DISCUS SUITE

Users Guide

Version 6.21

written by

Reinhard Neder
Email: reinhard.neder@fau.de

http://tproffen.github.io/DiffuseCode

Document created: December 22, 2025

Preface

Disclaimer

The DISCUS SUITE software described in this guide is provided without warranty of any kind. No liability is taken for any loss or damages, direct or indirect, that may result through the use of the DISCUS SUITE. No warranty is made with respect to this manual, or the program and functions therein. There are no warranties that the programs are free of error, or that they are consistent with any standard, or that they will meet the requirement for a particular application. The programs and the manual have been thoroughly checked. Nevertheless, it can not be guaranteed that the manual is correct and up to date in every detail. This manual and the DISCUS SUITE program may be changed without notice.
DISCUS SUITE is intended as a public domain program. It may be used free of charge. Any commercial use is, however, not allowed without permission of the authors.

Using DISCUS SUITE

Publications of results totally or partially obtained using the program DISCUS SUITE should state that DISCUS SUITE was used and contain the following reference:

> NEDER, R.B. in preparation - check website.

More information

This users guide can only provide program specific details. A broader discussion of simulation techniques and some DISCUS SUITE examples and macro files can be found in our book

> NEDER, R.B. & PROFFEN, TH. "Diffuse Scattering and Defect Structure Simulations - A cook book using the programs DISCUS", *IUCr Texts on Crystallography*, Oxford University Press, 2007.

Contents

Chapter 1

Introduction

1.1 What is DISCUS_SUITE ?

The DISCUS SUITE is a combination of the components of the DISCUS program suite. It includes the sections DISCUS, DIFFEV, REFINE, KUPLOT , and EXPERI , under a common command language. The command language is documented in the manual package_man.
In this combination, the DISCUS SUITE allows you to seamlessly switch between the individual sections.

- **DISCUS**: This is the main diffuse scattering and defect structure simulation section. This is the section that gave the package its name. Within this section crystal structures are build, defects are introduced, and the corresponding diffraction pattern can be calculated

- **DIFFEV**: This section allows the refinement of a set of high level parameters (*e.g.* input parameters of a DISCUS macro) based on a differential evolutionary algorithm.

- **REFINE**: This section section allows a classical least squares refinement.

- **KUPLOT**: This is is the plotting and data manipluation section of the package.

- **EXPERI**: This is the data processing section of the package.

See the DIFFEV and REFINE manuals for full details of the refinement process.
As of Versions 6.02.00 and later, the DISCUS SUITE is compiled with OpenMP, a standard for parallel processing. This will accelerate the calculations: powder, fourier and mmc within DISCUS. See the command language documentation for the commands `set` and `show` to control the number of parallel threads that will be executed.

1.2 Getting started

1.2.1 Windows

Once the DISCUS SUITE has been installed you should see an icon called `Discus_SUITE_WSL` on your personal desktop. Double click the icon to start the program of your choice.

At program start, DISCUS SUITE sets its starting folder to

    `C:\Users\your_name`

where `your_name` will be your user name. The string `Users` may be slightly different depending on the language settings. To work with data and macros that are stored in a separate folder, you need to change to this folder.

Within the DISCUS SUITE program window, type the command `cd` including a blank space after the `cd`. At this point do not hit the enter key. Open the desired folder with the Windows-Explorer. Left click in the top line that contains the path to your folder. This should highlight the full path to the folder.

Select the highlighted path to the folder with CTRL-c. Activate the DISCUS SUITE window. Press the SHIFT key and press the right button on the mouse. This should open a menu. Please choose the "Paste" option to place the full path into the program window. You can also press CTRL-SHIFT-v to insert the directory path. Incidentally this holds for any string that had been copied into the clipboard with CTRL-c.

Once you activate the window and hit the ENTER key the program will work in this folder.

Roughly every week the program will automatically update the Windows-Subsystem-Linux (Ubuntu) on which the discus-suite runs. This will be performed upon the exit from the suite and should not take long.

To install future updates simply type the discus_suite command `update`.

As the DISCUS SUITE runs on top of the Windows-Subsystem-Linux, a Linux distribution, access to the removable disks is slightly different than for a original Windows program. Unfortunately, to date the Windows-Subsystem-Linux does not perform an automatic mounting of removable drives such as a USB drive. The suite will do such a mount process if you use the command `cd f:` to access a drive `f:`. As the DISCUS SUITE cannot know if you do not need this drive any longer, the drive is not automatically dismounted once you change back to the `C:` drive or even when you exit the program. Other instances of the DISCUS SUITE might still need the drive. Thus it is left to you to dismount a drive with the `umount f:` command once your are finished.

