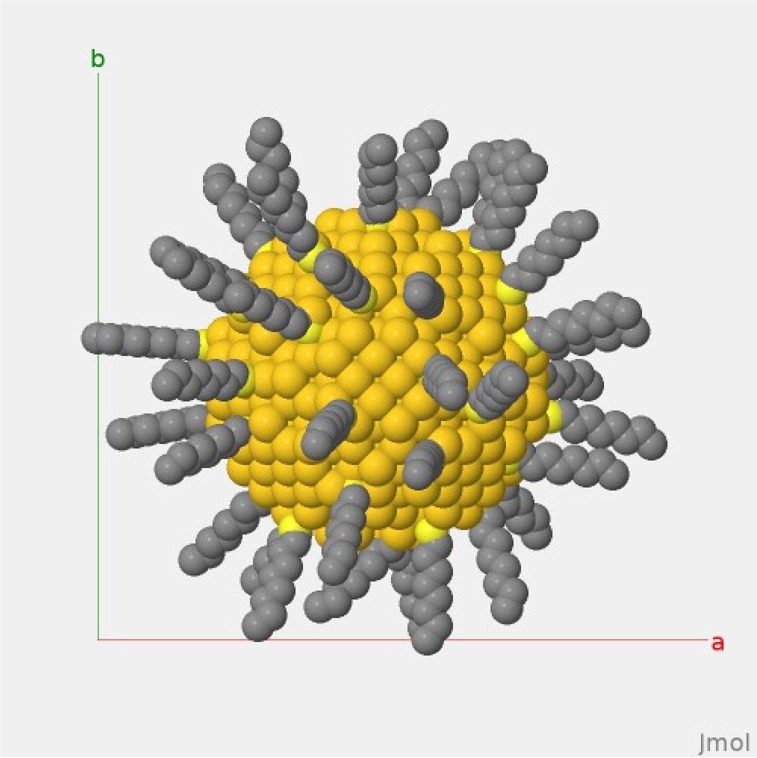
DISCUS Workshop
Building Nanoparticles

Reinhard B. Neder
Crystallography and Structural Physics
Friedrich-Alexander-Universität Erlangen-Nürnberg

reinhard.neder@fau.de

Building a Nanoparticle



Goal:

Build an ellipsoidal Gold nanoparticle decorated with a simple ligand

Calculate Powder pattern / PDF

Basic Info:

Diameter (s)

Number of Ligand molecules

A quick demonstration

Building a Nanoparticle

Open in Windows Explorer: Lectures\02_Building_Nanoparticle\ELLIPSOID

Start DISCUS_SUITE

You should see:

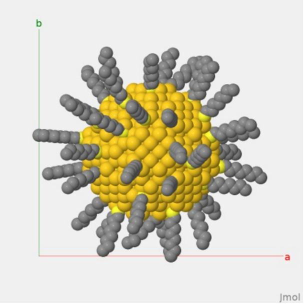
```
...
User macros in ...
System macros in ...
Start directory ...

suite >
```

```
suite > cd  Lectures\02_Building_nanoparticle\ELLIPSOID

suite > @top.mac  30.0,  30.0, 30.0, 0.01
```

Building a Nanoparticle



Basic Info:

- **Diameter (s)**
- **Number of Ligand molecules**

Detailed information

Gold structure : **Spacegroup**
Lattice parameter
Atom positions

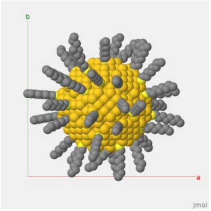
Ligand structure : **Spacegroup**
Lattice parameter
Atom positions

Building a Nanoparticle

Basic Info:

Diameter (s)

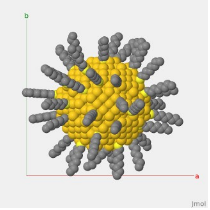
Number of Ligand molecules



top.mac

```
reset      ! Ensure clean start up conditions
#    ! Some useful variable definitions
@definitions.mac $1, $2, $3, $4
#    ! Create a Thio-octane molecule
@prepare_chain.mac
#    ! Step into discus section
discus
#    ! Build a gold unit cell from scratch
@prepare_gold.mac
```

