Development of crystallographic surfaces for modelling interactions

Peter S. Ford

Development of Crystallographic Surfaces for Modelling Interactions

Thesis submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy at the University of Durham

Peter S. Ford

June 13, 1997

Abstract

This thesis addresses two separate problems - an investigation of the interaction of probe molecules with crystalline rutile and an investigation of the environment of group IA and IIA elements in organometallic compounds.

Ab-initio Hartree-Fock calculations have been performed, aimed at investigating the interactions between the ionic surface of a crystal and an adsorbate molecule. Titanium dioxide, a material important for catalysis, electronic components and pigments, was chosen as the substrate, with carbon monoxide as the probe molecule. The calculations were carried out using the Crystal92 program, for the (110) surface of the Rutile polymorph of TiO_2, employing a slab with a thickness of 5 atomic layers. The calculations investigated two orientations of the CO molecule with the molecular axis perpendicular to the surface. Results are reported showing contour diagrams for slices through the energy hypersurface parallel and perpendicular to the surface of the substrate. In order to facilitate the work described above, a program 'Builder2' was developed . This provides a convenient means for generating models of slabs of material from crystal structure data. Part of the development of Builder2 was to devise computer code to decompose standard Space Group symbols into the underlying symmetry matrices. The code for Builder2 is proprietary to Oxford Materials Ltd. and forms part of a commercial product.

The environment of group IA and IIA elements in crystalline materials has not been the subject of any reported investigation. These elements, and organic ligands associated with them, play a significant role in biological systems. Around 16,000 atomic environments were extracted from the Cambridge Crystallographic Database to provide an up-to-date analysis of actual environments. The results are presented as histograms and tables,and suggestions are made for future extension of the analyses.

The following are due acknowledgement for their help throughout this project:

Professor Judith A.K. Howard, for leadership, encouragement and parties.

Dr. Elizabeth Colbourn and Oxford Materials Limited, for financial support and direction.

The Durham University Crystallography group and former members, especially Roy Copley, Christian Lehmann, Jason Cole, Garry Smith, Vanessa Hoy and Claire Wilson.

Dr. Nic Harrison at DRAL Daresbury, for technical advice on the use of CRYSTAL92.

EPSRC for funding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Connection

The nature of a material is, to a very good approximation, wholly dependent on the behaviour of the electrons in the atoms making up that material. An essential first step towards understanding the electronic structure of the material is to elucidate the atomic structure, from which assumptions about the behaviour of electrons can lead to a model of the material. Currently the most complete method of determining the structure of a solid is single-crystal X-ray or neutron diffraction, which can provide a precise explanation of the locations of atoms within a molecule or lattice structure. Using appropriate X-ray and neutron diffraction methods it is possible to describe in detail not only the positions of nuclei, but also the distribution of electrons in a system. Once the electron distribution pattern is known, the macroscopic properties

of any material should be open to understanding.

In practical terms, it is often not possible to study a material or a system with a diffraction experiment. Interactions between solids and fluids do not produce tidy crystalline snapshots of what is happening, so a precise determination of the structure of the system is not available. The premise that knowledge of the electron distribution implies full knowledge of the system still holds however, and an alternative method of understand the electron distribution should be sought. Computational and theoretical chemistry have both provided ways of approximating closely the complex interactions between atoms, although not always by consideration of electronic interactions. Once a suitable method of evaluating electron distribution, or total free energy of a system, has been established, computer simulation can produce very convincing models of how the real world behaves at a microscopic level.

It is still necessary, before a successful computer simulation can be prepared, to understand the atomic structure of a system. Since it is currently impossible to calculate analytically the interactions between atoms and electrons completely for all useful systems, computer programs apply iterative methods to make the system self-consistent, or minimise the overall energy of the system. To produce a realistic final model by an iterative process usually requires that the starting point is not too much different from the final result, otherwise most iterative processes will find false localised minima. It is realistic to assume that the crystalline form of a material lies at an energy minimum with respect to the parameters which define the structure, and

so for most modelling purposes, a structure derived from diffraction experiments will provide a good starting point, and additionally provide a check on the validity of the calculation method.

A simulation of a chemically interesting system will inevitably contain a large number of atoms, and hence an even larger number of electrons. The definitions of 'large' and 'even larger' are gradually increasing with improvements in computer technology, but it always makes sense to minimise the number of variables in a system where possible. Symmetry is applied as a matter of course to the refinement of crystallographic models, which are small calculations compared to most molecular simulations. It also makes sense in an electronic calculation to assume that where two orientations of a system are indistinguishable from each other, then symmetry can be applied to reduce the scale of the computational problem.

1.2 Visualisation

With computational experiments, the bias of the workload has moved from the effort of calculating the actual values, toward the interpretation of the data. In the experiments detailed in the present work, the calculation of the energy of the TiO_2 - CO interaction at each data point was a matter of hours of processing time on a sufficiently large computer. However, collating the information contained within the several hundred data values and extracting a meaning is somewhat more difficult. There are scientists who are able to

take pages of raw numbers and detect correlation and patterns within them. Most people find a graphical representation more informative.

Computer software developers have provided many solutions to this problem, and the options available to the scientist are many and varied. Truly spectacular images can be produced from almost any data set, incorporating bright colours and clever "lighting" effects. Some of these effects are useful in clarifying the three-dimensional nature of the subject, but it is all too easy to sacrifice information content for appearance.

Presentation of the results from the TiO_2 surface work proved a particularly difficult task. The data points vary in all three physical dimensions, with an energy value at each point. Representing a volume of data so that the value at each data point is discernible, is very difficult. An object, for example a sphere of size related to the value, can be placed at the co-ordinates of the data point, but these objects tend to obscure each other. Even using colour to represent value is not ideal, since the points deep inside the volume cannot be seen. Some improvement can be made by using transparency to allow the observer to see through the outer layers. In addition, it is possible to generate an isosurface linking points of equal value, and indeed several of these could be superimposed in the same graphic to give an effect of "contour" levels. All of these methods have been tried during the course of the analysis of surface-interaction data, until a graphical model useful to the author was developed. Unfortunately, the three-dimensional nature of this model does not translate well to the two-dimensional paper of this thesis,

and for some details slices through the volume, represented as contour maps, may provide more information.

Crystallographic structures are relatively easy to represent on a piece of paper, since the atom positions form a sparse data set with plenty of space between them. Three-dimensional effects are still very useful in this case, to help distinguish the depth differences in the diagram. Well known techniques most often used are hidden-line removal, depth-cueing and perspective drawing. All help to give the impression of some atoms being behind the others, although the ideal representation of the molecule is one which avoids hiding atoms from view. Most molecular graphics programs are capable of calculating the "best" view of a molecule, and letting the observer rotate to a desired orientation, often in real time.

Chapter 2

Main Group Elements Bonding Survey

2.1 Introduction

Group IA and IIA elements are currently attracting attention because of their role in biological systems. Transport of these ions across membranes, and their concentrations in cell structures, usually depends upon their association with organic ligands. Clinically and in research, ligands binding to these ions are used to monitor their movement, and in their assay. For example, fluorescent calcium-biding ligands are used to estimate calcium levels in body fluids. The effective bonding radii of d- and f- block metals have already been analysed, but no systematic analysis has yet been performed on the alkali and alkaline earth metals. Code was developed to facilitate the

interrogation of the Cambridge Crystallographic Database. 16,000 suitable entries were identified, and the results of this analysis are presented as tables and histograms. Examination of these results indicate that there are inconsistencies in the database, and that the data would be amenable to further analysis.

The use of electronic databases to store information about molecular structure stems from a statement by J.D. Bernal in 1948 [1] about the problems of disseminating scientific information. The Cambridge Crystallographic Data Centre (CCDC) came into being in 1965, with the aims of providing a database of organic and organo-metallic crystal structures, providing the software to access such a database, and to use the database for research into structural properties.

The database currently includes structures of all carbon containing compounds, determined by X-ray or neutron diffraction, published in the literature. It is now standard for most journals featuring crystallographic data to deposit the structural details with the CCDC. The database currently includes protein crystal structures in an agreement with the Brookhaven Protein Data Bank, but these structures are not stored to the same level of detail and are separate from the main database. Most of the entries in the main database contain full structural data, which has been error-checked before inclusion. Entries with no atomic coordinates, or poor accuracy, are marked as such and can be excluded from a search of the database.

The most recent published work concerning the analysis of bond lengths

in structures stored in the Cambridge Crystallographic Database appeared in 1987 [2], with a supplement being published in 1989 [3]. This search was initially limited to the non-metallic elements H, B, C, N, O, F, Si, P, S, Cl, As, Se, Br, Te and I. Mean values were presented for some 682 bond types identified between these elements, including separate entries for different functional groups; for example, carbonyl C-O is separated from alcohol C-O. The supplement extended the scope of the survey to include bonds involving *d*- and *f*-block metals; that is Sc-Zn, Y-Cd, La-Hg, Ce-Lu and Th-U. The September 1985 version of the database was used throughout this work. At this time, there were approximately 55000 structures stored in the database. The database searching and analysis of data was done on IBM3081D (part 1) and VAX 11/750 (supplement), using programs then present as part of the database distribution.

2.2 Methods and Programs

2.2.1 Introduction

The period between the previous study and the present work has seen a general increase in the rate of growth of the Cambridge Crystallographic Database. The continual development of the software available to access it, and the hardware on which to run the programs, has made the process of extracting such data from the Database much more rapid. The present work takes advantage of these improvements to fill a gap in the tables of

bond length data. The relatively low number of Group IA and IIA metal structures in the database means that a detailed analysis of the functional groups involved is not really possible, since the calculation of average bond lengths from such small samples would be statistically unrealistic. The data are ordered in terms of the "metal centre" and the bonded atom, indexed by coordination number about the metal centre.

The Cambridge Crystallographic Database is distributed to users on CD-ROM, and can readily be installed onto a local computer system. This facilitates very rapid searching of the database, since no network activity is required. The normal user interface is a full-function graphical screen which allows the user to sketch a search fragment and define various parameters interactively. While this is extremely useful for searching specific molecules or classes of molecules, large scale extraction of data is probably best performed using the non-interactive part of the QUEST program. The present work was performed on a Hewlett-Packard 9000 series 720 workstation, using version 5 of the QUEST program.

The interactive interface was used, in the early stages of this work, to establish a "question" which would extract the required information from the database. A new element type, named AZ, was defined to include the existing AA (any atom) type and also add H, which is specifically excluded from the AA type. A simple search fragment, consisting of one AZ atom linked to on metal atom by a bond of any type, was defined. The three-dimensional constraint menu was invoked to add a parameter label, M2AZ, to

the bond distance, thus ensuring that the metal–ligand distance was recorded for analysis.

Once a general form of this question was established, a more automatic sequence of programs could be used to do the lengthy extraction and tabulation of the data. A combination of UNIX C-shell and FORTRAN programs were used, with the C-shell programs handling the control of the process and the FORTRAN programs extracting data from the output of QUEST. A great advantage of this automated analysis of the data is that it can be readily repeated at any stage. The set of programs and scripts could be run to process later releases of the database, dynamically updating the information to include all new structures stored.

2.2.2 Program Details

Refer to Figure 2.1.

The C-shell script **start** was used to control the overall process of data extraction. A list of atom types was given on the command line. The types in the list were processed in order to produce a "virtual database": a subset of the full database containing only structures with the atom type in question, and only those entries where atomic coordinates are stored. The search is performed non-interactively, and the resulting files placed in a subdirectory bearing the name of the atom type. Extensive use was made of the **awk** programming language [4] for creating the input files; **awk** is a standard component of most UNIX operating systems.

Once the virtual database for each atom type was created, **start** prepares input files for a second search, this time on the newly created virtual database. Another C-shell program, **getdist**, is called to search for the appropriate type of Metal–Ligand bonds. The search fragment will locate all occurrences of the metal atom concerned, joined by any type of bond to any other atom. This includes metal–hydrogen bonds, all metal–non-metal bonds and also any metal–metal bonds incorporating the metal concerned. Where there is more than one fragment of this type in a single structure, all such fragments are reported.

The output from this search is filtered through the UNIX **sort** command and used as input for the FORTRAN program **getdist.x**. **getdist.x** sorts the entries in the QUEST output, grouping entries together for each type of ligand atom and each coordination number. The **getdist** script then merges this output into a final file, which is fed into another FORTRAN program, **table**. The output from **table** is in several forms, including plain text and LaTeX source code commonly used for scientific reports including the present work.

The output from each search is in two files. The table file contains an entry for each fragment which matched the search, which gives the reference code of the structure and the bond distance. The other file is in the CCD's proprietary "FDAT" format, which contains the crystal structure information for each molecule which match the search pattern. This file format, detailed in the database manual [5] is fairly complex, but the crystalline coordinates

can be extracted and compared with the bond length data in the other file to determine the coordination around the metal atom. **getdist.x** cross-references every fragment in the table file to the structural data in the FDAT file to determine what type of atom is bonded to the metal, and how many atoms are in the coordination sphere of each metal. For an extra check on the bond, the distance in the table file is compared with a distance computed from the atomic coordinates in the FDAT file.

The metal–ligand fragments are grouped into separate output files by total coordination number about the metal. Within these files the bond distances are grouped by atom type, and the mean and standard deviation recorded. The files are filtered within the **getdist** program using a more complex **awk** program and a final file is produced; the **tables** program then processes this file into a more readable form.

2.3 Results

The results of this database survey are presented in table form in Tables 2.2 to 2.17 (pages 25 to 40). The tables give, for each metal type, an entry for each ligand atom indexed against the total coordination number for the metal centre. Each entry gives the number (n) of examples of this metal–ligand bond found, and the mean (μ) and sample standard deviation (σ) of the bond length in Ångstrom. The survey is based upon the April 1996 release of the Cambridge Crystallographic Database, which contains 152464

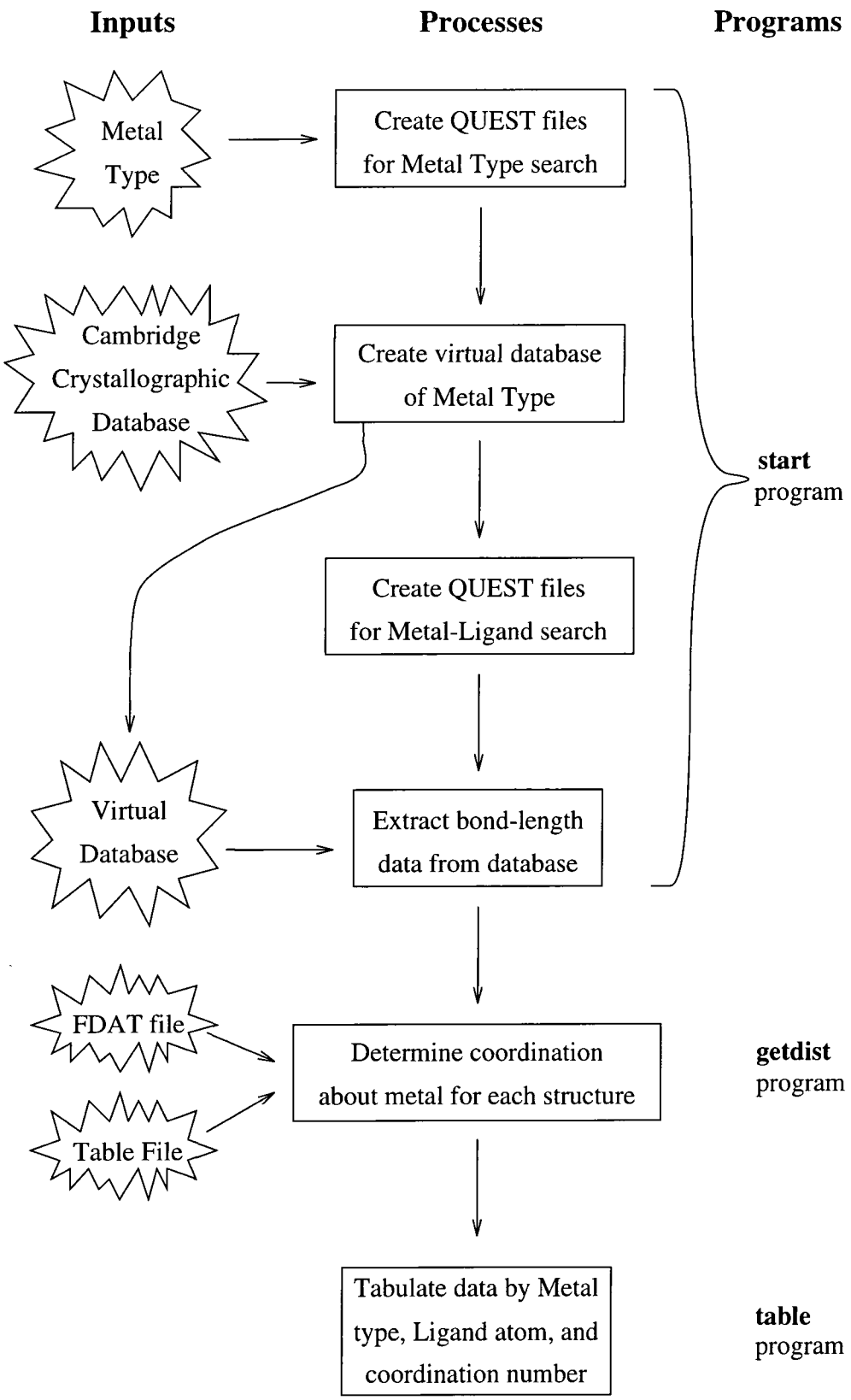


Figure 2.1: Flow diagram for data extraction method

entries, although not all of these are separate structures. A total of 2529 of these structures contain a metal from group IA or IIA bonded to another atom, and have crystallographic coordinates stored. Within these structures there are 15962 fragments that match the (Group IA,IIA Metal)-(Any Atom) pattern. This breaks down as given in Table 2.1.

Table 2.1: Breakdown of Group IA and IIA metal-containing structures in study

Element	Ba	Be	Ca	Cs	K	Li	Mg	Na	Rb	Sr
Entries	129	62	137	21	287	1019	404	400	27	62
Fragments	1152	286	887	153	2273	5839	2112	2667	197	494

The previous studies [2, 3] have needed to take care in the statistical survey, since the broader range of bond types in non metal–non metal interactions could lead to multiple peaks in the distributions. In this work, the nature of the metal atoms is such that all interactions could be considered to be single bonds, so the assumption of a unimodal distribution is probably more valid. Analysis of the data using the VISTA program (Figures 2.2, 2.3 and 2.4), reveals that the distribution of bond distances in most cases fits the assumed form. The data shown in Figures 2.2 - 2.4 can be loosely divided into two classes:

element	distance(Å)	element	distance(Å)	
Li	1.9	Be	1.6	
Na	2.4	Mg	2.0	Gaussian
		Ca	2.4	
K	2.8			
Rb	2.9	Sr	2.6	Non-Gaussian
Cs	3.0	Ba	2.8	

The long tails on the long bond-length side of the distributions for the heavier atoms possibly reflects more flexibility in the bonding electrons as a result of their increased atomic radius.

Table 2.2: Bond length statistics for lithium complexes

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Li-S	n	4	8	5	75	1	2	0	0	0
	μ	2.515	2.404	2.439	2.483	2.617	2.747	0.000	0.000	0.000
	σ	0.003	0.042	0.093	0.075	0.000	0.047	0.000	0.000	0.000
Li-O	n	39	105	394	1312	315	175	17	231	0
	μ	1.885	1.944	1.942	1.963	2.057	2.139	1.967	2.361	0.000
	σ	0.312	0.209	0.092	0.096	0.125	0.103	0.050	0.112	0.000
Li-N	n	12	159	406	964	165	85	15	1	4
	μ	2.068	2.035	2.052	2.104	2.105	2.180	2.122	2.357	2.385
	σ	0.079	0.096	0.078	0.075	0.099	0.090	0.085	0.000	0.033
Li-Mo	n	1	0	2	0	0	0	0	0	0
	μ	2.775	0.000	2.714	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.009	0.000	0.000	0.000	0.000	0.000	0.000
Li-Li	n	4	0	3	13	30	16	34	7	0
	μ	2.555	0.000	2.439	2.426	2.533	2.547	2.656	2.718	0.000
	σ	0.051	0.000	0.126	0.169	0.163	0.139	0.154	0.173	0.000
Li-Hg	n	2	0	0	0	0	0	0	0	0
	μ	2.573	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Li-C	n	11	40	79	189	104	71	59	25	6
	μ	2.259	2.223	2.308	2.260	2.315	2.313	2.299	2.368	2.347
	σ	0.079	0.072	0.478	0.269	0.122	0.131	0.072	0.108	0.034
Li-W	n	0	8	0	0	0	0	0	0	0
	μ	0.000	2.772	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.097	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Li-Pt	n	0	4	0	0	0	1	0	0	0
	μ	0.000	2.823	0.000	0.000	0.000	2.628	0.000	0.000	0.000
	σ	0.000	0.022	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 2.3: Bond length statistics for lithium complexes (continued)

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Li-P	n	0	10	47	108	12	2	6	0	0
	μ	0.000	2.481	2.527	2.605	2.624	2.651	2.842	0.000	0.000
	σ	0.000	0.045	0.063	0.079	0.050	0.006	0.230	0.000	0.000
Li-F	n	0	1	5	10	4	2	0	0	0
	μ	0.000	1.887	1.918	1.918	2.216	2.236	0.000	0.000	0.000
	σ	0.000	0.000	0.096	0.083	0.062	0.007	0.000	0.000	0.000
Li-Cl	n	0	7	9	135	11	4	0	0	0
	μ	0.000	2.345	2.391	2.374	2.517	2.553	0.000	0.000	0.000
	σ	0.000	0.030	0.048	0.070	0.117	0.021	0.000	0.000	0.000
Li-Br	n	0	1	4	46	5	0	0	0	0
	μ	0.000	2.517	2.614	2.530	2.726	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.110	0.073	0.273	0.000	0.000	0.000	0.000
Li-Bi	n	0	1	0	0	0	0	0	0	0
	μ	0.000	2.924	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Li-As	n	0	2	4	3	0	0	0	0	0
	μ	0.000	2.734	2.554	2.682	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.024	0.056	0.037	0.000	0.000	0.000	0.000	0.000
Li-Te	n	0	0	11	4	0	0	0	0	0
	μ	0.000	0.000	2.747	2.828	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.036	0.034	0.000	0.000	0.000	0.000	0.000
Li-Si	n	0	0	2	11	0	0	0	0	0
	μ	0.000	0.000	2.644	2.619	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.030	0.068	0.000	0.000	0.000	0.000	0.000
Li-Se	n	0	0	6	21	0	0	0	0	0
	μ	0.000	0.000	2.471	2.600	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.007	0.042	0.000	0.000	0.000	0.000	0.000

Table 2.4: Bond length statistics for lithium complexes (continued)

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Li-I	n	0	0	11	14	2	0	0	0	0
	μ	0.000	0.000	2.757	2.807	2.911	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.120	0.059	0.196	0.000	0.000	0.000	0.000
Li-H	n	0	0	5	12	6	7	4	0	0
	μ	0.000	0.000	1.820	1.892	1.919	2.017	1.972	0.000	0.000
	σ	0.000	0.000	0.077	0.098	0.246	0.064	0.037	0.000	0.000
Li-Fe	n	0	0	3	0	0	0	0	0	0
	μ	0.000	0.000	2.458	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.008	0.000	0.000	0.000	0.000	0.000	0.000
Li-B	n	0	0	3	3	12	4	9	0	0
	μ	0.000	0.000	2.395	2.327	2.342	2.199	2.447	0.000	0.000
	σ	0.000	0.000	0.053	0.103	0.047	0.034	0.105	0.000	0.000
Li-V	n	0	0	0	1	2	0	0	0	0
	μ	0.000	0.000	0.000	2.638	2.546	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.038	0.000	0.000	0.000	0.000
Li-Ta	n	0	0	0	1	0	0	0	0	0
	μ	0.000	0.000	0.000	2.631	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Li-Sn	n	0	0	0	2	1	0	0	0	0
	μ	0.000	0.000	0.000	2.872	3.012	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.011	0.000	0.000	0.000	0.000	0.000
Li-Pb	n	0	0	0	2	0	0	0	0	0
	μ	0.000	0.000	0.000	2.859	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.007	0.000	0.000	0.000	0.000	0.000
Li-Ni	n	0	0	0	2	1	0	0	0	0
	μ	0.000	0.000	0.000	2.714	2.389	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.123	0.000	0.000	0.000	0.000	0.000

Table 2.5: Bond length statistics for lithium complexes (continued)

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Li-Au	n	0	0	0	4	0	0	0	0	0
	μ	0.000	0.000	0.000	2.870	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.020	0.000	0.000	0.000	0.000	0.000
Li-Zn	n	0	0	0	0	2	0	0	0	0
	μ	0.000	0.000	0.000	0.000	2.453	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.007	0.000	0.000	0.000	0.000
Li-Y	n	0	0	0	0	2	0	0	0	0
	μ	0.000	0.000	0.000	0.000	3.026	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.018	0.000	0.000	0.000	0.000
Li-Sm	n	0	0	0	0	2	0	0	0	0
	μ	0.000	0.000	0.000	0.000	2.912	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.088	0.000	0.000	0.000	0.000
Li-Pd	n	0	0	0	0	3	0	0	0	0
	μ	0.000	0.000	0.000	0.000	2.611	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.116	0.000	0.000	0.000	0.000
Li-Cu	n	0	0	0	0	0	6	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	2.639	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.017	0.000	0.000	0.000
Li-Be	n	0	0	0	0	0	1	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	2.227	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Li-Al	n	0	0	0	0	0	2	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	2.675	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.034	0.000	0.000	0.000
Li-Ag	n	0	0	0	0	0	6	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	2.755	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.040	0.000	0.000	0.000

Table 2.6: Bond length statistics for lithium complexes (continued)

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Li-Na	n	0	0	0	0	0	0	3	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	2.583	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.038	0.000	0.000
Li-Ru	n	0	0	0	0	0	0	0	0	2
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.807
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.030

Table 2.7: Bond length statistics for beryllium complexes

		Coordination number about metal					
bond type		1	2	3	4	8	10
Be-N	n	2	3	7	58	0	0
	μ	1.783	1.668	1.648	1.747	0.000	0.000
	σ	0.002	0.084	0.081	0.058	0.000	0.000
Be-C	n	1	7	1	21	0	0
	μ	2.408	1.817	1.886	1.828	0.000	0.000
	σ	0.000	0.074	0.000	0.093	0.000	0.000
Be-S	n	0	2	2	0	0	0
	μ	0.000	2.098	1.989	0.000	0.000	0.000
	σ	0.000	0.009	0.002	0.000	0.000	0.000
Be-P	n	0	1	0	0	0	0
	μ	0.000	2.083	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000
Be-O	n	0	21	1	96	0	0
	μ	0.000	1.610	1.589	1.616	0.000	0.000
	σ	0.000	0.051	0.000	0.028	0.000	0.000
Be-H	n	0	1	0	8	4	4
	μ	0.000	1.228	0.000	1.484	1.528	1.722
	σ	0.000	0.000	0.000	0.070	0.049	0.039
Be-F	n	0	4	0	12	0	0
	μ	0.000	1.532	0.000	1.547	0.000	0.000
	σ	0.000	0.008	0.000	0.016	0.000	0.000
Be-Cl	n	0	7	2	2	0	0
	μ	0.000	1.904	1.977	1.926	0.000	0.000
	σ	0.000	0.047	0.008	0.061	0.000	0.000
Be-Zr	n	0	0	3	0	0	0
	μ	0.000	0.000	2.328	0.000	0.000	0.000
	σ	0.000	0.000	0.002	0.000	0.000	0.000
Be-B	n	0	0	2	0	4	6
	μ	0.000	0.000	2.050	0.000	1.975	2.072
	σ	0.000	0.000	0.005	0.000	0.041	0.014
Be-Li	n	0	0	0	1	0	0
	μ	0.000	0.000	0.000	2.227	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000
Be-Br	n	0	0	0	2	0	0
	μ	0.000	0.000	0.000	2.157	0.000	0.000
	σ	0.000	0.000	0.000	0.036	0.000	0.000

Table 2.8: Bond length statistics for sodium complexes

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Na-S	n	1	2	1	11	13	0	3	0	5
	μ	2.825	3.007	2.942	2.880	2.866	0.000	3.038	0.000	3.084
	σ	0.000	0.046	0.000	0.027	0.077	0.000	0.123	0.000	0.255
Na-O	n	6	33	146	272	382	485	384	284	2
	μ	2.340	2.322	2.342	2.388	2.411	2.447	2.481	2.522	2.425
	σ	0.033	0.095	0.100	0.163	0.139	0.118	0.118	0.109	0.181
Na-N	n	2	10	30	134	88	60	26	53	0
	μ	2.276	2.484	2.461	2.493	2.487	2.523	2.588	2.710	0.000
	σ	0.123	0.071	0.107	0.146	0.093	0.135	0.097	0.160	0.000
Na-C	n	2	8	2	13	9	6	18	15	0
	μ	2.554	2.555	2.494	2.601	2.670	2.700	2.745	2.834	0.000
	σ	0.075	0.010	0.007	0.131	0.069	0.068	0.067	0.114	0.000
Na-Ni	n	0	1	5	0	0	0	0	0	0
	μ	0.000	3.037	2.891	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.056	0.000	0.000	0.000	0.000	0.000	0.000
Na-Se	n	0	0	4	0	0	0	0	0	0
	μ	0.000	0.000	3.046	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.116	0.000	0.000	0.000	0.000	0.000	0.000
Na-I	n	0	0	1	0	7	2	0	0	0
	μ	0.000	0.000	3.164	0.000	3.103	3.070	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.081	0.018	0.000	0.000	0.000
Na-Fe	n	0	0	1	0	0	0	1	0	2
	μ	0.000	0.000	4.048	0.000	0.000	0.000	3.555	0.000	3.579
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.032
Na-F	n	0	0	3	2	0	3	26	8	0
	μ	0.000	0.000	2.339	2.369	0.000	2.287	2.369	2.469	0.000
	σ	0.000	0.000	0.033	0.013	0.000	0.109	0.104	0.085	0.000

Table 2.9: Bond length statistics for sodium complexes (continued)

		Coordination number about metal								
bond type		1	2	3	4	5	6	7	8	12
Na-Cl	n	0	0	2	12	9	26	6	0	0
	μ	0.000	0.000	2.838	2.781	2.822	2.854	2.947	0.000	0.000
	σ	0.000	0.000	0.002	0.129	0.071	0.067	0.062	0.000	0.000
Na-P	n	0	0	0	2	2	0	0	1	0
	μ	0.000	0.000	0.000	2.973	2.910	0.000	0.000	3.430	0.000
	σ	0.000	0.000	0.000	0.082	0.028	0.000	0.000	0.000	0.000
Na-As	n	0	0	0	2	0	0	0	0	0
	μ	0.000	0.000	0.000	2.950	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.013	0.000	0.000	0.000	0.000	0.000
Na-Na	n	0	0	0	0	3	1	2	1	3
	μ	0.000	0.000	0.000	0.000	2.984	2.741	3.118	3.277	4.052
	σ	0.000	0.000	0.000	0.000	0.066	0.000	0.170	0.000	0.164
Na-Li	n	0	0	0	0	2	1	0	0	0
	μ	0.000	0.000	0.000	0.000	2.609	2.529	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.000
Na-Br	n	0	0	0	0	0	4	0	1	0
	μ	0.000	0.000	0.000	0.000	0.000	2.960	0.000	2.820	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.133	0.000	0.000	0.000
Na-Au	n	0	0	0	0	0	6	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	3.376	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.071	0.000	0.000	0.000
Na-La	n	0	0	0	0	0	0	2	2	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	3.586	3.605	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.023	0.009	0.000
Na-B	n	0	0	0	0	0	0	1	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	2.882	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Na-H	n	0	0	0	0	0	0	0	3	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.336	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.055	0.000