Folders, Data storage

On a Windows operating system, the easiest might be to use a folder
`c:\Users\neder\Documents\DISCUS_WORK\`
for all data and macros that you plan to use within the DISCUS SUITE. Replace "neder" with your user profile name. Feel free to use any other name than "DISCUS_WORK" for your work folder. To navigate to this folder type the commands:
`cd $HOMEPATH`
and
`cd Documents\DISCUS_WORK`
at the the Suite prompt. Currently these have to be two separate commands, as DISCUS SUITE recognized the special name `$HOMEPATH` only at the first line.

As the DISCUS SUITE is developed at Linux computers and runs within the Linux environment WSL on your Windows computer, the performance can be improved if you do run the process

entirely within the WSL envirnoment. This is particularly effective if your macros do a lot of File I/O.

You have two choices how to run DISCUS SUITE within the WSL environment:

The first option is to run the Linux environment and to start DISCUS SUITE within this environment. This requires basic knowledge of a Linux operating system. For the second option, type
`cd $WSL_DOC`
at the use DISCUS SUITE prompt. This will make the directory `$HOME/Documents` your current directory. To copy data into and from this directory, start the WindowsExplorer. Choose the `Network` drive and type into the top line that contains the path to your folder:
`\\wsl.localhost\home\discus\Documents`
Here it is assumed that you created a WSL user name `discus`, adapt if necessary. Please make sure that you do not use any other directory than the `Documents` directory within WSL as this bears the risk of messing up WSL. With the Windows File Explorer you can copy any other Windows directory into the WSL `Documents` directory. See the Figure 1.1 for an example. In this example, a directory `DISCUS_WORK` has been placed into the WSL `Documents` directory.
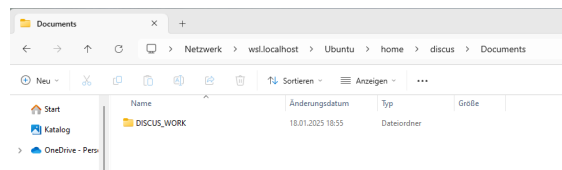

Figure 1.1: Example of the Windows File Explorer at WSL Documents directory.

The left bottom part of your Windoiws Explorere will probably contain a `Linux` icon as in Fig. 1.2. Once you click on this icon the `Ubuntu` directory will show up in the main explorerwWindow, and you can navigate into this directory and on to `home` and `Documents`.
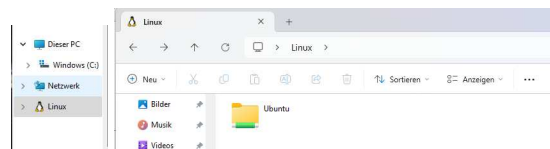
Figure 1.2: Example of the Windows File Explorer with Linux and resulting Ubuntu icon.

Within a macro, these graphic tools are not accessible. If you need to copy data to and from WSL use commands like

```
system cp /mnt/c/Users/neder/Documents/DISCUS_WORK/my_macro.mac /home/discus/Documents
system cp /home/discus/Documents /mnt/c/Users/neder/Documents/DISCUS_WORK/my_macro.mac
```

The first command copies the macro `my_macro.mac` from Windows into the WSL environment, the second command reverses this operation.

To navigate from Windows to WSL and back use commands like:

```
cd $WSL_DOC    ! Ends in /home/discus/Documents in the WSL environment
cp $HOMEPATH   ! Ends in C:\Users\neder\
```

The comment for the first command assumes that your WSL user name is `discus`. If you used a different name you will equivalently end up in `/home/user_name/Documents`. Likewise, the second command will return to your Windows User account with its proper name.

Please refer to the general command language Manual at package_man for a list of helpful Linux commands.

1.2.2 Linux

After the program *DISCUS_SUITE* has been installed properly and the environment variables are set, the program can be started by typing 'discus_suite' at the operating systems prompt.

To install future updates simply type the discus_suite command `update`.

1.2.3 MAC

After the program *DISCUS_SUITE* has been installed properly and the environment variables are set, the program can be started by first starting a terminal window and typing 'discus_suite' at the operating systems prompt.

To install future updates simply type the discus_suite command `update`.

1.2.4 General introduction

The program uses a command language to interact with the user. The command `exit` terminates the program and returns control to the shell. All commands of DISCUS SUITE consist of a command verb, optionally followed by one or more parameters. All parameters must be separated from one another by a comma ",". There is no predefined need for any specific sequence of commands. DISCUS SUITE is case sensitive, all commands and alphabetic parameters MUST be typed in lower case letters. If DISCUS SUITE has been compiled using the `-DREADLINE` option (see installation files) basic line editing and recall of commands is possible. For more information refer to the reference manual or check the online help using (`help command input`). Names of input or output files are to be typed as they will be expected by the shell. If necessary include a path to the file. All commands may be abbreviated to the shortest unique possibility. At least a single space is needed between the command verb and the first parameter. No comma is to precede the first parameter. A line can be marked as comment by inserting a "#" or `!` as first character in the line. All characters to the right of a "#" or `!` are treated as comment. This allows to place in-line comments into any command line.

