

Designspecifikation

Redaktör: Anders Rehult

Version 1.1

Status

Granskad	VB	18-05-25
Godkänd	Johan Jönsson	18-04-11

PROJEKTIDENTITET

2018/VT, Grupp 2
Linköpings Tekniska Högskola, IFM

Gruppdeltagare

Namn	Ansvar	Telefon	E-post
Anders Rehult	Projektledare (PL)	076-3161206	andre449@student.liu.se
Marian Brännvall	Dokumentansvarig (DOK)	070-7280044	marbr639@student.liu.se
Andreas Kempe	Sekreterare (SE)	073-9796689	andke133@student.liu.se
Viktor Bernholtz	Viktor Bernholtz (VB)	073-0386030	vikbe253@student.liu.se

Kund: IFM, Linköpings universitet, 581 83 Linköping

Kontaktperson hos kund: Rickard Armiento, 013-281249, rickard.armiento@liu.se

Kursansvarig: Per Sandström, 013-282902, persa@ifm.liu.se

Handledare: Johan Jönsson, 013-281176, johan.jonsson@liu.se

Innehåll

Dokumenthistorik	v
1 Inledning	1
1.1 Parter	1
1.2 Projektets bakgrund	1
1.3 Syfte och mål	1
1.4 Användning	2
1.5 Begränsningar	2
1.6 Definitioner	2
2 Översikt av systemet	3
2.1 Beroenden till andra system	3
2.2 Delsystem	3
2.3 Designfilosofi	4
3 Datakonvertering	4
3.1 HDF5	4
3.1.1 HDF5-struktur för VASP-data	6
3.2 VASP	7
3.2.1 Kristallstruktur	7
3.2.2 Elektrontäthet	8
3.2.3 Tillståndstäthet	9
3.2.4 Fermi-ytor	10
3.2.5 Dynamisk kristallstruktur	10
3.3 Elk	10
4 Visualisering	11
4.1 Utritning	11
4.2 Användarindata	12
4.3 Interaktivitet	13
4.4 HDF5-indata	13
4.5 Kristallstruktur	13
4.5.1 Översiktlig beskrivning	13
4.5.2 Inwiwo-nätverk för visualisering av kristallstruktur	13
4.6 Elektrontäthet	14
4.6.1 Översiktlig beskrivning	14

4.6.2	Inviwo-nätverk för visualisering av elektrontäthet	14
4.7	Tillståndstäthet	15
4.7.1	Översiktlig beskrivning	15
4.7.2	Inviwo-nätverk för visualisering av tillståndstäthet	15
4.8	Fermi-yltor	17
4.8.1	Inviwo-nätverk för Brillouin-zonuppritning	17
4.8.2	Inviwo-nätverk för Fermi-ylte-uppritning	18
4.9	Dynamisk visualisering av kristallstruktur	19
4.9.1	Översiktlig beskrivning	19
4.9.2	Inviwo-nätverk för dynamisk visualisering av kristallstruktur	19
Referenser		21
Bilaga A HDF5-datastruktur		22
Bilaga B HDF5-datastruktur i två delar		23

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2018-02-27	Första utkast.	Projektgruppen	SE
0.2	2018-03-05	Andra utkast.	Projektgruppen	DOK
0.3	2018-03-28	Tredje utkast.	Projektgruppen	VB
1.0	2018-03-28	0.3 godkänd.	Projektgruppen	VB
1.1	2018-03-28	Uppdaterat licens för bilder.	Projektgruppen	VB

1 Inledning

Dokumentet är en designspecifikation för kandidatprojektet i visualisering av elektronstrukturer. Visualisering av elektronstrukturer är ett av projekten i kursen TFYA75 vid Linköpings universitet. Designspecifikationen är en fördjupning av systemskissen och beskriver detaljerat hur systemet kommer att designas.

1.1 Parter

I Tabell 1 listas de olika parter som är inblandade i projektet.

Roll	Namn	E-post
Beställare	Rickard Armiento	rickard.armiento@liu.se
Expert	Rickard Englund	rickard.englund@liu.se
Handledare	Johan Jönsson	johan.jonsson@liu.se
Projektleddare	Anders Rehult	andre449@student.liu.se

Tabell 1: Parter i organisationen.

1.2 Projektets bakgrund

Ett viktigt verktyg inom teoretisk fysik är elektronstrukturberäkningar. Med hjälp av dessa beräkningar går det att förutsäga hur material med vissa egenskaper kommer att bete sig.

För att förstå beräkningarna är det viktigt att kunna analysera data som beräknats. I vissa fall förenklas detta med hjälp av visualisering och i andra fall krävs det mer eller mindre att man visualiserar sin data för att förstå den. Ett kraftfullt verktyg för att visualisera data är visualiseringsprogrammet Inviwo.

Inviwo är ett forskningsverktyg som utvecklats vid Linköpings universitet och ger användaren möjlighet att styra visualisering med hjälp av programmering i Python 3 eller grafiskt. Det tillhandahåller även användargränssnitt för interaktiv visualisering.

1.3 Syfte och mål

Målet med projektet är att utveckla ett system för visualisering av resultat av elektronstrukturberäkningar. Detta ska göras i visualiseringsverktyget Inviwo och systemets funktionalitet ska demonstreras genom att använda det för att illustrera resultat från befintliga beräkningar. I och med att den framtagna mjukvaran ämnas användas i forskningssammanhang måste projektet hålla en hög vetenskaplig och teknisk kvalitet.

Utöver det konkreta målet med framtagandet av mjukvara för visualisering ska även projektet ge projektmedlemmarna erfarenhet av att arbeta i projekt och utöka deras förmåga till analytiskt och fysikaliskt tänkande för att ge värdefull erfarenhet inför arbetslivet.

1.4 Användning

Denna produkt kommer användas vid Linköpings universitet för att analysera data från elektronstruktursberäkningar.

1.5 Begränsningar

I projektet kommer visualiseringsverktyget Inviwo och programmeringsspråken Python och C++ användas. Det kommer inte utredas om det är bättre att använda andra verktyg.

1.6 Definitioner

API (Application Programming Interface) är en specifikation av hur olika applikationer kan använda och kommunicera med en specifik programvara. Detta utgörs oftast av ett dynamiskt länkat bibliotek. [11]

BSD2 är en licens för öppen källkod. [3]

C++ är ett programmeringsspråk. [4]

I Inviwo används C++ för att skriva programkod till processorer.

Elk är ett program som använder sig av metoden linjäriserad förstärkt planvåg (Linearized Augmented Planewave (LAPW)) för att lösa ekvationer av täthetsfunktionalteori (Density Functional Theory (DFT)). [6]

Fermi-energi definieras som en energinivå där antalet tillstånd som har en energi lägre än Fermi-energin är lika med antalet elektroner i systemet. [2]

Fermi-yta är, för elektroners k-punkter i reciproka rummen, isoytan där elektronernas energi är lika med Fermi-energin. [12]

Git är ett decentraliserat versionshanteringssystem. [7]

GUI (Graphical User Interface) är ett grafiskt användargränssnitt. [13]

HDF5 är ett filformat som kan hantera stora mängder data. [8]

Inviwo (Interactive Visualization Workshop) är programvara för visualisering som tillhandahåller en nätverksredigerare för designen av dataflödesnätverk. Noderna i dessa dataflödesnätverk kallas processorer. Indata till nätverket behandlas i dessa processorer och utdata genereras. [15]

Processor är benämningen på ett visuellt funktionsblock i Inviwos nätverksredigerare. I detta dokument avser en processor alltid en inviwoprocessor om inte annat anges.

