrator privileges for this. Assume Dir as the location of CONUSS-`2.3.0_source.tar.gz`. Open Terminal app and enter

```
> cd Dir
> xattr CONUSS-2.3.0_source.tar.gz
>   ... list of extended attributes ...
> sudo xattr -c CONUSS-2.3.0_source.tar.gz
> Password:
> xattr CONUSS-2.3.0_source.tar.gz
>
```

4.2 Extract files

Depending on the available system utilities you may have several options to extract the CONUSS files. For example, the line command

```
> gzip -dc CONUSS-2.3.0_source.tar.gz | tar xf -
```

recovers the files on most systems. Also a double-click on `CONUSS-2.3.0_source.tar.gz` recovers the folder `CONUSS-2.3.0` in many cases.

4.3 Configure, compile, and install

Next, run the installation script to create the configuration-dependent files, to compile the source code, and to install the executables. It is generally less complicated to install locally, i.e., into a user directory. For system-wide installation, proper authorization is required. On MacOS, authorization is requested when needed during the installation. Otherwise the install script might have to be run as a 'sudo' command. The installation process can be terminated by Ctrl-c.

Several options can be specified to adjust to local circumstances. For most cases, options are probably not needed. Enter the following line commands to see available options.

```
> cd CONUSS-2.3.0
> ./install --help
Usage: install [OPTION]...
Compile and install CONUSS-2.3.0

      --foption=<list> colon separated list of compiler options
      --fortran=<exe>  use Fortran compiler <exe>
                       <exe> must exist in search path
      --nographics     disable graphics support
      --help           display this help and exit
      --loption=<list> colon separated list of ld options
      --prefix=<dir>   set <dir> as installation location
                       only used in 'make install'
                       preset location is <home directory>
      --spath=<list>   append or prepend directories to search path
                       to prepend add leading _ to directory name
                       <list> is a colon separated list of directories
      --xlarge         enforce 64bit pointers (large data section)
      --xnoquad        disable quad precision arithmetic
      --xsmall         accommodate 32bit pointers (small data section)
      --xstatic        create statically linked executables

Examples:
  install --spath=$HOME/bin    prepend $HOME/bin to search path
  install --prefix=/usr/local  install into directory /usr/local
>
```

Here follows a description of the functionality of each option for the installation script.

foption list of options that will be passed to the Fortran compiler. This might be needed for some compilers. The correct syntax would be, for example, '--foption=-x1=a:-x2=b'. Everything after the first '=' character will be passed to the compiler with all ':' characters being replaced by spaces.

fortran name of the Fortran compiler executable. By default the configuration script assigns gfortran as compiler if is found in the search path (see description below). Change the compiler by '--fortran=/crazypath/fort'.

nographics disable run-time graphics support.

loption list of options that will be passed to the linker; see 'foption' above for syntax rules.

prefix set the location for installation of the CONUSS executables and their support files. By default the installation location is the installer's home directory: under MacOS the CONUSS app is copied into $HOME/Application and support files are copied into $HOME/Library/NRIXS/CONUSS; under other Unix systems the executables are copied into $HOME/bin and the support files are copied into $HOME/.NRIXS/`CONUSS`. If the prefix is set to another directory it is important to have appropriate permissions as installer, e.g., '--prefix=/usr/local' requires administrator or root privileges. Under MacOS support files are installed either into $HOME as described above or into '/' where the CONUSS application is copied into '/Applications' and support files are copied into '/Library/NRIXS/CONUSS'.

spath append or prepend directories to the search path for essential support programs including the compiler and run-time graphics programs. By default the path contains the following directories: /bin, /usr/bin, /usr/local/bin, /opt/bin, /opt/local/bin, /opt/homebrew/bin, /opt/X11/bin, /usr/X11/bin. If the install script fails to locate essential support programs in this path (reported upon execution of the script) directories have to be added. The correct syntax would be, for example, '--spath=/crazypath/dir1:~/dir2', where '~' symbolizes the user's home directory.