Table 2.10: Bond length statistics for magnesium complexes

		Coordination number about metal						
bond type		1	2	3	4	5	6	7
Mg-O	n	6	27	287	170	105	409	52
	μ	2.082	2.059	2.074	2.046	2.082	2.093	2.174
	σ	0.019	0.064	0.047	0.074	0.082	0.067	0.078
Mg-N	n	2	22	89	167	92	58	12
	μ	2.156	2.089	2.145	2.135	2.160	2.213	2.158
	σ	0.037	0.087	0.090	0.076	0.112	0.102	0.127
Mg-Cl	n	1	0	11	21	9	117	5
	μ	2.326	0.000	2.434	2.361	2.461	2.507	2.399
	σ	0.000	0.000	0.089	0.052	0.121	0.044	0.045
Mg-C	n	5	16	21	153	57	40	11
	μ	2.191	2.205	2.196	2.255	2.223	2.312	2.388
	σ	0.097	0.102	0.072	0.242	0.078	0.143	0.084
Mg-Si	n	0	1	0	5	0	0	0
	μ	0.000	2.631	0.000	2.641	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.020	0.000	0.000	0.000
Mg-P	n	0	2	2	4	0	0	0
	μ	0.000	2.494	2.766	2.624	0.000	0.000	0.000
	σ	0.000	0.007	0.004	0.033	0.000	0.000	0.000
Mg-Mo	n	0	1	1	0	0	0	0
	μ	0.000	2.737	2.853	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Mg-Br	n	0	5	4	31	8	16	0
	μ	0.000	2.580	2.650	2.530	2.597	2.654	0.000
	σ	0.000	0.033	0.091	0.089	0.075	0.091	0.000
Mg-I	n	0	0	2	7	0	1	0
	μ	0.000	0.000	2.681	2.813	0.000	2.872	0.000
	σ	0.000	0.000	0.009	0.049	0.000	0.000	0.000

Table 2.11: Bond length statistics for magnesium complexes (continued)

		Coordination number about metal						
bond type		1	2	3	4	5	6	7
Mg-H	n	0	0	1	11	5	1	0
	μ	0.000	0.000	2.068	1.976	1.923	1.183	0.000
	σ	0.000	0.000	0.000	0.086	0.025	0.000	0.000
Mg-F	n	0	0	6	0	0	0	0
	μ	0.000	0.000	1.944	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.055	0.000	0.000	0.000	0.000
Mg-B	n	0	0	2	0	0	0	0
	μ	0.000	0.000	2.430	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.047	0.000	0.000	0.000	0.000
Mg-Ti	n	0	0	0	1	0	0	1
	μ	0.000	0.000	0.000	2.768	0.000	0.000	2.763
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Mg-Te	n	0	0	0	2	0	0	0
	μ	0.000	0.000	0.000	2.717	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.003	0.000	0.000	0.000
Mg-Se	n	0	0	0	2	0	0	0
	μ	0.000	0.000	0.000	2.491	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.009	0.000	0.000	0.000
Mg-S	n	0	0	0	4	0	0	0
	μ	0.000	0.000	0.000	2.401	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.026	0.000	0.000	0.000
Mg-Mg	n	0	0	0	1	2	0	0
	μ	0.000	0.000	0.000	2.966	2.822	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.019	0.000	0.000
Mg-Co	n	0	0	0	2	0	0	0
	μ	0.000	0.000	0.000	2.526	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.045	0.000	0.000	0.000

Table 2.12: Bond length statistics for magnesium complexes (continued)

		Coordination number about metal						
bond type		1	2	3	4	5	6	7
Mg-As	n	0	0	0	2	0	0	0
	μ	0.000	0.000	0.000	2.594	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.004	0.000	0.000	0.000
Mg-Al	n	0	0	0	1	0	0	0
	μ	0.000	0.000	0.000	2.839	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Mg-Ni	n	0	0	0	0	2	0	0
	μ	0.000	0.000	0.000	0.000	2.615	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Mg-Z	n	0	0	0	0	0	2	0
	μ	0.000	0.000	0.000	0.000	0.000	2.598	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.013	0.000
Mg-Ge	n	0	0	0	0	0	4	0
	μ	0.000	0.000	0.000	0.000	0.000	2.718	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.006	0.000
Mg-Cu	n	0	0	0	0	0	0	3
	μ	0.000	0.000	0.000	0.000	0.000	0.000	2.754
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.006

Table 2.13: Bond length statistics for potassium complexes

		Coordination number about metal									
bond type		1	2	3	4	5	6	7	8	9	10
K-O	n	3	28	64	125	64	415	321	702	59	130
	μ	2.639	2.766	2.758	2.780	2.846	2.809	2.806	2.829	2.843	2.919
	σ	0.063	0.089	0.108	0.096	0.132	0.084	0.090	0.072	0.166	0.104
K-N	n	1	12	9	24	15	28	35	171	2	0
	μ	2.816	2.816	2.723	2.921	2.917	2.769	2.909	2.981	2.918	0.000
	σ	0.000	0.076	0.115	0.124	0.164	0.160	0.179	0.070	0.007	0.000
K-C	n	0	4	2	2	2	0	1	2	0	0
	μ	0.000	2.999	2.875	2.950	3.360	0.000	2.746	3.154	0.000	0.000
	σ	0.000	0.101	0.049	0.019	0.107	0.000	0.000	0.109	0.000	0.000
K-Se	n	0	0	0	4	0	0	0	0	0	0
	μ	0.000	0.000	0.000	3.397	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.022	0.000	0.000	0.000	0.000	0.000	0.000
K-S	n	0	0	0	3	0	1	5	0	0	0
	μ	0.000	0.000	0.000	3.166	0.000	3.246	3.192	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.004	0.000	0.000	0.084	0.000	0.000	0.000
K-P	n	0	0	0	1	0	0	0	0	0	0
	μ	0.000	0.000	0.000	3.230	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
K-F	n	0	0	0	1	0	0	0	12	5	0
	μ	0.000	0.000	0.000	2.644	0.000	0.000	0.000	2.775	2.936	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.178	0.218	0.000
K-Cl	n	0	0	0	4	4	0	0	0	0	0
	μ	0.000	0.000	0.000	3.091	3.117	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.013	0.038	0.000	0.000	0.000	0.000	0.000
K-I	n	0	0	0	0	0	0	2	1	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	3.491	3.448	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.023	0.000	0.000	0.000
K-H	n	0	0	0	0	0	0	0	0	6	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.748	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.057	0.000

Table 2.14: Bond length statistics for calcium complexes

		Coordination number about metal											
bond type		1	2	3	4	5	6	7	8	9	10	11	12
Ca-N	n	1	2	30	12	13	21	16	43	0	4	0	0
	μ	2.454	2.516	2.407	2.526	2.447	2.483	2.516	2.519	0.000	2.471	0.000	0.000
	σ	0.000	0.245	0.083	0.088	0.060	0.115	0.110	0.074	0.000	0.027	0.000	0.000
Ca-O	n	0	12	56	45	45	125	89	201	27	14	1	4
	μ	0.000	2.345	2.336	2.431	2.429	2.344	2.408	2.464	2.493	2.530	2.437	2.339
	σ	0.000	0.082	0.052	0.101	0.139	0.088	0.076	0.078	0.073	0.114	0.000	0.005
Ca-C	n	0	10	7	14	2	6	0	4	0	2	10	20
	μ	0.000	2.622	2.650	2.766	2.716	4.815	0.000	2.671	0.000	2.798	2.759	2.858
	σ	0.000	0.056	0.018	0.157	0.006	2.819	0.000	0.091	0.000	0.097	0.021	0.173
Ca-Te	n	0	0	2	0	0	0	0	0	0	0	0	0
	μ	0.000	0.000	3.195	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ca-Sn	n	0	0	1	0	0	0	0	0	0	0	0	0
	μ	0.000	0.000	3.272	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ca-P	n	0	0	2	5	0	0	0	0	0	0	0	0
	μ	0.000	0.000	2.884	2.915	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.003	0.040	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ca-I	n	0	0	1	0	4	1	0	0	0	0	0	0
	μ	0.000	0.000	3.106	0.000	3.184	3.109	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.059	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ca-Cl	n	0	0	6	0	1	6	0	0	0	0	0	0
	μ	0.000	0.000	2.758	0.000	2.579	2.798	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.052	0.000	0.000	0.021	0.000	0.000	0.000	0.000	0.000	0.000

Table 2.15: Bond length statistics for calcium complexes (continued)

		Coordination number about metal											
bond type		1	2	3	4	5	6	7	8	9	10	11	12
Ca-S	n	0	0	0	0	0	1	0	0	0	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	2.941	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ca-As	n	0	0	0	0	0	2	0	0	0	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	2.994	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.023	0.000	0.000	0.000	0.000	0.000	0.000
Ca-F	n	0	0	0	0	0	0	0	8	0	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.393	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.147	0.000	0.000	0.000	0.000
Ca-H	n	0	0	0	0	0	0	0	0	0	6	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.458	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.069	0.000	0.000
Ca-B	n	0	0	0	0	0	0	0	0	0	4	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2.823	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.107	0.000	0.000

Table 2.16: Bond length statistics for rubidium complexes

		Coordination number about metal								
bond type		2	3	4	5	6	7	8	9	10
Rb-Si	n	4	0	0	0	0	0	0	0	0
	μ	3.566	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.040	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rb-O	n	0	5	15	0	18	31	38	23	20
	μ	0.000	2.999	2.941	0.000	2.980	2.942	2.914	2.984	3.035
	σ	0.000	0.047	0.069	0.000	0.068	0.101	0.054	0.065	0.114
Rb-N	n	0	1	5	3	6	11	10	4	0
	μ	0.000	2.946	2.980	3.111	2.984	3.048	3.018	3.137	0.000
	σ	0.000	0.000	0.033	0.016	0.046	0.101	0.032	0.005	0.000
Rb-C	n	0	0	0	2	0	0	0	0	0
	μ	0.000	0.000	0.000	3.423	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.062	0.000	0.000	0.000	0.000	0.000

Table 2.17: Bond length statistics for strontium complexes

		Coordination number about metal									
bond type		1	2	3	4	5	6	7	8	9	10
Sr-O	n	2	2	11	17	25	57	28	104	77	70
	μ	2.499	2.516	2.531	2.567	2.622	2.515	2.558	2.610	2.670	2.701
	σ	0.024	0.027	0.082	0.038	0.112	0.070	0.103	0.082	0.097	0.061
Sr-N	n	0	3	12	12	0	2	5	22	13	0
	μ	0.000	2.663	2.583	2.718	0.000	2.596	2.701	2.695	2.715	0.000
	σ	0.000	0.095	0.097	0.074	0.000	0.015	0.105	0.101	0.069	0.000
Sr-C	n	0	1	0	2	0	0	0	8	0	0
	μ	0.000	2.834	0.000	2.854	0.000	0.000	0.000	2.815	0.000	0.000
	σ	0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.084	0.000	0.000
Sr-Se	n	0	0	2	0	0	0	0	0	0	0
	μ	0.000	0.000	3.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.062	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Sr-S	n	0	0	1	0	0	0	0	0	0	0
	μ	0.000	0.000	2.951	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Sr-I	n	0	0	1	1	0	0	0	0	0	0
	μ	0.000	0.000	3.322	3.230	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Sr-P	n	0	0	0	4	0	5	0	0	0	0
	μ	0.000	0.000	0.000	3.053	0.000	3.115	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.017	0.000	0.082	0.000	0.000	0.000	0.000
Sr-As	n	0	0	0	0	0	2	0	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	3.124	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.023	0.000	0.000	0.000	0.000
Sr-Sr	n	0	0	0	0	0	0	1	1	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	3.815	3.820	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Sr-Ba	n	0	0	0	0	0	0	1	1	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	3.985	3.988	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 2.18: Bond length statistics for caesium complexes

		Coordination number about metal							
bond type		1	2	4	6	7	8	10	13
Cs-N	n	1	0	3	0	1	2	0	0
	μ	2.915	0.000	3.045	0.000	3.193	3.070	0.000	0.000
	σ	0.000	0.000	0.020	0.000	0.000	0.012	0.000	0.000
Cs-Si	n	0	4	0	0	0	0	0	0
	μ	0.000	3.802	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.031	0.000	0.000	0.000	0.000	0.000	0.000
Cs-O	n	0	0	25	18	20	22	30	16
	μ	0.000	0.000	3.101	3.139	3.097	3.120	3.226	3.321
	σ	0.000	0.000	0.117	0.085	0.115	0.187	0.110	0.133
Cs-C	n	0	0	0	0	0	0	0	10
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.536
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.098

2.4 Conclusions

The primary limitation of this method of extracting data from the CSDS database is that it relies on the metal–ligand interaction being flagged as a bond within the database. There is known to be some inconsistency in this respect, especially where structures reported before the computerisation of the database are concerned. It is quite reasonable that an interaction considered to be ionic has not been recorded in the database as a bond, and many of the metals featured in this study are commonly encountered as ions. To establish the extent of this problem, a small additional study was made to discover the scale of this problem. The metal atom chosen was caesium, since it occurs least in the main study, and is commonly found as an ion in structural models. The CSDS analysis program, VISTA, was used to plot histograms of the distribution of Cs-ligand distances including both bonded and non-bonded contacts. The non-bonded contacts were found using a feature of the QUEST program, where a search can be made using the van der Waals radius of the atoms concerned as a limiting radius for a contact. This means that within a small tolerance, atoms which are closer than the sum of their van der Waals radii are considered to be in contact.

In the histogram of the distances, Figure 2.5 there is a fairly clear division, splitting the distribution into two approximately normal distributions superimposed. One is centred around 3.18Å, probably corresponding to truly bonded interactions, while the other is centred around 4.04Å and represents

the non-bonded interactions. However, when the bonded interactions are analysed separately (Figure 2.7) the height of the peak, although centred in much the same place as the left-hand peak in the full analysis, is much less. It seems that the contacts recorded in the database as bonded form a very small part of the number of contacts where the distance suggests a bond. For comparison, the non-bonded contacts, Figure 2.6 also shows this double-Gaussian-type distribution, with the two peaks much the same place as in the full analysis, so a large number of cases where caesium is probably bonded to another atom are not recorded as bonds. This is likely to be the case for the other Group IA and IIA metals, which share the ambiguity of whether a bond is actually present or not.

2.5 Future Work

The conclusions above indicate that the current analysis of the Cambridge Crystallographic Database is non-exhaustive since it relied upon the flagging in the database of metal-ligand bonds. A future analyis based upon computed interatomic distances and a rule-based division into bonded and non-bonded interactions would be more thorough and authoritative. It would also increase the number of data points in the analyses, and thus make them more reliable.

A visual evaluation of the tables 2.2 - 2.19 indicates that binning all of the contacts to a metal only as a function of the metal atom (i.e. the histograms

in Figures 2.2 - 2.4) conceals much fine details in the data. Referring, for example, to table 2.2, it is clear from row one that there is (not unexpectedly) a link between coordination number and bond length. In the same table column 4 shows a variation of bond length with ligand atom. A future appraisal of the database might seek to perform a 3-way analysis, giving most-probable values of metal-ligand distances as a functon of metal atom, ligand atom and metal coordination number. Such an analysis could be useful in devising useful new ligands, and as benchmarks against which new structures could be judged.

An additional conclusion to this work is that the Cambridge Crystallographic Database would be more useful if the compilers of the database were able to update retrospectively the information stored, flagging as bonds all metal–ligand interactions falling within a defined set of criteria. These criteria may, of course, be decided on with reference to a study such as the present work. The Cambridge Crystallographic Data Centre have already shown an interest in the results of this work and the relative ease with which it could be repeated with an updated version of the database distribution. With such an update, where all interactions conforming to a well defined set of rules were designated as bonds, a study such as described here could be used to refresh dynamically the crystallographic community's idea of metal–ligand bond distances. It would also be relatively easy to make such an automated search for specific atom types involved in bonding. A crystallographic refinement package might then be able to make more accurate assessments of

bond distances found in the structural mode, rather than rely on incomplete data hard-wired into it.

Table 2.19: Bond length statistics for barium complexes

		Coordination number about metal											
bond type		1	2	3	4	5	6	7	8	9	10	11	14
Ba-O	n	1	9	3	47	49	82	81	138	186	189	66	10
	μ	2.719	2.704	2.543	2.696	2.744	2.682	2.725	2.790	2.822	2.840	2.885	2.873
	σ	0.000	0.147	0.032	0.121	0.131	0.109	0.111	0.111	0.099	0.079	0.108	0.141
Ba-N	n	1	0	5	20	16	31	20	53	49	32	8	0
	μ	2.761	0.000	2.805	2.861	2.952	2.915	2.912	2.925	2.933	2.955	3.066	0.000
	σ	0.000	0.000	0.123	0.139	0.036	0.085	0.084	0.084	0.083	0.050	0.038	0.000
Ba-C	n	0	13	4	0	0	2	0	0	0	0	0	0
	μ	0.000	2.980	2.936	0.000	0.000	3.112	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.056	0.016	0.000	0.000	0.041	0.000	0.000	0.000	0.000	0.000	0.000
Ba-Te	n	0	0	0	1	0	0	0	0	0	0	0	0
	μ	0.000	0.000	0.000	3.382	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ba-P	n	0	0	0	0	0	2	0	0	0	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	3.174	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.016	0.000	0.000	0.000	0.000	0.000	0.000
Ba-I	n	0	0	0	0	0	3	0	0	0	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	3.478	0.000	0.000	0.000	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.080	0.000	0.000	0.000	0.000	0.000	0.000
Ba-S	n	0	0	0	0	0	0	4	0	2	0	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	3.309	0.000	3.261	0.000	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.022	0.000	0.029	0.000	0.000	0.000
Ba-Ba	n	0	0	0	0	0	0	0	1	4	3	1	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.942	4.010	4.032	4.036	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.069	0.062	0.000	0.000
Ba-Cl	n	0	0	0	0	0	0	0	0	2	6	0	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.312	3.197	0.000	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.013	0.049	0.000	0.000
Ba-Sr	n	0	0	0	0	0	0	0	0	0	0	2	0
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.987	0.000
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000
Ba-Cu	n	0	0	0	0	0	0	0	0	0	0	0	4
	μ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.599
	σ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.072

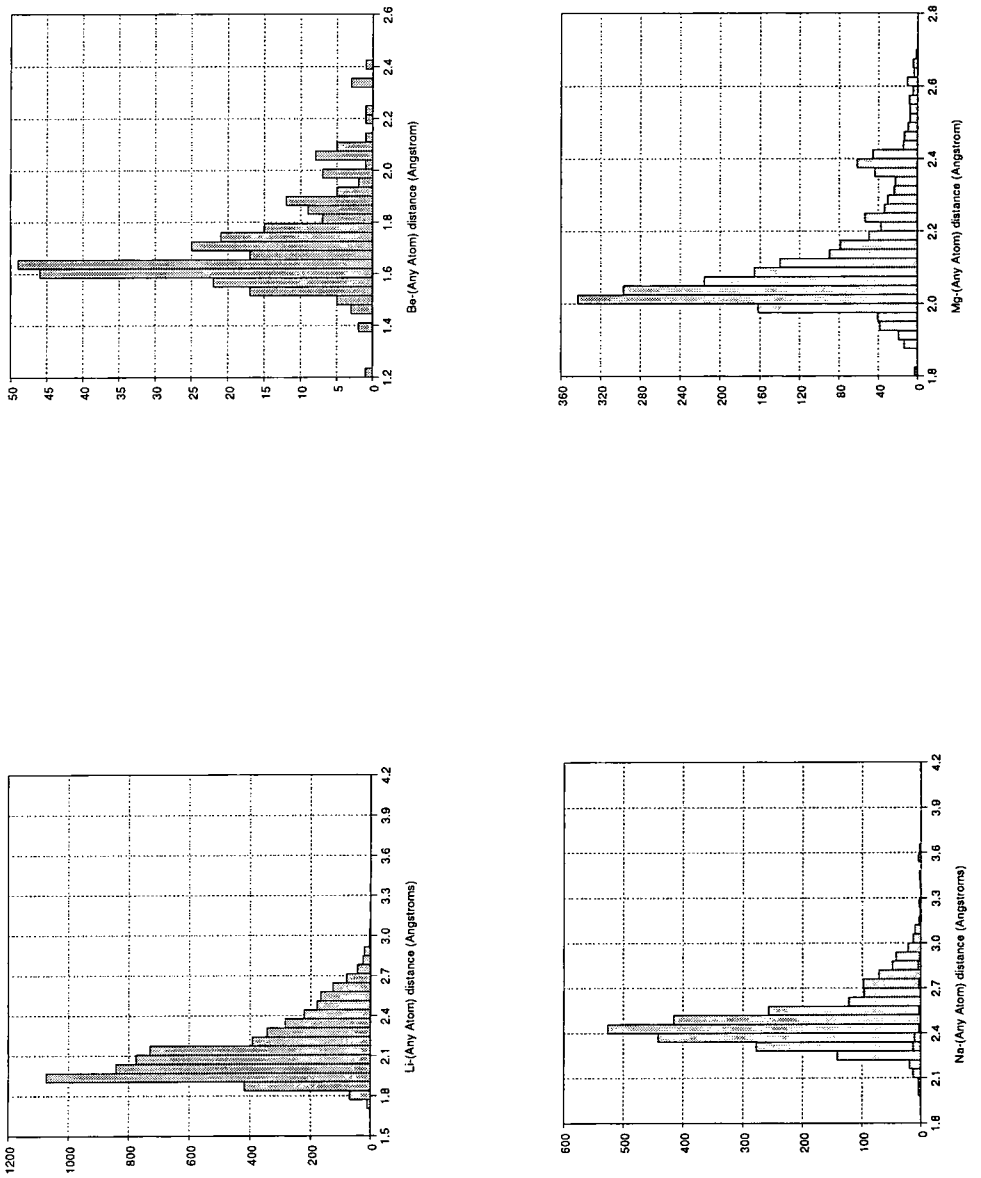


Figure 2.2: Histograms of Metal–(Any Atom) bond lengths for Li, Be, Na, Mg.

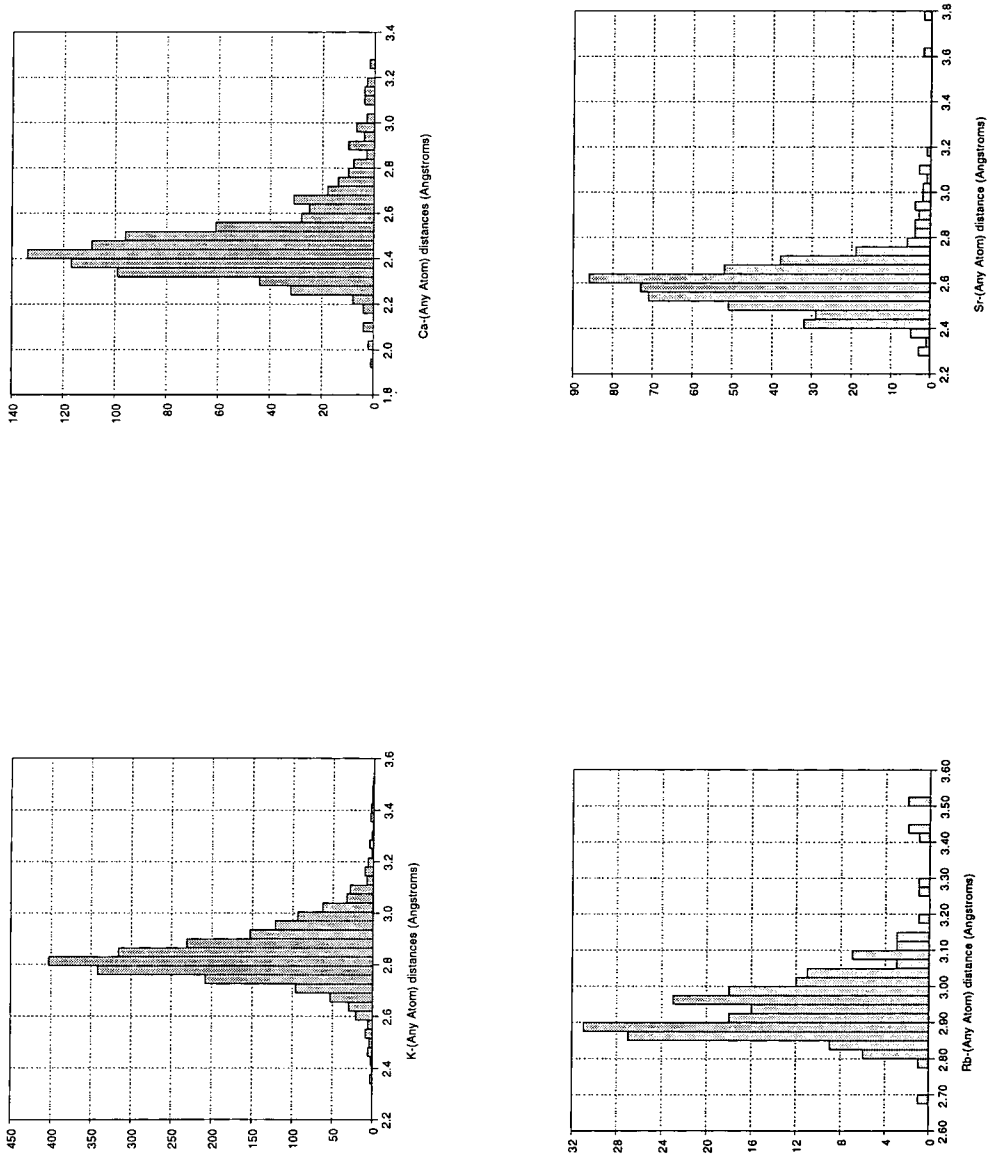


Figure 2.3: Histograms of Metal–(Any Atom) bond lengths for K, Ca, Rb, Sr.

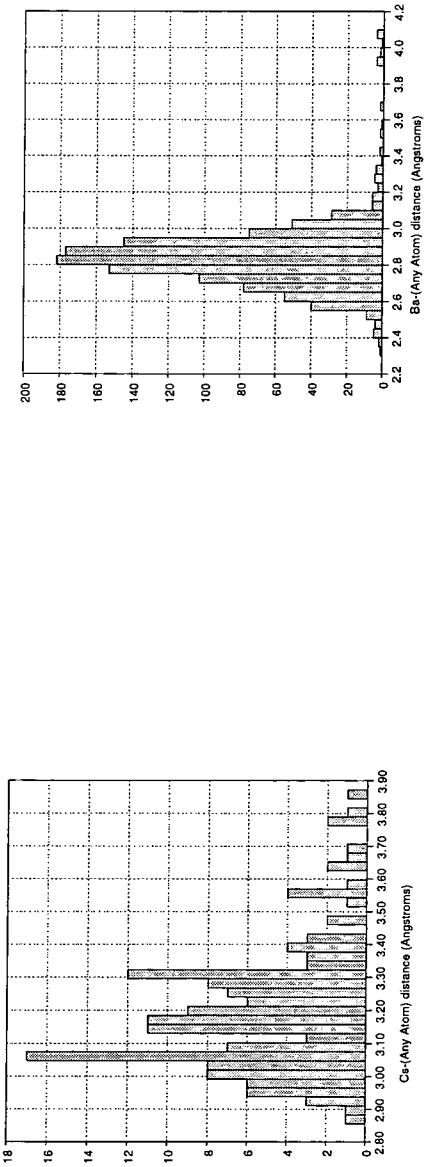


Figure 2.4: Histograms of Metal–(Any Atom) bond lengths for Cs, Ba.