Python är ett programmeringsspråk. [16]

I Inviwo används Python för att knyta samman processorer.

Reciproka rummen, även känd som k-rummet, är en fouriertransformering av det normala rummet. I reciproka rummen motsvarar varje punkt ett visst k-värde för en partikel. [14]

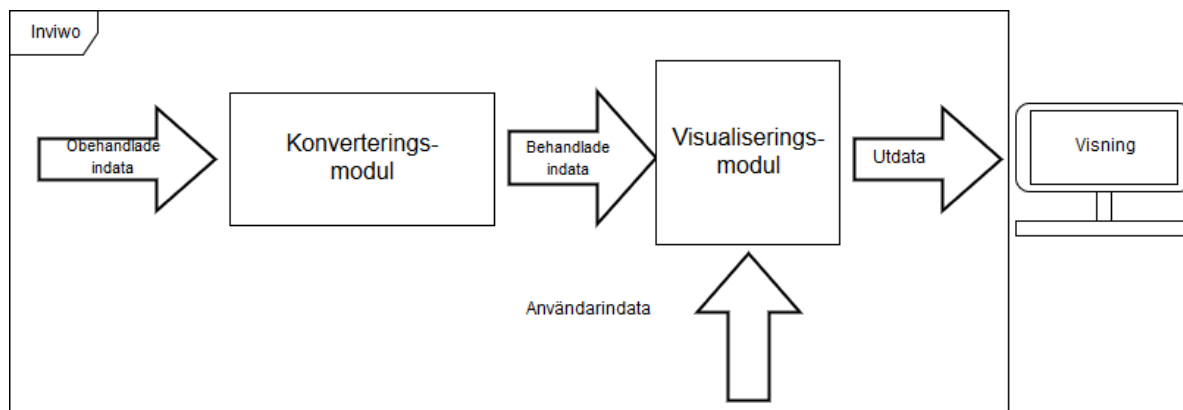
VASP är ett program för modellering på atomnivå, för t.ex. elektronstruktursberäkningar och kvantmekanisk molekylodynamik. [1]

Wigner-Seitz-radie är en radie tillhörande en atom definierad så att summan av volymerna av alla sfärer, bildade för varje atom i enhetscellen utifrån denna radie, är samma som enhetscellens totala volym. I en kristall bestående av flera atomslag finns inget entydigt sätt att välja denna radie. [17]

2 Översikt av systemet

Det färdigställda systemet, se Figur 1, ska kunna hantera indata från VASP och, i mån av tid, även från Elk. Datan kommer från elektronstrukturberäkningar och ska konverteras till HDF5-format för att därefter visualiseras. Användaren ska kunna välja vad som ska visualiseras, kontrollera renderingens utseende och ställa in egenskaper såsom färg, transparens och rotation.

Systemet har delats in i delsystem och dessa kommer utvecklas och testas enskilt i den mån det inte finns beroenden till andra delsystem.



Figur 1: Grov design av systemet

2.1 Beroenden till andra system

Projektet använder sig av Inviwo för att tillhandahålla ett gränssnitt, behandla data och rendera visualiseringar.

2.2 Delsystem

Detta avsnitt behandlar de olika delsystemen projektet innehåller.

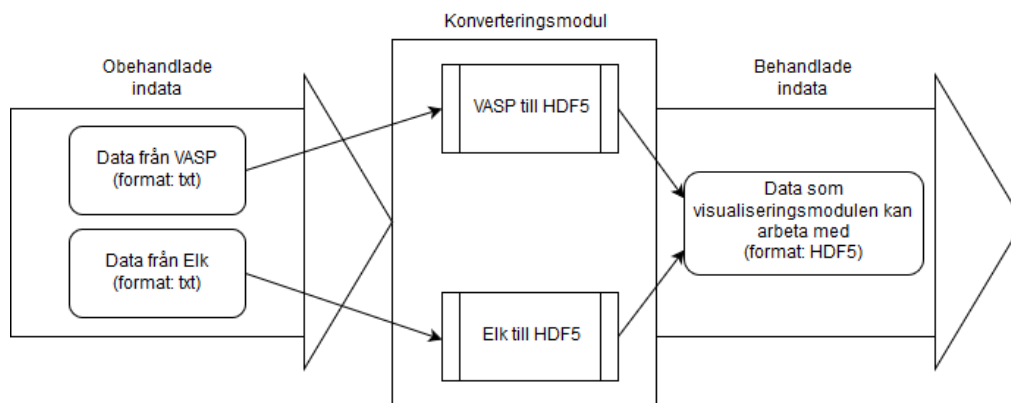
- Datakonvertering VASP
- Datakonvertering Elk
- Visualisering av kristallstruktur
- Visualisering av elektrontäthet
- Visualisering av total och partiell tillståndstäthet
- Visualisering av Fermi-ytor
- Dynamisk visualisering av kristallstruktur

2.3 Designfilosofi

Projektet kommer att drivas med hjälp av versionshanteringsystemet Git och koden kommer vara licensierad med BSD 2, men även utvecklas under Inviwos utvecklaravtal för att, om önskvärt, kunna officiellt integreras i programvaran. Tester kommer skrivas löpande i den mån det är möjligt.

3 Datakonvertering

Detta avsnitt behandlar delsystemet datakonvertering, figur 2 ger en grov översikt av delsystemet. Obehandlad data skickas in, obehandlad data är i form av textfiler med data från beräkningar gjorda i VASP eller Elk. Indata behandlas sedan i konverteringsmodulen. Från konverteringsmodulen kommer behandlad data i HDF5-format.



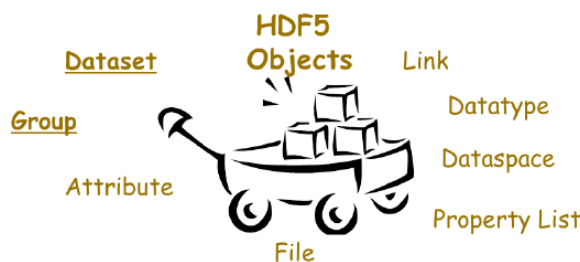
Figur 2: Grov design av konverteringsmodulen

3.1 HDF5

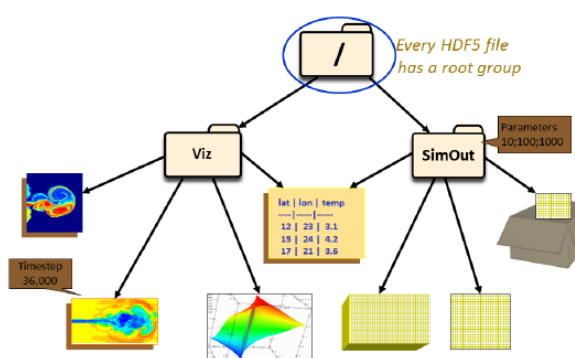
Systemet måste kunna hantera de olika utdatafilerna från VASP. Det bör även kunna hantera utdatafiler från Elk. Detta görs genom att konvertera datan till HDF5-format.