Building a Nanoparticle



Detailed information

Gold structure : **Spacegroup**
Lattice parameter
Atom positions

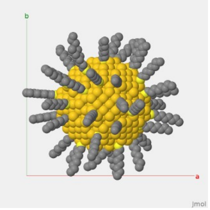
DISCUS Building a unit cell

Read a CIF file

Read a DISCUS file

Build from scratch

Building a Nanoparticle



Detailed information

Gold structure : Spacegroup
Lattice parameter
Atom positions

prepare_gold.mac

```
read                               ! Read, freely defined space
  free  4.0782, 4.0782, 4.0782, 90,90,90, Fm-3m
!                                  ! Exception, read menu exits automatically
insert Au,  0.00, 0.00, 0.00,  bvalue  ! First Gold atom position
#
save                               ! Save unit cell file to disk
  outfile gold.cell                ! Filename
  omit all                         ! Write basic header only
  run                              ! Perform the actual task
exit                               ! Back to main DISCUS sec
```

READ Menu load a structure“

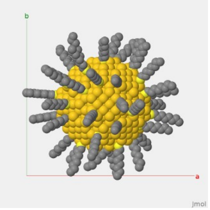
Commands:

structure file_name	Reads a structure „as is“
cell file_name { na, nb, nc}	Read asymmetric unit, expand to full unit cell if na, nb, nc > 1 expand to block of multiple unit cells
free a, b, c, alpha, beta, gamma, spacegroup	Create an empty space with the given unit cell parameters, defaults to cartesian

Options for **cell** and **structure**

“**unique**:<**qualifier>**“	Define details to distinguish unique atom types
“**unique**:**site**“	Each atom is a unique atom type
“**unique**:**name**“	All atoms with identical chemical symbol are treated as an identical atom type
“**unique**:**charge**“	All atoms with identical chemical symbol and charge are treated as an identical atom type

Building a Nanoparticle



Detailed information

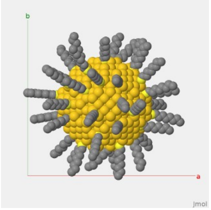
Gold structure :

Spacegroup
Lattice parameter
Atom positions

gold.cell

```
title freely created structure
spcgr Fm-3m           , setting:abc
cell    4.078200,  4.078200,  4.078200, 90.000000, 90.000000, 90.000000
format numbers,XYZBP
atoms  x,          y,          z,          Biso,    Property
AU    0.000000,   0.000000,   0.000000,  0.050000,       1
```

Building a Nanoparticle



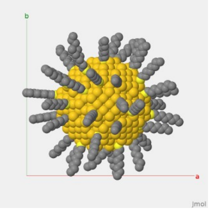
Detailed information

Gold structure : **Spacegroup**
Lattice parameter
Atom positions

top.mac

```
read                    ! Read and expand to one unit cell
  cell gold.cell        ! This helps to know the lattice parameters
```

Building a Nanoparticle



Detailed information

Gold structure : **Spacegroup**
Lattice parameter
Atom positions

top.mac

```
read              ! Read and expand to one unit cell
  cell gold.cell  ! This helps to know the lattice parameters
#                 ! Determine required unit cell numbers
ncell_a = int(max(diam_a, diam_b, diam_c)/lat[1]) + 2
ncell_b = int(max(diam_a, diam_b, diam_c)/lat[2]) + 2
ncell_c = int(max(diam_a, diam_b, diam_c)/lat[3]) + 2
read              ! Read and expand to full crystal
  cell gold.cell, ncell_a, ncell_b, ncell_c
```

Building a Nanoparticle

You should see:

Open in Windows Explorer: Lectures\02Building_Nanoparticle\ELLIPSOID

```
suite > discus

discus > read
discus/read >    cell   gold.cell
discus/chem >    chemistry
discus/chem >        element
discus/chem >    exit
discus > show atom, 1
discus > show atom, all
discus > show cdim
```

Try „**structure**“ instead of „**cell**“

Building a Nanoparticle

You should see:

Open in Windows Explorer: Lectures\02Building_Nanoparticle\ELLIPSOID

```
suite > discus

discus > read
discus/read >    cell   gold.cell,  ncell_a,  ncell_b,  ncell_c
discus/chem >    chemistry
discus/chem >        element
discus/chem >    exit
discus > show atom, 1
discus > show atom, n[1]
discus > show cdim
```