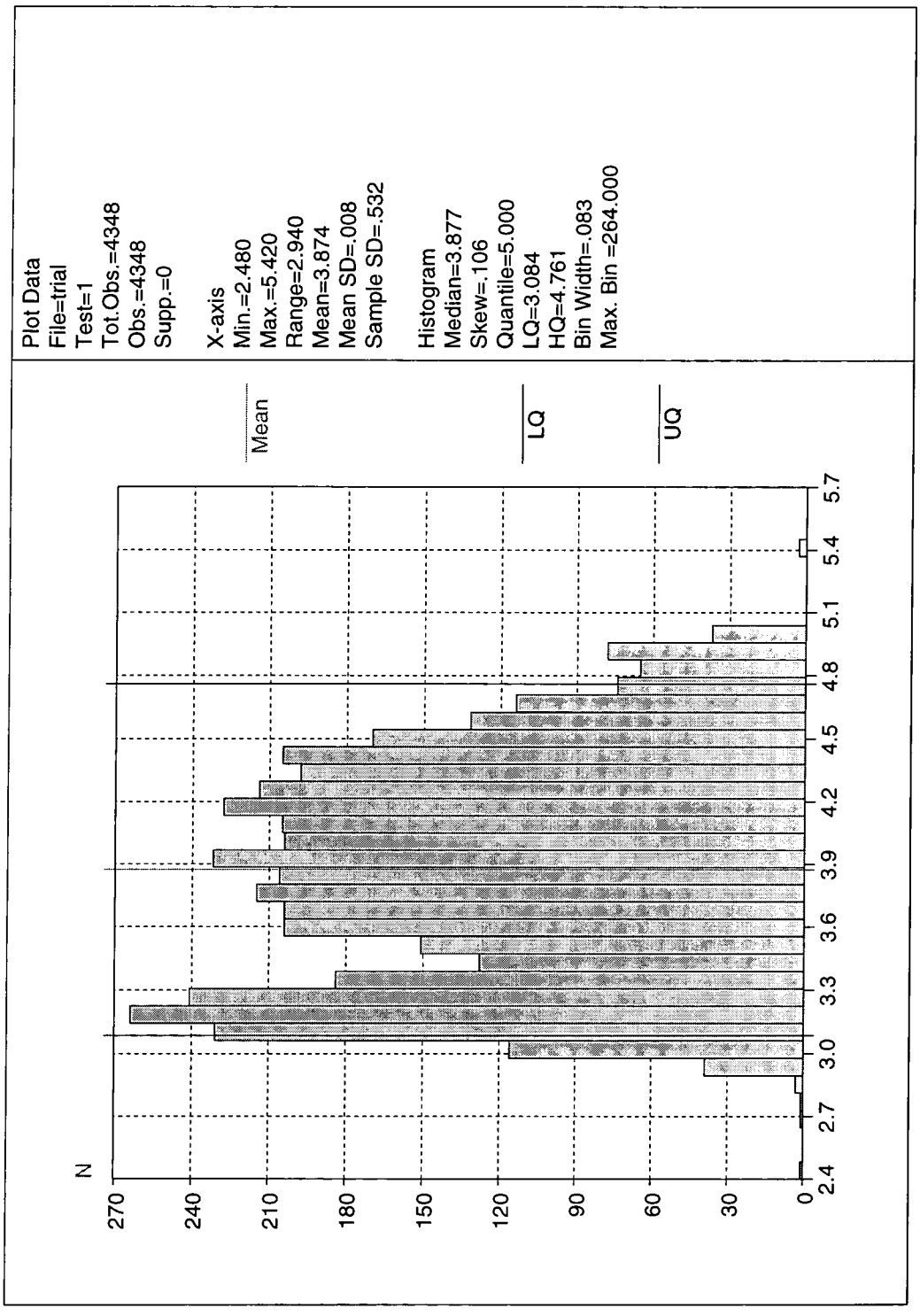


Figure 2.5: Histogram of Cs-Any atom distances including non-bonded contacts

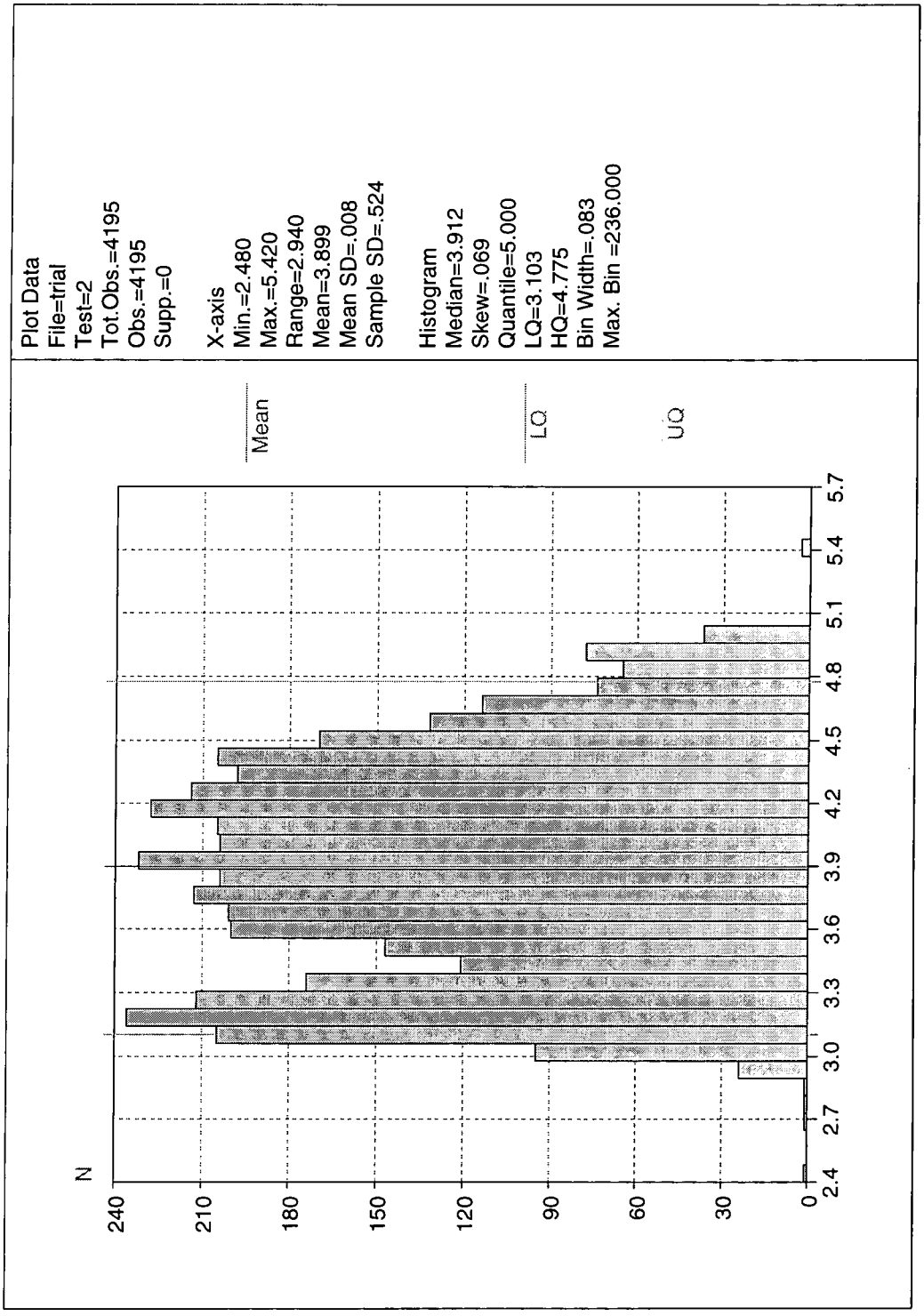


Figure 2.6: Histogram of Cs-Any atom non-bonded contact distances

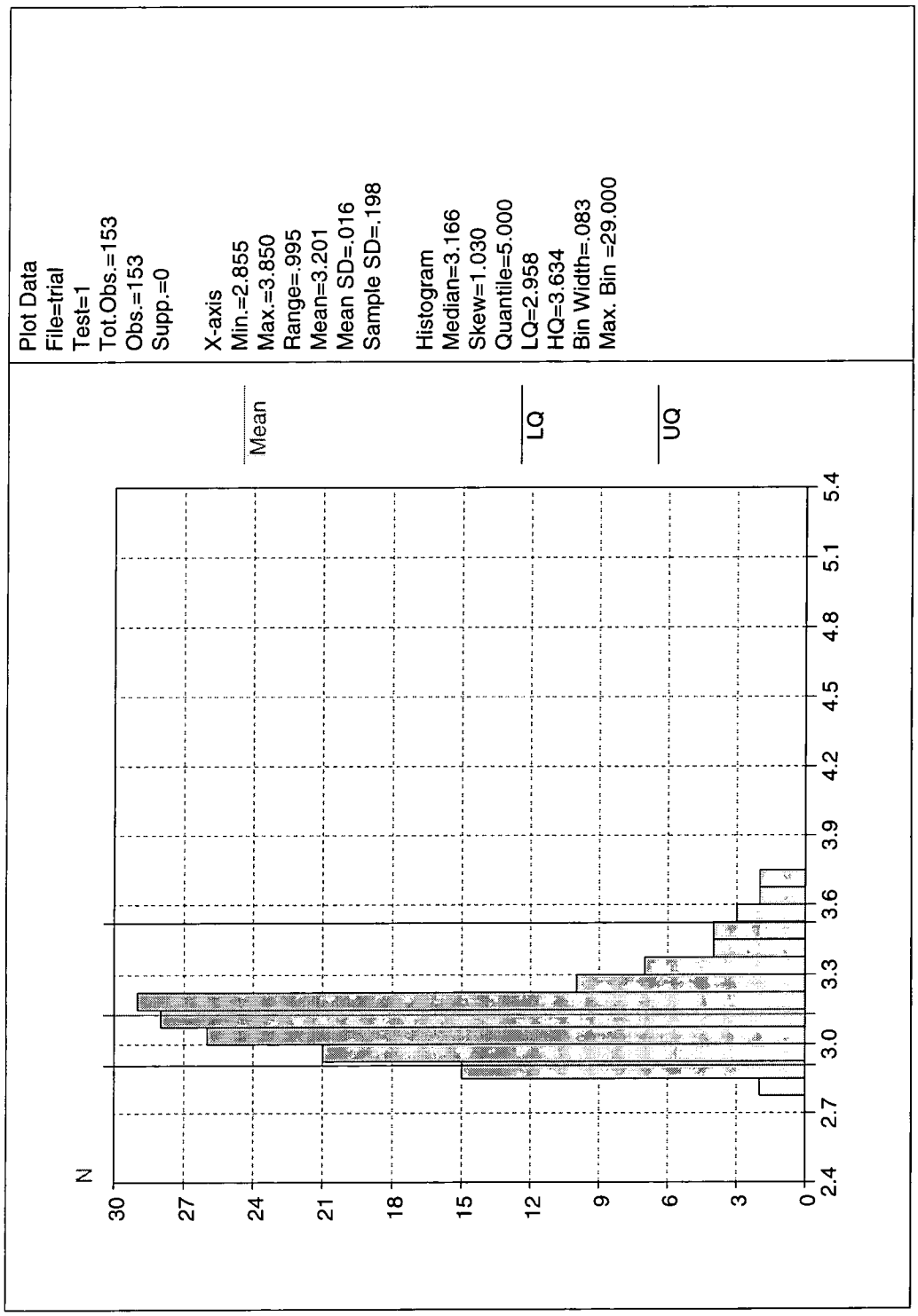


Figure 2.7: Histogram of Cs-Any atom bond distances

Chapter 3

Symmetry Determination

3.1 Introduction

The use of space groups in crystallography provides a shorthand notation to describe the symmetry of the repeating unit in the crystalline lattice. Space groups allow the structure to be described by the minimum amount of information regarding atomic positions, and also simplify the task of calculating Fourier maps, Least Squares matrices and other mathematical tools used in crystallography. This minimum descriptor is known as the asymmetric unit, since it contains no symmetry-equivalent atom positions. In minerals this often consists of only a few atomic sites, since mineral crystals usually crystallize with high symmetry structures, while organic compounds tend to have complete molecules in the asymmetric unit. As examples, consider sodium chloride, space group $Fm\bar{3}m$, which has very high symmetry and

thus only two atom positions form the asymmetric unit. Alternatively, crystalline potassium penicillin V has four molecules, each of 25 atoms, in the asymmetric unit, but the space group is $P1$ so there is no further symmetry to be considered.

When the need for modelling a system or even visualising a structure arises, the symmetry of the system must be expanded to place all of the atoms comprising the unit cell into their correct positions - few people can picture the lattice structure from the asymmetric unit alone. In theory there is enough information in the symbol given for a space group to determine all of the symmetry operations required to expand an asymmetric unit into a full unit cell. Unfortunately, the symbols most commonly used for space groups are the International Short Symbols, from which some of the information is missing. This symbol usually includes, or implies, enough about the symmetry elements present, but often gives no clue about their relative positions within the unit cell.

3.2 Development

Two very different strategies were considered for the task of decoding space group symbols. The first was to construct a database of symbols and their corresponding symmetry operations. This could be searched by the program to match the symbol input with one in the database, and the resulting operators printed. The alternative was to determine some systematic approach

to decoding the symbol, and using group theory combined with matrix arithmetic to produce the complete set of operations.

The database method has two main disadvantages. Firstly the database would be fairly large if it were to include all of the operators for all of the possible space groups. Some of the higher-symmetry cubic space groups require 192 symmetry operations to describe them fully, and the database would need to store 12 matrix elements for each operator. Using a binary file for the database might reduce the size, but would cause problems when trying to port the program from one computer system to another. The second disadvantage is that it is difficult to ensure completeness of this kind of database. Not only are there 230 space groups for three-dimensional lattices, but many of these have alternative origin choices and orientations which are sometimes used by crystallographers when reporting the structure. A program was tried initially using this database method, with information about the space groups gathered from International Tables Volume A [6], but the problems of database completeness and portability suggested that a more elegant solution was preferable.

As part of the present work, a system has been devised to allow the interpretation of simple space group symbols to yield the symmetry operations. This system relies on the assumption that a truly systematic approach will work, and then corrects any problems using additional information gleaned from *IT*. The system has been comprehensively tested against all of the space groups reported in the Cambridge Crystallographic Database, including those

with non-standard settings. Some theoretically possible space groups have not been observed experimentally, and the results for these have been compared with *IT* to ensure that the system works correctly.

The system works as follows:

1. The symbol is separated into its component parts, describing the lattice centring and symmetry operations present on the possible axes.

2. The crystal class is determined to help decide which directions are referred to by the symbols.

3. Matrices corresponding to the operations explicitly given in the symbol are extracted from a look-up table.

4. A minimum set of generator matrices is taken from the possible operations, and these are adjusted to give the correct setting as per *IT*.

5. The generators are multiplied together exhaustively to produce all of the symmetry operations.

6. If the lattice centring implies further symmetry then these operations are included.

7. Each atom of the asymmetric unit is multiplied by each matrix, and checked for duplication, to get the full unit cell.

3.3 Input

The input string is required to be in an unambiguous form. Some ambiguity may arise in certain crystal classes, and the program attempts to resolve this. However, it is not easy to differentiate between, for example, $P12_11$ and $P121$ when the character set of the input doesn't allow for subscripts. Screw axes are input with the subscripts either in brackets or preceded by an underscore. Thus a $P2_1$ would be entered as "P 2(1)" or "P 2_1". Rotation-inversion $\bar{n}$ operations are input as "-n". The program is faithful to the 1983 edition of International Tables Volume A, and so in the case of some Cubic space groups the 3-fold axis is written as a "$\bar{3}$", whereas the older editions used a "3" [6, p. 16]. If the space group is Monoclinic, then the unique axis will be assumed to be the b-axis unless explicitly padded with 1's; for example "P 2" would be taken as the b-axis unique, while "P 1 1 2" would force the c-axis unique orientation.

As well as the space group symbol, the code requires an array of atom positions in fractional coordinates, which will be expanded and returned as a full unit cell. Since the atom positions are checked for duplication, the input set may be more than the actual asymmetric unit. However, the coordinates will be adjusted to fit within the unit cell dimensions.

3.4 Technical Details

3.4.1 Space group symbols

Three-dimensional space groups have a centring symbol; a capital letter from the set P, A, B, C, I, F, R; and a set of characters similar to a point-group symbol. These correspond to symmetry operations referring to a symmetry direction or set of equivalent symmetry directions (Table 3.4.1). The symmetry direction to which each symbol corresponds depends upon the lattice symmetry and naming conventions.

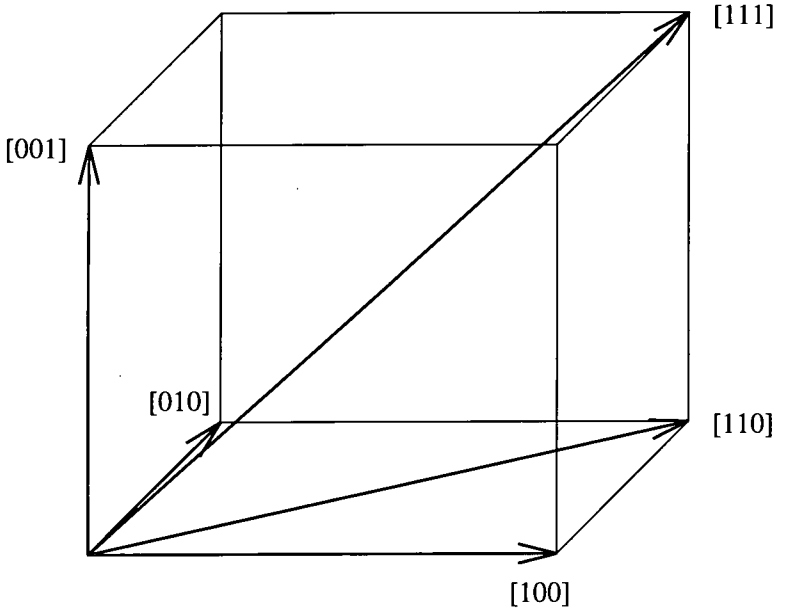


Figure 3.1: Symmetry directions in orthogonal space groups

A symbol in any of these positions may take one of the following forms:

1. A number **n** is an n-fold rotation axis parallel to the symmetry direc-

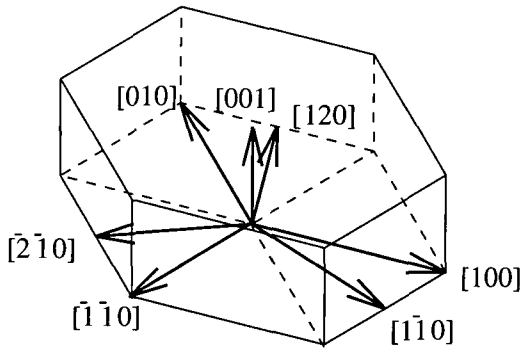


Figure 3.2: Symmetry directions in hexagonal space groups

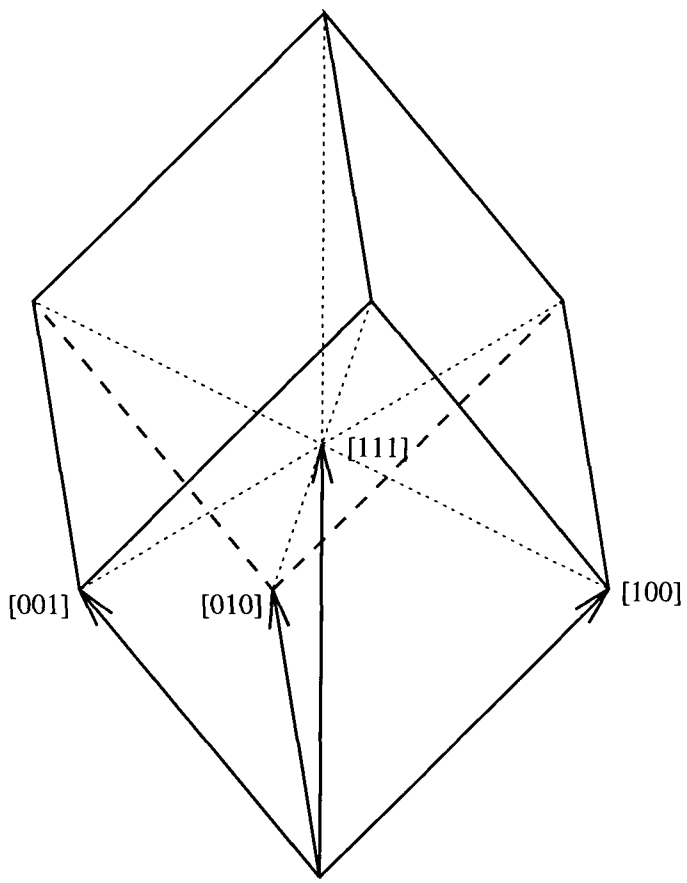


Figure 3.3: Symmetry directions in rhombohedral space groups

tion; for example "2".

2. A subscripted number n_j is a screw axis; an n-fold rotation combined with a translation of $\frac{j}{n}$ parallel to the symmetry direction; for example "3_2".

3. A barred number $\bar{n}$ is a rotation-inversion axis; an n-fold rotation combined with an inversion centre. Note that a "$\bar{1}$" represents an inversion centre.

4. A letter **p** is a mirror plane or glide plane perpendicular to the symmetry direction. The letter represents the direction of translation (if any) following the reflection operation. The symbol m is a simple reflection, a, b and c have translation of $\frac{1}{2}$ in the directions [100], [010] and [001] respectively, n has translation $\frac{1}{2}$ in both directions perpendicular to the symmetry direction (e.g. [110] for an n glide on the [001] symmetry direction), and d is a complicated double glide found only in I-centred and F-centred lattices.

For example, the Orthorhombic space group *Pbca* contains a P for a primitive lattice, and has glide planes perpendicular to the [100], [010] and [001] directions respectively. For a glide plane, the translation vector lies in the plane, perpendicular to the symmetry direction. A c-glide cannot exist for the [001] symmetry direction, and so on.

For a more complicated example, the Cubic space group Fm$\bar{3}$m has a face-centred lattice, with mirror planes perpendicular to the a, b and c cell

axes, a 3-fold rotation-inversion axis along the body diagonals, and a mirror plane perpendicular to the face diagonals.

3.4.2 Crystal Classes

Determination of the crystal class of the space group is necessary to correctly identify which symmetry operations are present in the group. The components of the International Symbol can only be interpreted with reference to the crystal class, but the crystal class is contained within the components of the space group symbol. It is possible, however, to perform a sequence of tests to decide on the crystal class by a process of elimination.

1. A check is made that the class is not triclinic. A flag will have been set during the symbol decoding if anything other than a "1" or "$\bar{1}$" was found. If the flag is not set, the class is Triclinic.

2. Look at the centring symbol; if it is "R" then the class is Rhombohedral.

3. Check for a six-fold axis among the symbols, which would imply a Hexagonal class.

4. If a four-fold axis is present the class may be Cubic or Tetragonal. If a three-fold axis is also present then the class is Cubic, else it is Tetragonal.

5. Another possible form of space group in the Cubic crystal class has a 3 or -3 as the second part of the symbol, and no "1" symbols.

6. If a space group with a three-fold axis has not been eliminated by the above, then it is Trigonal.

7. If all of the parts of the symbol represent two-fold axes or symmetry planes then the class must be Orthorhombic.

8. Anything else is taken to be Monoclinic.

If no sense can be made of the input at this stage, then an error message is given.

3.4.3 Matrices

In the set of crystallographic space groups, there are 26 symmetry operations and 10 symmetry directions in which they can act. Each combination of symmetry operation and symmetry direction has a can be described by a unique matrix, although not all combinations are possible.

The 26 operations include the identity operation, which is implied in the absence of any other. Table 3.4.3 indicates which symmetry directions an operation can be applied to.

Each of the X's in Table 3.4.3 indicates the presence of a non-zero matrix in the look-up table used by the program. The space group symbol is dissembled to yield the operator symbols, to which the matrices can be assigned. Some or all of these matrices will be used as generators for the space group symmetry operations.

Table 3.1: Symmetry directions in terms of the lattice vectors

Lattice	first symbol	second symbol	third symbol
Triclinic	No symmetry directions; only point operations		
Monoclinic	Only one symbol; Direction is along unique axis [100][010]or[001]		
Orthorhombic	[100]	[010]	[001]
Tetragonal	[001]	[100] & [010]	$[1\bar{1}0]$ & [110]
Trigonal	[001]	[100], [010],$[\bar{1}\bar{1}0]$	$[1\bar{1}0]$, [120], $[\bar{2}\bar{1}0]$
Hexagonal	[001]	[100], [010],$[\bar{1}\bar{1}0]$	$[1\bar{1}0]$, [120], $[\bar{2}\bar{1}0]$
Rhombohedral	$[11\bar{1}]$	$[1\bar{1}0]$, $[01\bar{1}]$, $[\bar{1}01]$	
Cubic	[100], [010], [001]	$[111][1\bar{1}\bar{1}]$ $[\bar{1}1\bar{1}][\bar{1}\bar{1}1]$	$[1\bar{1}0][110][01\bar{1}]$ $[011][\bar{1}01][101]$

Table 3.2: The ten symmetry directions

Direction	**Miller Index**	**Description**
1	[100]	Along a cell axis
2	[010]	Along b cell axis
3	[001]	Along c cell axis
4	$[1\bar{1}0]$	Along face diagonal (Cubic, Tetragonal)
5	$[1\bar{1}0]$	Along face diagonal (Trigonal)
6	[100]	Along a cell axis (Trigonal)
7	[111]	Along body diagonal for cubic and rhombohedral
8	$[1\bar{1}0]$	Along face diagonal (Rhombohedral)
9	[100]	Along a cell axis (Hexagonal)
10	$[1\bar{1}0]$	Along face diagonal (Hexagonal)

Table 3.3: The 26 symmetry operations used in space group symbols

Operation	1	2	3	4	5	6	7	8	9	10
1	X	X	X	X	X	X		X	X	X
$\bar{1}$	X	X	X							
2	X	X	X	X	X	X		X	X	X
m	X	X	X	X	X	X		X	X	X
2_1	X	X	X							
a		X	X							
b	X		X							
c	X	X		X	X	X		X	X	X
n	X	X	X	X						
d	X	X	X	X						
3			X				X			
$\bar{3}$			X				X			
3_1			X							
3_2			X							
4			X							
$\bar{4}$			X							
4_1			X							
4_2			X							
4_3			X							
6			X							
$\bar{6}$			X							
6_1			X							
6_2			X							
6_3			X							
6_4			X							
6_5			X							

The matrices used to describe space group symmetry operations are $4 \times$ 4matrices, which contain a 3×3 rotation matrix R_{ij}, a translation vector T_i, and a set of dummy values to facilitate matrix multiplication:

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The translation vector is a combination of two vectors; that arising from screw axes and glide planes, and that arising from the point, vector or plane of action in the unit cell. For example, the matrix for a two-fold rotation acting in the [100] direction through the cell origin (0,0,0) is:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & 0 \\ 0 & 0 & \bar{1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

but if this were to act through a point $(0, \frac{1}{4}, \frac{1}{4})$ it would need a translation component corresponding to the position:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & \frac{1}{2} \\ 0 & 0 & \bar{1} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

If, in addition, this were a two-fold screw axis with its implicit $\frac{1}{2}$ unit translation along the symmetry direction, then the matrix would become:

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & \bar{1} & 0 & \frac{1}{2} \\ 0 & 0 & \bar{1} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Initially, the present code does not attempt to include the position-vector part of the translation vector, but flags those parts of the translation vector which may be changed to allow for the position. The rotation matrix has to be modified for the direction vector on which it operates, but it should be noted that each type of operation has a unique set of determinant and trace for the rotation matrix, independent of orientation.[7, p. 43] Once the symmetry operations have been deduced from the symbol, they are then used to index a look-up table of matrices of the form shown above. At this stage, a check can be made on whether the space group is centrosymmetric, since multiplying the known symmetry matrices together will yield new operations

not explicit in the symbol. If one of these should be an inversion centre, identifiable by its determinant and trace, then a flag is set.

3.4.4 Generators

To be absolutely sure of getting all of the correct operations at their correct locations in the unit cell, it is necessary to reduce the set of matrices derived from the symbol to a minimum set of "generators". The rules for selection of generators are hinted at in *IT*, but are not exhaustive or without exceptions. Therefore a set of heuristic selection rules were devised, partly from those given in *IT*, and partly from empirical observations during testing of the code. The initial selection uses a list of operations in their apparent order of preference as generators. This order is effectively:

$$\bar{1} > 4 > 6 > [a, b, c, n, d] > 2_1 > m > 2 > 3 > 1$$

Once the best set of operations have been selected as generators, their matrices are then adjusted by applying translations to ensure that the generators are a consistent set. This means that the repeated multiplication of the generators, and all of the resulting matrices from such multiplication, should result in an identity matrix with no translation:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In the cases of centrosymmetric space groups, this is usually achieved by ensuring that the point of inversion lies on the origin, although there are one or two exceptions even to this principle. The code which adjusts the generators deals with small families of space groups, since there are no overriding general rules. The space groups are separated into those with common point groups. The point group is derived from the rotation-only parts of the operations given by the symbol. Hence *Pbca* and *Pmmm* share the common point group *mmm*, while $P2_12_12_1$ has point group *222*. Within these point groups some general trends can be determined, but as inspection of the code would illustrate, there are many exceptions. One notable exception is *Fddd* for which no set of rules appears to hold.

3.4.5 The Full Set of Operators

Once a complete and consistent set of generators has been found, the hard work is complete. Simple matrix multiplication will expand the set of generators into a full group, with checks made to ensure that no duplications occurs. This is a matter of comparing each new matrix with all of those

already found, since there cannot be two identical operations. The operations which correspond to the lattice centring are also taken into account here, to produce a further set of matrices, which are themselves checked for duplication.

3.4.6 Lattice Centring

If the first part of the symbol implies that the lattice is not primitive, then up to three extra symmetry operations are included in the set of matrices to get the full unit cell. These matrices are simply identity matrices with a translation component corresponding to the centring vector. For example the C-centring is

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

while an I-centring matrix is

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These matrices are used in the same way as the other symmetry operations.

3.4.7 The Complete Unit Cell

Only at this stage are the atoms introduced. The position vector of each atom in the asymmetric unit is operated on by the complete set of matrices. Checks are made with reasonable tolerances (0.05 fractional units) to ensure that atoms on special positions are not duplicated.

3.5 A Worked Example - $P2_1/c$

Separate into component parts: The lattice centring symbol is "P", so the lattice is primitive, and no centring operations will be required. There is a symbol for only one symmetry direction, which has two parts; a two-fold screw axis parallel to the [010] direction, and a glide plane perpendicular to the [010] direction with a glide in the [001] direction. By convention, in the absence of other indications, the symmetry direction is assumed to be [010].

Deduce the crystal class: First it is noted that there are symbols other than "1" or "$\bar{1}$", so this is not Triclinic. There are no 6-fold, 4-fold or 3-fold axes, and it is not Rhombohedral. The presence of symbols for only one symmetry direction implies that the other two directions have symmetry "1", and hence this is not Orthorhombic. By elimination therefore this space group is Monoclinic.

Obtain the matrices for the symmetry operations: The symbol contains a 2_1 screw axis along the [010] direction; the matrix for this is:

$$\begin{pmatrix} \bar{1} & 0 & 0 & * \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & \bar{1} & * \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and a c glide perpendicular to the [010] direction; the matrix for this is:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & * \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The * symbols indicate that these values may be changed to adjust the position in the unit cell. Hence a 2_1 screw axis along [010] can be moved in either the [100] or [001] directions, but a glide plane perpendicular to [010] can only be moved along the [010] direction. It can also be seen that

$$
\begin{array}{ccccc}
2_1 & \times & c & = & \bar{1} \\
\begin{pmatrix} \bar{1} & 0 & 0 & * \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & \bar{1} & * \\ 0 & 0 & 0 & 1 \end{pmatrix} & \times & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & * \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} \bar{1} & 0 & 0 & * \\ 0 & \bar{1} & 0 & * \\ 0 & 0 & \bar{1} & * \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
\frac{1}{2} \text{ for screw axis} & & \frac{1}{2} \text{ for glide plane} & & \text{translation vector} \\
\text{translation} & & \text{translation} & & \text{undefined}
\end{array}
$$

and hence there is an inversion centre implied in this space group, characterised by the rotation part of the resulting matrix.

Select the generators: Following the order of preference the newly discovered $\bar{1}$ operation and the c glide are the best to use as generators. For this point group $(2/m)$ only two generators are needed. However, to ensure that the inversion centre is on the origin we need to make sure that the translation part of the 2_1 cancels that of the c glide, so the c glide is given a position vector $(0, \frac{1}{2}, 0)$; i.e. the * is replaced by $\frac{1}{2}$.

Multiply the generators:

$$\begin{array}{ccccc} \bar{1} & & c & & 2_1 \text{ on } [010] \\ \begin{pmatrix} \bar{1} & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & 0 \\ 0 & 0 & \bar{1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \times & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} \bar{1} & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & \bar{1} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

The resulting 2_1 is translated off the origin in the [001] direction. Any other combinations of matrices result in duplicates or the identity matrix. Note that the identity must have a zero translation component: this is checked by multiplying

$$2_1 \times c \times \bar{1} = I$$

as long as the translations are consistent.

Check lattice centring: This space group is Primitive, so no further operation are present.

The unit cell can now be constructed: Multiply each atom position by each symmetry matrix.

$$\begin{array}{ccccc} 1 & & \mathbf{x} & & \mathbf{x} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \times & \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} & = & \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \end{array}$$

$$\begin{array}{ccccc} c & & \mathbf{x} & & \mathbf{x}\prime \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} & \times & \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} & = & \begin{pmatrix} x \\ \bar{y}+\frac{1}{2} \\ z+\frac{1}{2} \\ 1 \end{pmatrix} \end{array}$$

$$\begin{array}{ccccc} 2_1 & & \mathbf{x} & & \mathbf{x}\prime\prime \\ \begin{pmatrix} \bar{1} & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & \bar{1} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} & \times & \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} & = & \begin{pmatrix} \bar{x} \\ y+\frac{1}{2} \\ \bar{z}+\frac{1}{2} \\ 1 \end{pmatrix} \end{array}$$

$$\begin{array}{ccccc} \bar{1} & & \mathbf{x} & & \mathbf{x}\prime\prime\prime \\ \begin{pmatrix} \bar{1} & 0 & 0 & 0 \\ 0 & \bar{1} & 0 & 0 \\ 0 & 0 & \bar{1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \times & \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} & = & \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \\ 1 \end{pmatrix} \end{array}$$

The resulting atom positions x, x', x'' and x''' correspond to the number of asymmetric units in the unit cell, and are checked for duplication in case any of the atoms are on special positions. For example, if an atom was on the cell origin, it would be generated twice, because the inversion operation would make x''' the same as x.

3.6 Program Details

The symmetry program, *cracker* , was originally written in Fortran77 on a Digital MicroVAX computer, then transferred to a Hewlett-Packard UNIX workstation, and finally rewritten in C to allow better integration into the *builder2* program (Chapter 4). A full source listing of a stand-alone version of *cracker* is given in Appendix B.

3.6.1 Cracker

Parameters

Command line parameters (should be space group symbol only).

Returns

None.

Description

The *main* routine for Cracker is essentially a wrapper around the symmetry code to handle the command line parameters passed to the program. All parameters passed on the command line are concatenated to form a space group symbol - this makes the program tolerant of spaces on the command line. When the space group symbol has been extracted from the command line it is stored in a global variable **spgp**, and control is passed to the *crack* routine. If *crack* is unsuccessful an error message will be printed.

3.6.2 Crack

Parameters

None

Returns

True for success, **False** otherwise (defined in*crack.h*, appendix B.14).

Description

crack calls *crunch* to make sense of the space group symbol and split it into the component parts. If *crunch* fails *crack* returns **False**. *matax* is then called to allocate matrices for all of the symbols found in the space group symbol. If *matax* returns zero operators *crack* returns **False**. The global flag **centric** is set by calling *invpt* to determine whether the space group is centrosymmetric.

selgen is called to do the main work of selecting generators for the space group, and checking the translation parts of the matrices representing them The matrices are stored in the global array **gmats**, and the translation parts are corrected to fall in the range 0 to 1. The generators are then used in *symeqs* to produce a set of symmetry operations for a Primitive cell, then the centring part of the space group symbol is used to determine how *centre* is called to produce the full set of operations for the space group. An error here causes *crack* to return **False**. The operators are then printed out, and stored in a double-precision array, which is used when the subroutine is incorporated into the *builder2* program.