HDF5 är ett filformat som är designat för att hantera stora mängder data på ett flexibelt sätt [8].

HDF5 har flera olika datatyper, se Figur 3, och ett HDF5-objekt som antingen är lagrat på disk eller hålls i minnet är uppdelat i två huvudsakliga underobjekt, nämligen grupper och dataset. [9, s. 3-4]



Figur 3: Överblick av de ingående objektstyperna i HDF5-formatet. Tagen från dokumentet *High Level Introduction to HDF5* [9, s. 3]. Bilden fri att använda enligt HDF Group [5].



Figur 4: Exempel på en HDF5-datastruktur. Tagen från dokumentet *High Level Introduction to HDF5* [9, s. 5]. Bilden fri att använda enligt HDF Group [5].

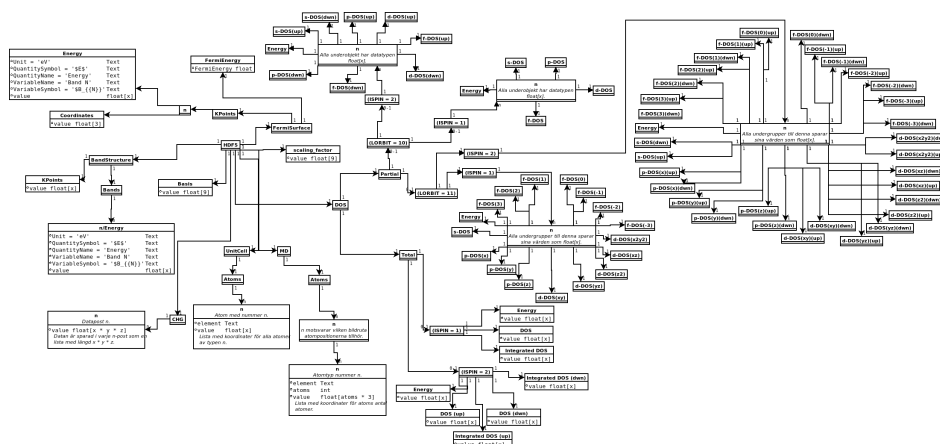
Alla HDF5-objekt har en rotgrupp som äger alla andra objekt i datastrukturen. Denna grupp innehåller i sin tur all övrig data i form av andra grupper, länkar till andra grupper eller dataset. Dataset innehåller rådata av något slag. Rådata kan i sammanhanget vara bilder, utdata från beräkningar, programdata, etcetera. [9, s. 4-5]

I Figur 4 ses ett exempel på hur en HDF5-datastruktur kan se ut. Den har en rotnod betecknad med ett snedstreck som har två undergrupper med namnen Viz och SimOut. De två undergrupperna innehåller sedan data av olika slag. [9, s. 5]

Strukturen i Figur 4 kan liknas vid ett Unix-filsystem [9, s. 5]. Noden Viz kan således kommas åt via sökvägen /Viz, medan noden SimOut kan kommas åt via sökvägen /SimOut [9, s. 5]. Skulle en extra grupp kallad NewGroup läggas till under SimOut skulle den således kommas åt via sökvägen /SimOut/NewGroup.

Två sökvägar kan peka på samma dataset då data kan delas mellan grupper [9, s. 6].

De övriga objektstyperna går inte igenom i detalj i detta dokument, men finns väl beskrivna i *High Level Introduction to HDF5* [9].



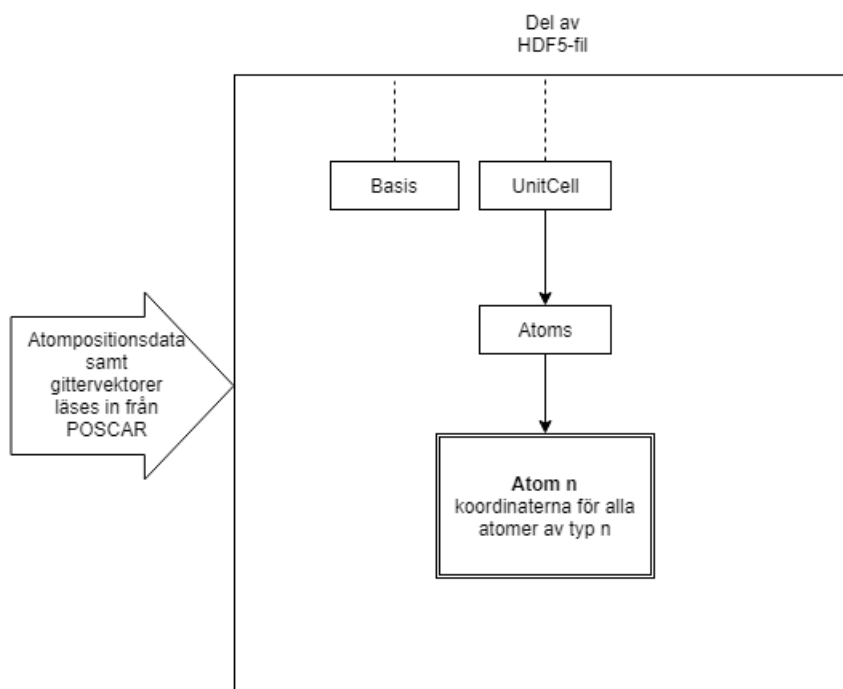
3.2 VASP

Från beräkningsprogrammet VASP fås en rad olika utdatafiler och dessa listas nedan.

- POSCAR innehåller data för enhetscellen samt atompositionsdata.
- CHG innehåller laddningstäthetsdata.
- DOSCAR innehåller tillståndstäthetsdata.
- EIGENVAL innehåller data för alla energier i k-rummet.
- POTCAR innehåller data om atomtyper.
- OUTCAR innehåller all utdata. I detta fall är information om Fermi-energi, den irreducibla 1:a Brillouin-zonens basvektorer och den 1:a Brillouin-zonens symmetriegenskaper vad som söks.
- XDATCAR innehåller data om enhetscell, atompositionsdata för varje beräkningssteg och även atomtyp.
- CONTCAR på samma format som POSCAR men CONTCAR fylls med information om atompositioner uppdaterats.

3.2.1 Kristallstruktur

Atompositionsdata läses in från POSCAR. Dessa kan vara angivna i kartesiska koordinater och måste därför konverteras till koordinater med gittervektorer som bas. Gittervektorer läses in från POSCAR och läggs in i gruppen Basis i HDF5-filen. Atompositionsdatan läggs in i gruppen Unitcell och sorteras in efter atomslag.