The symbols used throughout this manual to describe commands, command parameters, or explicit text used by the program DISCUS SUITE are listed in Table 1.1. There are several sources of information, first DISCUS SUITE has a build in line help, which can be accessed by entering the command `help` or if help for a particular command <cmd> is wanted by `help <cmd>`. This manual describes background and principle functions of DISCUS SUITE and should give some insight in the ways to use this program.

Symbol	Description
"text"	Text given in double quotes is to be understood as typed.
<text>	Text given in angled brackets is to be replaced by an appropriate value, if the corresponding line is used in DISCUS SUITE. It could, for example be the actual name of a file, or a numerical value.
'text'	Text in single quotes exclusively refers to DISCUS SUITE commands.
[text]	Text in square brackets describes an optional parameter or command. If omitted, a default value is used, else the complete text given in the square brackets is to be typed.
{text | text}	Text given in curly brackets is a list of alternative parameters. A vertical line separates two alternative, mutually exclusive parameters.

Table 1.1: Used symbols

1.2.5 Overview

Autorun a startup macro

As of version 6.03.00 and later the DISCUS SUITEwill look for a macro file called `autorun.mac` within the starting folder. If this macro is found, the commands within this macro are executed prior to any further action.
Use this feature to customize default behavior. At Windows the macro "autorun.mac" must be placed into the folder:

```
C:\Users\your_name
```

Make sure to delete, move or rename this macro file if you want to disable the autorun feature. For Linux and MacOS, the DISCUS SUITE looks for the macro file in the current directory at which you start the program. You can disable its use by moving, renaming, or deleting the macro. Alternatively, if you use the command line option `-noautorun` a macro `autorun.mac` will be ignored. See the secion 1.2.5 for further details on command line options.

Command line options

Several command line options are available for the Linux and MAC version. The most important one is to start the execution of a macro:

```
discus_suite -macro useful.mac
discus_suite -macro useful_with_params.mac  par1 par2 par3
```

The first line would start DISCUS SUITE and begin the automatic execution of macro `useful.mac`. Likewise in the second line, the macro `useful_with_params.mac` is started, which takes three parameters. The actual strings that you provide for par1, par2, par3 are handed down as parameters to this macro. Note that there are no comma between the parameters.
another command line option is `-noautorun`. If used as in:

```
discus_suite -noautorun
discus_suite -noautorun -macro useful.mac
discus_suite -noautorun -macro useful_with_params.mac  par1 par2 par3
```

the macro `autorun.mac` is ignored, even if it is present within the start directory. If the command line option is omitted the macro `autorun.mac` if performed prior to any further action of DISCUS SUITE.

The macro started through the command line is placed into the command history, including its parameters. For `useful_with_params.mac` the first line in the command history would be:

```
@useful_with_params.mac  par1, par2, par3
```

For info on further command line options the the build-in help for the Command language.

1.2.6 Major Updates in this last version

In DISCUS SUITE version 6.02 several of the time consuming calculations have been moved to an automatic parallel calculation using openMP, a standard for parallel calculations. See the section `Parallel computing` in the general package manual DISCUS package for further details. This internal parallel mode should not be confused with the explicit parallel calculation as described in section 1.3 in this manual.

In DISCUS SUITE version 6.00 several new features were introduced:

A completely new refinement section REFINE allows you to run a fast Least-Squares refinement as an alternative to the population based DIFFEV section. The same algorithm now works behind the scenes of the KUPLOT `fit` command.

Within DISCUS the calculation of the powder PDF has been moved from the `pdf` menu to the powder diffraction menu. In versions 5 and prior, the PDF was explicitly calculated using the histogram of interatomic distances. As this is an approximation in the case of X-ray and electron diffraction, DISCUS SUITE version 6 calculates the PDF via the powder pattern. The normalized powder pattern is transformed into the PDF. See Neder and Proffen (2020) for details.

The powder menu has been thoroughly revised and improved. Besides the new PDF output, the powder menu can now be used to calculate powder pattern for multiple phases.

In an analogue fashion, the 3D PDF or 3D Delta PDF is calculated by a Fourier transform of the (3/2)dimensional single crystal diffraction pattern.

To help adapt single crystal data to experimental values, a convolution of the single crystal diffraction pattern with a resolution function has been introduced.

The 3D output of a single crystal diffraction pattern can be stored as an HDF5 format file. This file is compatible with Yell. The `load` command in KUPLOT can read these files. The `mouse` menu in KUPLOT has been augmented to allow to view these files.

The space groups have been expanded to include all alternative setting for the orthorhomic space groups.

Noteworthy new features in the command language include:

On a more internal technical note, the internal accuracy of the interpreter language has been increased.