3.6.3 Crunch

Parameters

None

Returns

True for success, **False** otherwise (defined in*crack.h*, appendix B.14).

Description

The main function of *crunch* is to dissemble the space group symbol given in **spgp** into symbols corresponding to the lattice centring and symbols corresponding to the symmetry directions. The information is stored in two

variables, **centring** and the array **parts[3][2]**. The parts array separates rotation axes and mirror planes for each of the three symmetry direction appropriate to the lattice type. A flag for each type of operation is also allocated for each axis to help with later processing. *crunch* expects any subscripts to be in brackets or preceded by '_', but otherwise accepts standard space group symbols and is tolerant of spaces. To improve the scope of this, the symbol is pre-processed by *spgpex* . Each character is then read in turn and processed by a "switch-case" structure:

P,A,B,C,F,I,R,H: Store as centring symbol **centring**.

1: This will be present either as a spacer or in triclinic groups.

- (bar): Check that the next character is valid for a rotation-inversion axis, and return an error if not.

2,3,4,6: Store the symbol, then check for a following '(' or '_' which implies a subscript (a screw axis). If the subscript is not valid return an error, otherwise store it.

/ (slash): Check the current axis symbol for validity (i.e. make sure that an n/m operation is allowed), and then backspace the axis counter to apply the plane processing to the appropriate axis.

m,a,b,c,n,d: Consider possible glide plane symbols.

r,h: Special cases for rhombohedral groups to help determine the type of setting (rhombohedral or hexagonal) required - ignored in this routine.

default (none of above): Ignore spaces but generate an error at anything else

From what has been extracted, it should be possible to determine the crystal class of the space group using the flags set during the processing. A special flag **nottri** is set to **False** before processing, and if anything which does not fit into a Triclinic cell is found, then the flag is set to **True**. At the end of the symbol processing the flags are examined in the following sequence:

- If **nottri** is still **False** then the cell must be Triclinic (P1 or P-1).
- If the centring symbol is 'R' then it must be Rhombohedral.
- If a six-fold was found then it must be hexagonal.
- If four-fold was found then check for a three-fold to decide between Cubic and Tetragonal. If a three-fold is found in the second axis and the others are not ones then it must be Cubic. Correct for old-style symbols in Cubic groups where 3 should be $\bar{3}$ by checking with the point group.
- If a three-fold was found (other than Cubic or Hexagonal) then it must be Trigonal.
- If all the axes are either two-folds or planes then the cell is Orthorhombic.

- Otherwise the cell must be Monoclinic; this is the hardest to test, so use a process of elimination.

3.6.4 Spgpex

Parameters

in - raw space group symbol, **out** - processed space group symbol.

Returns

None.

Description

spgpex applies a set of rules to the space group symbol to accommodate several possible ways of representing the symbol using standard ASCII characters, such as those used in keyboard input. Much of this routine was designed to accept the concise forms of space group symbol used by the Cambridge Crystallographic Data Centre, primarily to simplify the testing procedure.

The routine search a table of translations for the components of the input string and produces an output string in a well-defined form. The translations are performed in a specific order, to ensure that the result is a space group which really exists, rather than some alternate combination of symbols which has no meaning. For example, the input `P3121` might appear ambiguous, and could be interpreted as the invalid symbol $P312_1$ rather than the correct form

$P3_121$.

3.6.5 Matax

Parameters

Logical flag **multi** - should n-fold axes be expanded to give n operations.

Returns

Number of matrices produced.

Description

matax uses the information stored for the crystal class (i.e. Monoclinic, Orthorhombic etc.) and the subsymbols stored in the **parts** array to determine which symmetry operations are explicitly given in the space group symbol. *matax* handles ambiguous monoclinic space groups by applying the common convention of making the b-axis unique unless a c-glide is present. Then *symget* is called to actually recover the matrix for the operation. If *symget* returns a zero matrix then the operation is not valid, and *matax* returns zero.

If **multi** was set to **True** then operations with order greater than 2 must be repeatedly applied to themselves to generate the implied symmetry operations.

3.6.6 Symget

Parameters

Integers: **op** - numerical code for operation type

ax - numerical code for symmetry direction

x, y - matrix indices

Returns

Integer value of matrix element M_{xy}

Description

This routine simply decodes the matrices stored in a compressed form declared in `matrix2.h`, appendix B.16.

3.6.7 Invpt

Parameters

None.

Returns

True if inversion centre exists, **False** otherwise (defined in *crack.h*, appendix B.14).

Description

To detect an inversion centre, the matrices obtained from the symmetry operations explicitly given in the space group symbol are multiplied together. The existence of an inversion centre is checked by calculating the determinant and trace of the rotation part of each matrix using the *detntr* routine.

A couple of cases allow this process to be bypassed. One exception is if the $\bar{1}$ symbol already exists in the space group symbol, as in $P\bar{1}$. The other is when a $\bar{3}$ operation is present, since all such space groups have a centre of symmetry.

If an inversion operation is discovered, *invpt* returns **True**, and returns **False** otherwise.

3.6.8 Detntr

Parameters

Integers	**imat**	- pointer to matrix,
	det	- pointer to returned determinant value,
	trace	- pointer to returned trace value.

Returns

None (see Parameters).

Description

detntr simply calculates the determinant and trace of a 3×3 integer matrix. To simplify indexing of the matrix, a macro **I2D** is defined, where **I2D(M,x,y)** references matrix element M_{xy}.

3.6.9 Selgen

Parameters

None.

Returns

ngen - number of generators selected.

Description

selgen is the routine which does the selection and correction of the generator matrices required to produce the full space group symmetry. After initialisation of the matrices, *point* is called to get the point group symbol corresponding to the space group. Next the matrix **pno** is loaded with a preference number, which acts as a weighting when choosing the generators. Also, a matrix of flags, **genflg**, is initialised to false - this will be used to flag operations to be used as generators. *onemat* is used to extract the matrices for the operations known from the space group symbol. If an inversion centre has been detected, then an additional matrix will be added to the genera-

tors and flagged as a generator, since an inversion centre is always used as a generator.

The crystal class is used as a switch for the next part of the processing, see source code (Appendix B.10):

Monoclinic: Some adjustment is made to the translations to ensure the origin and inversion (if present) are on the origin.

Orthorhombic: These need to be further split by point group into 222, $mm2$ (including $m2m$ and $2mm$), and mmm. A few special cases needed to be handled separately, notably $Fddd$.

Tetragonal: Also split into point groups 4 & $\bar{4}$, $4/m$, 422, $4mm$, $\bar{4}2m$ and $\bar{4}m2$, and $4/mmm$.

Trigonal: The three-fold axis always lies on the origin, as does a $\bar{3}$ inversion centre. If the three-fold is a screw axis, an translation adjustment needs to be made to any other operators.

Hexagonal: Point groups $6/m$, $6/mmm$ and 622 need some adjustment, while in $P\bar{6}c2$ and $P\bar{6}2c$ the inversion centre is not on the origin.

Cubic: Split into point groups 23 and $m\bar{3}$, 432, $\bar{4}3m$ and $m\bar{3}m$. $F4_132$ needs special adjustment.

Rhombohedral, Triclinic: No adjustments are necessary - any operator that is not an identity is flagged as a generator.

The matrices flagged as generators are then copied to the global array **gmats**, including an identity matrix, and the number of generators is returned.

3.6.10 Point

Parameters

ptgrp - point group symbol returned.

Returns

None (see Parameters).

Description

point uses a simple look-up table to translate space group symbols to point group symbols. These are the rotation-only equivalents, i.e. screw-axes become rotation axes, and glide planes become mirror planes. Thus, for example, $P2_1/c$ belongs to the point group $2/m$.

3.6.11 Onemat

Parameters

char	**axsym**	- symbol of operator.
int	**ax**	- symmetry direction.

Returns

MATRIX structure.

Description

matax returns a **MATRIX** structure containing the matrix corresponding to symmetry operation **axsym** in symmetry direction **ax**. If an error occurs, because the symbol or direction is not valid, then a zero matrix (all elements are zero) is returned.

3.6.12 Symeqs

Parameters

None.

Returns

Number of symmetry operations.

Description

symeqs exhaustively multiplies the generator matrices together to create new symmetry operations. Each new operation is compared with existing operations by converting it to a string with *pretty* and comparing the string representations. All of the new symmetry operators are also included in the

expansion process. The final number of symmetry operators is returned, and the matrices themselves are stored in the array **isym**.

3.6.13 Pretty

Parameters

int **pmat** - pointer to input matrix,

char **output** - string generated by routine,

Returns

None (see Parameters).

Description

pretty is used to convert the matrix representation of a symmetry operation into the form used in International Tables. Each row of the matrix is analysed in turn, and the rotation part of the matrix is converted to x, y or z as appropriate, with the correct sign, then the translation part is converted to a fraction and added on to the string. The three strings for the rows of the matrix are put together, separated by commas, and the result returned in the parameter **output**.

3.6.14 Centre

Parameters

char **ctype** - Type of centring to apply.

int **tnmats** - Number of operations generated.

Returns

None.

Description

centre multiplies all of the existing matrices by a centring operation if the cell is not primitive. New matrices are checked for duplication using the string generated by *pretty* and added to the array of operations. In some cases, the centring operation may have been generated by the expansion of the symmetry operations in *symeqs* , so this is checked before proceeding further. An error is produced if the total number of matrices stored exceeds 192, since this is the maximum number of operators allowed for any legal three-dimensional crystallographic space group.

3.7 Limitations

The present code has been tested extensively and gives the correct result for all space groups tried so far. The prime failing of the program arises when the space group has more than one origin choice. These space groups

are those where there is an inversion centre present, and also another point of very high symmetry. The present code will always chose the inversion centre as the site for the origin, but clearly other programs may chose the alternate origin position. When this happens, there is a simple translation which can be applied to all of the atom positions, which effectively transforms the origin. This is considered simpler than attempting to make a consistent set of generators in the alternate choice of origin.

The program currently will only produce a set of symmetry operations, in the form of matrices, for the space group. Other information related to the space group is not determined. For crystallographic work, information about the systematic absences in diffraction data for the space group could be determined from the space groups. Also such things as the extent of the asymmetric part of a Fourier map can be determined from the space group.

3.8 Testing

For the program presented here to be truly useful, it must provide the correct information for a wide range of inputs. Rigorous testing of the space group interpretation code was largely performed by another program, using the Cambridge Crystallographic Database (CSD) as a source of space group symmetry information. Almost all of the known space groups have been observed in X-ray structure experiments, and all but a few are present in this database. Using a small program to set up a search for each space group

by number, rather than name, ensured that alternative settings and orientations were retrieved. The program was designed to handle all possible space group symbols, including those not explicitly described in International Tables, to allow for experiments where data has been collected for a structure in a non-standard way.

Every structure in the CSD can be extracted as a file in the CSD proprietary FDAT format. This file contains, amongst other things, the structure of the molecule, the space group symbol and number, and a minimal set of space group generator matrices. From these generators, the full set of space group symmetry operations can be produced. The testing program then compares the information produced by feeding the space group symbol into the symmetry code with the symmetry information derived from the FDAT file. Any inconsistencies are reported to allow checking by hand, since there were generally few of these. Only a small proportion of these errors were due to bugs in the symmetry code; most were caused by the way the space group symbol was stored in the CSD, which was not consistent with the input strategy for the symmetry code.

Chapter 4

The Builder2 program

4.1 Introduction

To provide the link between crystallographic structural models and *ab-initio* calculations of surface systems, one first must use the structural data to generate a model of the surface required. This sounds relatively simple, but the nature of crystallographic experiments means that advantage is taken of any symmetry within the lattice. In all but the simplest cases, unscrambling crystallographic symmetry is a non-trivial task, since it requires knowledge of the matrices corresponding to the symmetry operations present. From these, it is possible to use simple, but tedious, matrix and vector multiplication to generate the whole repeat unit from the more-often encountered asymmetric unit. For many organic compounds, the symmetry is simple but the number of atoms involved is large, while for inorganic materials and minerals a small

number of atom sites combine with a large set of symmetry operators to produce the full unit cell.

This is only part of the task, since it is not often that the surface Miller index corresponds to one of the simple Miller planes forming the sides of the unit cell. A surface direction needs to be calculated, the surface unit cell with a suitable thickness needs to be prepared, and for convenience the coordinate system changed to give the surface plane some convenient geometric definition. In addition, stable surface models will generally require no net charge, and no dipole moment along the axis perpendicular to the plane. Surface directions which result in charge or dipole imbalance are likely to reconstruct if energy optimisation is allowed.

The *builder2* program has grown out of a requirement of Oxford Materials Ltd. for a tool to accept the most widely published crystal structure data, and produce a user-specified model representing a slab or block of material, upon which further calculations can be performed. it is partly derived from an earlier program, *figeac*, written at ICI [8], and is designed to be incorporated into *Crysalis*, a package marketed by Oxford Materials. The program contains an effective algorithm for decoding conventional crystallographic space group symbols to generate all symmetry-related atom sites from the minimal asymmetric unit information. The present version of the program is written throughout in an ANSI-conforming version of the C programming language, and as such is expected to be portable across hardware and operating system variations. Among the many advantages of using the

C programming language is that it is the language of choice for most modern graphical applications. While the program is at present a free-standing application, it would not be difficult to convert it into a subprogram within a three-dimensional graphics system, thus allowing more interaction in the creation of the desired model structure. Indeed, during the development of the program, a subroutine was incorporated to permit viewing of the structure using the *AVS* visualisation package.

The input to the program consists of a command line, with several required and optional parameters, and a data file. Currently two formats of data file are supported, a proprietry format called ".crys", and a modified form of the existing standard known as CSSR. Since the output is in the CSSR form, output from the program should be readable by the program. The ".crys" form is ideal for initial work on a structure. This form contains atom types, atom positions, unit cell information and space group information. The CSSR format contains no space group information, but this can be given on the command line if necessary. Detection of input format is, as far as possible, automatic. The file is tried as ".crys" format, and if this fails is tried as CSSR format. The program can be told to expect CSSR format with another command line option.

The data set required for generation of a model is as follows:

1. Unit cell dimensions, assumed to be in Ångstroms and degrees. These can be given as 1 1 1 90 90 90 if the atom data are already in orthogonal Ångstrom coordinates.

2. A list of atom positions and types describing the crystallographic asymmetric unit, optionally with charges and/or connectivity information.

3. A space group symbol, which is $P1$ by default.

4. A Miller index to specify a plane with which to align the surface.

5. A thickness for the slab of atoms produced, which can be specified as an overall thickness, or distances above and below the origin.

6. A set of up to three multipliers to specify the number of surface units cells to build the final structure with. If there are three numbers specified, the thickness parameter is not required and ignored.

7. Optional flags for specifying zero overall charge and zero overall dipole.

4.2 Data input

A filename is taken from the command line options, and the program attempts to open that file for input. An error message will be issued if that is not possible for whatever reason. Both file formats are human readable and editable ASCII text. After opening, a few tests are performed to determine the type of file present, and the relevant information is read from the file. At this stage, the structure is assumed to be a minimal asymmetric unit, as would be reported in a crystal structure publication or produced by one of the crystal structure refinement packages. In addition to this information,

parameters specifying the final output structure are parsed. These will give the Miller plane and dimensions of the surface model.

4.2.1 The ".crys" file format

Records are free format, each section labelled with a keyword

gheader This identifies the file as ".crys" format; the following line carries a title.

abc Cell contents; following line contains a b c (in Ångstroms) and α β γ (in degrees).

ions Atom/Ion types used; each following line contains a reference number, charge, and an element label

basis Atom/Ion positions in fractional coordinates; each following line contains the atom type number, and fractional x, y, and z coordinates.

sgrp Space group symbol; the following line contains a conventional space group symbol written as described in 3.3.

4.2.2 The "CSSR" file format

A fixed format file - formats are given following Fortran convention.

Record 1: format (22X,A6,10X,3F8.3)

refcode 6 character reference code,
taken as a segment name by *Chem-X*

a, b, c unit cell length parameters

Record 2: format (21X,3F8.3)

alpha
beta unit cell angles
gamma

Record 3: format (2I4,1X,A60)

natoms number of atoms

inorm 1 for cartesian coordinates
0 for fractional coordinates.

title1 used as the structure name

Record 4: format (4X,I2,1X,A60)

iqset 0 for no charges, 1 if charges specified

title2 another title.

Record 5: format (I4,1X,A4,1X,1X,3(F9.5,1X),8I4,1X,F7.3,1X,I3)

atom information:

serial	
label	serial and label form a reference
x,y,z	coordinates
(con(j),j=1,8)	connectivity table
chg	charge of the species
group	not used.

Appended after last atom data line: (only read if inorm = 1, that is, if atom coordinates are in cartesian coordinates)

b1x, b1y, b1z	bulk lattice vectors in same
b2x, b2y, b2z	cartesian coordinate system
b3x, b3y, b3z	as atom coordinates.

4.3 Surface Construction

The expected input to the Builder2 program is a crystallographic asymmetric unit - the part of the unit cell containing no crystallographic symmetry. In most cases this will require symmetry expansion to produce a full unit cell. The code described in another chapter 3 was developed to perform this task. While it is true that computer programs already exist for the interpretation of symmetry (for example, within Crystals[9]), they are not complete for all space groups, and the additional problem of writing code for a commercial project means that use of public-domain or other existing code was not possible. The alternative approach, taken by some crystallographic packages [10], is to rely on the user having access to International Tables

[6], in which the symmetry operations for each space group can be found. Experience has shown that in the case of high-symmetry mineral structures this can be extremely time-consuming and laborious.

The full unit cell provides the bulk basis required to create the surface. What is then required is that the unit cell be cut by the desired Miller index to produce a top layer and bottom layer. The resulting surface unit can then be expanded to create a slab of atoms forming the model. The next task is to calculate the lattice vectors of the new surface cell relative to the old bulk unit cell. The Miller indices are taken to define a vector perpendicular to the desired surface. This vector is defined as the c-axis of the surface unit, so that points above the surface have a positive z-coordinate. The in-plane vectors, corresponding to the a and b axes of the surface cell, are defined to result in a right-handed set aligned as far as possible with the original bulk unit cell. A matrix is constructed to transform bulk coordinates to surface coordinates, and the atom positions of the bulk cell are converted to the surface cell. The space enclosed by the surface cell is searched for atom positions in adjacent bulk cells which might fall into the surface unit. In addition, if connectivity data was present in the input, then the search is extended to include whole molecules in the surface unit.

The desired surface thickness, and optional conditions on dipole and charge for the surface, will require an adjustment to the final surface basis. The thickness is satisfied by defining a zero layer in the surface unit, and then adjusting the length of the c-axis of the surface unit to fit. The atoms

contained within the space produced by this are included in the surface unit. To ensure zero charge and zero dipole, several possible values of thickness close to that requested are explored. The charge and/or dipole of the unit is calculated for each, and the best fit to the desired conditions is used. This may result in the final surface thickness being different to the value input.

4.4 Model Output

The model resulting from the above manipulation is saved to a human-readable ASCII file, in CSSR format. The primary target for this output is Oxford Materials' *Crysalis* program, which enables visualisation of the model among other functions. Unfortunately, *Crysalis* is not portable across computer systems, since it makes use of a graphics system only available on Silicon Graphics workstations. For the development of the program an alternative output routine was added to produce a file readable by the *AVS* visualisation system. This format is a pure graphics format, containing no chemical information, and is therefore not useful as input to any other system. The standard output format contains the surface unit cell dimensions and all of the atom positions required to construct the surface. The coordinates of the atom positions are in orthogonal Ångstrom coordinates, allowing true measurement of distances on the model. The slab is defined as having positive z above the surface, with the surface level itself defined as $z = 0$. If appropriate, charges and connectivity for each atom is also stored.

Chapter 5

Modelling TiO_2 surfaces

5.1 Introduction

Computer modelling is now established as a valuable tool in chemistry research, with applications in drug design, catalysis research and material science among many others. The computer model allows microscopic examination of the environment in which molecules form and react, and also permits calculations on energy and forces which can be used to explain physical properties.

The extent to which computer modelling can be used is still restricted by the types of model available, and also by limits on system size dictated by the current state of the art in computer hardware. While it is perfectly possible to model small organic molecules in isolation on a desktop workstation, applications requiring large numbers of atoms or extensive molecular

interactions need the sort of data storage and computing power still only found on mainframe computers. This work aimed to develop a method of applying high-level electronic calculations to a complex system, and produce a numerical model on which meaningful simulation experiments could be performed.

Metal oxides, especially transition metal oxides, are widly used in commerce and research. Titanium dioxide is used in catalysis and, most importantly, as a white pigment in paints; in the paint industry it is the primary component of many surface treatments since it has the best opacity of any simple material. The surface chemistry of titanium dioxide is beginning to be an important topic for research, as an understanding of it's catalytic properties is sought. The primary catalytic use for TiO_2 is in large-scale methanol synthesis, and some very recent work [11] has looked at the orientation of CO_2 on TiO_2 surfaces.

5.2 Previous Work in the Literature

5.2.1 Introduction

There are two approaches to theoretical structural modelling of chemical systems. One is to devise a molecular mechanics scheme, whereby atoms are treated as objects attracted and repelled by forces governed by Newtonian mechanics. The other is to attempt to solve the electrodynamics equations that detail the populations and orientations of atomic orbitals, commonly

known as the *ab-initio* approach.

These two broad areas have their respective advantages. Molecular mechanics has fairly low computational overheads, since the optimisation of structures requires the minimisation of simple energy functions. These are made up of a superposition of potential energy models representing each inter atomic interaction. While this leads to the minimisation of a many-dimensional function, it is still within the limits of most desktop computers for small molecules. The disadvantages of this approach is that the interaction between each type of atom in the model must be defined and parameterised. This has been done extensively for carbon, nitrogen and oxygen and other components of organic systems, but for ionic structures the potential functions often produce a poor agreement with experimental results. In addition, molecular mechanics is based on the motion of solid bodies connected by forces. These bodies are subject to Newtonian mechanics, so potential and kinetic energy can be calculated and minimised, the motion of the atoms can be reproduced and the effects of distorting the structure can be studied. Electronic effects (conduction, magnetism and spectroscopic details) are not available, since the model does not treat atoms as electron-containing bodies at all.

Electronic calculations by *ab-initio* methods account for the interactions between the nuclei and the electrons which make up the system by attempting to solve the total wavefunction created by bringing together the wavefunctions of the component atoms. The results of these calculations are generally

very close to experimental measurements, both for geometric measurements and electronic effects. The cost of these calculations is extremely high because the wavefunctions cannot be solved analytically, or even exactly, for real systems. Hence most *ab-initio* methods rely on iterative methods to produce self-consistent results, leading to long cycles of computation and high demands on disk storage space. *Ab-initio* calculations are currently limited to small molecules or structures with small repeating units, such as ionic crystals and polymers. They are also constrained to atoms of low atomic number because as the total number of electrons in the model is increased, the complexity and computation time increases rapidly. This can be compared with the limited number of molecular mechanics studies of heavy atom systems, which is due to the small amount of data from which interaction parameters can be developed.

With reference particularly to ionic oxide systems, many apparently simple but important materials represent the extremes of possibility for either of the above methods. The problem is that such materials contain ions which are not handled well by molecular mechanics, and too many electrons for *ab-initio* calculations to be applied. As will be shown below, the level of complexity reached in the study of simple metal oxides does not yet reach that required to understand the macroscopic properties of these materials in the fields of geochemistry, nuclear power and catalysis where they are so important. A particular increase in difficulty is encountered when the metal in question contains d-orbitals. Producing suitable basis sets to model the

anisotropy associated with d-orbitals appears to be an art rather than a science. Unfortunately such metals are often part of the most useful materials, with properties most difficult to explain qualitatively.

5.2.2 *Ab-initio* studies

The work detailed here covers the study of titanium dioxide by electronic calculations. The level of approximation applied tends to decrease with time, so that the earliest studies use a modified electron gas (MEG) model which treats the species as ions and the valence electrons as distributed throughout the solid. In more recent studies, Extended Huckel (EHT) calculations and HartreeFock (HF) methods have become more popular, since they apply fewer assumptions about the electronic structure of the materials. There is also a further division of *ab-initio* work into those studies which use finite clusters to simulate infinite lattices, and those studies which use periodic boundaries and Bloch functions to model wavefunctions in a truly infinite lattice.

Theoretical electronic properties of TiO_2 (Rutile) (001) **and** (110) **surfaces.**

R.V. Kasowski and R.H. Tait. *Physical Review B*20, 12:5169-5177, 1979

This is one of the first *ab-initio* studies of TiO_2; the authors apply the Linear Combination of Muffin-Tin Orbitals (LCMTO)[12, 13] method to mod-

elling Rutile, both in bulk TiO_2 form and as thin surface films. The LCMTO method has been applied to metals in infinite lattices and derive information about energy bands in the structure. The authors point out that for a structure such as TiO_2 the full LCMTO description is extremely complicated and computationally very expensive (although this report is dated 1979 so this may not be such a drawback now). They simplify the basis set required by ignoring p-orbitals in the Ti atoms, justifying this by pointing out that the p-orbitals are some 7eV above the conduction band edge, and not important in their study. In addition, since the main interest was to compare bulk TiO_2 with surface layers, the same basis set was used for both bulk solid and layer models, eliminating the need for absolute accuracy. Also the authors demonstrate a reasonably good agreement between the theoretical results for bulk TiO_2 and Ti_2O_3 and experimental ultraviolet photoelectron spectra for these materials.

The density of states (DOS) results for bulk TiO_2 show a band gap between -9.24eV and -12.59eV, with hybridisation of the Ti(3d) and O(2p) orbitals, although the valence band is mainly O(2p) and the conduction band is mainly Ti(3d). For Ti_2O_3 the band gap is similar, but the conduction band is partially occupied by Ti 3d electrons as a result of the lower oxidation state (Ti^{III} compared with Ti^{IV} in TiO_2). Ti_2O_3 is therefore considered metallic from the *ab-initio* results.

The Rutile TiO_2 (001) film studied consisted of a slab of five layers arranged in a supercell, with enough vacuum space between slabs in the repeat

lattice to avoid interaction between slabs. The DOS results show a band gap much narrower than in the bulk material. This appears to be due to a large number of O(2p) and Ti(3d) surface states which are a result of the surface Ti atoms being only 4-fold coordinated (they are 6-coordinate in the bulk solid), and therefore having "dangling bonds". Such a surface is considered to be unstable in this form, and requires reconstruction to achieve thermodynamic stability. The report also points out that this reconstruction must be more than just the relaxation of oxygen positions, since this cannot result in an increase in the Ti coordination. A much greater reconstruction, or possibly faceting of the surface, is required to stabilise the (001) surface. It is also noted in support of this that 4-fold coordination is rare in titanium-containing oxides.

The TiO_2 (110) surface was modelled as a three-layer slab, although the layers in this projection are not completely planar. The band gap is wider than the (001) surface, but is narrowed relative to the bulk structure by some of the O(2p) states. These states are not due to coordination defects at the surface, since the Ti atoms in the (110) layer are 5 and 6-coordinated, but rather because the O atoms protrude from the surface. If they are allowed to relax inward by 0.2Å by shortening the Ti-O bonds from 1.85Å to 1.75Å, the band gap widens to approximately that of the bulk structure. With this slight relaxation, no dramatic reconstruction is required to make the (110) surface as stable as the bulk material, corresponding to the experimental observations of this surface.

Ionic modelling of mineral structures and energies in the electron gas approximation: TiO_2 polymorphs, quartz, forsterite, diopside.

J.E. Post, C.W. Burnham., *American Mineralogist*, 71:142-150, 1986

This paper studies a range of materials including the three common TiO_2 polymorphs; Rutile, Anatase and Brookite. The method uses a modified electron gas (MEG) model, which assumes that the species are wholly ionic. This seems a reasonable approximation for TiO_2, although the other materials studied in the paper include SiO_4 minerals where the bonding is usually considered to be covalent. As has been found in later work (see below), there is also a considerable amount of covalent bonding in TiO_2.

For a nearly ionic species, this MEG model should compare favourably with molecular mechanics models, since it does attempt to include the electrons in the calculation. In addition, there is no need for empirical measurements to determine any of the parameters. The limitation is that only closed-shell ions can be used in the calculation, so many of the transition metals are not amenable.

The general method is to separate the short-range interactions between the ions into kinetic, exchange, correlation and non-point Coulomb contributions; each is expressed as a functional of the electron densities. The net electron density of each ion pair is assumed to be a superposition of individual densities calculated from Hartree-Fock atomic wavefunctions. Note that these are not the Hartree-Fock crystal wavefunctions used in the periodic

lattice methods mentioned below. Full details of the MEG method can be found in the work of Gordon et al.[14, 15, 16]. The cation-cation short range interactions were considered insignificant (the octahedra of oxygen atoms which make up TiO_2 insulate the Ti ions from each other quite well), and these terms were not included in the model.

The structures of the TiO_2 polymorphs were optimised with respect to energy by varying the cell parameters a and c, and the oxygen position. In common with most other studies, the c dimension is over-estimated; by 3.5% in Rutile and 6.1% in Anatase. The distortion of the octahedra in Rutile is not quite as experimental data suggests, since the calculated axial bonds are around 0.03Å shorter than the equatorial bonds, whereas the experimental data demonstrates that the opposite is true. In Anatase the sense, but not the magnitude, of the distortion is predicted by calculation. Brookite has a more complicated unit cell requiring 12 parameters to define. The space group is *Pcab* (orthorhombic) and there are two oxygen ions in the asymmetric unit. The results of the calculation are close to the observed structure, with the cell parameters within 1.5% of the experimental values and the Ti-O distances matching very well.

The conclusions reached from this work are that although the purely ionic model gives fair agreement with experiment, the distortions of the octahedra are probably linked to some covalent bonding effects. The calculated energies of the structures are strictly only valid at zero temperature and pressure, but suggest that Rutile is more stable than Anatase, with Brookite least stable;

this agrees with many later predictions and also the relative abundance of these three materials in nature.

Pseudopotential periodic Hartree-Fock study of Rutile TiO_2.

B. Silvi, N. Fourati, R. Nada, C.R.A. Catlow. *J. Phys. Chem. Solids* 52, 8:1005-1009, 1991

This appears to be the first HF study of titanium dioxide, using *Crystal92* [17] and employing the Durand-Barthelat effective core pseudopotential (ECP)[18, 19, 20, 21] to overcome the computational cost of an all-electron calculation for titanium. In addition they believe that using the ECP minimises the basis-set superposition error.

The optimised basis set refined for the Ti atom yields an energy very close to the experimental value for the first four ionisation potentials - the error (0.05 Hartree) is ascribed to correlation and relativistic effects.

For oxygen they use a PS-31G basis set (again employing a pseudopotential for the core). The energy using an optimised basis set is within 0.3 Hartree of the experimental value for the first six ionisation potentials of oxygen, again this difference is ascribed largely to the correlation energy, estimated to be 0.258 Hartree.

Having determined suitable basis sets for the ions, the authors go on to optimise the structure by varying a selection of three parameters that describe the TiO_2 unit cell. Rather than selecting the obvious choices of a and c (cell parameters) plus a fractional coordinate of oxygen x, they choose

two bond lengths and an angle to describe the geometry; labelled r_1, r_2 and θ in Figure 5.1.