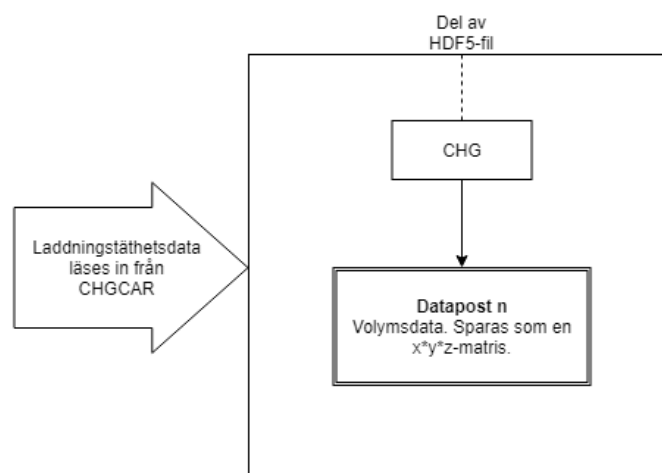


Figur 6: Schematisk bild över sorteringen av data från POSCAR fil.

Figur 6 illustrerar var i HDF5-filen som gittervektorer och atompositionsdata hamnar. Den dubbelstreckade rektangeln representerar datasetet för ett atomslag n och innehåller koordinaterna för alla atomer av det slaget.

3.2.2 Elektrontäthet

Elektrontäthetsdata läses in från utdatafilen CHGCAR. Det är alltså volymsdata som behöver läsas in. Denna volymsdata förs sedan in i HDF5-filen i gruppen CHG och sparas som en matris av dimensionen $x * y * z$.

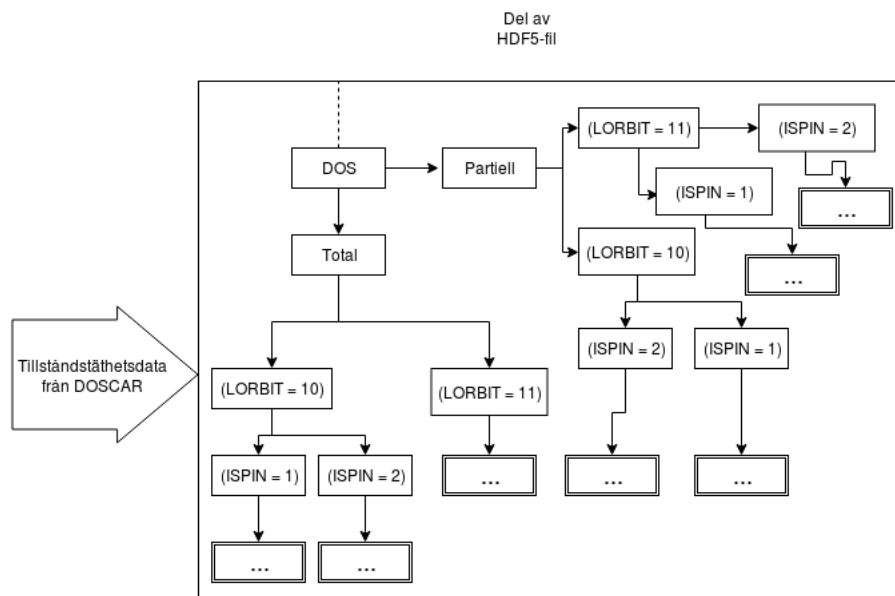


Figur 7: Schematisk bild över sortering av data från CHGCAR fil.

Figur 7 illustrerar sorteringen av volymdatan från CHGCAR.

3.2.3 Tillståndstäthet

Tillståndstäthetsdata läses in från utdatafilen DOSCAR. Flaggorna ISPIN, RWIGS och LORBIT sätts i INCAR-filen och avgör vad som skrivs i DOSCAR-filen. ISPIN-flaggan informerar om spinn har tagits hänsyn till vid beräkningar, RWIGS-flaggan specificerar Wigner-Seitz-radie för varje atomtyp och LORBIT-flaggan (kombinerat med RWIGS) avgör om PROCAR- eller PROOUT-filer skrivs. Vad flaggorna är satta till medför att datan sorteras in på olika sätt i HDF5-filen. Datan förs in i HDF5-filen enligt Figur 8.



Figur 8: Schematisk bild över sortering av data från DOSCAR fil.

Figuren illustrerar en del av HDF5-filen som en stor rektangel och inuti den beskrivs uppdelningen av den inlästa datan från DOSCAR-filen. Detaljer kan läsas ur Figur 5.

3.2.4 Fermi-yltor

Varje k-punkts möjliga energier parsas från EIGENVAL-filen. OUTCAR-filen parsas för att hitta Fermi-energin och basvektorer för den irreducibla 1:a Brillouin-zonen i kartesiska koordinater, samt den irreducibla zonens symmetrigrupp.

3.2.5 Dynamisk kristallstruktur

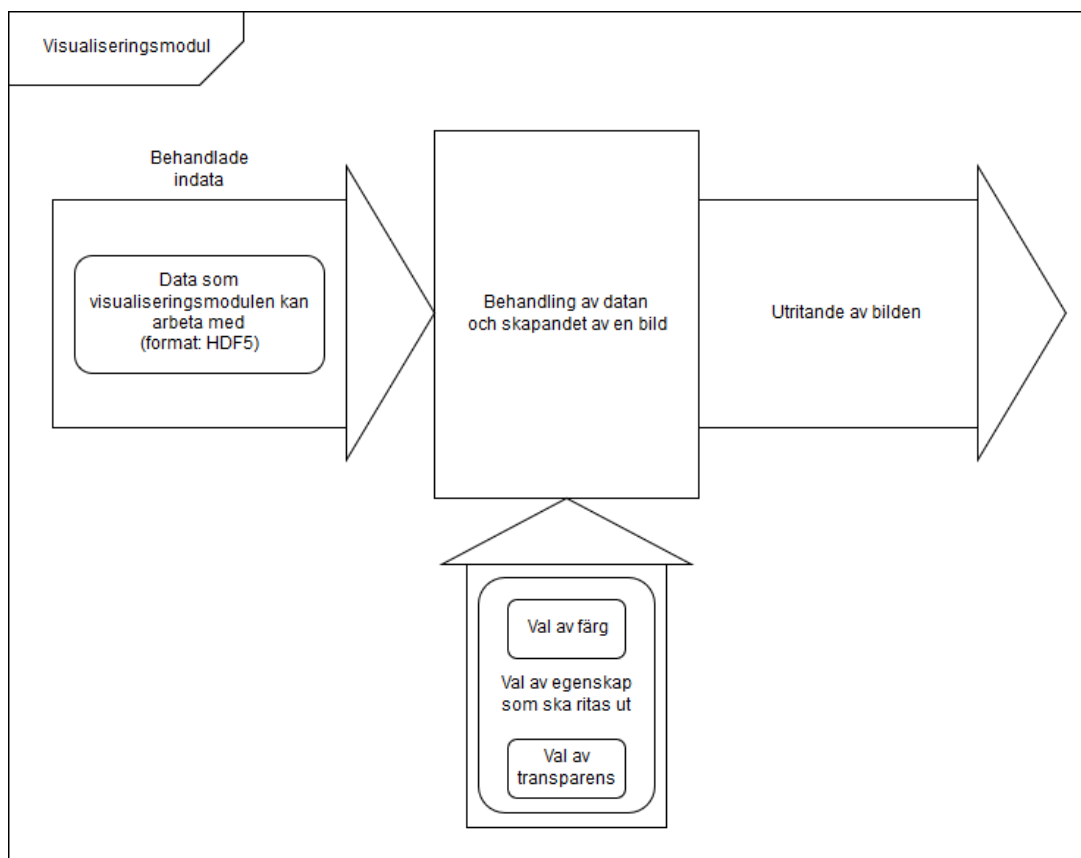
Delsystemet läser in utdatat från XDATCAR. Även data för enhetscell och atomposition fås från XDATCAR. Den data som fås är alltså en lista med data om enhetscell och atomposition för varje beräkningssteg där någon atomposition har ändrats.