xlarge override default 32bit pointer size. Some systems by default use a pointer size of 32bit instead of 64bit which would be the native size for most modern operating systems. This puts a constraint on the maximum size of data fields defined at compilation time. A typical error message contains something like '`relocation truncated to fit: R_X86_64_PC32`' and the produced executables (if any) are defective.

xnoquad disable quad precision arithmetic. Not every system permits the compiler to use quad precision expressions. However, only calculations involving nuclear Bragg/Laue reflections require quad precision and would potentially fail with this option.

xsmall reduce the size of data fields. This may restrict some calculational capabilities such as number of sites (kfor and kref) and the number of angles/thicknesses (kfor, kgin, and kref). These settings can be fine-tuned by adjusting parameters in the file 'CONUSS-2.3.0`/src/conuss.h`' followed by re-compilation and re-install.

xstatic created statically linked executables. This option is meant for developers and not recommended for normal use.

Execute the install script with appropriate options.

```
> ./install [OPTIONS...]
  ....
  (messages)
  ....
>
```

The installation script may be re-run with different options as needed. The last install command is saved in the file 'installcmd'. The script produces various messages on the progress of the configuration, compilation, and installation. After installation, you have the option to run a test of the CONUSS executables, see section 7 for more details. The test may take more than 30 minutes and can also be run at a later time.

The install requires write access to the installation directory, by default the users home directory. If DIR is the installation directory the following files are copied into DIR/bin: kctl, kctl-`2.3.0`,

kdec, kdec-2.3.0, kfan, kfan-2.3.0, kfmf, kfmf-2.3.0, kfor, kfor-2.3.0, kgin, kgin-2.3.0, kgmf, kgmf-2.3.0, kmcc, kmcc-2.3.0, kmco, kmco-2.3.0, kmix, kmix-2.3.0, kpnr, kpnr-2.3.0, kref, kref-2.3.0, krmf, krmf-2.3.0, kvzz, kvzz-2.3.0, conuss_test, mca2exp .

The installed executables are only accessible directly by line command if DIR/bin is part of the '`path`' setting in the login resource file in your home directory. This can be tested by typing '`echo $PATH`' or '`echo $path`' at a terminal prompt. If the directory DIR/bin is not part of the listing then the login resource file, usually something like '`.bash_profile`', '`.profile`', or '`.login`', must be edited to include DIR/bin in the '`path`' setting. After that you have to logout and login again to update the '`path`' settings.

4.3.1 Search path

Pay attention to the search path.

```
> ./install --spath
/bin:/usr/bin:/usr/sbin:/usr/local/bin:/opt/local/bin:/opt/X11/bin:/usr/X11/bin
>
```

If the location of the compiler or your xmgrace executable is not included in this search path you must add those locations to the path. For example, you installed xmgrace into directory '~/mybin' ('~' symbolizes the user's home directory) and want the install script to use it then you execute the install script with option

```
> ./install --spath=~/mybin
  ....
  (messages)
  ....
>
```

The directory is now appended to the original search path. Sometimes you need to overwrite an installation present in the search path. Then you must prepend the directory to the search path using a preceding underscore

```
> ./install --spath=_~/mybin
  ....
  (messages)
  ....
>
```

Use the prepend option with care to avoid unwanted command overrides.

5 File locations

5.1 Binaries

The executable binaries are placed into the directory 'DIR/bin' for a install into directory DIR, i.e., './install --prefix=DIR'. The default for DIR is the home directory.

5.2 Support files

For MacOS operating systems, support files are placed into directories '$HOME/Library/NRIXS/-CONUSS' or '/Library/NRIXS/CONUSS' for a system-wide install. On Unix-type systems, support files are placed into directories '$HOME/.NRIXS/CONUSS' or 'DIR/share/NRIXS/CONUSS' for a system-wide install into directory DIR. If a particular directory exists the content will be saved into a same-name directory with a number appended.

6 How to uninstall

The CONUSS program package is uninstalled by the command

```
> cd CONUSS-2.3.0
> ./uninstall
  ....
  (messages)
  ....
>
```

This step removes all files that were created during installation and is recommended prior to installation of a new version of CONUSS.