The option to use continuations of a long line across several lines in a macro.

Vector and matrix variables along with the usual matrix manipulation tools were introduced.

The new `update` command will perform an update to the latest version.

The Windows version of the DISCUS SUITE has migrated onto the Windows-Subsystem-Linux platform. This presents an improvement in calculation speeds by about 30%.

Automatic installer scripts for all operating systems should make the installation much ore straightforward.

1.3 Parallel execution

The typical application of the DISCUS SUITE is a run on a massive parallel system. You will have to check for details with your local administrator. In general you should expect to start DISCUS SUITE along the lines of:

```
mpiexec -n 192 discus_suite -macro refine.mac > /dev/null
```

The command `mpiexec` initiates the MPI system. Here in this example we request 192 CPUs `-n 192`. The MPI system then starts the DISCUS SUITE and read its input from the macro `refine.mac`. All output is discarded to `/dev/null` in order to minimize I/O. During development of your macro you might redirect the output to a logfile.
The example line above works fine on a single multi-core computer with Linux. The number of processes would be more reasonably placed around the number of CPUs on your computer. Within the Windows version, parallel execution of such a macro is performed by DISCUS SUITE command *parallel*:

```
parallel refine.mac
parallel 6, refine.mac
```

The first command version instructs DISCUS SUITE to run a parallel refinement via the macro *suite.mac*. The number of parallel threads is determined automatically. Alternatively you can tell DISCUS SUITE how many parallel threads you would like to run by adding the number of threads prior to the macro name. In the example above DISCUS SUITE is instructed to use 6 threads.

1.4 Command language

The program includes a FORTRAN style interpreter that allows the user to program complex modifications. A detailed discussion about the command language which is common to all DISCUS package programs can be found in the separate package_man reference guide which is included with the package. At the moment, DISCUS SUITE does not include any specific variables.

Chapter 2

Concepts

2.1 Quick overview

The purpose of the DISCUS SUITE is to run a combination of the sections: DIFFEV, DISCUS, REFINE, and KUPLOT within a single program. The DISCUS SUITE will be the initial section upon program start. This level serves as a simple rooftop to the individual sections, which do the main work.

As the main tasks are still performed within the sections DIFFEV, DISCUS, REFINE and KUPLOT, the actual set of commands within DISCUS SUITE is very limited. Besides the general command of the common command language the commands are just the names of the three sections, Table 2.1.

Within the four sections, a few details are slightly different compared to their stand alone versions. The main command syntax and typical command sequences are, however, identical. As DISCUS SUITE is a single program, the global variables $i[]$, $r[]$, $res[]$ are identical throughout all sections. The same holds for user defined variables. Thus, for example, if $i[0]$ is set to one value in any section, all other sections will see this new value. Be careful when you combine old DISCUS and KUPLOT macros. Their local variables my interfere with each other.

Besides the three variables $i[]$, $r[]$, $res[]$ and the user defined variables, each of the four section includes specific variables that are valid and meaningful within this section only. These variables specific to any of the sections are visible within that section only.

Command	Description
diffev	Switch to the `diffev` section. in many cases this will be the only command used within the top menu of the suite.
discus	Switch to the `discus` section.
refine	Switch to the `refine` section.
kuplot	Switch to the `kuplot` section.
parallel	Run a parallel refinement.
manual	Open the manual file.

Table 2.1: DISCUS SUITE command verbs.

2.2 Obtaining help

The DISCUS SUITE offers help in form of the manuals and as build in help menu. Check the manual for the Command language for full details in section Help.

2.3 Modified concepts

In order to fully utilize the common program architecture, a few details should be used in a modified way within the different sections. These details mostly concern disk I/O related parts.

2.3.1 DIFFEV

The stand alone DIFFEV used to communicate with DISCUS and KUPLOT via (small) files that contain the trial parameters and the cost function results. To avoid writing these to disk use:

```
init silent
```

The new `silent` parameter tells the DIFFEV section with the DISCUS SUITE not to write an initial set of trial parameters to disk. These values are automatically transferred to the slave sections.
Closely related is the modified command:

```
trialfile silent
```

which likewise tells DIFFEV not to write trial files during the refinement cycles.
In both cases the DISCUS SUITE will use the internal variables to communicate the trial parameters to the slave sections. The trial parameters are set up within DIFFEV as variable names and can be referenced with these names within the other sections. Likewise, the information on the current generation is transferred with special variable names. See the DIFFEV manual for full details.
As an example, to define the a and c lattice parameters for a Wurtzite type structure use the commands:

```
newpara P_lata,  3.85, 3.95, 3.88, 3.92
newpara P_latc,  6.45, 6.55, 6.48, 6.52
```

See the chapter Example in the DIFFEV manual for a fully documented example. The remainder of this section is obsolete for versions 5.22 and later.
For backwards compatibility the variables `ref_para[entry_number]` are retained. Entries in this array range from 1 to the number of parameters to be refined, as defined by tt pop_dimx in DIFFEV. As of Versions 5.7 the data are still written to the array, `r` as well. As a deliberate break compared to the general flexibility in all DISCUS SUITE sections, the trial values are stored in entries starting at `r[201]`. Use of these entries is meant to preserve backwards compatibility but will be discontinued in future releases.