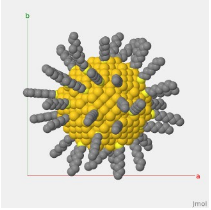
A quick demonstration

```
discus > @build_pure_ellipsoid.mac   yes
```

Building a Nanoparticle

Shaping the nanoparticle

build_pure_ellipsoid.mac



```
if('$1'=='yes') then         ! Compare two strings
  @display.mac  gold_cube    ! Display the initial block
endif
```

The conditional enables a optional display of plot and chemical info

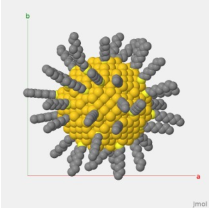
display.mac

```
@plot.mac $1     ! Interactive plot with JMOL
chemistry        ! Check composition
  element        ! List elements and number of atoms in structure
exit             ! Back to main DISCUS section
wait return      ! Wait for user confirmation
```

Building a Nanoparticle

Shaping the nanoparticle

build_pure_ellipsoid.mac



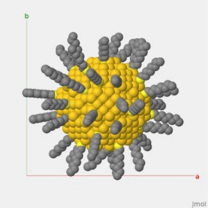
```
if('$1'=='yes') then        ! Compare two strings
  @display.mac  gold_cube   ! Display the initial block
endif
```

plot.mac

```
plot
  program jmol                ! Define the plotting program
  select all                  ! All atom types
  extend all                  ! Full extend over all crystal dimensions
  outf   $1_plot.cif          ! output CIF file
  run plot:inter, kill:yes    ! Write CIF file, make interactive plot
                              !   kill previous windows
exit
```

Building a Nanoparticle

Shaping the nanoparticle **build_pure_ellipsoid.mac**



```
if('$1'=='yes') then       ! Compare two strings
  @display.mac  gold_cube  ! Display the initial block
endif
#
surface                         ! Create an ellipsoid
  boundary ellipsoid, diam_a, diam_b, diam_c
# boundary ell, diam_a,diam_b,diam_c,first:[0,0,1], third:[1,0,0]
exit
#
purge
```

SURFACE Menu shape crystal “arbitrarily“

Main command „**boundary**“

Possible boundaries are:
Single face “**hkl**“; set of symmetrically equivalent faces „**form**“
Ellipsoid „**ellipsoid**“; ”**sphere**”; ”**cylinder**“
Defined by local environment „**local**“

Options

Last parameter:
- “**inside**“; default, keep atoms inside the desired form
- „**outside**“; keep atoms outside the form

Better:
- “**keep**:**inside**“ ”**keep**:**outside**“

Position of center:
- “**centx:value**“ ”x”, “y”, “z” will be at „**value**“; can be string „**com**“
- „**center:value**“; x,y, and z all at “**value**“

SURFACE Menu shape crystal “arbitrarily“

Main command „**boundary**“

Possible boundaries are:
Single face “**hkl**“; set of symmetrically equivalent faces „**form**“
Ellipsoid „**ellipsoid**“; ”**sphere**”; ”**cylinder**“
Defined by local environment „**local**“

Options

Orientation; Ellipsoid:

“**first:[u,v,w]**“;	Direction of the first principal ellipsoid axis
„**third:[u,v,w]**“	Same for third axis; or [h,k,l,r] for axes in reciprocal space

Cyclinder:

“**axis**:**[u,v,w]**“	Direction of cylinder axis

Complex surfaces:

“**accum:{init | add}**“	Start collection of multiple rules for several surfaces
„**exec:{run | hold}**“;	Run execution with this surface or hold for further conditions

SURFACE Menu shape crystal “arbitrarily“

Main command „**boundary**“

Possible boundaries are:
Single face “**hkl**“; set of symmetrically equivalent faces „**form**“
Ellipsoid „**ellipsoid**“; ”**sphere**”; ”**cylinder**“
Defined by local environment „**local**“

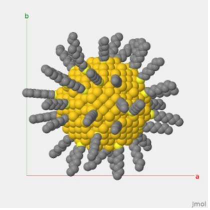
Options

Complex surfaces:

“**accum:init**“, „**exec:hold**“	Start collection of multiple rules for several surfaces
“**accum:add**“, „**exec:hold**“	Continue to add rule
“**accum:add**“, „**exec:run**“	Last condition, Execute the combined rules with logical and for all conditions