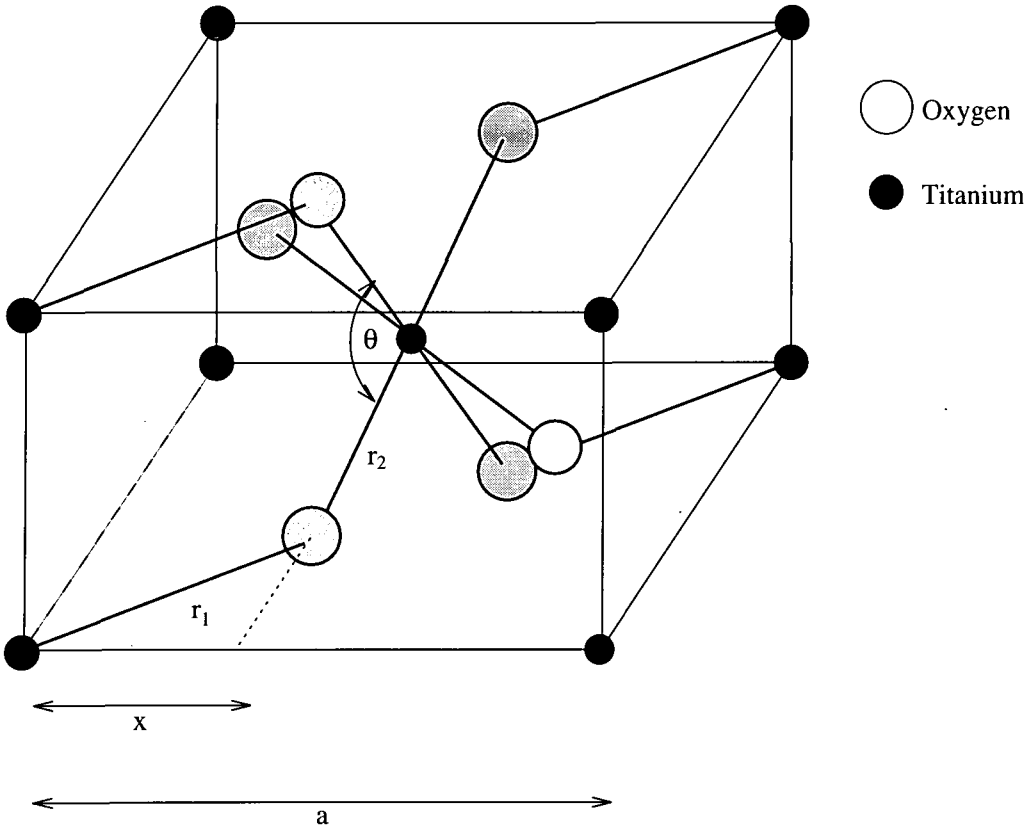


Figure 5.1: The unit cell of Rutile TiO_2

The results of this geometry optimisation were "very satisfactory", with the resulting values for a, c and x only around 1% away from the experimental (crystallographic) coordinates.

The energy calculated for this optimised structure was -69.788179 Hartree. Subtracting the values obtained for the energies of the free atoms calculated

using this basis set gave a value for the binding energy of -0.6615 Hartree, which differs from a thermodynamics calculation based on experimental results by 0.7715 Hartree. This difference is ascribed to correlation effects, which clearly form a major part of the error encountered with HF calculations. A remaining source of error, after some allowance for correlation had been made, was suggested to be the basis set superposition error.

Analysis of the calculated Mulliken population showed that Rutile TiO_2 is partially ionic (some previous studies have assumed fully ionic species), with net charges of +2.600 (Ti) and -1.300 (O). An electron density calculation illustrated the charge transfer to oxygen, and also that the Ti atom contributes the bonding population. This agreed in general with an experimental study of the density[22].

Analysis of the band structure shows a predictably flat topology, with the topmost states being those of the oxygen lone pairs, then a block of Ti states, and the oxygen 2s orbitals providing the lowest states. The band gap is calculated to be around 10eV, which is much larger than the experimental values that are of the order of 3eV. This is explained as a common effect in HF calculations. TiO_2 is, however, predicted to be an insulator by the authors. Density of states assessment indicates that the Ti 3d orbitals contribute to the bonding as well as forming much of the conduction band.

The authors, in summary, claim a substantial improvement in structural accuracy for their HF calculations over previous pair potential studies. In addition, the pair potential method is unable to produce the electron distri-

bution results presented. The authors also feel that they have proved the reliability of HF calculations on *transition metals* (their italics), especially when combined with effective core pseudopotentials.

Theoretical analysis of the structures of titanium dioxide crystals.

A. Fahmi, C. Minot, B. Silvi, M. Caus, *Physical Review B*47, 18:11717-11724, 1993

In this more recent study of TiO_2, the authors discuss the previous results for Rutile TiO_2, and also give results for the Anatase polymorph. The authors present a qualitative argument for the relative stability of the two polymorphs and then proceed to compare them by HF calculation.

Table 5.1: Experimental thermodynamic data for TiO_2 polymorphs at 298.15K

	Rutile	Anatase
ΔH_f^0 (kcal mol^{-1})	225.8	224.6
ΔH_f^0 (kcal mol^{-1})	212.6	211.4
S^0 (cal K^{-1} mol^{-1})	12.03	11.93

The authors take the axial Ti-O bond lengths (d_{ax}) and the equatorial Ti-O bond lengths (d_{eq}) to be approximately equal to their average (1.95Å) and hold this constant while analysing the effects of change the bond angles of the distorted TiO_6 octahedra. It was thought that the Coulomb repulsion between oxygen atoms would have the main effect on the energy of the system and so the cell would expand to minimise these interactions.

The angles are defined in Figure 5.2 for the Rutile structure, and are defined similarly for Anatase.

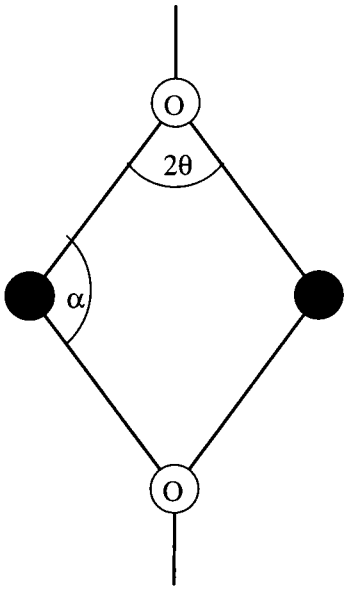


Figure 5.2: Angle definitions in Rutile and Anatase structures

The angle α was found to have little effect on the distance between oxygen atoms and hence no significant effect on the Coulomb repulsion, although the cell volume increases with decreasing α.

What does appear to affect the optimal volume is the coordination around the atoms. In both polymorphs, the coordination type of each species would require the O-Ti-O angle α to be 90 degrees, while the Ti-O-Ti angle 2θ should be 120 degrees (the oxygen atoms are effectively 3-coordinate). In Rutile undistorted octahedra would result in 2θ=90 degrees, while in Anatase the undistorted structure requires that 2θ=180 degrees. The real structures show a compromise between undistorted octahedra and the three-fold co-

ordination for oxygen, by increasing the 2θ in Rutile to 98.88 degrees and decreasing 2θ in Anatase to 156.16 degrees. The value of α is decreased correspondingly, flattening the octahedra and increasing the volume.

The study also looks at the bonding distances by applying Extended Huckel type calculations on Ti-O clusters. The authors point out that although EHT calculations don't give good results for geometry optimisation, they are effective for analysing the factors influencing the geometry. The calculations on the $[TiO_6^{8-}]$ octahedral cluster optimise the Ti-O bond lengths to be 1.965Å for d_{ax} and 1.82Å for d_{eq}. This optimisation was carried out against the O-O repulsion, and although the magnitudes do not agree with experiment, the sense of the distortion is correct.

The essential meaning of the qualitative analysis and EHT calculations is that the main influence on the energy of the structure is the coordination type of the oxygen, and it's distortion from the optimal 120 degrees which results in O-O repulsion. In Anatase, this angle is more distorted than in Rutile, and so Anatase is a higher energy structure than Rutile. The EHT calculations, although often inaccurate, have been shown to produce the correct sense of this energy difference.

The HF study of Anatase is presented to compare with the previous study of Rutile, and again the authors claim very good agreement with experimental geometric parameters. They point out that their results are a noticeable improvement over previous MEG models, where the species were considered to be purely ionic. The HF analysis is initially not perfect, since the energy

for Anatase is approximately 10^{-3} Hartree per TiO_2 lower . The correct sense of the energy difference is only reproduced when correlation is taken into account. The greater density of Rutile places more significance on correlation than is found in Anatase . The correlation density functionals applied were those of Perdew[23] and Colle and Salvetti[24, 25].

The Mulliken population analysis of Anatase compares very closely with Rutile, and the electron density shows similarities. The oxygen atoms have almost spherical density and the density around the Ti atoms is polarised into regions extending in the bonding directions (note that the density study in this paper is a total valence electron density rather than a deformation density). The band structure of Anatase is found, unsurprisingly, to be very similar to that of Rutile, and again the band gap is over-estimated at around 10eV, and the density of states projection draws similar conclusions about the contribution of the Ti 3d orbitals in the bonding.

The parameters of the HF calculation were essentially the same for this Anatase study as for the Rutile work. The only difference appears to be a slightly adjusted Ti basis set, where the Ti 4s orbitals were absorbed into the 4sp shell used for polarisation, since they are not occupied.

5.2.3 Molecular mechanics and pair potential studies

Inter-atomic potentials for oxides.

C.R.A. Catlow, C.M. Freeman, M.S. Islam, R.A. Jackson, M. Leslie, S.M. Thomlinson, *Philosophical Magazine A*,58, 1:123-141, 1988

This paper provides a good overview of the application of the pair potential model to ionic oxides, and includes Rutile TiO_2. The main point of the TiO_2 work is that most pair potential models for TiO_2 model the geometry well or the dielectric properties well, but not both. Mention is made of examples[26, 27] where a potential system was devised to account for the dielectric properties of Rutile TiO_2 based on the experimental crystal coordinates, but when the models were relaxed to a zero-strain geometry the dielectric properties were wildly inaccurate. Conversely, another example[28] modelled the equilibrium condition well and also reproduced the phonon dispersion curves, but did not accurately model the dielectric properties.

The main problem encountered in modelling Rutile TiO_2 is that the c/a ratio for the unit cell usually turns out to be too large. This effect is blamed on the use of simple pair potentials, since cation-cation repulsion between the edge-sharing octahedra pushes the structure apart along the c axis. The authors suggest that a many-bodied potential is needed to retain the rigidity of the octahedra.

Unfortunately there is no obvious set of three atoms in the Rutile structure for which a bond-bending potential could be devised; this would be the

simplest form of many-bodied potential. The solution that the authors propose is to model the general covalency with a potential for three Ti ions, rather than model specific covalent bond interactions. They claim to have reduced the error on the c/a ratio to 2%, compared to around 7% in previous work.

5.3 *Ab-Initio* Hartree-Fock

5.3.1 Background

Molecular mechanics studies of titanium dioxide have shown that a conventional potential system using bond stretching, bend and torsion potentials has limited application to largely ionic structures. Extending such a system to examine interactions at the surface of an ionic solid, especially a transition metal compound, is unlikely to produce realistic results in any form of simulation. The alternative would be to simulate using a full electronic calculation for each step of simulation, but that would be extremely time consuming.

This work was directed towards producing a numerical potential system for interaction using electronic calculations, which could then be applied to a simulation. A grid of energy values covering the space above the surface could be calculated by locating the appropriate molecule in the model of the surface and performing a single point calculation on the system. The grid could then be applied to simulation, with some form of interpolation used to

smooth the grid out.

The project followed a simple plan:

- Establish basis sets for the atom species used in the calculation.
- Create a model of the bulk solid using the geometry of the known crystal structure, and optimise the basis sets in this new environment.
- Perform an energy minimisation on the bulk solid, allowing the structural parameters to vary, but keeping the basis sets fixed. Use this to check the validity of the computational method.
- Returning to the crystalline coordinates, generate the required surface structure from the bulk structure, and refine the basis sets again.
- Choose a molecule to explore the surface with, and also establish basis sets for the atoms in this molecule.
- Insert the molecule into the surface model, and systematically explore the energy hypersurface of the interactions between surface and molecule.

5.3.2 Basis sets

Ab-initio Hartree-Fock

calculations attempt to describe the electron distribution around the atoms with a set of simple probability distribution functions. The distribution functions in this calculation have a form related to the gaussian function

in three dimensions, and are intended to approximately model the shapes of atomic orbitals. As such they are often referred to as s, p and d functions depending on their form, although they do not have a direct equivalence with these orbitals. The Linear Combination of Atomic Orbitals (LCAO) approximation is invoked to allow the construction of molecular distribution functions from these gaussian-type functions. The gaussian functions are integrated over space to generate an electron density function for the molecule, and from the electron density the properties of the system can be extracted.

Hartree-Fock calculations need to refine the initial input parameters of the basis set to accommodate variations in the distribution of electrons throughout the molecule. Each nucleus will affect the local electron distribution, and the electrons themselves are susceptible Pauli exclusion and electrostatic forces. The many-bodied problem that is created by these interactions is solved by a Self Consistent Field (SCF) method (figure 5.3), where the model is tweaked to minimise the energy of the system and then recalculated until that the electronic distribution is self-consistent to within a convergence criterion. The application of Hartree-Fock Self Consistent Field methods to periodic systems is outlined by Pisani and Dovesi [29].

From an early stage it was realised that the basis sets used in this form of calculation would be restricted by the scale of the final calculations intended. Additionally, the *Crystal92* program used has some limitations on the extent of the basis sets it can use.

To help reduce the size of the calculation, an effective core pseudopotential

(ECP) was used for both the titanium and oxygen ions. The assumption made is that the core electrons are heavily shielded by the valence electrons, and so need not be included in the calculation, but rather gathered together with the nucleus. The form of the pseudopotential is chosen such that there is a seamless join between it and the gaussian-type functions used to model the outer shells. The ECP formalism used in this work is that proposed by Durand and Barthelat [18, 19, 20, 21], and gives titanium a core equivalent to a neon atom, and oxygen a core equivalent to a helium atom. Taking these core electrons out of the model results in a dramatic decrease in the scale of the calculation. The primary drawback is that the energies computed are no longer absolute energies, since the energy of the core electrons is lost. The results can still be compared with calculations on systems using the same ECP and basis set configuration in a relative way, and so they are still useful for generating a potential. The energy lost by using the ECP should be constant whatever the environment around the atoms, and could be added later to the results to find absolute energies.

Optimisation of basis sets was performed with a local modification of the *Crystal92* program to embed it in a minimisation program using Powell's method [30]. This allowed the use of a template file from which an input file for *Crystal92* was generated with the values of the variables inserted. A calculation of the energy was then carried out, and the result fed back into the minimisation procedure. Each single point calculation for a single atom took of the order of 10 seconds, so this minimisation was not terribly time-

consuming. The final basis sets for the isolated neutral atoms were as given in Tables 5.3.2 and 5.3.2.

Table 5.2: Basis set for isolated Ti atom with Durand-Barthelat ECP (Ne core)

Type	exponent	s-coefficient	p-coefficient	d-coefficient
3sp	0.8230	1.0	1.0	-
4sp	0.3329	1.0	1.0	-
3d	10.017188	-	-	.129841
	2.762720	-	-	.406390
	.801286	-	-	.841634
	.127760	-	-	1.325561

Table 5.3: Basis set for isolated O atom with Durand-Barthelat ECP (He core)

Type	exponent	s-coefficient	p-coefficient
2sp	31.628698	-.008830	.009580
	8.647912	-.091500	.069600
	3.206664	-.040200	.206500
	1.116974	.379000	.347000
3sp	.506947	1.000000	1.000000
4sp	.176225	1.000000	1.000000

5.3.3 Bulk Model

The crystal structure of Rutile titanium dioxide has been determined very accurately by Restori *et. al.* [22] in a charge density study. The structure crystallises in the tetragonal space group $P4_2/mnm$ (Figure 5.4), with

measured cell parameters of $a = 4.593659(18)$, $c = 2.958682(8)$, and oxygen fractional coordinates of $0.30479(10), 0.30479(10), 0$. It is a reasonable assumption that the experimental crystal structure is at least in a local minimum; the existence of the polymorphs Anatase and Brookite suggests that there is more than one such local minimum. A good model should therefore fit fairly closely to the crystal structure given the same constraints on the symmetry.

Inserting the atoms modelled above into the crystal structure, along with symmetry information, means that only geometrical parameters need to be varied. However, the isolated atom basis sets are not quite right for a crystalline ionic solid. In the case of Rutile, there is evidence [31] that the system is not purely ionic, based on the Mulliken populations of the atom sites. The basis sets must therefore be refined to account for some covalent interaction, or bond electron density, between the atom species. The minimisation was again performed using Powell's method, since *Crystal92* can only produce energy values and not the derivatives needed to use a conjugate-gradient method of minimisation.

The high symmetry of the Rutile system means that only three geometrical parameters are needed to describe it; the most obvious are the two cell dimensions and the oxygen coordinate. The Ti atoms lie on the origin and symmetry-related special positions, while the O atoms lie on the face diagonal of the (xy0) plane and symmetry-related positions. The time required for each single-point calculation was reasonably short, and this optimisa-

Table 5.4: Comparison of different models of Rutile TiO_2

Method	Energy/a.u.	a/Å (% error)	c/Å (% error)	x (% error)	c/a ratio (% error)	$V/Å^3$
Crystal Structure	N/A	4.593659 (-)	2.958682 (-)	0.30479 (-)	0.644 (-)	62.43
Silvi *et al.*	-69.7882	4.55498 (-0.84)	2.99784 (1.32)	0.30613 (0.43)	0.658 (2.17)	62.20
This work	-69.5734	4.6551 (1.34)	3.00088 (1.69)	0.3063 (0.50)	0.6463 (0.36)	65.20

tion of the geometry within the Hartree-Fock approximation converged well. However, it was not possible to match the final results of Silvi *et al.* [31], despite the fact that the same program was used. For an unknown reason, the *Crystal92* program refused to accept the 5d/3s basis set which Silvi claims to have used. Table 5.4 compares the results of the calculations with the crystal structure data.

Agreement with the Silvi *et al.* data is not perfect, but the present study is roughly as close to the measured parameters as their work. The cell volume is a little over-estimated, but the ratio between a and c is closer. It is possible that a slightly more contracted basis set (i.e. more ionic nature) would reduce this inflation of the cell, but this basis set was optimised when the geometry was fixed at the crystal coordinates. The fairly large discrepancy in energy between Silvi's results and the present work is almost certainly due to the smaller basis set. This geometry optimisation was intended as a proof of concept, and the rest of the work used the crystalline coordinates for the model.

5.3.4 Surface Structure

Crystal92 is capable of using three-dimensional structure data, and applying a slicing operation to generate a surface slab. In the present study, it was considered unclear as to the exact nature of this slab. Visualising a structure generated in this way was extremely difficult, since the unit cell produced was not an obvious cluster of atoms. It was felt that this was not satisfactory, primarily because location of an adsorbate molecule above this surface could be very confusing. Using a pre-release version of the *builder2* program (see Chapter 4) allowed the construction of a surface unit which was comprehensible as a miniature surface in its own right. Fractional coordinates referenced against the edges of such a cell had meaningful values in real space, and a clear interpretation of both the thickness and the surface of such a slab was possible.

The surface chosen for this study was the Rutile (110) surface. This surface has been observed experimentally [32, 33] as the most prevalent in micro-crystalline Rutile powder, and the most thermodynamically stable [34]. The actual surface is fairly close to the form of a simple slice along the (110) plane of the crystal structure. Some previous modelling work [35] has suggested that there is a slight relaxation of the surface on cleaving, but for the present study this has been ignored. Other surfaces observed in experimental work are the (100) and (001) planes, which undergo much more reconstruction to achieve stability. The symmetry of the (110) projection is rectangular, with two perpendicular mirror planes and a rotation axis; plane

group *p2mm*, see Figure 5.5. The dimensions of the surface cell are 6.49Å and 2.96Å, and the thickness of the slab in the z direction in this model is 6.49Å. At the surface, the topmost titanium atoms are five-coordinate, rather than six-coordinate in the bulk, while the oxygen atoms are two- rather than three-coordinate.

This model was equivalent to a thin film of Rutile of infinite extent in the xy plane, and five layers of atoms thick in the z plane. The *Crystal92* program has a function whereby it will search for symmetry in an input model, and this was used to help reduce the scale of the calculation; a model containing sixteen atoms including several transition metal atoms is large by the current standards of *ab-initio* Hartree-Fock calculation. The basis sets for the atoms were inherited from the bulk system, but again the coefficients needed to be optimised for the surface structure. In this case the optimal coefficients were not so different from the starting values, which was fortunate since each step of the optimisation now took much longer. The Mulliken populations for the atoms in the centre layer of the surface model were compared with those for atoms in the bulk model. The populations were considered sufficient evidence that the middle layer of the model was largely unaffected by the surface, so the model needed no further layers of atoms.

5.3.5 Adsorbate Molecule

One of the primary limitations of the Hartree-Fock approximation is that it ignores electron correlation and no account is made of induced dipoles in

the calculation. This rules out the use of most of the simplest molecules which could be used to explore the surface potential. Dihydrogen, dihalides, dinitrogen and dioxygen are useless, as are the noble gases. Some initial work using a single argon atom as an adsorbate demonstrated the problem. The energy of the system was calculated at a sequence of points as the argon atom was moved away from the surface. The curve traced by this process showed that the energy only decreased as the atom-surface distance increased; there was no energy increase associated with an attractive force on the argon atom.

The two most obvious possibilities for simple dipolar molecules are hydrogen fluoride and carbon monoxide. Carbon monoxide was chosen in this case because it is more likely to be used in an experimental study of oxide surface adsorption, with which the results of calculation could be compared. The basis sets chosen for the CO molecule were standard $6-21G\star$ molecular basis sets, and were optimised for the free molecule within the *Crystal92* program. For the purposes of this work the bond distance was fixed at an idealised value of 1.13859Å.

5.3.6 Adsorbate and Surface System

The complete surface-adsorbate system was constructed by simply inserting atom coordinates for the adsorbate into the surface structure, and including the basis set details for the molecule in the input file. The CO molecule in vacuo is cylindrically symmetric, so there are five degrees of freedom for the orientation of a CO molecule with respect to the surface. The surface unit

has $2mm$ symmetry, but the actual symmetry of the system is dependent on the location and orientation of this molecule. The first positions explored were high-symmetry sites, with the molecule oriented perpendicular to the (110) plane (i.e. normal to the surface), and located above the centre of the surface unit. To preserve the symmetry of the whole slab model, a second CO molecule was added below the surface at a site related to the first by a reflection in the $z = 0$ plane. Figures 5.6 and 5.7 illustrate the geometry of this model.

The symmetry of the surface cell means that only one quarter of the surface unit needs to be explored, although the z-coordinate is open ended. Higher symmetry sites are, as expected, computationally less expensive than general position, so any site lying on one of the two-fold rotation operations is least expensive, followed by sites lying on one of the mirror planes, and finally all of the general positions are most expensive.

The *Crystal92* package is split into three main components; integral calculation, self-consistent field minimisation, and properties computation. In these terms, the symmetry, and hence overall complexity of the system, affects the integral calculation most strongly. Less symmetry requires that more separate overlap integrals must be computed and stored. Thus a general CO position in the Rutile-CO system requires four times as many integrals to be calculated as the highest symmetry (0.5,0.5,z) site. Up to 4 gigabytes of hard disk space was required for storage of integrals for these general positions of this system, which are then available for the SCF part

of the program. An alternative version of the program which is able to calculate the integrals afresh for each step of the SCF calculation was found to require excessive memory resources and time, even when the program was parallelized across several machines.

The surface asymmetric unit (that part which is not related by symmetry to another part of the surface cell) was covered in steps of 0.1 fractional units, that is 0.649Å in x and 0.296Å in y. The aim was to explore the full cell in a relatively low-resolution manner, and fill in with smaller steps if time permitted. The volume above the surface was planned to be explored in steps of 0.2Å, starting at an z-coordinate for the CO carbon of 4.0Å from the cell origin (the cell origin in z is the central layer of the slab). This starting distance is fairly arbitrary, but when the carbon atom is at (0.5,0.5,4.0) the nearest neighbours are O_4 at 2.21Å, O_2 at 2.21Å and Ti_4 at 1.66Å. A theoretical study of H_2O adsorbed on TiO_2 by Jug *et al.* [36] suggested that the oxygen atom of an adsorbed water at this site was between 1.8Å and 2.05Å from the titanium atom.

5.3.7 Results and Discussion

The CO molecule was tried in two off the possible orientations relative to the surface. Both had the CO axis perpendicular to the surface (110) surface plane, but one orientation was with carbon closest to the surface (referred to as TiCO), and the other had the oxygen closest to the surface (referred to as TiOC.)

The first set of results were obtained for the TiCO system, and demonstrated that at a z-coordinate of 4.6Å the x and y coordinate made very little difference to the energy of the system. On the basis of this it was assumed that the CO molecule was effectively not interacting with the surface. To provide a useful measure of the energy of absorption, the energy value at this distance was subtracted from the other results, and it is this relative energy which has been plotted in the figures.

With reference to figure 5.8, it can be seen that there are channels for the CO molecule to approach the surface, starting above the centre of the surface cell, and slipping sideways toward the Ti sites on the surface. The volume directly above the surface O sites is strongly repulsive at z-coordinates less than 4.0, while further away from the surface the energy range is around 2 hartree.

The view from below (figure 5.9) shows where the CO molecule heads for, and the channel is still visible as a region of "daylight" through the isosurface representation.

The countour maps (figures 5.10 - 5.12) show that sites above the Ti atoms (**X**) are unfavourable compared with the infinite distance energy, whereas **A** and **B** are favourable; arund 1 Hartree lower in the case of **A**. **B** turns out to be a dead end, although it does provide a pathway or lateral movement across the surface. Approaching the surface, (figure 5.11) the CO molecule is directed through sites **C** towards low energy sites **D** close to the oxygen atoms, with Ti atoms sharply repelling the approach to site **Y**. There is a

small energy barrier, around 0.15 Hartree, to movement from **C** to **D**. At closer approeach (figure 5.12), the bulk of the O atoms **Y**,**Z** pushes the CO into the favourable sites **E**. The cross-section through O-O in figure 5.13 shows that the is a low-energy path to the inner adsorption sites, delimited by the zero-energy (relative to CO in vacuum) contour coloured white.

In contrast to the carbon-down attitude, adsorption of the CO molecule approaching oxygen-down (TiOC) is less favourable, and the zero-energy iso-surface (figure 5.14) closes any paths to the surface from the non-interacting region. The channels open up at an energy of some 1.5 hartree above zero, giving a measure of the energy barrier to be overcome. The view from below (figure 5.15) shows that the absorbtion sites are likely to be very similar to those for the TiCO, which is consistent with the fact that carbon monoxide has a small dipole moment. It is possible that the source of the energy barrier is to do with a change in the dipole moment of the CO molecule induced by the approach to the surface.

Site **A** in figure 5.16 is favourable by around 1 Hartree compared to the infinite distance energy, but to progress to the very favourable site marked **C** in figure 5.18 it must overcome an energy barrier of around 1.9 Hartree - that is site **B** in figure 5.17 is around 0.9 Hartree higher in energy than a site at infinite distance.

Movement across the surface is likely to follow the diagonal path from **A** to another **A** site in the neighbouring cells, avoiding the unfavourable regions marked **X** and **Y** in 5.16.

The visualisation of results for these experiments requires that information about a densely-populated volume of data be represented in a simple graphic. This has been extremely difficult to achieve, and the figures in this work are considered the best compromise.

The images were generated using the AVS program, using a sequence of program modules to process the raw data into the form of a three-dimensional array of values. The data was first read into the program using the File Descriptor module, then interpolated using a tri-linear algorithm to expand the dataset to four times it's original size in each direction. This smoothes the sharper edges of the data, and when fed into the Isosurface module a smooth surface is produced. The data were also filtered through a thresholding module, which allows the user to select upper and lower bounds for the data to be represented. Values above the upper bound take the value of the upper bound, and values below the lower bound take the lower bound value. The result of this filter was then represented as an array of spheres on the data points, each with radius determined by the value at that point. For the main figures, the lower bound was set to zero, since a zero-sized sphere is invisible.

The main pictures show an oblique view of the Rutile (110) surface of the titanium dioxide, with the atoms represented as spheres of approximately half their van der Waals radii. Above this is shown the volume probed with the carbon monoxide molecule. The grey surface is the isosurface joining the points where the energy of the interaction is zero. The blue spheres show the points where the energy of the interaction is positive (unfavourable), and the

clear space represents the areas accessible to the adsorbate molecule.

The Hartree-Fock Self Consistent Field calculation method has a number of limitations, in part due to the approximations made to enable the method to be used in real systems. Firstly, only ground-state systems are calculated because the method does not handle open electronic shell well. Also electronic correlation is neglected. This has the effect of ignoring any induced-dipole – induced-dipole interactions. Finally, there is the problem of the Basis-Set Superposition Error (BSSE) (see, for example, [37] and references therein), which is due to overlapping basis functions from neighbouring atoms, and leads to an under-estimation of the overall energy of the system. Allouche notes [38] in a recent study of NaCl with acetylene that the correlation and BSSE errors may be quite small, and possibly negligible, in periodic systems.