3.3 Elk

Om tid finns så kommer även utdata för elektrontäthet från Elk, ett annat beräkningsprogram än VASP, behandlas. Från Elk ska elektrontäthet visualiseras. För detta måste utdatafilerna GEOMETRY.OUT och RHO3D.OUT läsas. I GEOMETRY.OUT finns bland annat data för atompositioner för att kunna bygga upp geometrin. Från filen RHO3D.OUT fås elektrontäthetsdatan.

4 Visualisering

Figur 9 visar hur delsystemet Visualisering är uppbyggt. Behandlad samt användarindata skickas in. Användarindata består av val av färg, val av egenskap som ska ritas ut, val av transparens etc. Dessa behandlas sedan i visualiseringsmodulen som skapar en bild utifrån dem. Den utritade bilden skickas sedan ut från modulen. Detta är en allmän beskrivning av visualiseringen, för de olika egenskaperna som ska visualiseras kommer mittenrektangeln i Figur 9 att vara uppbyggd på olika sätt med olika processorer. Detta går mer in på i avsnitt 4.5-4.8.



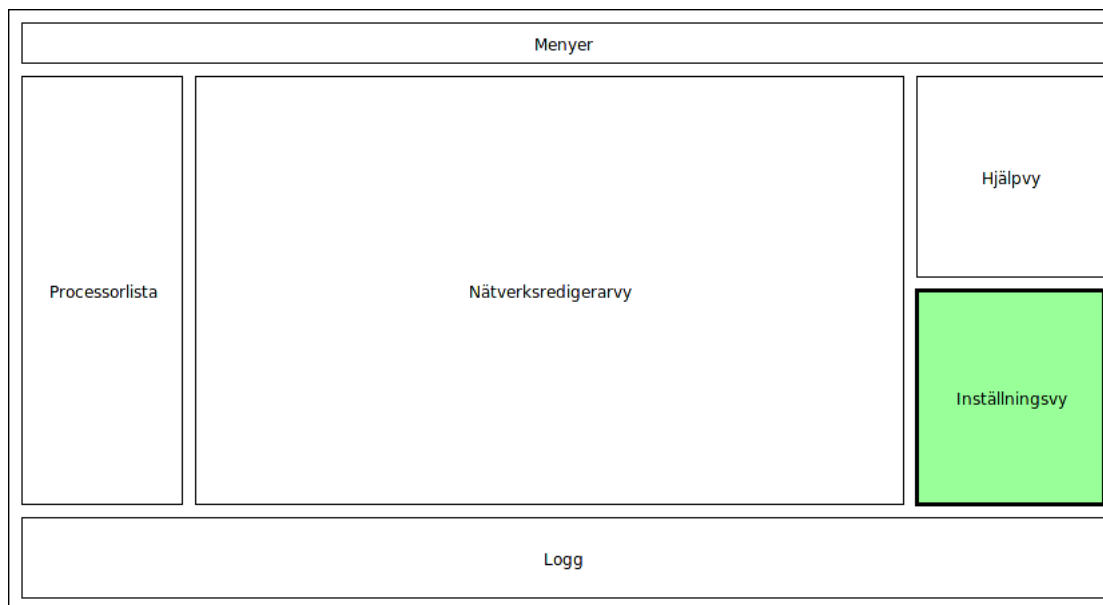
Figur 9: Grov design av delsystemet Visualisering

4.1 Utritning

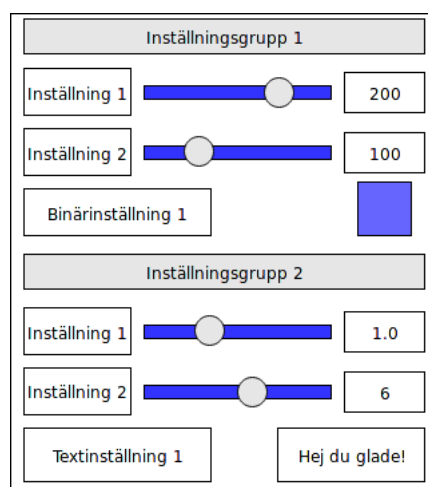
Delsystemet ska rita ut bilder utefter den behandlade datan. Utritningen skall ge en visualisering av den egenskap som modulen behandlar och kan till exempel vara en volymrendering eller en vanlig 2D-graf, beroende på vilket som är mest lättolkat.

Utritningen görs via Inviwos inbyggda funktionalitet för att rendera via OpenGL. Grafisk behandling görs också med Inviwo och OpenGL, där möjligheten finns att både använda färdig funktionalitet och att utveckla nya renderingsfunktioner.

4.2 Användarindata



Figur 10: Skiss som visar det tänkta användargränssnittet.



Figur 11: Skiss som visar den tänkta inställningsrutan i Inviwo.

Användaren kommer kunna ändra inställningar som kontrollerar utseendet på visualiseringarna. Denna indata matas in i Inviwos användargränssnitt, se Figur 10, och i inställningsrutan för en processor, se Figur 11.

Visualiseringsfunktionaliteten kommer att läggas till i form av processorer som via Inviwo tillhandahåller inställningsmöjligheter på det sätt som beskrivits ovan. Således blir inte användarinställningar ett eget separat delsystem, utan kommer att bakas in i visualiseringssystemen. Detta görs genom att processorerna skrivs att exponera de inställningar som användaren är menad att justera. Inviwo kommer då automatiskt lägga till inställningarna i rutan skisserad i Figur 11.

4.3 Interaktivitet

Användaren ska kunna modifiera visualiseringen genom att reglera ett intervall av värden för någon egenskap, där full transparens fås för alla värden inom intervallet, rotera 3D-bilder etc.

Ny användarindata, som skickas in efter att den första bilden har ritats upp, skickas tillbaka till visualiseringsmodulen för att utföra en ny rendering.