Another predefined section of the internal variables is the range:

Entry	Description
`i[201]`	Current generation number
`i[202]`	Size of the population
`i[203]`	Size of the child population
`i[204]`	Number of parameters

The use of this predefined section of the internal variables is discouraged and has been replaced by the variables:

Entry	Description
`REF_GENERATION`	Current generation number
`REF_MEMBER`	Size of the population
`REF_CHILDREN`	Size of the child population
`REF_DIMENSION`	Number of parameters
`REF_KID`	Current child to work on
`REF_INDIV`	Current individual repetition of the current child

To retain the backwards compatibility, the actual child number and the number of the individual repetition are transferred as parameters on the `run_mpi` command line.

In the DISCUS SUITE version of KUPLOT, a new internal variable is used to store and transfer the weighted R-value from KUPLOT to DIFFEV. Thus, the need to write and read the result file ceases to exist. Do this via the commands:

```
resultfile  silent
...              ! further initialization
run_mpi ...      ! Calculate with Discus and Kuplot
compare silent
```

The first and last lines instruct the DIFFEV section not to read the resultfile but to use the value that is transferred internally.

MPI has been implemented in the current Windows installation as well starting with Version 5.7. Use the icon `DISCUS_Suite Parallel_Version` for parallel refinements.

Even in the single session version of the DISCUS SUITE, the refinement should use the `run_mpi` command in an identical fashion.

In order to use the suite efficiently, the `branch` command is available within DIFFEVas well, and you can branch to either DISCUSor KUPLOT . This command should replace the `system discus < discus_macro_name` construction that is used in a serial refinement set up:

```
 1  diffev
 2     ... SETUP ...
 3     init silent
 4     branch discus
 5        @discus_macro.mac
 6        exit
 7     branch kuplot
 8        @kuplot_macro.mac
 9        exit
10     compare silent
11  exit
```

In this simplified example, we switch to the diffev section with the `diffev` command in line 1. After initial setup done by DIFFEV and the initialization of generation zero in line 3, the

DIFFEV section branches off to the DISCUS section in line 4. Now control is at DISCUS and the loop over all children and all calculations are performed by the macro `diffev_macro.mac` in line 5. The `exit` command in line 5 returns to the DIFFEV section. If you use a previous DISCUS macro, the `exit` line might be part of the DISCUS macro! Lines 7 to 9 do the same for KUPLOT. The `compare` command (line 10) at the DIFFEV level does the comparison between the generations, and finally we leave the DIFFEV section in line 11. In a full refinement there would be a loop over all refinement cycles that includes lines 4 to 10 of this example.

2.3.2 DISCUS

The performance of DISCUS is not directly affected by the DISCUS SUITE. To make full use of the DISCUS SUITE, a couple of concepts should be adhered to.

Use the trial parameter value through the variable names that are defined within DIFFEV . Use the variables REF_GENERATION, REF_MEMBER, REF_CHILDREN, REF_KID, REF_INDIV to obtain the current generation number, the size of the population, the number of children and the number of trial parameters. Do not open and read the file GENERATION. This file is created within each refinement cycle, and has been augmented by a lot of information that DIFFEV reads to continue an interrupted refinement. Withinhte sections DISCUS and KUPLOT there is, however, no longer any need to read it.

If the initial asymmetric unit is not extremely long, it is best to build the initial asymmetric unit from scratch. To build a Wurtzite type asymmetric unit use the following command sequence:

```
1    read
2       free P_lata, P_lata, P_latc, 90.00, 90.00, 120.00, P63mc
3    insert Zn, 1./3.,  2./3, P_zn_z, P-biso
4    insert S , 1./3.,  2./3, 0.00  , P-biso
5    save
6       outf internal.wurtzite.cell
7       omit all
8       write scat
9       write adp
10      run
11   exit
12   read
13      cell internal.wurtzite.cell, 5,5,5
```

In this example it is assumed that P_lata contains the a lattice parameter value, P_latc the c lattice parameter, P_zn_z the z-position of Zn and P_biso a common atomic displacement parameter B. The last parameter in line 2 tells DISCUS that the structure belongs to space group P63mc. The asymmetric unit is stored internally into computer memory in lines 5 to 11, and then expanded into a block of 5x5x5 unit cells in lines 12 and 13. By providing the space group symbol as last parameter to the `free` command, the expansion in line 13 will ensure that the two atoms Zn and O are properly copied to create all atoms in the unit cell. This macro section does not perform any disk I/O and will be much more efficient on a large parallel system.