Building a Nanoparticle

Shaping the nanoparticle **build_pure_ellipsoid.mac**



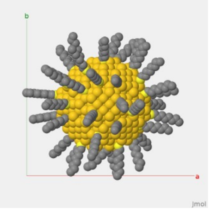
```
if('$1'=='yes') then           ! Compare two strings
  @display.mac  gold_cube      ! Display the initial block
endif
#
surface                                  ! Create an ellipsoid
  boundary ellipsoid, diam_a, diam_b, diam_c
# boundary ell, diam_a,diam_b,diam_c,first:[0,0,1], third:[1,0,0]
exit
#
purge
```

! Removes all “VOID” atoms from computer memory
! Sequence of atoms will be changed

A quick demonstration

discus > **@build_pure_ellipsoid.mac yes**

Building a Nanoparticle



Detailed information

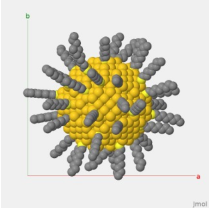
Gold structure : **Spacegroup**
Lattice parameter
Atom positions

top.mac

```
. . .
read                 ! Read and expand to full crystal
  cell gold.cell, ncell_a, ncell_b, ncell_c
#
@build_pure_ellipsoid.mac no  ! Shape into an ellipsoid
#
@debye.mac gold_ellipsoid.mac ! Calculate Powder pattern, PDF
```

Building a Nanoparticle

Building a Nanoparticle



Detailed information

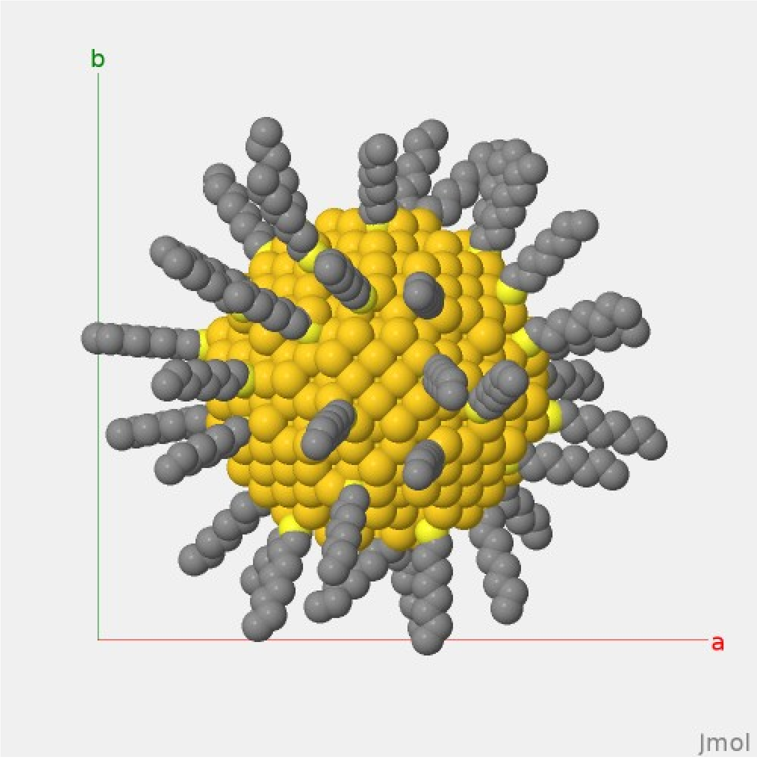
Gold structure : **Spacegroup**
Lattice parameter
Atom positions

top.mac

```
. . .
read                  ! Read and expand to full crystal
  cell gold.cell, ncell_a, ncell_b, ncell_c
#
@build_pure_ellipsoid.mac no  ! Shape into an ellipsoid
#
@debye.mac gold_ellipsoid.mac ! Calculate Powder pattern, PDF
#
@decorate.mac                 ! Add ligand molecules
```

Building a Nanoparticle

Building a **strained** Nanoparticle



Goal:

Build an ellipsoidal Gold nanoparticle decorated with a simple ligand

Introduce strain by Au-S-Au bond angle

A quick demonstration

Building a **strained** Nanoparticle

Open in Windows Explorer: Lectures\02_Building_Nanoparticle\STRAINED_ELLIPSOID

Start DISCUS_SUITE

You should see:

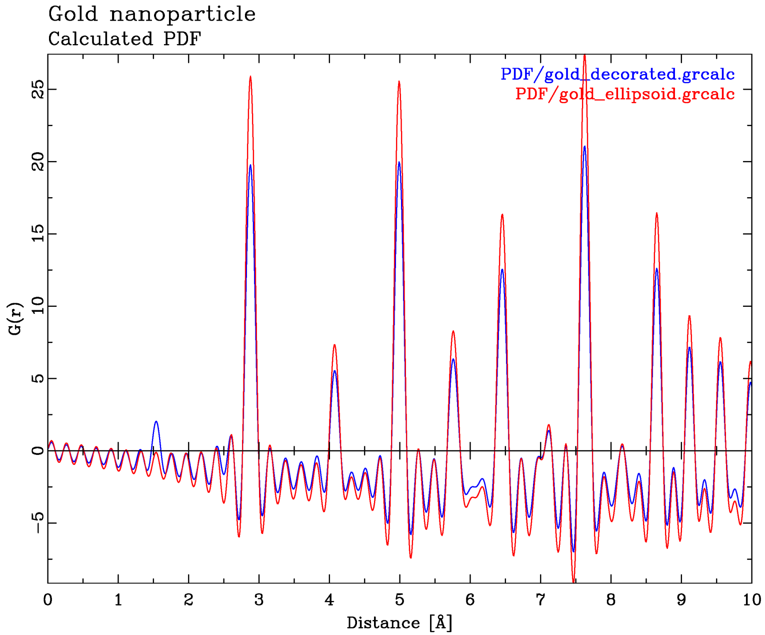
```
...
User macros in ...
System macros in ...
Start directory ...

suite >
```

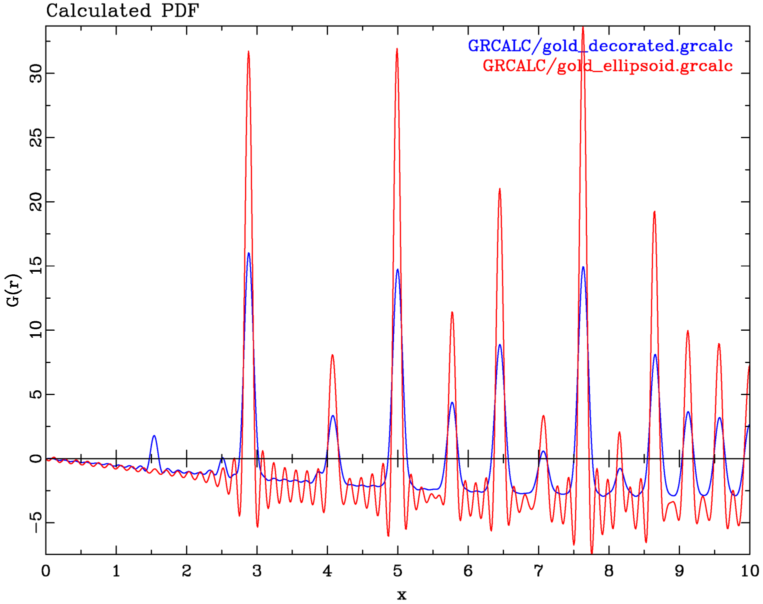
```
suite > cd  Lectures\02_Building_nanoparticle\STRAINED_ELLIPSOID

suite > @top.mac  30.0,  30.0, 30.0, 0.01
```

Building a **strained** Nanoparticle



Non - strained



Strained

Building a **strained** Nanoparticle

top.mac

```
. . .
read                  ! Read and expand to full crystal
  cell gold.cell, ncell_a, ncell_b, ncell_c
#
@build_pure_ellipsoid.mac no  ! Shape into an ellipsoid
#
@debye.mac gold_ellipsoid.mac ! Calculate Powder pattern, PDF
#
@decorate_head.mac            ! Add ligand „head“ == Sulfur
#
@strain.mac                   ! Strain the core-shell object
#
@decorate_tail.mac            ! Add Carbon chain on top of Sulfur
```

Building a **strained** Nanoparticle