4.4 HDF5-indata

I Inviwo 0.9.9 finns en modul som tillåter importering av data från HDF5-fil [10]. Denna kommer användas för att ladda in data som sedan ska visualiseras. Se avsnitt 3.1 för beskrivning av uppbyggnaden av HDF5-filen.

4.5 Kristallstruktur

Kristallstruktur ska visualiseras som atompositioner i enhetscellen. Sfärer som representerar atomer ska ritas ut genom volymsrendering. Nedan följer en översiktlig beskrivning av hur detta ska byggas upp.

4.5.1 Översiktlig beskrivning

Koordinaterna (tre dimensioner) för atomerna läses in från HDF5-fil. Utifrån dessa positioner ritas sedan atomerna ut. Färginställningar för atomerna kan ändras av användaren.



Figur 12: Översiktligt flödesdiagram över dynamisk visualisering av kristallstruktur.

I Figur 12 beskrivs detta med ett flödesschema. Inviwo-nätverket näst sist i flödesschemat skickar ut en 3D-bild av kristallstrukturen där atomerna har placerats ut på sina positioner som hämtats från HDF5-filen.

4.5.2 Inviwo-nätverk för visualisering av kristallstruktur

En 3D-bild ska ritas ut vilket kommer kräva ett antal steg. Varje steg kommer att implementeras i Inviwo med en eller flera processorer.

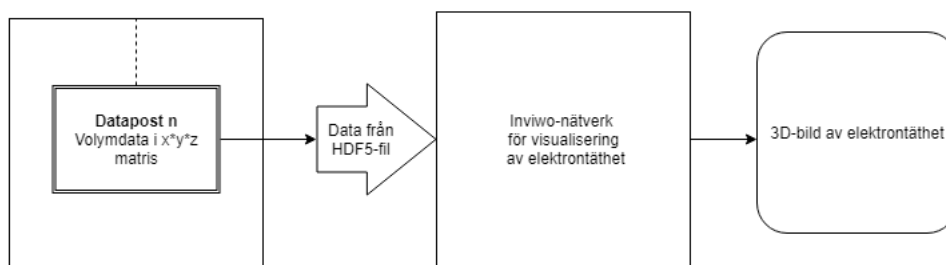
- HDF5-inläsning
 - Ladda in data från HDF5-fil.
- Koordinatläsning
 - Läs in koordinaterna för atomerna från HDF5-fil.
- Mesh-konstruering
 - Bygga upp ett mesh utifrån koordinaterna från koordinatläsar-processorn.
- Canvas
 - Visa 3D-bilden.

4.6 Elektrontäthet

Elektrontäthet visualiseras genom volymsrendering. Nedan följer en översiktlig beskrivning av hur detta ska byggas upp.

4.6.1 Översiktlig beskrivning

Datan som ritas ut är sannolikheten för att hitta elektroner på olika positioner. Sannolikheterna visualiseras med färger och skiftande transparens.



Figur 13: Översiktligt flödesdiagram över visualisering av elektrontäthet.

4.6.2 Inviwo-nätverk för visualisering av elektrontäthet

En 3D-bild ska ritas upp vilket kräver ett antal steg som implementeras med processorer i Inviwo.

- HDF5-inläsning
 - Ladda in data från HDF5-fil.
- Konvertering till volym

- Konvertera datan till en volym.
- Canvas
 - Visa 3D-bilden.

4.7 Tillståndstäthet

Tillståndstätheten, total och partiell, ska visualiseras med en 2D-graf. Nedan följer en översiktlig beskrivning av hur detta ska byggas upp.

4.7.1 Översiktlig beskrivning

Data hämtas från dataseten i gruppen tillståndstäthet i HDF5-filen. För partiell-tillståndstäthet hämtas energi, tillståndstäthet och så vidare för varje atom. Detta ska sedan plottas i en 2D-graf med tillståndstätheten på y-axeln och energin på x-axeln. Energin ska ha sitt nollställe vid Fermi-energin.



Figur 14: Översiktligt flödesdiagram över visualisering av tillståndstäthet.

I figur 14 beskrivs förloppet översiktligt. För partiell-tillståndstäthet hämtas alltså datan i första rektangeln i figuren för varje atom. För total-tillståndstäthet är detta dataseten för gruppen Total. Ett Inviwo nätverk för visualisering av tillståndstäthet måste skapas, detta nätverk illustreras i figur 14 av en stor kvadrat som skickar ut en 2D-graf på en canvas.

4.7.2 Inviwo-nätverk för visualisering av tillståndstäthet

En 2D-graf ska ritas upp vilket kräver ett antal steg. Dessa implementeras i Inviwo på en eller flera processorer.

- HDF5-inläsning
 - Ladda in data från HDF5-fil.
- Plottning
 - Plotta en graf av den laddade datan.

- Canvas
 - Visa den plottade grafen.

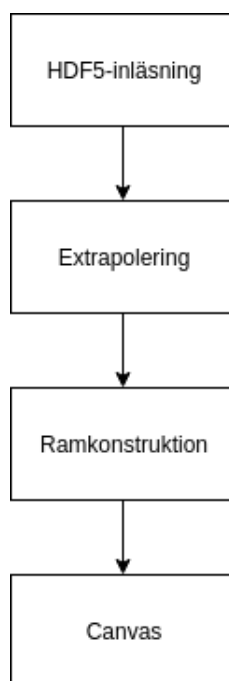
4.8 Fermi-yltor

Fermi-yltor och godtyckliga energi-isoytor ska kunna ritas upp i reciproka rymden, k -rummet. Detta görs genom att den 1:a Brillouin-zonen ritas upp och inuti denna ritas ytan.

4.8.1 Inviwo-nätverk för Brillouin-zonuppritning

Fermi-yltor ritas upp inuti 1:a Brillouin-zonen, som är den samling k -punkter i reciproka rymden vars närmsta gitterpunkt är origo. Den irreducibla 1:a Brillouin-zonen är den minsta samling vektorer som behövs för att extrapolera den fullständiga zonen, som kan byggas upp från kristallens symmetrigrupp. Brillouin-zonuppritningsnätverkets uppgift är att ta den irreducibla 1:a Brillouin-zonens basvektorer i kartesiska koordinater samt dess symmetriegenskaper, beräkna alla k -punkter som ingår i den 1:a Brillouin-zonen och rita upp en ram för denna. Basvektorerna parsas från OUTCAR-filen av en VASP-beräkning rörande Fermi-yltor, och återfinns på HDF5-format.

Inviwo-nätverket för att rita upp den 1:a Brillouin-zonen består av ett antal steg som beskrivs nedan. Se figur 15 för en skiss av hur dessa förhåller sig till varandra. Varje steg implementeras i Inviwo i form av en eller flera processorer.