Do use the internal storage whenever you need to save a temporary crystal structure. Every time a file name is preceded with the string `internal`, DISCUS will save / read the structure to / from internal memory.

To perform any further manipulations with the calculated data like multiplication by a scale factor, the addition of a background etc., KUPLOT needs to be executed once the DISCUS section is finished. To allow efficient use of massive parallel systems the new `branch` command has

been added to DISCUS and KUPLOT. It allows to switch directly from the DISCUS section to the KUPLOT section within DISCUS SUITE.

```
0   ...                       ! Discus initialization
1   do indiv=1,nindiv         ! Repeat nanoparticle generation nindiv times
2      @discus.zno.mac kid     ! do the actual simulation
3   enddo
4   branch   kuplot           ! Switch to KUPLOT
5      @kup.diffev.mac ., kid  ! Contains a final exit
6   exit                      ! Finish DISCUS
```

In this DISCUS macro section the simulation of nanoparticles is repeated several (nindiv) times. This might be necessary to average out the defect distribution within the nanoparticles. Under these circumstances the individual calculated diffraction pattern / PDFs need to be averaged after the DISCUS section is finished. The branch to the KUPLOT section in line 4 allows DISCUS to do this while remaining on the identical slave process and thus on the identical compute node. On many massive parallel systems data transfer from a compute node to the central disk is time consuming and many of these systems have local fast (RAM) disks. The DISCUS section can write the individual diffraction pattern / PDF to this local disk without much of a penalty. But then, KUPLOT needs to operate on the same node in order to find the data. This is ensured by the `branch` construct.
In the alternative set up :

```
0   ...                         ! Diffev initialization
1   run\_mpi discus, discus\_macro.mac  " Discus calculation
2   run\_mpi kuplot, kuplot\_macro.mac  ! Kuplot calculation
```

DIFFEV will distribute the DISCUS jobs onto all nodes, and then do the same with the KUPLOT jobs. As of version 5.18 the DISCUS SUITE will ensure taht the two jobs for a given child are performed on the identical node.

2.3.3 KUPLOT

In the same fashion as DISCUS , the KUPLOT section should rely on the variables `ref_para[]` REF_GENERATION REF_MEMBER, REF_CHILDREN, REF_KID, ref_INDIV to determine the population size and if necessary the trial parameters. Do not read the GENERATION file.
Do not write a result file. The KUPLOT instruction:

```
rvalue 1, 2
```

stores the weighted R-value internally and instructs the DISCUS SUITE to transfer the value from the slave process back to the master process. The master process can then use this in the `compare silent` command.
The `branch` command is available within KUPLOT as well, and you can branch to either DIFFEV or DISCUS . Most application will probably not need to do this.

Chapter 3

Example

This chapter takes you through all steps of a nanoparticle refinement. The example will use data published in Chory et al. (2007), which describes the refinement of ZnSe nanoparticles. We will quickly describe the main aspects of the model that is used to describe the experimental data. The main focus will be on the use of the DISCUS SUITE and on the differences to a refinement with DIFFEV.

The ZnSe nanoparticles can be described as a disordered Zincblende structure. Stacking faults introduce a local Wurtzite structure. TEM images showed that the particles are of approximately ellipsoidal shape with one long and two short axes. Bulk ZnSe crystallizes in the Zincblende structure, and these nanoparticles show a typical high defect concentration. For this example we choose to describe the structure as a stack of layers based on a hexagonal metric. Thus the stacking faults are restricted to one of the symmetrically equivalent [111] axes of the bulk cubic structure. A perfect ABCABC sequence of these layers would correspond to the cubic Zincblende structure, a perfect ABAB sequence would yield the hexagonal Wurtzite structure. For further details see Chory et al. (2007) and Neder and Proffen (2007).

3.1 Parallel refinement

A main advantage of DISCUS SUITE compared to the stand alone programs is achieved in a parallel refinement. A stand alone refinement with DIFFEV needs to start DISCUS and KUPLOT as slave programs within each refinement cycle. Thus at each program start, the macros and the experimental data need to be read again. Furthermore, these separate programs need to communicate the results via files on the hard disk, which slows down the performance. As the DISCUS SUITE is a single program, internal communication is straightforward and the makes most disk I/O obsolete. The DISCUS SUITE performs its tasks within the individual sections and thus most parts of the actual refinement macros do not need to be changed.

To run in parallel mode, the DISCUS SUITE must have been compiled with the MPI option. The suite will then be started with a command like:

```
mpiexec -n 192 discus_suite -macro suite.mac  > /dev/null
```

If you use the Windows operating system, start a parallel refinement via:

```
parallel suite.mac
```

see chapter 1.3 for further info on the invocation.