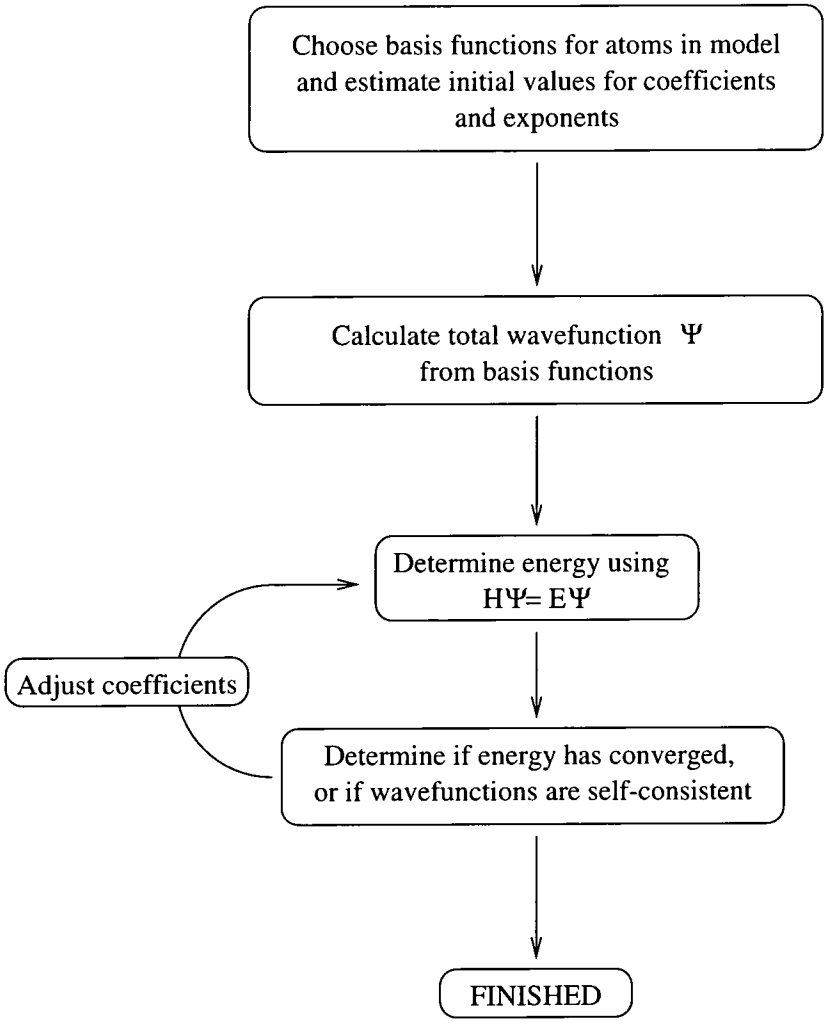


Figure 5.3: Flow diagram outlining HF/SCF calculation process

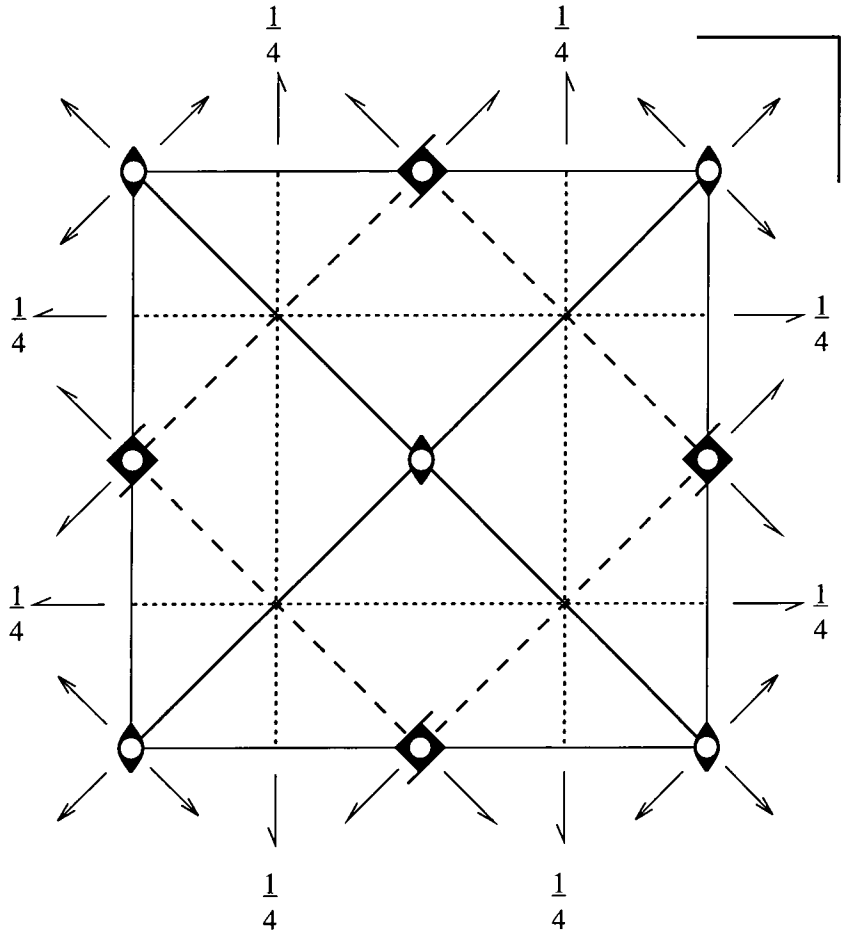


Figure 5.4: Space group diagram for $P4_2/mnm$

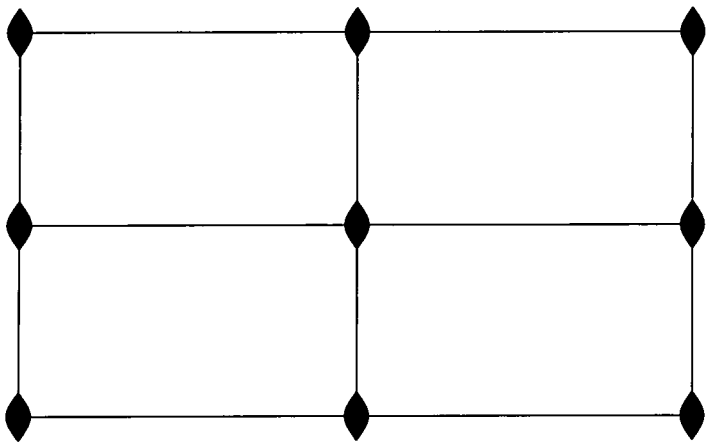


Figure 5.5: Plane group diagram for $p2mm$

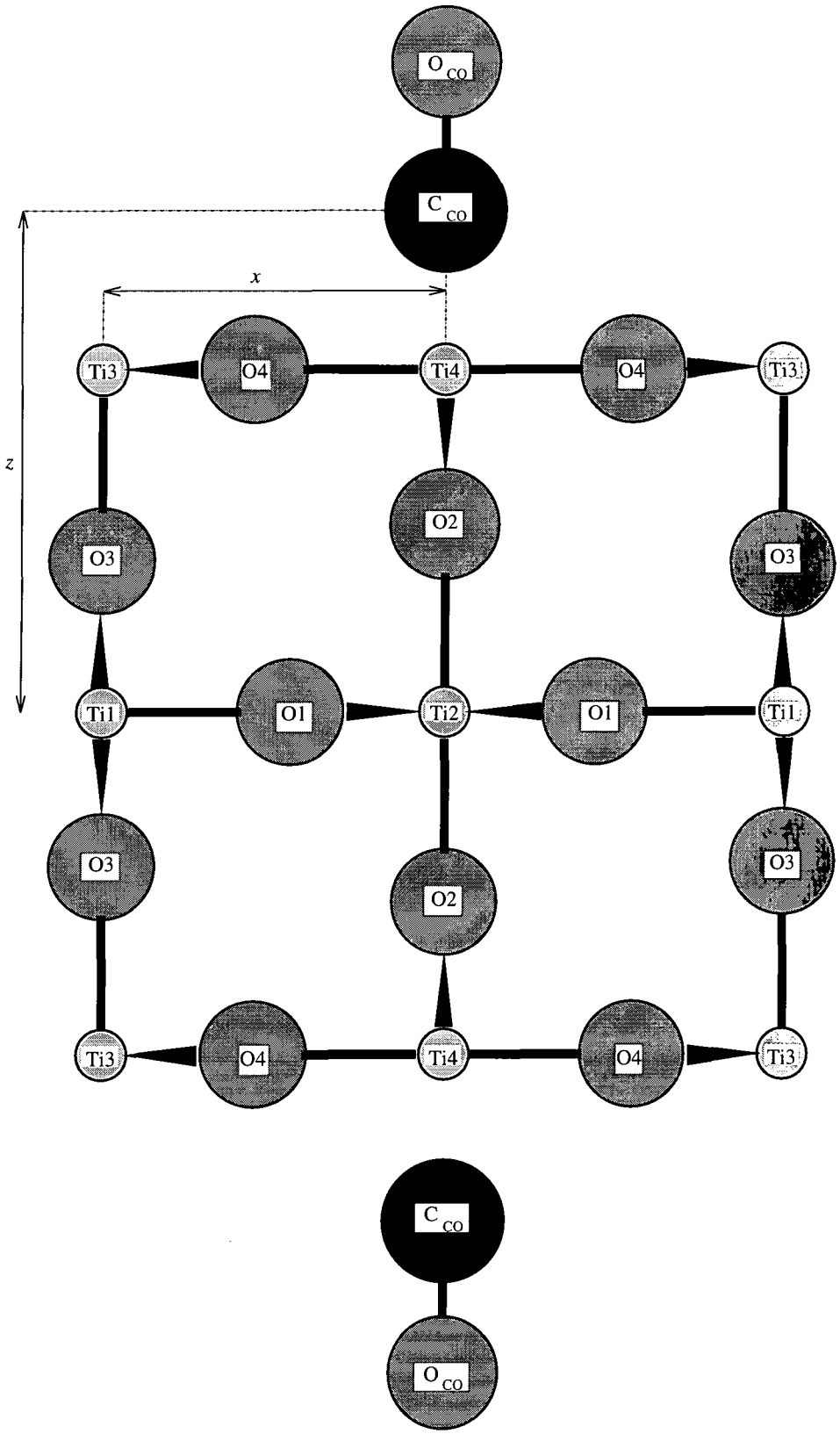


Figure 5.6: Cross-section of surface-adsorbate system as modelled

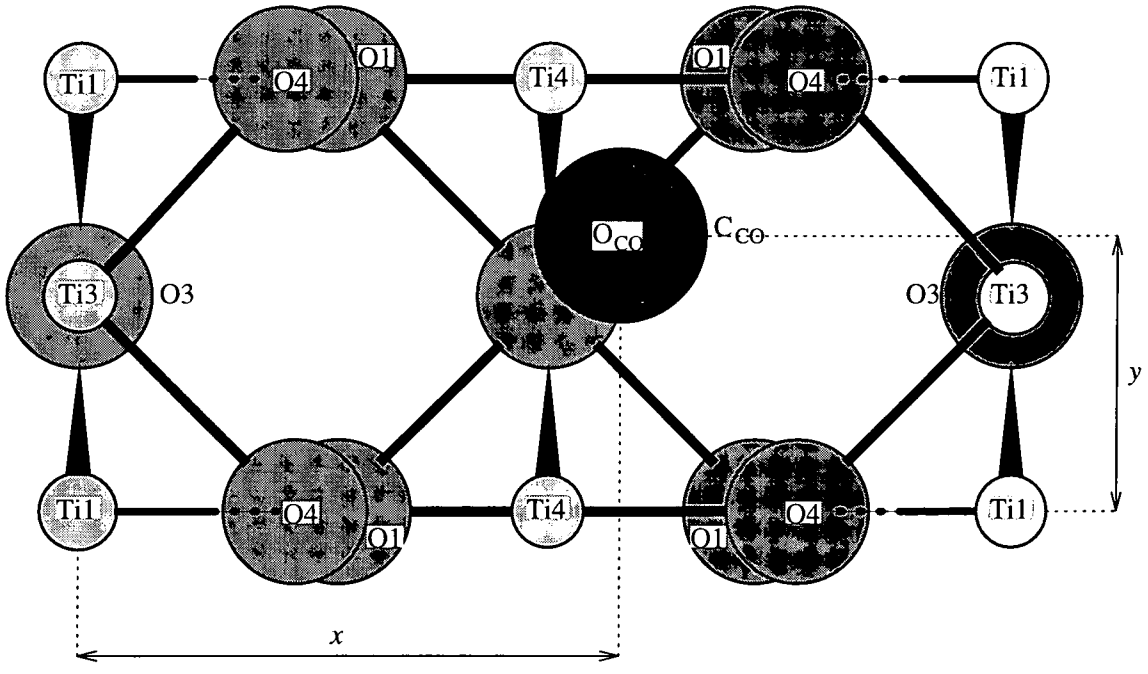


Figure 5.7: Projection of surface-adsorbate system along (110) vector

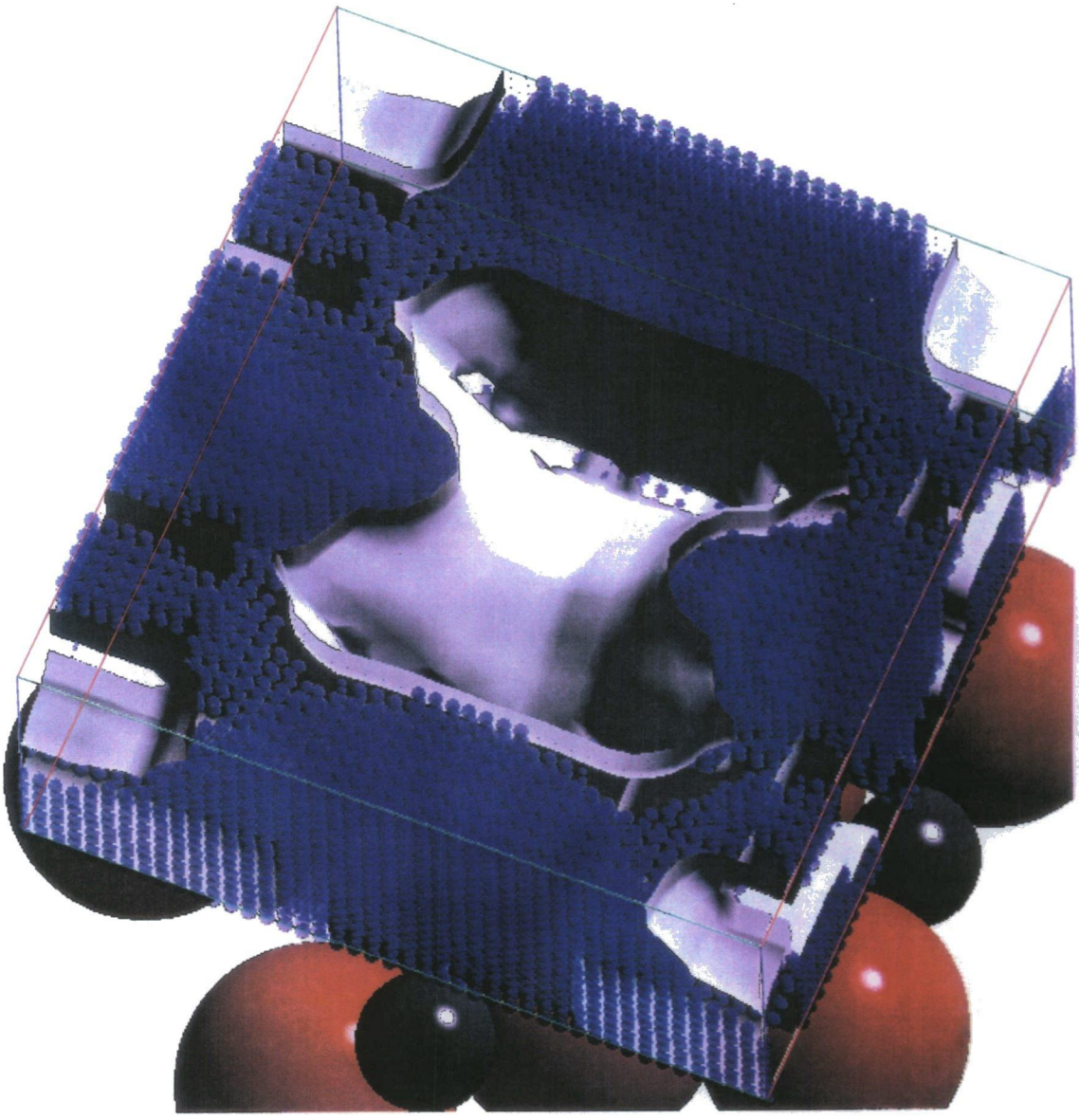


Figure 5.8: Illustration of energy hypersurface above TiO_2 (110) surface, with carbon closest to surface plane. Blue volumes are regions of positive energy, and surface represents the zero adsorption-energy isosurface. Red spheres are O atom positions and grey spheres are Ti atom positions in the surface.

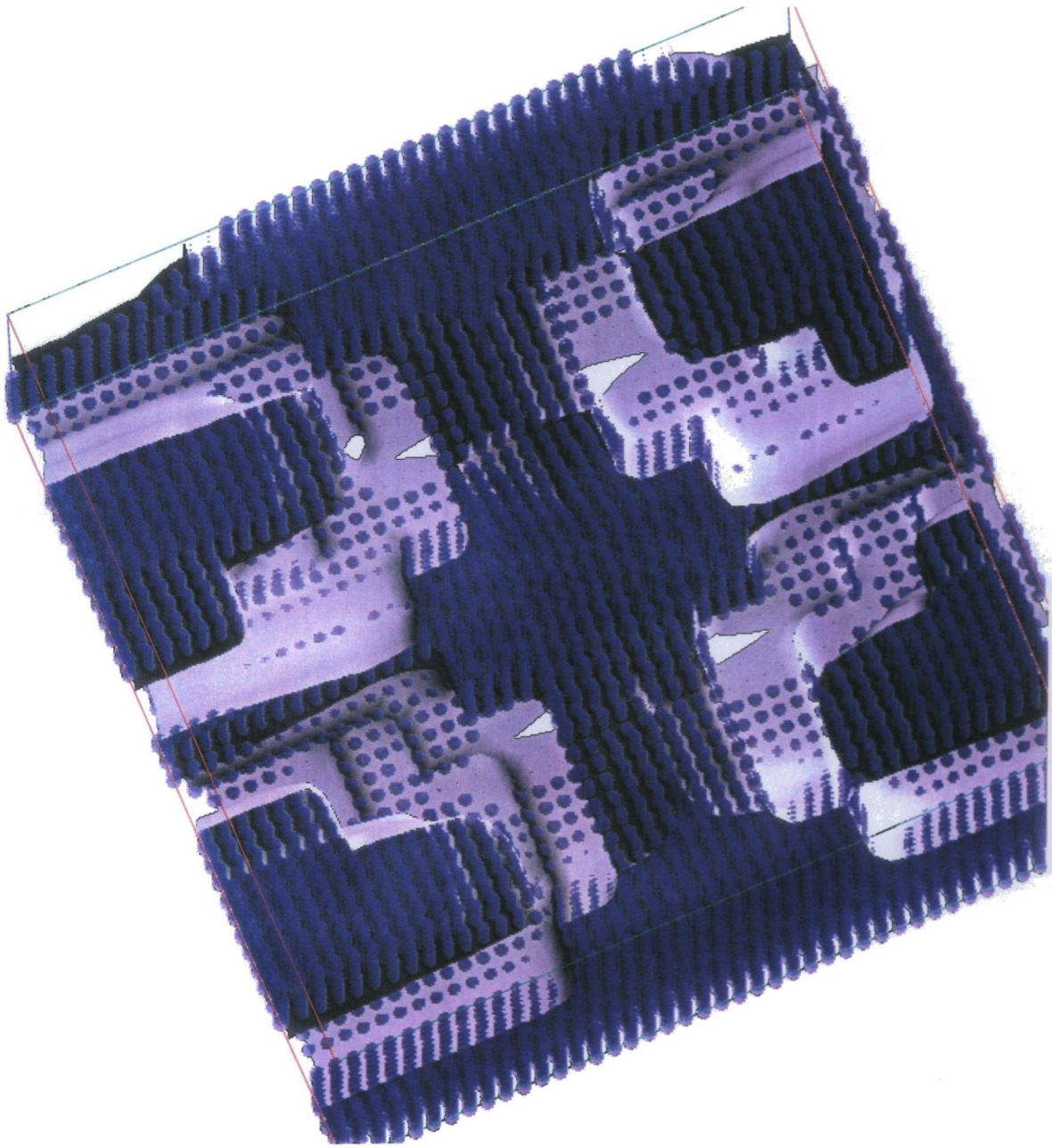


Figure 5.9: Illustration of energy hypersurface above TiO_2 (110) surface with carbon closest to surface plane, from the viewpoint of the surface itself. The atom positions have been omitted.

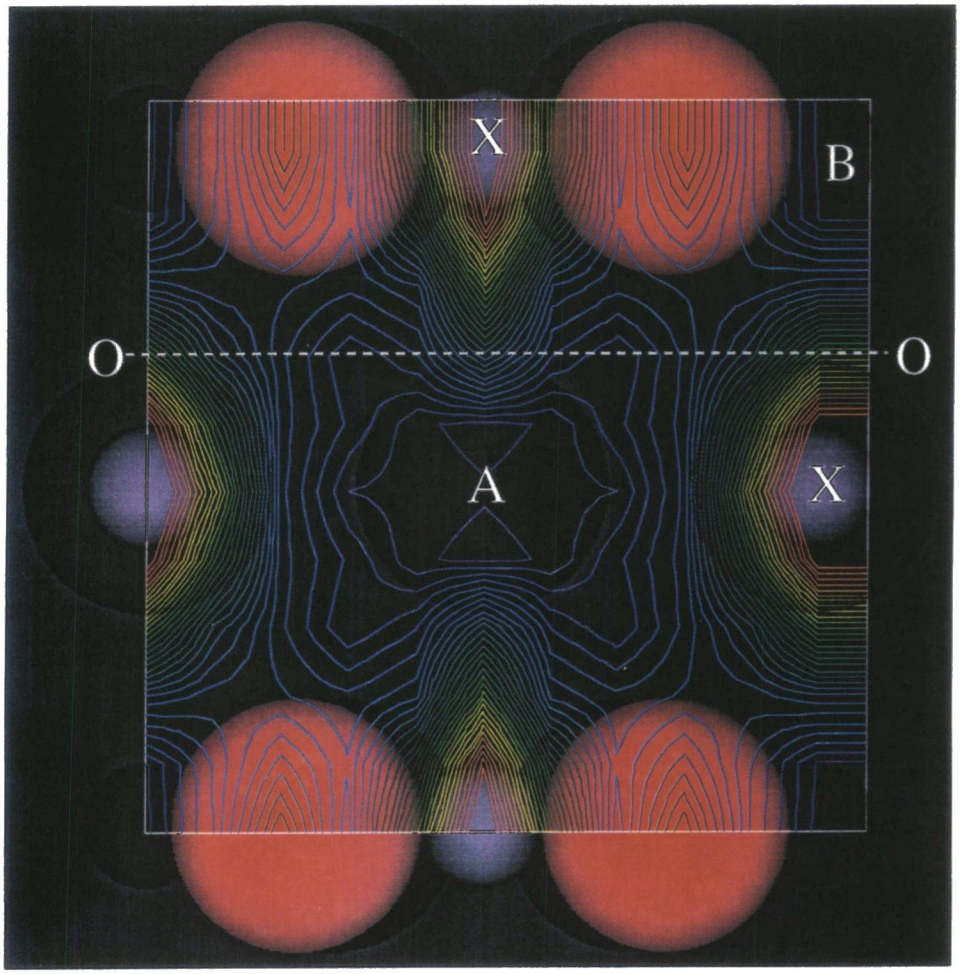


Figure 5.10: Contour slice of energy hypersurface above TiO_2 (110) surface with carbon closest to surface plane, at a distance of 1.555Åfrom the surface layer.

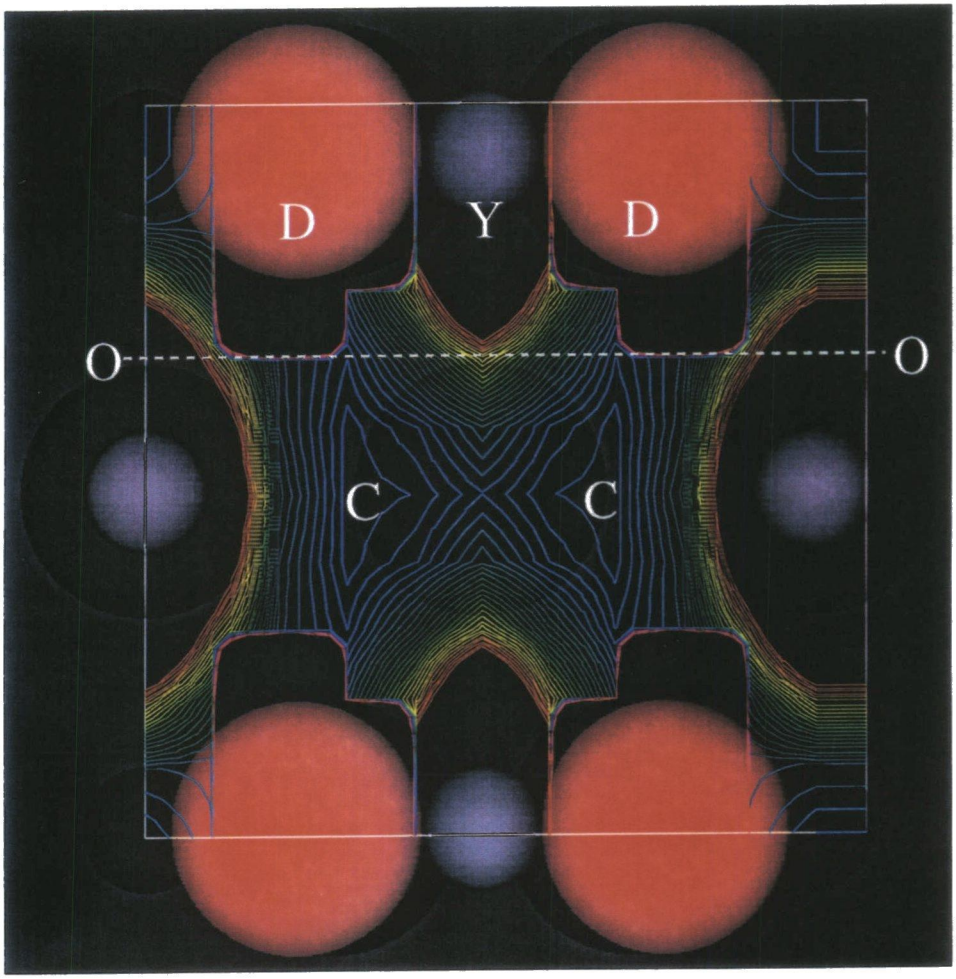


Figure 5.11: Contour slice of energy hypersurface above TiO_2 (110) surface with carbon closest to surface, at a distance of 0.955Åfrom the surface plane.

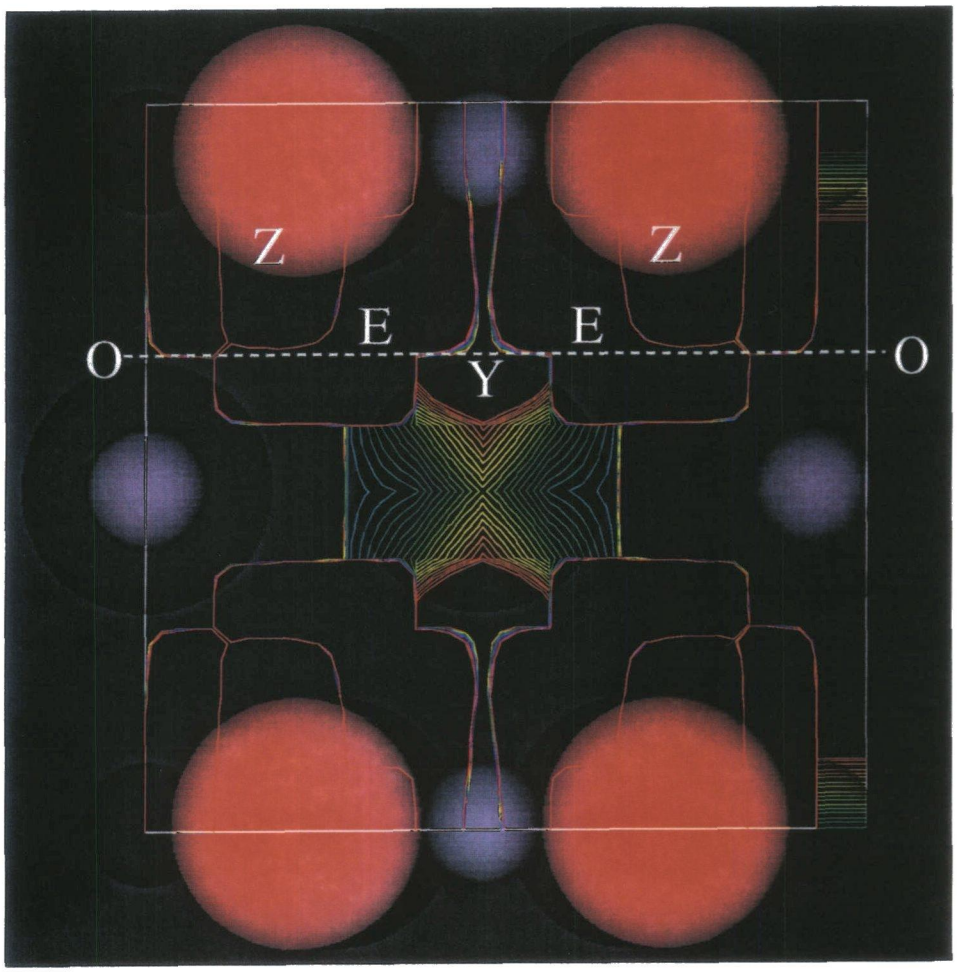


Figure 5.12: Contour slice of energy hypersurface above TiO_2 (110) surface with carbon closest to surface, at a distance of 0.355Åfrom the surface plane.

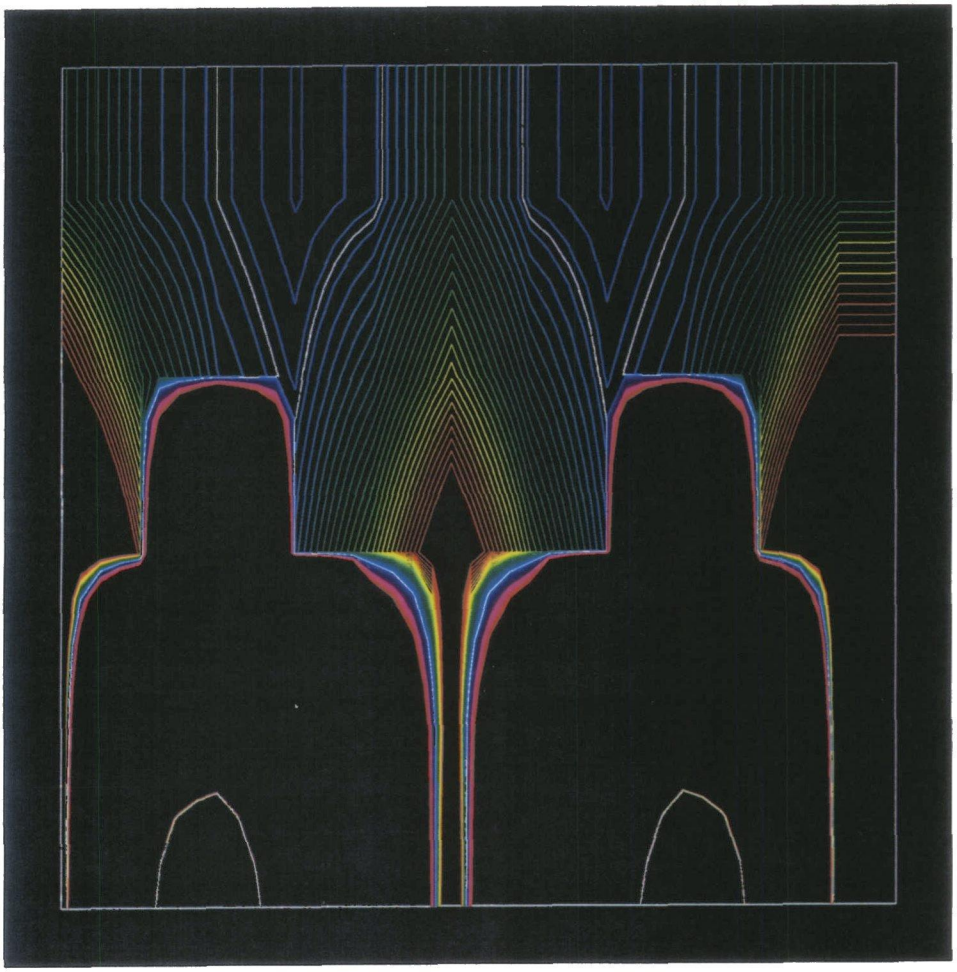


Figure 5.13: Contour plot perpendicular to surface through section O-O in figures 5.10 - 5.12. The white contour marks zero energy relative to CO at infinite distance

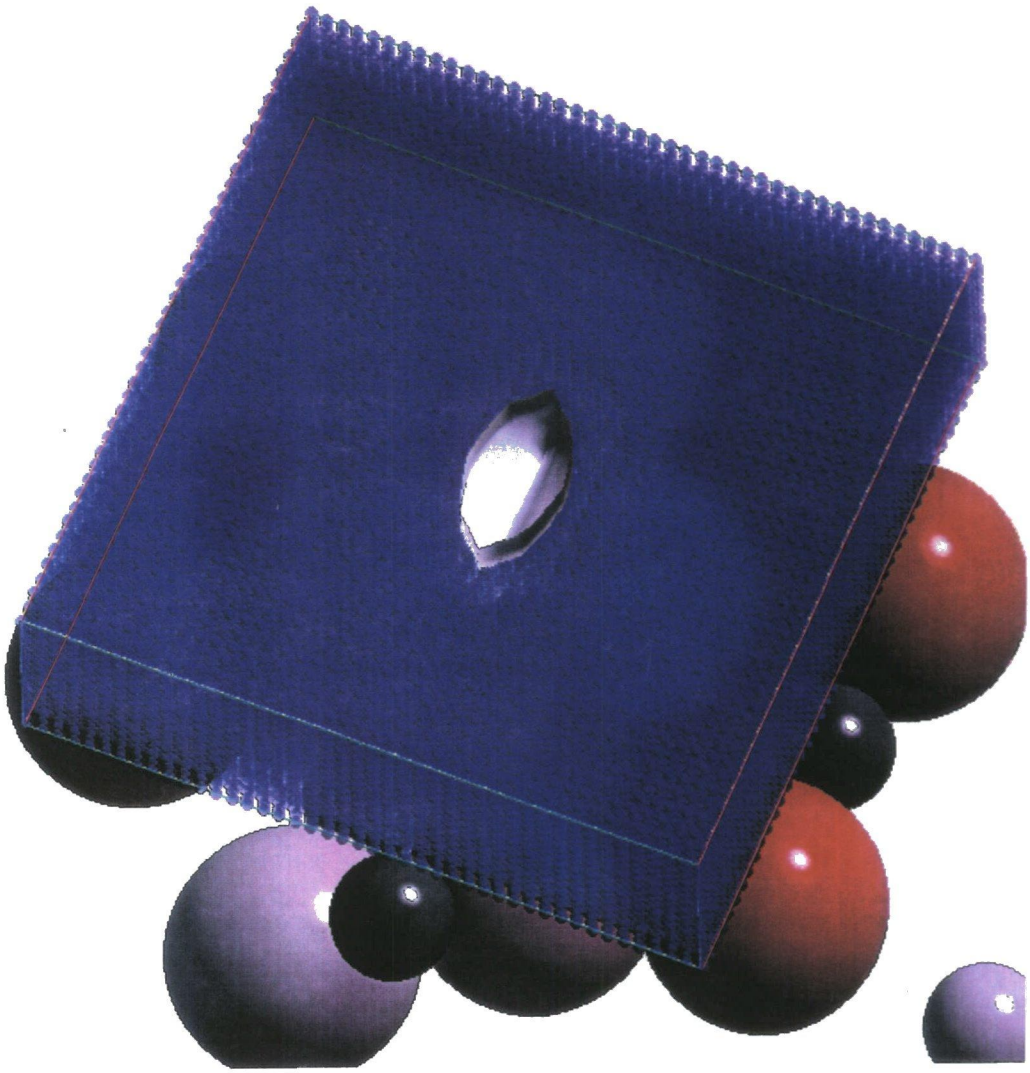


Figure 5.14: Illustration of energy hypersurface above TiO_2 (110) surface, with oxygen closest to surface plane. Blue volumes are regions of positive energy, and surface represents the zero adsorption-energy isosurface. Red spheres are O atom positions and grey spheres are Ti atom positions in the surface.