Figur 15: Översiktlig beskrivning av de nödvändiga stegen för Brillouin-zonuppritning

- HDF5-inläsning
 - Hämtar data om basvektorerna för den irreducibla 1:a Brillouin-zonen
- Extrapolering
 - Använder basvektorerna och symmetrigruppen från HDF5-inläsningen för att extrapolera vilka punkter som ingår i den 1:a Brillouin-zonen

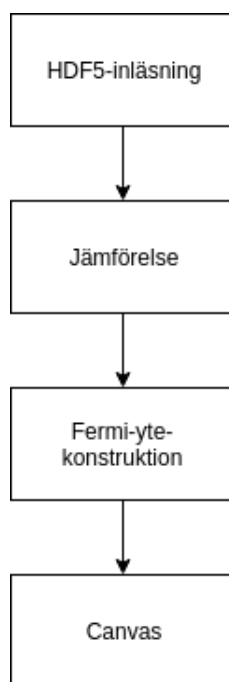
- Ramkonstruktion
 - Beräknar vilka k-punkter som utgör “ramen” för 1:a Brillouin-zonen
- Canvas
 - Ritar upp ramen

4.8.2 Inviwo-nätverk för Fermi-yte-uppritning

En sorts utdata från VASP-beräkningar är möjliga energier för alla k-punkter i reciproka rummet. Fermi-yte-uppritningsnätverkets uppgift är att ta dessa punkter med tillhörande energier och beräkna samt rita upp isoytan där energin är lika med Fermi-energin inuti ramarna för 1:a Brillouin-zonen. Energierna för alla k-punkter parsas från EIGENVAL-filen. Fermi-energin parsas från OUTCAR-filen. Dessa data återfinns på HDF5-form.

Denna visualisering ska även kunna göras för en godtycklig energi, inte bara för Fermi-energin. Denna energi väljs med hjälp av en slider eller genom att manuellt skriva in önskad energi.

Inviwo-nätverket för att rita upp Fermi-energin består av ett antal steg som beskrivs nedan. Se figur 16 för en skiss av hur dessa förhåller sig till varandra. Varje steg implementeras i Inviwo i form av en eller flera processorer.



Figur 16: Översiktlig beskrivning av Inviwo-nätverket för Fermi-yte-uppritning

- HDF5-inläsning
 - Hämtar data om Fermi-energin samt de möjliga energierna hos alla k-punkter inuti 1:a Brillouin-zonen
- Jämförelse

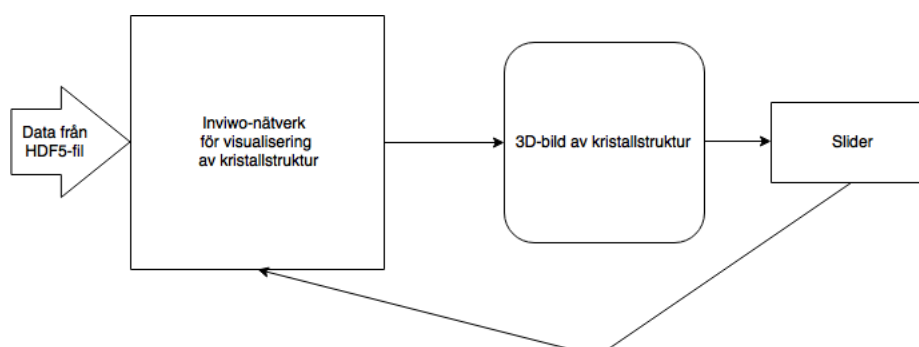
- Jämför de möjliga energierna för varje k-punkt med Fermi-energin för att se vilka k-punkter som ingår i Fermi-ytan
- Fermi-yte-konstruktion
 - Beräknar vilka k-punkter som utgör Fermi-ytan
- Canvas
 - Ritar upp Fermi-ytan

4.9 Dynamisk visualisering av kristallstruktur

Dynamisk visualisering baserad på en serie atompositioner. Användaren ska med hjälp av en slider kunna ändra visualiseringen.

4.9.1 Översiktlig beskrivning

Koordinaterna (tre dimensioner) för atomerna läses in från HDF5-fil. Utifrån dessa positioner ritas sedan atomerna ut. Användaren kan sedan med hjälp av en slider ändra tiden och se hur visualiseringen ändrar sig därefter. Se figur 17 för ett enklare flödesschema.



Figur 17: Översiktligt flödesdiagram över dynamik.

4.9.2 Inviwo-nätverk för dynamisk visualisering av kristallstruktur

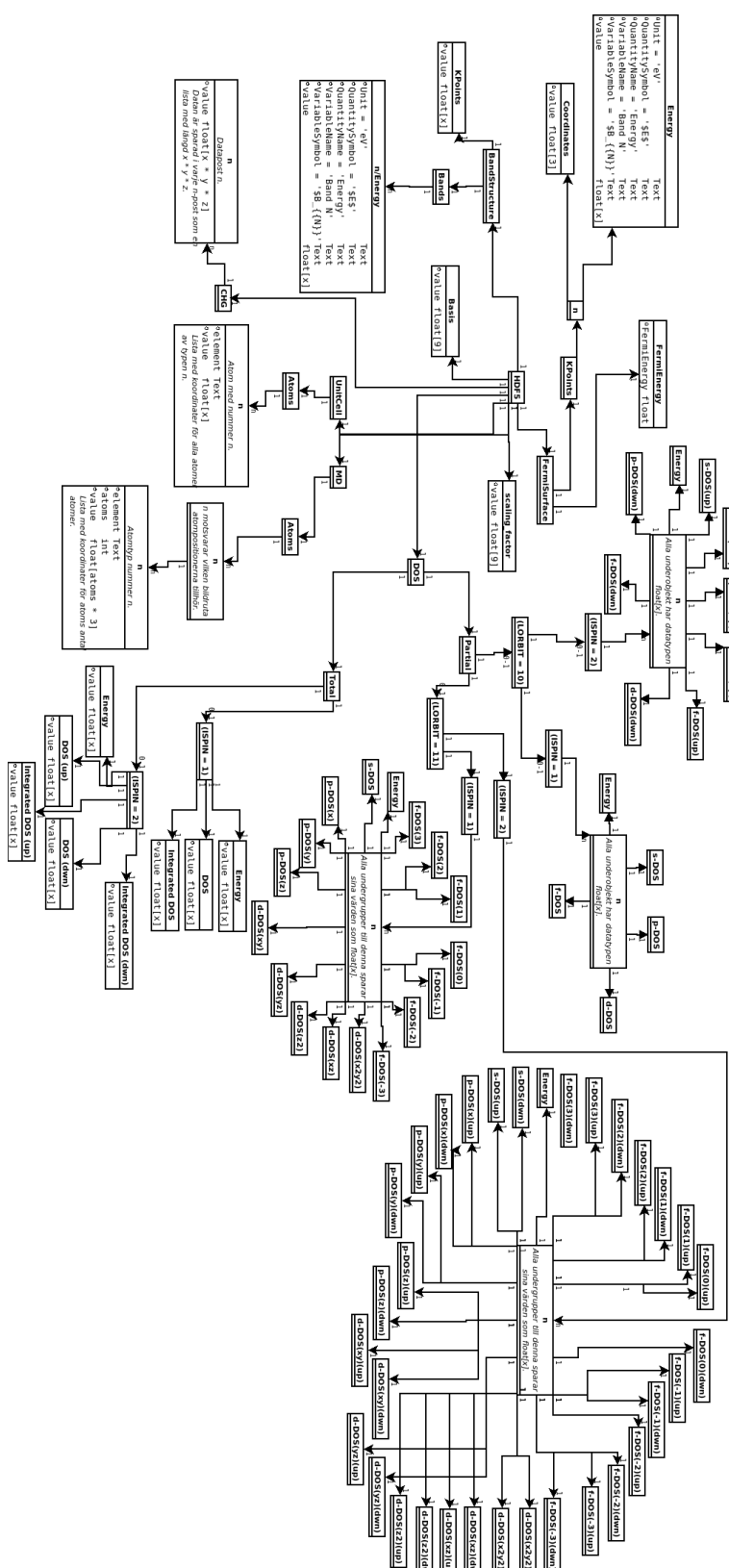
En 3D-bild ska ritas ut med hjälp av ett antal steg. Det är denna 3D-bild som sedan kommer att uppdateras om användaren väljer att justera slidern för tid. Varje steg kommer att implementeras i Inviwo med en eller flera processorer.