The main DISCUS SUITE macro can be very short:

```
1   diffev
2       @diffev.mac
3   exit
4   exit
```

In line 1 we switch to the DIFFEV section, perform the -slightly modified - macro `diffev.mac` and return to the main DISCUS SUITE menu via the `exit` in line 3. In line 4 we finally finish the suite itself.

This macro nesting is not a necessary feature. You could of course place all these lines into the `diffev.mac` macro as well.

The `diffev.mac` macro needs just few changes compared to a stand alone version:

```
1    set error,exit
2    @cleanup.mac
3    #
4    @setup.mac
5    @diffev_setup.mac
6    #
7    init silent
8    #
9    do i[0]=1,100
10       echo "GENERATION %4d",REF_GENERATION
11       run_mpi discus, nano.znse.mac, repeat:REF_NINDIV , compute:serial, logfile:/dev/null
12       compare silent
13   enddo
```

In line 1 we instruct DISCUS SUITE to be picky with errors and to stop the execution is any error should occur. I do not want to run a refinement for several hours just to find out that somewhere there there has been an erroneous calculation.

The *cleanup.mac* macro may be used to remove all files related to an earlier refinement.

The *setup.mac* macro is used to define the number of individual repetitions that DISCUS should perform for each member of the population.

Macro *diffev_setup.mac* defines the size of the population, the number and range of the parameters to be refined etc.

In line 7 we initialize the parameters to their starting values. The qualifier *silent* instructs the DISCUS SUITE not to write these trial parameters onto the disk. They will instead be passed silently between the DIFFEV and DISCUS sections.

The *run_mpi* command in line 11 starts the simulation and the assessment of the current agreement between experimental and calculated data. The DIFFEV section will start the DISCUS section and within DISCUS operate the macro *nano.znse.mac*. The next parameter, here a zero, tells DIFFEV if the individual repetitions are to be carried out in parallel or if they are to be done in serial operation by the DISCUS section. If the parameters is zero, as in this example, any repetition is left to the DISCUS section. If the parameter is larger than one, DIFFEV will calculate the individual repetitions in parallel on your system.

Finally in line 12 we tell DIFFEV that the final agreement factor between experimental data and the calculated data shall not be read from a file on your disk. Instead KUPLOT will pass this parameters silently back to DIFFEV.

Appendix A

SUITE commands

A.1 Summary

You can switch to the individual sections "discus", "diffev", "kuplot" and "refine" by typing the respective section name. To return to the suite type "exit" at the main menu of each section.
The variables i[*], r[*] and res[*] are global variables, a change in any section will be seen in any other section as well. The same holds for all user defined variables!
The section specific variables are local within each section.
If an output filename in "discus" starts with "kuplot", the data are written directly into the next available KUPLOT data set. This is available for Fourier output, powder, pdf.

A.2 News

The individual help topics contain information on changes, improvements and bug fixes. Further news are available at the DISCUS, KUPLOT, DIFFEV, REFINE and Command_language help sections.

2020_November

Added options to the 'update' command

2020_October

As of version 6.02.00 the DISCUS_SUITE is compiled with support for OpenMP, an internal parallelization that affects at the moment Single crystal Fourier, MMC, and Powder diffraction through the Debye scattering equation.
In version 6.02.01 a small bug was corrected. The number of parallel threads is by default limited to the number of physical cores, you can change this vis the ==> 'set parallel' command.

2019_November

Added a global 'reset' command

2019_June

Added more stringent error controls on MPI initialization

2018_June

Revised the reaction to a CTRL-C
Added a ==> 'set error, ... , "save" option

2018_Jan

The logical comparisons may now take the operators: $<, <=, ==, /=, >=, >/$ The classical fortran77 operators are still valid
New logical functions "isvar" and "isexp" can be used within an "if" construction. See help entry ==>'function' in the general "Command_lang" section

2017_Sep

Throughout the program the internal calculation of random numbers was changed to the FORTRAN 90 intrinsic function.

2016_Dec

At a few select points colors are introduced into the output. Currently these are just the error messages.

2016_Oct

A new command 'parallel' has been added to the Windows version This allows to execute a macro in parallel.

2016_June

The SUITE may now be interruted gracefully with a CTRL-c. This will cause the DISCUS part to write the current structure, and DIFFEV to shut down MPI if active.

2015_December

The branch command within the sections discus, diffev, kuplot may now take the form : branch discus -macro macro_name par1, par2, ...

2015_June

Starting with Version 5.1, we have migrated to a X-Window environment for WINDOWS as well. As a small side effect, the technique to jump to the desired folder has changed slightly. See the help entry on "cd" in the general "Command_lang" section for further information. The process is described in the package manual as well.

A.3   help

```
help [<command> [, <subcommand>] ]
```

The 'help' command is used to display on-line help messages. They are short notes on the command <command>. The command may be abbreviated. If the abbreviation is not unique, only the first help topic that matches the command is listed.