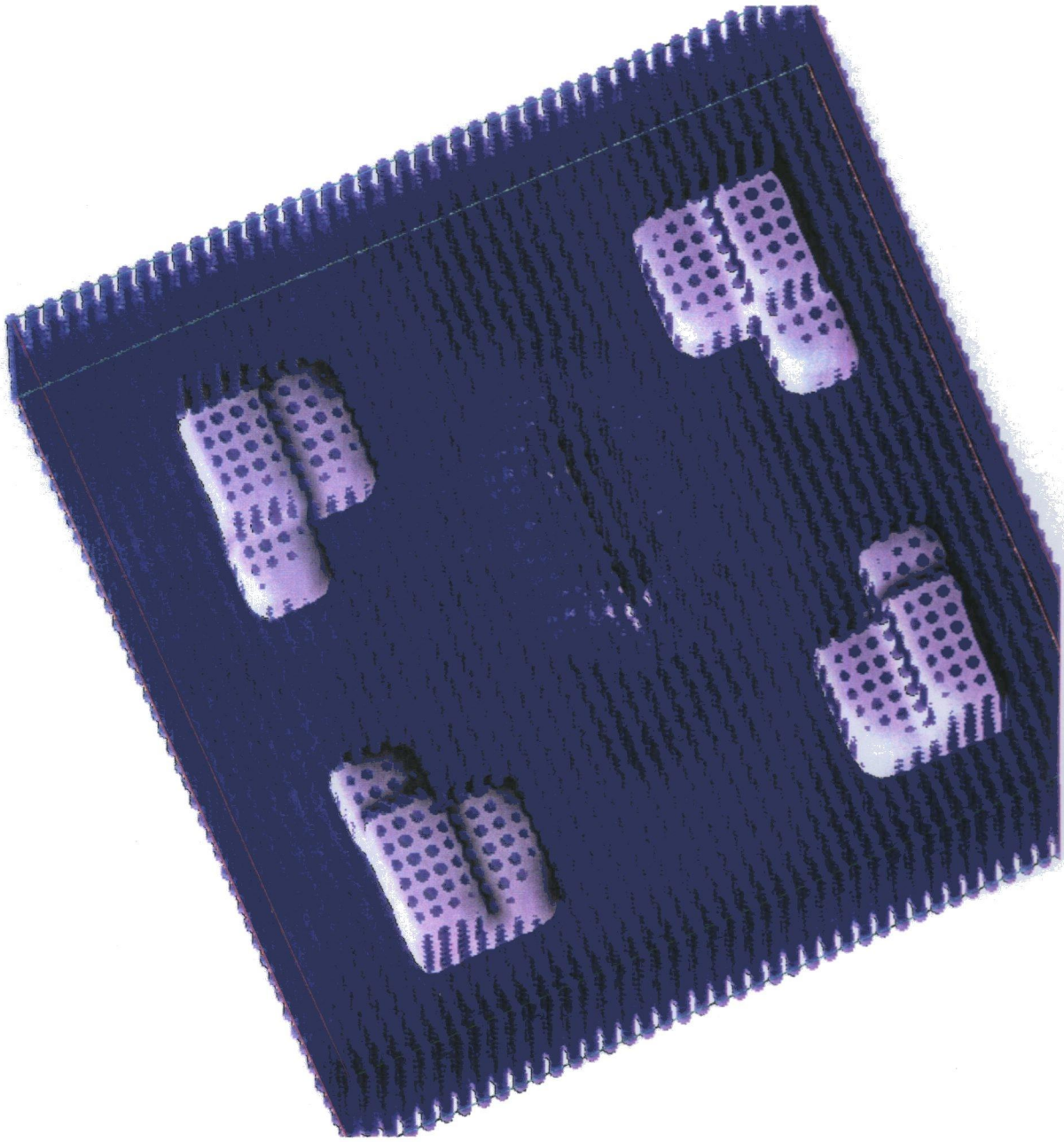


Figure 5.15: Illustration of energy hypersurface above TiO_2 (110) surface with oxygen closest to surface plane, from the viewpoint of the surface itself. The atom positions have been omitted.

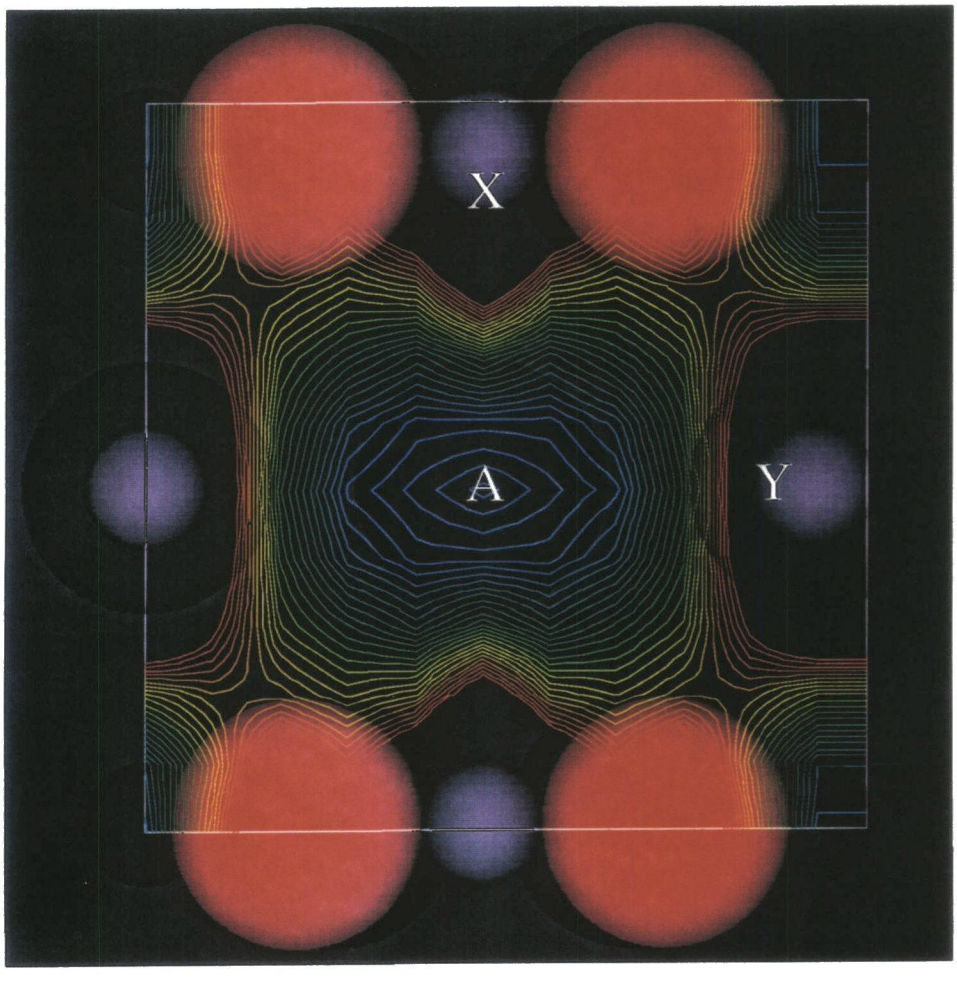


Figure 5.16: Contour slice of energy hypersurface above TiO_2 (110) surface with oxygen closest to surface plane, at a distance of 0.955Åfrom the surface layer.

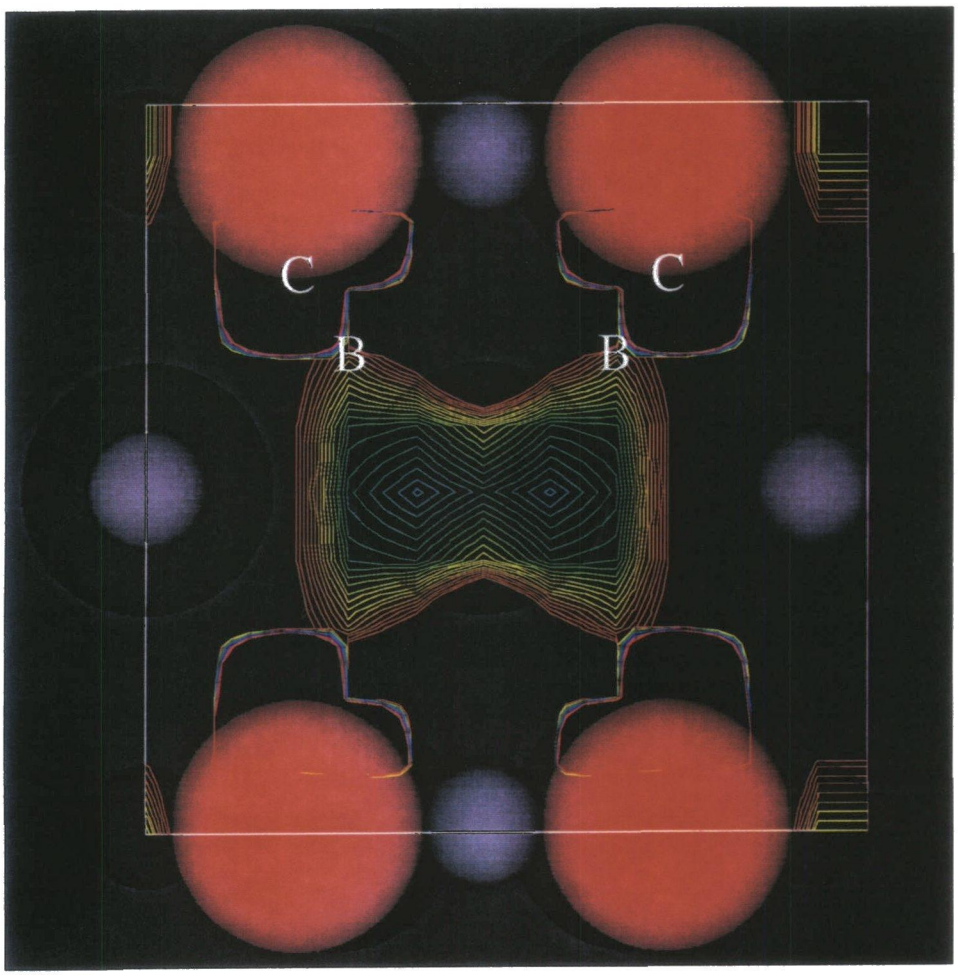


Figure 5.17: Contour slice of energy hypersurface above TiO_2 (110) surface with oxygen closest to surface plane, at a distance of 0.535Åfrom the surface layer.

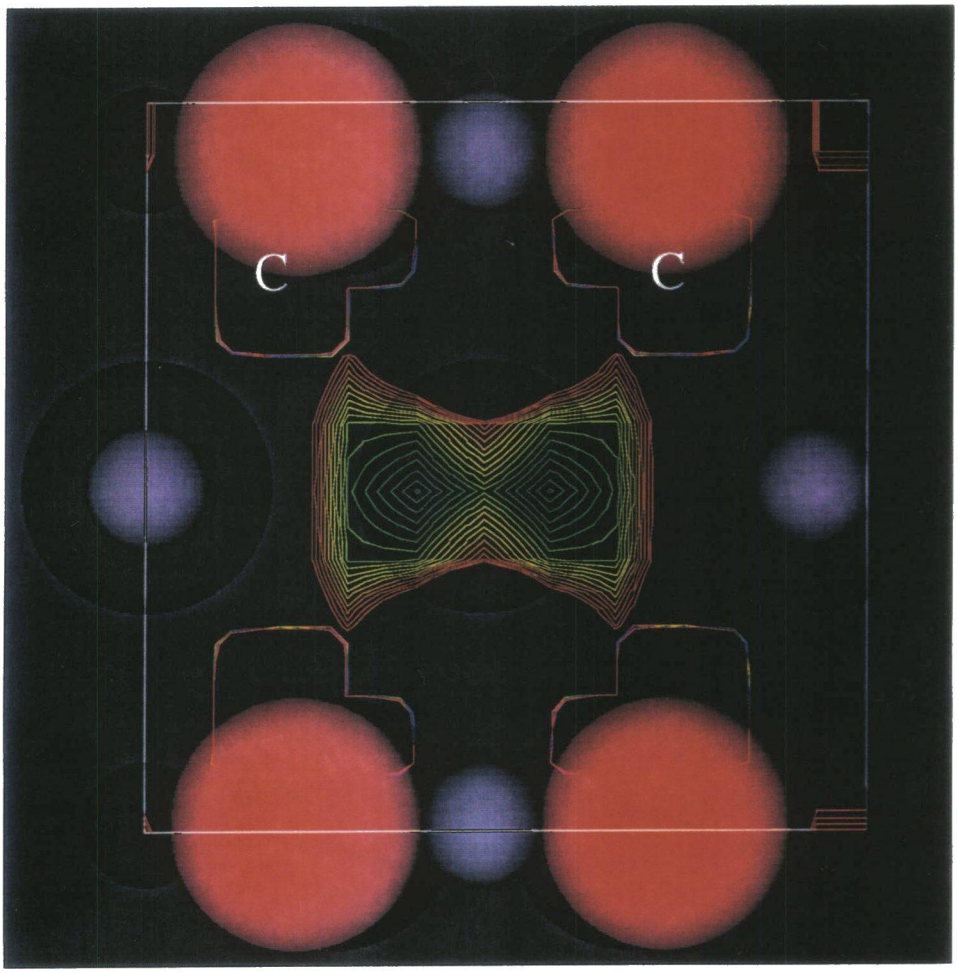


Figure 5.18: Contour slice of energy hypersurface above TiO_2 (110) surface with oxygen closest to surface plane, at a distance of 0.355Åfrom the surface layer.

Chapter 6

Visualisation of Data

6.1 Introduction

The use of computers in theoretical and experimental chemistry has resulted in the production of large amounts of numerical data, describing molecular structures, electron distribution, and a whole range of physical properties. Making use of this data requires it to be presented in such a way as to make sense to the scientific community - a conclusion is useless if no-one else can see what it means.

Graphical representations have long been the norm for picturing molecular structures as ball-and-stick models, and the so-called ORTEP diagrams are commonly used to highlight the thermal motion parameters for atoms in a structural model. Molecular modelling on computers relies on graphical means of constructing models, and most programs allow measurements to

be made on the model simply by selecting atoms or bonds on the computer screen. In addition, colour-coded and wire mesh representations of properties such as electronegativity and solvent-accessible surfaces can be superimposed on the traditional ball-and-stick model.

While ball-and-stick computer graphics could be considered a high-tech replacement for three-dimensional models made from beads and wire, some things are much more difficult to construct in real life. The application of both X-ray diffraction and *ab-initio* calculations to determining the charge density of a structure calls for a method of displaying the results diagrammatically. While it is possible to get a feel for the electron density around simple molecules from two-dimensional slices displayed as contour maps, a three-dimensional isosurface model can show the whole molecule in one image. Add to this the potential for rotating the model in real time, and peeling away layers to reveal internal structure, and the computer model becomes very appealing.

The main drawback of advanced three-dimensional computer modelling is that it is difficult to represent away from the computer screen. Publishing in journals, and presenting at lectures or on posters, require two-dimensional projections of the model. A few methods have been employed to portray three-dimensional models as stereographic images capable of introducing an impression of depth. These methods provide a separate image for each eye and a means of fooling the brain into believing that these two images are in fact the same. Two common methods are cross-eyed stereograms and red-

green stereo images. Cross-eyed images require the viewer to cross his eyes in order to focus on two separate images, at which point the sensation of depth becomes apparent. Red-green stereo superimposes the two images, with one printed in red and the other in green. The viewer then uses spectacles with coloured filters to ensure that one eye sees the red image and the other eye sees the green image, which the brain merges into one picture. These methods are very good a representing line drawings, but viewing stereo images of surfaces is not so easy. The method described in Section 6.5 provides a possible solution to this problem.

6.2 *AVS* and X-windows Molecular Graphics

Use of AVS

The market for computer software able to represent molecular structure in a clear and easily manipulated way is already very large. Use of such software by pharmaceutical companies for drug design and evaluation is one of the primary areas for sales, and as such tends to determine the nature of the programs available. Most of the best-selling programs suffer from a combination of high cost, over-specialisation and inflexible input formats. Another common limitation is that such programs tend to be hardware-specific, since they often make use of specialised graphics hardware and software. Since the work detailed in Chapter 5 required visualisation of an inorganic structure unrelated to present drug research, a more general approach was sought.

The AVS program is intended to aid visualisation of a wide variety of data types, through two- and three-dimensional display tools and a range of input methods, filters and output formats. Perhaps the most useful feature of this system is that it is possible to integrate new methods and programs as "modules" which can be connected to the system. A suite of library functions, accessible from both C and FORTRAN programming languages, allow access to all of the data structures AVS is designed to handle, and if necessary create new data structures. It should be noted that the AVS package is not unique in these features; Khoros [39] and Silicon Graphics' Explorer product provide much the same functionality.

In the case of molecular graphics, the supported modules for handling chemical data supplied with AVS were not especially suitable for the present work, since the PDB input format expected is overly complicated and primarily directed a protein molecules. A much simpler input format was developed for a new module, which generated a three-dimensional ball-and-stick model for direct visualisation by the Geometry Viewer module built into AVS. The format requires only the atomic types and coordinates, either fractional coordinates with a crystallographic unit cell, or in orthonormal coordinates in Ångstrom units. The program looks up a suitable radius and colour for representing the atoms, and also uses the van der Waals radius to calculate bonds between them. It then generates a geometry object for AVS to display, allowing the user to rotate, translate and zoom the image to view the model. This precluded the need for coding new graphics and object manipulation

routines, but does limit the program to merely displaying the model, and does not let the user change the model.

Use of X-windows

A related project was intended to assist the visualisation of crystallographic structures during structure refinement work. Although several programs exist for this purpose, they are restricted in their use by being commercial packages released in machine-specific binary form. What was sought was a program to allow users of the public-domain SHELX 93 [10] refinement program to see the structure on screen, and manipulate the structure produced by the program. Such changes would then be written to an instruction file for SHELX-93 for further refinement. Additionally, it would be useful to generate symmetry-expanded models and packing diagrams to look for intermolecular relationships. As such, the program would be essentially similar to the Siemens Analytical X-ray Instruments program, XP [40], but with the major additions of widespread portability using X-windows graphics, and a more intuitive mouse-and-menu driven user interface.

The program, provisionally named Xpete, does not yet include all of the planned functionality, but the essential graphical part of the program is in place. The three dimensional model is projected onto a two dimensional display by a series of matrix operations on the fractional coordinates in the input file. Mouse inputs allow for the rotation of the model about all axes, and selection of atoms for editing their labels or types, deleting or hiding

of atoms in the model. Multiple atoms can be selected by clicking, or from pop-up lists. Entire atom types can be selected and deleted, hidden from view or changed to another type. This is particularly valuable in the first stages of model refinement, where electron density peaks usually designated as type Q can be changed wholesale to carbon, for example. Two modes of display are currently supported. The standard mode represents the atoms as flat circles and is the fastest mode for redrawing the model as it is rotated. The rendering mode makes the atoms appear as shaded spheres, which helps to overcome some ambiguities in complicated models, but slows redrawing of the model.

6.3 Development work on *builder2* using AVS

The *builder2* program is intended to be a part of a larger package which includes a powerful graphical user interface. Owing to the way this interface has been written, it currently only operates on Silicon Graphics computer systems. Such a system was not available during the development of the *builder2* program, and so an alternative method of visualising the results was required. Again, the AVS package formed the basis of the display system, but was also used to provide the input values needed to define the surface. AVS provides functions which generate graphical input devices for integers, real numbers and strings. The can be included in user-defined modules to allow the module to accept the information required to run the program.

In this way a user-friendly interface can be built for an application, in the form of an AVS module 'wrapper' around the application program. The wrapper handles the input and output form the underlying program, and the program itself can be developed in this environment without large-scale changes to the program code itself. The *builder2* program requires the user to input a filename for the input data, a miller index, thickness and extents for the model, and flags to switch on zero-overall-charge and zero-dipole-moment requirements for the model; see 4.2.

6.4 Electron density surfaces

One of the fastest-growing areas of crystallographic research is the study of the charge density of a molecule in the crystalline form. Charge density studies use X-ray diffraction at very high resolution to get information on how the electronic charge is localised within the crystal structure. In contrast to the common forms of X-ray structure models which are represented by balls and sticks, charge density experiments yield a three-dimensional grid of values corresponding the charge at each point. In addition, calculation of the Laplacian of the charge density

$$\nabla^2\rho = \frac{\partial^2\rho}{\partial x^2} + \frac{\partial^2\rho}{\partial y^2} + \frac{\partial^2\rho}{\partial z^2}$$

gives important information about bonding and non-bonding interactions

```
 * MAXX(f), MAXY(f) and MAXZ(f) give the size of the field f
 * in each direction.
 * I2D(f,i,j) references the field element f(ij) (2 dimensions)
 * and I3D(f,i,j,k) references the field element f(ijk) (3 dimensions)
 *
 * The convention for AVS modules is to return 1 for success and 0 for failure
 */

int Laplacian_compute( Field_In, Field_Out)
AVSfield_float *Field_In;
AVSfield_float **Field_Out;
{
    int dims0[2], dims1[3], sx, sy, sz, i, j, k, n;
    float avge;
/*
 * Store the size of the field
 */
    sx = MAXX(Field_In);
    sy = MAXY(Field_In);
    sz = MAXZ(Field_In);
    printf ("%d %d %d\n",sx,sy,sz);
/*
 * If sx or sy are zero, then the field is one-dimensional (or worse)
 * so return failure
 */
    if (sx == 0 || sy == 0)
    {
```

```
        AVSerror("Input field not two or three dimensional.");
        return(0);
    }

/*
 * If a space is already allocated (from a previous run) then clear it
 */
    if (*Field_Out) AVSfield_free(*Field_Out);
/*
 * If the third dimension has zero size, treat as a 2D array
 */
    if (sz == 0)
    {
        printf("2D field\n");
        dims0[0] = sx;
        dims0[1] = sy;
/*
 * Allocate space for 2D output field
 */
        *Field_Out = (AVSfield_float *) AVSdata_alloc(
                "field 2D 2-space 1-vector float",dims0);
        if (*Field_Out == NULL)
        {
            AVSerror("Allocation of output field failed.");
            return(0);
        }
```

```
/*
 * use 5-point approximation - avoid problems with exceeding array
 * bounds by using fewer points at field edges
 */
      for (j=0; j<dims0[1]; j++)
      {
         for (i=0; i<dims0[0]; i++)                                    70
         {
            avge = 0.0; n = 0;
            if (i > 0)
            {
               avge += I2D(Field_In,i-1,j);
               n++;
            }
            if (j > 0)
            {
               avge += I2D(Field_In,i,j-1);                            80
               n++;
            }
            if (i+1 < dims0[0])
            {
               avge += I2D(Field_In,i+1,j);
               n++;
            }
            if (j+1 < dims0[1])
            {
               avge += I2D(Field_In,i,j+1);                            90
```

```
                n++;
            }
            I2D(*Field_Out,i,j) = I2D(Field_In,i,j) - avge/n;
        }
    }
}

/*
 * input is in three dimensions
 */                                                          100
  else
  {
    printf("3D field\n");
    dims1[0] = sx;
    dims1[1] = sy;
    dims1[2] = sz;
/*
 * Allocate space for a 3D output field
 */
    *Field_Out = (AVSfield_float *) AVSdata_alloc(           110
            "field 3D 3-space 1-vector float",dims1);
    if (*Field_Out == NULL) {
        AVSerror("Allocation of output field failed.");
        return(0);
    }
    printf("Data allocated\n");
```

```
/* use 7-point variant of 5-point approx. */
for (k=0; k<dims1[2]; k++) {
for (j=0; j<dims1[1]; j++) {                                    120
for (i=0; i<dims1[0]; i++) {
avge = 0.0; n = 0;
if (i > 0) {
avge += I3D(Field_In,i-1,j,k);
n++;
}
if (j > 0) {
avge += I3D(Field_In,i,j-1,k);
n++;
}                                                               130
if (k > 0) {
avge += I3D(Field_In,i,j,k-1);
n++;
}
if (i+1 < dims1[0]) {
avge += I3D(Field_In,i+1,j,k);
n++;
}
if (j+1 < dims1[1]) {
avge += I3D(Field_In,i,j+1,k);                                  140
n++;
}
if (k+1 < dims1[2]) {
avge += I3D(Field_In,i,j,k+1);
```

```
n++;
}
I3D(*Field_Out,i,j,k) = I3D(Field_In,i,j,k) - avge/n;
}
}
}                                                          150
}
/*
* Output field is passed from module when the routine returns
*/
      return(1);
   }
```

The Laplacian routine has been compared with analytical Laplacian results and has been demonstrated to produce reasonable results despite the approximation. The advantages of this method are the applicability to data which cannot be differentiated analytically, and that this approximate method is much faster and computationally less expensive than an analytical method. examples....

6.5 Autostereograms for three-dimensional visualisation

Autostereograms are also known as Single Image Random Dot Stereograms, which is a much more accurate description of the technique. To create a convincing three-dimensional image, the viewer's two eyes need to see slightly different aspects of the object, just as in normal vision the separation of his eyes causes them to see slightly different pictures. The viewer's brain is equipped to merge the two images into a scene containing solid objects.

An Autostereogram image combines the two pictures required into a single image, which in general does not appear to contain any information at all. Most of these images are a mass of randomly coloured dots, with no apparent structure, although careful inspection might reveal a repeating pattern across the image. In fact, a more random the dot pattern will improve the viewing of the actual object.

To see the object in the picture, the viewer must cause his eyes to diverge from the normal focusing angle which causes the image to appear as a flat surface. In doing so, his eyes can refocus on the pseudo-repeating pattern of the dots, which will fool his brain into perceiving depth in the image. Many methods of actually achieving this divergence have been described [41], and with practice most people are able to see the third dimension in the image. There are a few obvious limitations; the small proportion of people who have sight in only one eye, or have certain kinds of astigmatism will not see

the image correctly. Long-sightedness may often be an advantage, since the image is supposed to be focused as if it were further away than it actually is.

The principle is that any point on the surface of a three-dimensional object can be projected onto two points on a flat plane, each point lying on a line from one eye to the object. When constructing the image, the two points in the image which correspond to a point on the object must be given the same colour; it is not important what that colour is.

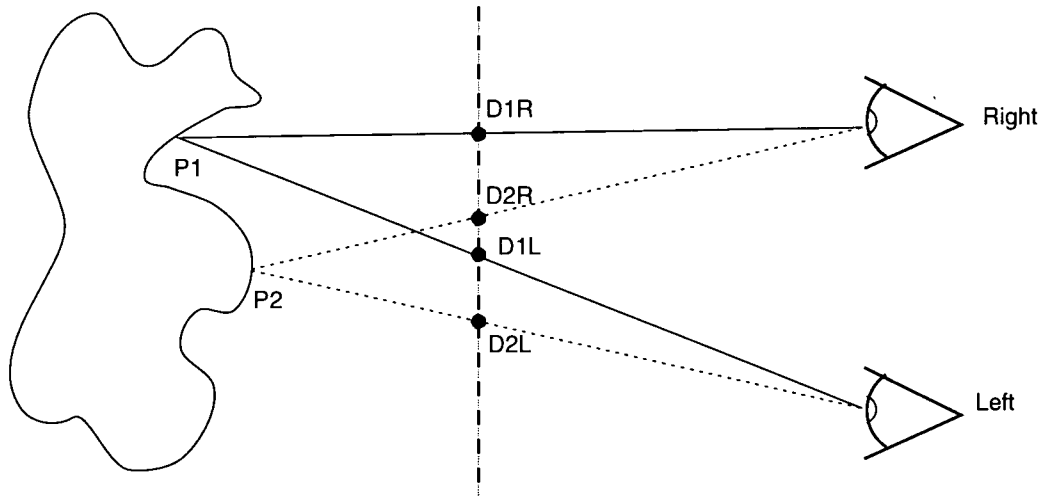


Figure 6.1: Principles of Stereographic Projection

Referring to figure 6.1, point $P1$ on the surface of the object is projected onto the viewing plane at points $D1R$ and $D1L$. The distance $D1R$ - $D1L$ is related to the distance of point $P1$ from the viewing plane. In the construction of the stereogram, a colour must be chosen for the pixels drawn at $D1R$ and $D1L$. Similarly, for point $P2$ on the plane, the points $D2R$ and $D2L$ are the projection onto the view plane of the point. The distance $D2R$

- $D2L$ will be different to $D1R$ - $D1L$ since $P2$ is at a different depth. If, for example, point $D2R$ coincided with point $D1L$, then the colour chosen for all four points would have to be the same, ensuring that the pairs of points have the same colour. Since it is not important for the stereo effect what this colour is, then this is not a serious limitation. In the most simple form a random choice of either black or white is used, while in commercially-produced images a carefully chosen source image provides the initial colour choice for the pixels, subject to slight modifications where clashes occur.

A flat plane represented at some distance behind the plane of the image will result in a simple repeating pattern of dots. Any deviation from the plane would cause that repeating pattern to be disrupted, and it is this which leads to the three-dimensional representation. Using am image rather than random dots for an Autostereogram results in parts of the image being distorted as the coloured dots are paired up to form the surface.

The image must be constructed by scanning a horizontal line across the object, and generating a horizontal line in the image. This means that the stereoscopic effect will only work if the image is viewed in one orientation. The program written for this work is designed for use with the *AVS* visualisation system. The system allows modules to be written to process its defined data structures, and these modules can be combined with existing modules to form an application program. For the generation of Autostereogram images, the input needs to be in the form of an array of values representing the depth of the nearest surface at each point in the scene. The Geometry

Viewer package included in the AVS system provides an output containing just this information. The flexibility of the AVS Geometry Viewer allows for many different forms of data to be viewed as a solid model which can be rotated and scaled in real time by the user. When the desired scene has been created, the depth information can be output from this module into the Autostereogram module.