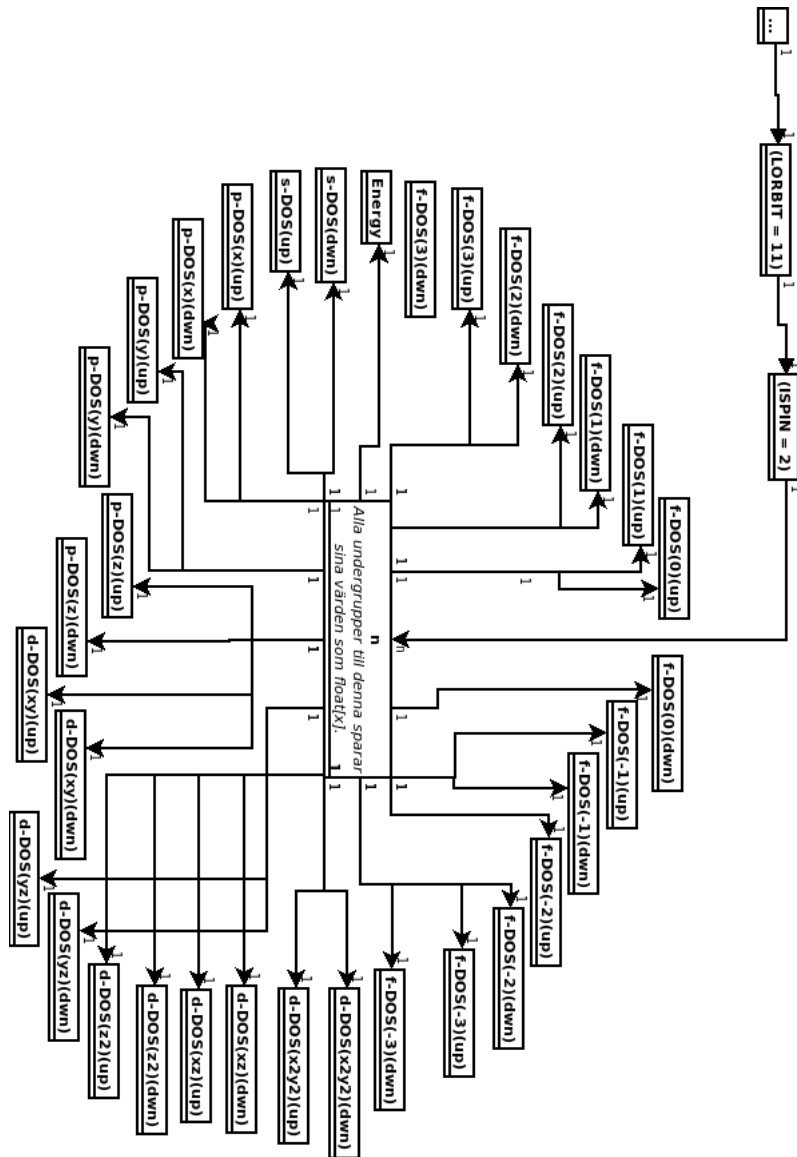
- HDF5-inläsning
 - Ladda in data från HDF5-fil.
- Koordinatläsning
 - Läs in koordinaterna för atomerna från HDF5-fil.
- Mesh-konstruering

- Bygga upp ett mesh utifrån koordinaterna från koordinatläsar-processorn.
- Canvas
 - Visa 3D-bilden.
- Slider
 - Ändra tiden.
- Koordinatläsning...

Referenser

- [1] *About VASP*. URL: <https://www.vasp.at/index.php/about-vasp/59-about-vasp> (hämtad 2018-02-22).
- [2] Neil Ashcroft och David Mermin. *Solid State Physics*. 1976, s. 141.
- [3] *BSD2*. URL: <https://opensource.org/licenses/BSD-2-Clause> (hämtad 2018-02-23).
- [4] *C++*. URL: <http://www.cplusplus.com/info/description/> (hämtad 2018-02-23).
- [5] *Copyright Notice and License Terms for HDF5 (Hierarchical Data Format 5) Software Library and Utilities*. URL: <https://support.hdfgroup.org/ftp/HDF5/releases/COPYING> (hämtad 2018-05-25).
- [6] *ELK*. 31 mars 2017. URL: <http://elk.sourceforge.net/elk.pdf> (hämtad 2018-02-22).
- [7] *Git*. URL: <https://git-scm.com> (hämtad 2018-02-23).
- [8] The HDF Group. *Hierarchical Data Format, version 5*. 1997-2018. URL: <https://support.hdfgroup.org/HDF5/> (hämtad 2018-02-21).
- [9] The HDF Group. *High Level Introduction to HDF5*. 23 sept. 2016. URL: <https://support.hdfgroup.org/HDF5/Tutor/HDF5Intro.pdf> (hämtad 2018-02-21).
- [10] *Inviwo Release. HDF5 module*. URL: <https://www.inviwo.org/2017/10/04/inviwo-0-9-9-released/> (data) (hämtad 2018-02-27).
- [11] *Nationalencyklopedin. API*. URL: <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/api-> (data) (hämtad 2018-02-23).
- [12] *Nationalencyklopedin. Fermi-yta*. URL: <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/fermi-yta> (hämtad 2018-02-23).
- [13] *Nationalencyklopedin. GUI*. URL: <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/api-> (data) (hämtad 2018-02-23).
- [14] *Nationalencyklopedin. reciproka rymden*. URL: <http://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/reciproka-rymden> (hämtad 2018-02-27).
- [15] *Overview - Inviwo*. URL: <http://www.inviwo.org/overview/> (hämtad 2018-02-23).
- [16] *Python*. URL: <https://www.python.org> (hämtad 2018-02-23).
- [17] *VASP. Wigner-Seitz-radie*. URL: <https://cms.mpi.univie.ac.at/vasp/vasp/RWIGS.html> (hämtad 2018-03-08).





Figur 20: Dataformatet som används när VASP konverteras till HDF5 del 2.