The first line of the help text gives the syntax of the command that is explained in the following lines. For a few commands the syntax line is repeated for different set of possible parameters.

After the text is displayed, you are in the HELP sublevel of PROG and there are the following commands possible:

```
<command> : Display help for <command> of current help level.
".."      : Go up one help level.
"?"       : Prints list of help entries of the current level.
<RETURN>  : Exit help sublevel.
```

Within this help, text that is written as: <text> : in pointed brackets : This text is to be understood as text or command or number that you have to type when using the actual command. Omit the pointed brackets. "text" : in quotation marks : This text is to be understood as the explicit test that is given. Omit the quotation marks. 'command' : in single quotes : Refers to commands of suite, discus etc.

A.4   diffev

Switches to the "diffev" section.
Within this section any standard DIFFEV command can be given.
The "diffev" command 'run_mpi' will start a discus/kuplot calculation at the specified section.
See the diffev manual / help for further information.
Use an 'exit' to return to the suite.

A.5   discus

Switches to the "discus" section.
Within this section any standard DISCUS command can be given.
Within the discus section you can use the command 'branch kuplot' to switch to the kuplot branch.
One can write an output file directly into the KUPLOT data sets. The number of data sets in KUPLOT is automatically incremented. Currently this is implemented for the PDF and the powder output. Single crystal diffraction pattern to follow shortly.
Use an 'exit' to return to the suite.

A.6   kuplot

Switches to the "kuplot" section.
Within this section any standard KUPLOT command can be given.

Within the kuplot section you can use the command 'branch discus' to switch to the discus branch.
Use an 'exit' to return to the suite.

A.7 refine

Switches to the "refine" section.
Within this section you can refine model parameters to a data set.
Use an 'exit' to return to the suite.

A.8 parallel

```
parallel {<numprocs>, },<macro.mac> {, <para_1...}
```

Starts an MPI driven parallel calculation. See the diffev help on a full explanation of parallel processing.
The parallel refinement will execute file <macro.mac>, which must reside in the current directory. Make sure you have used cd <path> to change to the proper directory prior to the use of the 'parallel' command. The macro name must be given in full, including the ".mac" extension. If the macro requires parameters you must specify these following the macro name.
Optionally you can place the number or processes that MPI shall start prior tot he macro name. The numebr defaults to the value of the SHELL variable NUMBER_OF_PROCESSORS on your system. If this variable is not set, discus_suite will start 4 processes.

A.9 manual

```
manual ["section:"{"suite"  | "discus"  | "diffev" |
                  "kuplot" | "package" | "mixscat"}
        [,"viewer:"<name>]
```

Opens a PDF viewer for one of the Manuals
The section defaults to the current program section that you are working with. On Linux systems, the viewer defaults to "qpdfview", on Windows system it defaults to "firefox". If DISCUS does not find the default or the user provided viewer, DISCUS will search a list of common PDF viewers. If none is found an error message points to the folder that contains the manuals.

A.10 show

```
show {"error" | "parallel" | "res" | "variables"}
```

The show command displays settings onto your screen. The individual programs discus, kuplot, diffev, refine all have specific parameters to the show command as well, see the help at the main program level for details of the 'show' command: "help Command_language".

A.11   update

```
update {"install=fetch" | "install=local" | "install="<archive.tar.gz>}
       {,"code=pre" | "code=git" | "code=current" |
         "code="<archive.tar.gz>}
```

This command updates the DISCUS_SUITE.

Without the optional parameters, the suite will download the installer ("install=fetch") and use a precompiled code ("code=pre") if available for the current operating system.

The options allow you to: "install=local" Use the installer in the current directory $HOME/DIFFUSE_INSTALL "install="<archive.tar.gz> Unpack the archive, expecting a directory called DIFFUSE_INSTALL "code=git" Use the code from the current release at GIThub, compile "code=current" Use the current source code in $HOME/DIFFUSE_INSTALL/develop and compile "code="<archive.tar.gz> Use the source code in the archive

Bibliography

C. Chory; R. B. Neder; V. I. Korsunskiy; F. Niederdraenk; Ch. Kumpf; E. Umbach; M. Schumm; M. Lentze; J. Geurts; G. Astakhov; W. Ossau; G. Meuller. Influence of liquid-phase synthesis parameters on particle sizes and structural properties of nanocrystalline ZnO powders. phys. stat. sol c **4** (2007) 3260–3269.

R. N. Neder; T. Proffen. Exact and fast calculation of the x-ray pair distribution function. J. Appl. Cryst. **53** (2020) 710–721.

R.B. Neder; Th. Proffen. *Diffuse scattering and defect structure simulations - A cook book using the program DISCUS* . Oxford University Press, Oxford, UK 2007.