7 How to test

Examples are provided with the CONUSS package. They are located in the directory 'CONUSS-2.3.0/`examples`' and after installation also in the 'examples' directory as support files. The CONUSS self-test script is located in the directory 'CONUSS-2.3.0/bin' and can be run as follows

```
> cd CONUSS-2.3.0
> bin/conuss_test --dir=examples
  ....
  (messages)
  ....
>
```

The complete test may take more than 30 minutes depending on your computing system. The test can be canceled at any time with Ctrl-c. The test of each example produces a report with extension '.log' which contains potential deviations as well as execution-time output.

```
> ls -px *.log
kctl-GINS.log kctl-MBS1.log kctl-MBS2.log   kctl-NBS1.log
kctl-NBS2.log kctl-NFS1.log kctl-NFS2.log   kctl-NFS3.log
kctl-NFS4.log kctl-TDI.log  kdec-COMPAS.log kdec-MBS1.log
kdec-MBS2.log kfan-1.log    kfan-2.log      kfmf-MBS.log
kfmf-NFS.log  kfor-NFS.log  kgmf-GINS.log   kgmf-GIS.log
kmco-GINS.log kmco-NBS.log  kmco-NFS.log    krmf-NBS1.log
krmf-NBS2.log
>
```

7.1 Individual examples

Your computer system can only access the CONUSS executables if the directory that they are located in is part of the '`path`' defined for your system at the time of login. You can test this, e.g., by trying something like

```
> cd ~
> which kctl
/usr/local/bin/kctl
```

If the second command doesn't succeed the location of the CONUSS executables can be made known permanently to your computer. This is done by modification of the '`path`' setting in the login resource file in your home directory, usually something like '`.bash_profile`', '`.profile`', or '`.login`'. You have to logout and login again to update the '`path`' settings.

Now change into an '`examples`' directory (you need write access). If you don't have write access to the examples directory you should copy the content of an '`examples`' directory to an accessable location, e.g., your home directory. For example, enter something like

```
> cd ~/CONUSS-2.3.0/examples/kctl-NFS1
> ls -px
Fe.dat Fe.mif Results/ in_kctl in_kfit in_kfor in_kmix
> kctl --help
Usage: kctl [OPTION]...
Run CONUSS executable kctl-2.3.0

      --geometry=<XxY+U+W> set the window geometry for
                           graphics display.
      --help               display this help and exit
      --infile=<file>      use <file> as input file
                           the default input file is 'in_kctl'
      --nographics         disable runtime visualization
      --quiet              suppress most print out
      --version            display version number and exit
      --vtool              list supported visualization tools
      --vtool=<tool>       use <tool> for data visualization

Examples:
  kctl --infile=MyFile  use 'MyFile' as input file
  kctl --vtool=xmgrace  use xmgrace for visualization

> kctl

++ CONUSS-2.3.0 Copyright (C) 2023 Wolfgang Sturhahn ++

-- CONUSS module KCTL data section size: 883 Mb
-- execution starting..

  ....
  (messages)
  ....

-- CPU time :  user   0.59 s  system   0.05 s
-- CONUSS module KCTL finished

> ls -px
Fe_BestFit/      Fe.dat           Fe.mif
Fe_exp.dat       Fe_exp_pwr.dat   Fe_fit.dat
Fe_fit_pwr.dat   Fe_fsh.dat       Fe_kctl.csv
Fe_kctl_ptl.txt  Fe_kfor_let.dat  Fe_kfor_log.txt
Fe_rsd.dat       Fe_rsd_pwr.dat   Results/
in_kctl          in_kfit          in_kfor
in_kmix          kforParms.txt
>
```

Several files were created during this fit of an forward scattering time spectrum of iron metal at ambient conditions. Compare the content with files provided in the '`Results`' directory.The meaning of file contents is explained briefly at the end of the Fe_kctl_ptl.txt file and in more detail in the CONUSS Manual.