The Autostereogram program contains the algorithm for scanning the lines of information derived from the scene, and preparing the lines of the output image. There are several possible algorithms for picking a colour for a dot in the output, but the important part of the program is that which ensures that pairs of dots corresponding to the projection of a point onto the image plane have the same colour. Each scan line of the image is independent of the others, so the algorithm has only to work on one line at a time.

1. The number of pixels *npix* on the line is determined by the size and resolution of the output image.

2. Two arrays of size *npix* are initialised; *colour*[*npix*] will contain the colour value for each pixel on the line, and *same*[*npix*] will contain the number of the pixel which must be the same colour. Thus if pixel number 20 needed to be the same colour as pixel number 40, then *same*[20] contains the value 40. Note that, in this example, *same*[40] would not necessarily contain 20. The *same* array is initialised to point to itself; that is *same*[x] contains the value x.

3. The input data are scanned: for each point on the scan line

   (a) A separation distance between the two projected positions is calculated.

   (b) The positions i and j along the line for these two points are determined.

   (c) If the point i is within the output image, the *same* array is traced to see which points need to be the same as both i and j.

4. The colour array is filled in reverse order. For each point x in the output image:

   (a) If $same[x] = x$, i.e. the colour for x does not depend on any point to the right of it, then a new random colour is generated for point x.

   (b) If $same[x] <> x$ then point x needs to be coloured the same as the point $same[x]$.

It should be noted that some three-dimensional models are better suited to autostereogram representation than others. The best models are those which consist of continuous surfaces and large objects. Conventional ball-and-stick representations of molecules do not work terribly well, and such models are probably better represented as red-green stereo pairs or cross-eyed stereo pairs. Space-filling representations of molecules fare much better,

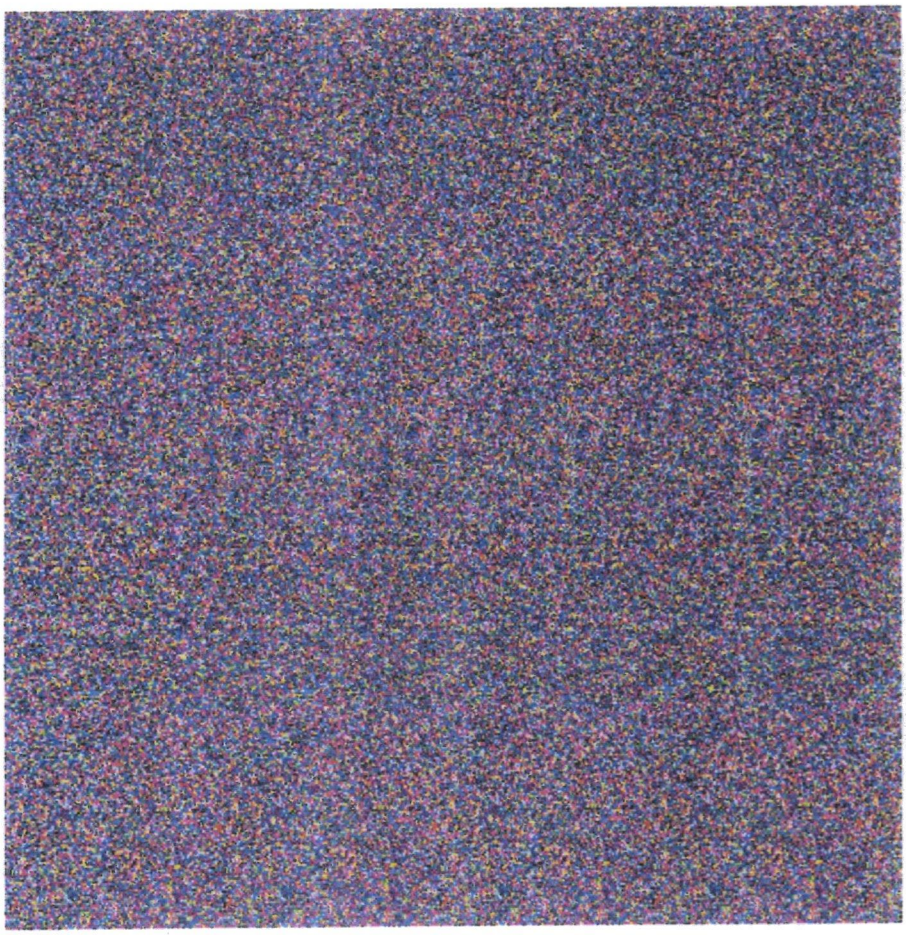


Figure 6.2: Example of Autostereogram Image

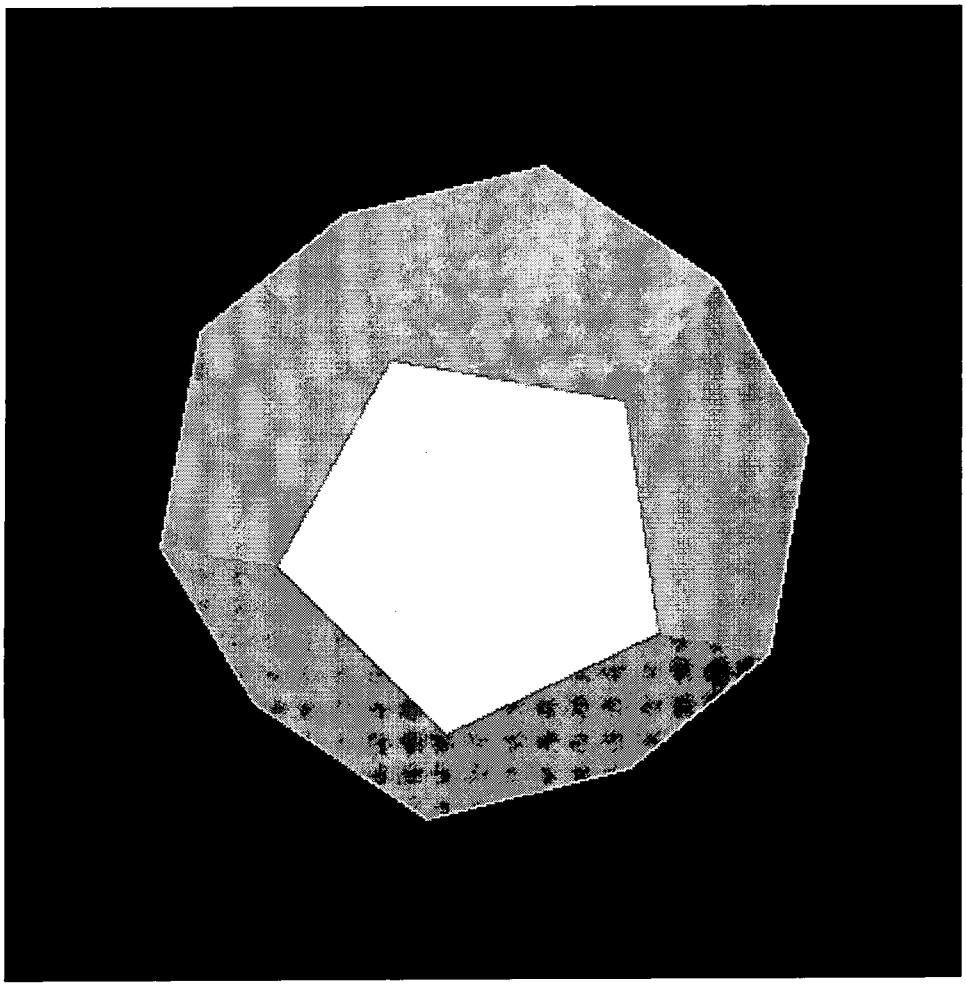


Figure 6.3: Model used for 6.2

and some good results have been obtained using models of C_{60} with atoms represented as spheres of radius slightly larger than the van der Waals radius.

The program allowed some tuning of the depth-scale and depth resolution of the resulting model, since some observers found that models with too much depth resolution were difficult to comprehend. The pixel size used in the final image has some bearing on the resolution of depth, since this quantizes the separation between paired pixels. On occasion, especially where the model was particularly large in the z-direction, planes of depth were apparent in the three-dimensional interpretation of the model.

Chapter 7

Experimental X-ray diffraction

7.1 Crystal structures determined by X-ray diffraction

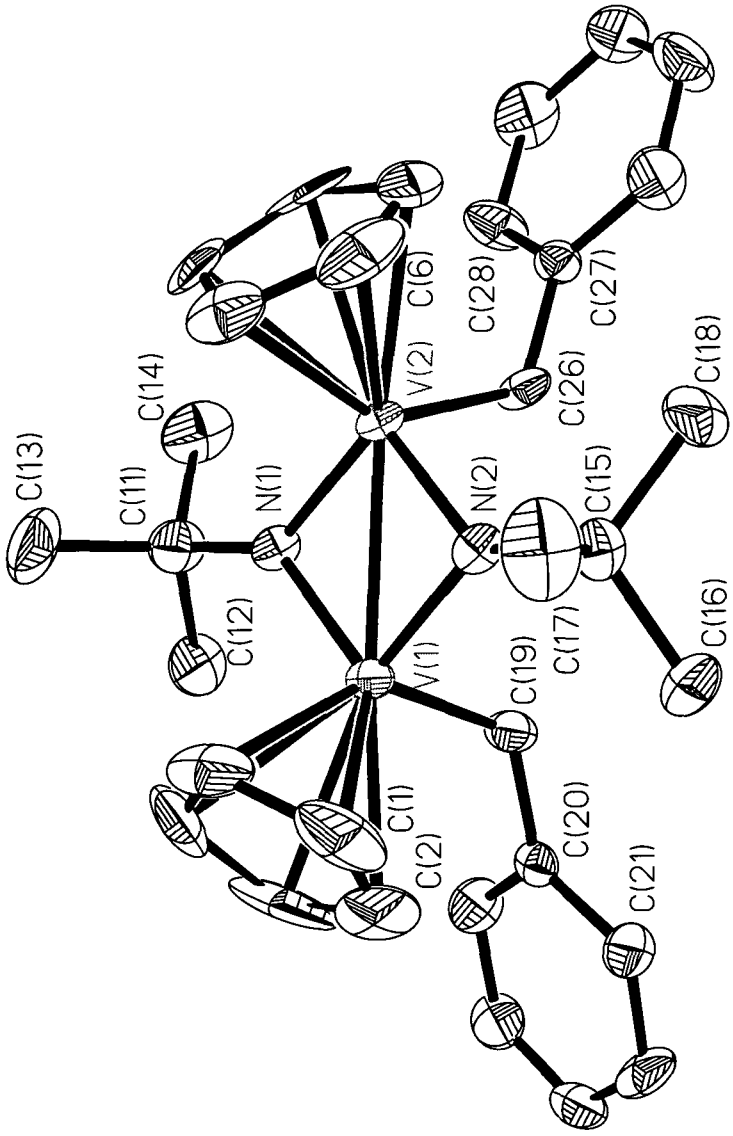
The work described in this Chapter covers expermental X-ray diffraction studies performed during the period of research. All of this work was carried out on behalf of the X-ray diffraction service in Durham University, in collaboration with the synthetic chemists who made and crystallised the samples. One of these structures, 7.1.4, has been incorporated into a paper submitted to the Journal of Organometallic Chemistry, which has been accepted for publication.

7.1.1 bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV})

The compound was believed to be air-sensitive and was kept in a Schlenk tube under nitrogen. The crystal chosen was transferred, under positive nitrogen pressure, from the Schlenk tube to a glass fibre by enveloping it in viscous perfluoropolyether oil. On cooling, this oil increases it's vicosity to hold the sample on the fibre for the duration of the data collection. The data were collected using Molybdenum Kα radiation on a Rigaku AFC6S four-circle diffractometer, and processed using the Texsan package. Structure solution was performed by direct methods, using SHELXS on a MicroVAX 3300 minicomputer. Refinement of the model was performed on a Hewlett-Packard 9000/720 workstation using SHELXL-93.

Although represented here with a vanadium-vanadium bond (Figures 7.2 and 7.1), it is thought unlikely that such a bond actually exists. The molecule is neutral overall, and this implies an oxidation state of +4 for both vanadium centres. The vanadium, nitrogen and immediately bonded atoms of the *tert*-butyl groups all lie in the same plane, suggesting delocalisation of the nitrogen lone pair.

Figure 7.1: Structure diagram for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl)vanadium^{IV}), hydrogen atoms omitted for clarity.



Crystal data and structure refinement for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV})

Identification code	95srv011
Empirical formula	V_2 N_2 C_{32} H_{42}
Structural formula	(η^5-Cp) ($PhCH_2$) V (μ-N ^{t}Bu)$_2$ V ($PhCH_2$) (η^5-Cp)
Formula weight	556.56
Temperature	293(2)K
Wavelength	.71070Å
Crystal system	Monoclinic
Space group	$P2_1/c$

Unit cell dimensions						
	a	=	8.615(7)Å	α	=	90°
	b	=	34.88(2)Å	β	=	111.73(6)°
	c	=	10.139(7)Å	γ	=	90°
	Volume	=	2830(3)Å^3			

Z	4
Density (calculated)	$1.306 Mg/m^3$
Absorption coefficient	$0.683 mm^{-1}$
F_{000}	1176
Crystal size	$0.40\times\ 0.30\times\ 0.10$mm
Theta range for data collection	2.55 to 24.02^o
Index ranges	$-1 <= h <= 9$
	$-1 <= k <= 37$
	$-10 <= l <= 11$
Reflections collected	4567
Independent reflections	$4567(R_{int} = 0.0000)$
Refinement method	Full-matrix least-squares on F^2
Data / restraints / parameters	4132/ 0/ 325
Goodness-of-fit on F^2	1.011
Final R indices ($I > 2\sigma(I)$)	
R_1	0.0673
wR_2	0.1678
R indices (all data)	
R_1	0.1524
wR_2	0.2090
Largest diff. peak and hole	.582 and -.500eÅ^{-3}

Table 7.1: Atomic coordinates and equivalent isotropic displacement parameters for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}).
U_{eq} is defined as one third of the trace of the orthogonalized U_{ij} tensor.

Atom	x	y	z	$U_{eq}/Å^2$
V(1)	.21863(14)	.15413(3)	.38697(11)	.0200(3)
V(2)	.15039(14)	.09360(3)	.24121(11)	.0207(3)
N(1)	.2558(6)	.13901(14)	.2259(5)	.0201(13)
N(2)	.1084(6)	.10957(14)	.4007(5)	.0206(13)
C(1)	.0166(10)	.1986(2)	.3048(9)	.043(2)
C(2)	.1597(12)	.2171(2)	.3055(9)	.045(2)
C(3)	.2683(11)	.2199(2)	.4455(10)	.055(3)
C(4)	.1910(13)	.2044(2)	.5313(9)	.050(2)
C(5)	.0346(12)	.1929(2)	.4465(10)	.051(2)
C(6)	-.1029(11)	.0961(2)	.0620(9)	.053(3)
C(7)	.0150(14)	.0876(3)	-.0006(7)	.062(3)
C(8)	.0776(13)	.0524(4)	.0454(11)	.070(3)
C(9)	.003(2)	.0389(3)	.1284(11)	.071(3)
C(10)	-.1070(12)	.0638(3)	.1395(9)	.058(3)
C(11)	.3385(9)	.1553(2)	.1319(7)	.034(2)
C(12)	.4775(10)	.1829(2)	.2149(9)	.047(2)
C(13)	.4219(11)	.1221(2)	.0803(8)	.042(2)
C(14)	.2113(12)	.1763(2)	.0034(8)	.054(2)
C(15)	.0270(9)	.0934(2)	.4962(7)	.028(2)
C(16)	.0559(11)	.0498(2)	.5090(8)	.042(2)
C(17)	.1080(11)	.1099(2)	.6457(7)	.042(2)
C(18)	-.1582(10)	.1020(3)	.4383(9)	.054(2)
C(19)	.4590(9)	.1420(2)	.5453(7)	.027(2)
C(20)	.5686(8)	.1694(2)	.6537(6)	.024(2)
C(21)	.6883(9)	.1907(2)	.6277(7)	.037(2)
C(22)	.7923(11)	.2162(2)	.7289(8)	.049(2)
C(23)	.7695(10)	.2212(2)	.8566(8)	.044(2)
C(24)	.6511(11)	.2006(2)	.8834(7)	.047(2)
C(25)	.5518(10)	.1744(2)	.7851(7)	.041(2)
C(26)	.3637(9)	.0579(2)	.3425(7)	.027(2)
C(27)	.3975(8)	.0193(2)	.2937(6)	.022(2)
C(28)	.4923(9)	.0146(2)	.2116(8)	.036(2)
C(29)	.5217(10)	-.0211(2)	.1643(7)	.039(2)
C(30)	.4501(10)	-.0529(2)	.1970(7)	.038(2)
C(31)	.3569(10)	-.0494(2)	.2797(8)	.037(2)
C(32)	.3310(9)	-.0141(2)	.3311(7)	.033(2)

Table 7.2: Bond lengths for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}).

Atom		Atom	Length/Å
V(1)	-	N(2)	1.853(5)
V(1)	-	N(1)	1.854(5)
V(1)	-	C(19)	2.141(7)
V(1)	-	C(1)	2.249(7)
V(1)	-	C(5)	2.326(8)
V(1)	-	C(2)	2.336(7)
V(1)	-	C(4)	2.351(7)
V(1)	-	C(3)	2.367(7)
V(1)	-	V(2)	2.519(2)
V(2)	-	N(1)	1.861(5)
V(2)	-	N(2)	1.868(5)
V(2)	-	C(26)	2.141(7)
V(2)	-	C(6)	2.266(8)
V(2)	-	C(7)	2.302(7)
V(2)	-	C(10)	2.316(8)
V(2)	-	C(8)	2.341(7)
V(2)	-	C(9)	2.342(8)
N(1)	-	C(11)	1.497(8)
N(2)	-	C(15)	1.500(8)
C(1)	-	C(2)	1.389(11)
C(1)	-	C(5)	1.401(11)
C(2)	-	C(3)	1.384(11)
C(3)	-	C(4)	1.386(12)
C(4)	-	C(5)	1.363(12)
C(6)	-	C(10)	1.379(13)
C(6)	-	C(7)	1.414(13)

Table 7.3: Bond lengths for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}) (continued).

Atom		Atom	Length/Å
C(7)	-	C(8)	1.354(14)
C(8)	-	C(9)	1.322(14)
C(9)	-	C(10)	1.318(14)
C(11)	-	C(12)	1.524(10)
C(11)	-	C(14)	1.541(10)
C(11)	-	C(13)	1.553(10)
C(15)	-	C(18)	1.512(10)
C(15)	-	C(17)	1.527(9)
C(15)	-	C(16)	1.538(10)
C(19)	-	C(20)	1.499(9)
C(20)	-	C(21)	1.373(9)
C(20)	-	C(25)	1.403(9)
C(21)	-	C(22)	1.402(10)
C(22)	-	C(23)	1.390(11)
C(23)	-	C(24)	1.355(11)
C(24)	-	C(25)	1.390(10)
C(26)	-	C(27)	1.500(9)
C(27)	-	C(28)	1.375(10)
C(27)	-	C(32)	1.409(10)
C(28)	-	C(29)	1.392(10)
C(29)	-	C(30)	1.368(11)
C(30)	-	C(31)	1.365(11)
C(31)	-	C(32)	1.386(10)

Table 7.4: Bond angles for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}).

Atom		Atom		Atom	Angle/^{o}
N(2)	-	V(1)	-	N(1)	95.1(2)
N(2)	-	V(1)	-	C(19)	99.1(2)
N(1)	-	V(1)	-	C(19)	99.8(3)
N(2)	-	V(1)	-	C(1)	105.0(3)
N(1)	-	V(1)	-	C(1)	103.4(3)
C(19)	-	V(1)	-	C(1)	144.6(3)
N(2)	-	V(1)	-	C(5)	92.9(3)
N(1)	-	V(1)	-	C(5)	138.6(3)
C(19)	-	V(1)	-	C(5)	118.9(3)
C(1)	-	V(1)	-	C(5)	35.6(3)
N(2)	-	V(1)	-	C(2)	139.9(3)
N(1)	-	V(1)	-	C(2)	92.4(3)
C(19)	-	V(1)	-	C(2)	118.3(3)
C(1)	-	V(1)	-	C(2)	35.2(3)
C(5)	-	V(1)	-	C(2)	57.7(3)
N(2)	-	V(1)	-	C(4)	113.9(3)
N(1)	-	V(1)	-	C(4)	148.3(3)
C(19)	-	V(1)	-	C(4)	88.5(3)
C(1)	-	V(1)	-	C(4)	58.2(3)
C(5)	-	V(1)	-	C(4)	33.9(3)
C(2)	-	V(1)	-	C(4)	57.3(3)
N(2)	-	V(1)	-	C(3)	147.5(3)
N(1)	-	V(1)	-	C(3)	115.0(3)
C(19)	-	V(1)	-	C(3)	88.2(3)
C(1)	-	V(1)	-	C(3)	57.8(3)
C(5)	-	V(1)	-	C(3)	56.6(3)
C(2)	-	V(1)	-	C(3)	34.2(3)
C(4)	-	V(1)	-	C(3)	34.2(3)
N(2)	-	V(1)	-	V(2)	47.7(2)
N(1)	-	V(1)	-	V(2)	47.4(2)
C(19)	-	V(1)	-	V(2)	103.0(2)
C(1)	-	V(1)	-	V(2)	112.4(2)
C(5)	-	V(1)	-	V(2)	127.3(2)
C(2)	-	V(1)	-	V(2)	127.4(2)
C(4)	-	V(1)	-	V(2)	159.2(3)
C(3)	-	V(1)	-	V(2)	160.1(3)
N(1)	-	V(2)	-	N(2)	94.3(2)
N(1)	-	V(2)	-	C(26)	100.2(2)
N(2)	-	V(2)	-	C(26)	99.5(2)
N(1)	-	V(2)	-	C(6)	104.8(3)
N(2)	-	V(2)	-	C(6)	104.2(3)
C(26)	-	V(2)	-	C(6)	143.8(3)

Table 7.5: Bond angles for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}) contiuned).

N(1)	-	V(2)	-	C(7)	93.4(3)
N(2)	-	V(2)	-	C(7)	139.9(4)
C(26)	-	V(2)	-	C(7)	117.7(4)
C(6)	-	V(2)	-	C(7)	36.1(3)
N(1)	-	V(2)	-	C(10)	139.5(3)
N(2)	-	V(2)	-	C(10)	93.4(3)
C(26)	-	V(2)	-	C(10)	117.6(4)
C(6)	-	V(2)	-	C(10)	35.0(3)
C(7)	-	V(2)	-	C(10)	57.4(3)
N(1)	-	V(2)	-	C(8)	116.0(4)
N(2)	-	V(2)	-	C(8)	147.2(4)
C(26)	-	V(2)	-	C(8)	87.8(3)
C(6)	-	V(2)	-	C(8)	57.8(3)
C(7)	-	V(2)	-	C(8)	33.9(3)
C(10)	-	V(2)	-	C(8)	55.6(4)
N(1)	-	V(2)	-	C(9)	147.8(4)
N(2)	-	V(2)	-	C(9)	114.9(4)
C(26)	-	V(2)	-	C(9)	88.6(3)
C(6)	-	V(2)	-	C(9)	56.7(3)
C(7)	-	V(2)	-	C(9)	55.7(3)
C(10)	-	V(2)	-	C(9)	32.9(3)
C(8)	-	V(2)	-	C(9)	32.8(4)
N(1)	-	V(2)	-	V(1)	47.2(2)
N(2)	-	V(2)	-	V(1)	47.2(2)
C(26)	-	V(2)	-	V(1)	103.5(2)
C(6)	-	V(2)	-	V(1)	112.6(2)
C(7)	-	V(2)	-	V(1)	128.1(3)
C(10)	-	V(2)	-	V(1)	127.9(3)
C(8)	-	V(2)	-	V(1)	160.7(4)
C(9)	-	V(2)	-	V(1)	159.3(4)
C(11)	-	N(1)	-	V(1)	137.0(4)
C(11)	-	N(1)	-	V(2)	137.6(4)
V(1)	-	N(1)	-	V(2)	85.4(2)
C(15)	-	N(2)	-	V(1)	137.6(4)
C(15)	-	N(2)	-	V(2)	137.2(4)
V(1)	-	N(2)	-	V(2)	85.2(2)
C(2)	-	C(1)	-	C(5)	107.5(7)
C(2)	-	C(1)	-	V(1)	75.8(4)
C(5)	-	C(1)	-	V(1)	75.2(4)
C(3)	-	C(2)	-	C(1)	107.4(8)

Table 7.6: Bond angles for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}) (continued).

C(3)	-	C(2)	-	V(1)	74.1(4)
C(1)	-	C(2)	-	V(1)	69.0(4)
C(2)	-	C(3)	-	C(4)	108.5(8)
C(2)	-	C(3)	-	V(1)	71.7(4)
C(4)	-	C(3)	-	V(1)	72.3(4)
C(5)	-	C(4)	-	C(3)	108.1(7)
C(5)	-	C(4)	-	V(1)	72.1(4)
C(3)	-	C(4)	-	V(1)	73.6(4)
C(4)	-	C(5)	-	C(1)	108.2(8)
C(4)	-	C(5)	-	V(1)	74.1(5)
C(1)	-	C(5)	-	V(1)	69.2(4)
C(10)	-	C(6)	-	C(7)	105.1(8)
C(10)	-	C(6)	-	V(2)	74.5(5)
C(7)	-	C(6)	-	V(2)	73.3(4)
C(8)	-	C(7)	-	C(6)	107.2(9)
C(8)	-	C(7)	-	V(2)	74.7(5)
C(6)	-	C(7)	-	V(2)	70.6(4)
C(9)	-	C(8)	-	C(7)	108.3(10)
C(9)	-	C(8)	-	V(2)	73.6(5)
C(7)	-	C(8)	-	V(2)	71.4(5)
C(10)	-	C(9)	-	C(8)	110.8(9)
C(10)	-	C(9)	-	V(2)	72.5(5)
C(8)	-	C(9)	-	V(2)	73.6(5)
C(9)	-	C(10)	-	C(6)	108.6(9)
C(9)	-	C(10)	-	V(2)	74.7(5)
C(6)	-	C(10)	-	V(2)	70.5(5)
N(1)	-	C(11)	-	C(12)	110.5(5)
N(1)	-	C(11)	-	C(14)	111.2(6)
C(12)	-	C(11)	-	C(14)	109.5(6)
N(1)	-	C(11)	-	C(13)	108.6(5)
C(12)	-	C(11)	-	C(13)	106.9(6)
C(14)	-	C(11)	-	C(13)	110.0(6)
N(2)	-	C(15)	-	C(18)	110.7(5)
N(2)	-	C(15)	-	C(17)	110.4(6)
C(18)	-	C(15)	-	C(17)	109.8(6)
N(2)	-	C(15)	-	C(16)	109.0(6)
C(18)	-	C(15)	-	C(16)	110.1(7)
C(17)	-	C(15)	-	C(16)	106.8(6)
C(20)	-	C(19)	-	V(1)	126.5(5)
C(21)	-	C(20)	-	C(25)	117.7(6)
C(21)	-	C(20)	-	C(19)	121.2(6)
C(25)	-	C(20)	-	C(19)	121.1(6)

Figure 7.2: Two dimensional schematic diagram for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}).

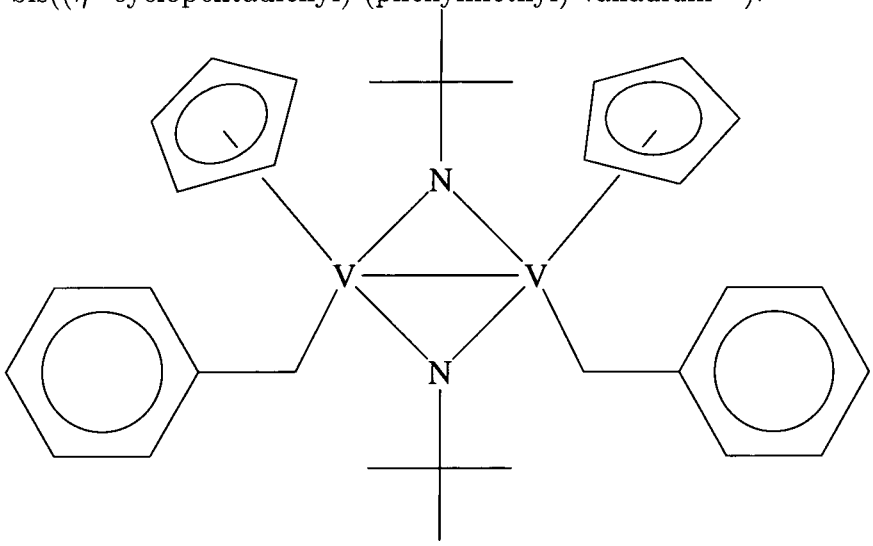


Table 7.7: Bond angles for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}) (continued).

C(20)	-	C(21)	-	C(22)	121.6(7)
C(23)	-	C(22)	-	C(21)	119.2(7)
C(24)	-	C(23)	-	C(22)	119.8(7)
C(23)	-	C(24)	-	C(25)	120.9(7)
C(24)	-	C(25)	-	C(20)	120.6(7)
C(27)	-	C(26)	-	V(2)	127.0(4)
C(28)	-	C(27)	-	C(32)	117.3(6)
C(28)	-	C(27)	-	C(26)	122.4(6)
C(32)	-	C(27)	-	C(26)	120.3(6)
C(27)	-	C(28)	-	C(29)	122.6(7)
C(30)	-	C(29)	-	C(28)	119.1(7)
C(31)	-	C(30)	-	C(29)	119.8(7)
C(30)	-	C(31)	-	C(32)	121.6(7)
C(31)	-	C(32)	-	C(27)	119.6(7)

Table 7.8: Anisotropic displacement parameters ($Å^2$) for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}).
The anisotropic displacement factor exponent takes the form:
$-2\pi^2(h^2a*^2U_{11} + ... + 2hka*b*U_{12})$

Atom	$U_{11}/Å^2$	$U_{22}/Å^2$	$U_{33}/Å^2$	$U_{23}/Å^2$	$U_{13}/Å^2$	$U_{12}/Å^2$
V(1)	.0209(7)	.0137(6)	.0257(6)	-.0009(5)	.0091(5)	.0017(5)
V(2)	.0199(7)	.0179(6)	.0225(6)	-.0032(5)	.0058(5)	-.0002(5)
N(1)	.016(3)	.019(3)	.027(3)	.000(2)	.011(3)	-.003(3)
N(2)	.013(3)	.021(3)	.027(3)	.002(2)	.007(3)	-.003(3)
C(1)	.022(4)	.030(4)	.066(6)	-.003(4)	.004(4)	.010(4)
C(2)	.079(7)	.012(4)	.054(5)	.013(4)	.037(5)	.022(4)
C(3)	.050(6)	.004(4)	.094(7)	-.005(4)	.009(6)	.006(4)
C(4)	.078(7)	.029(5)	.039(5)	-.007(4)	.017(5)	.021(5)
C(5)	.050(6)	.028(5)	.089(7)	.004(5)	.042(6)	.014(4)
C(6)	.033(5)	.036(5)	.063(6)	-.019(5)	-.014(5)	.020(4)
C(7)	.083(8)	.074(7)	.014(4)	-.001(4)	-.001(5)	-.045(7)
C(8)	.047(6)	.085(9)	.060(6)	-.063(6)	.000(6)	.009(6)
C(9)	.060(7)	.042(6)	.065(7)	-.013(5)	-.031(6)	-.024(6)
C(10)	.039(6)	.084(8)	.043(5)	-.026(5)	.006(5)	-.031(6)
C(11)	.038(5)	.032(4)	.036(4)	-.001(3)	.020(4)	-.005(4)
C(12)	.050(6)	.040(5)	.066(5)	.000(4)	.042(5)	-.012(4)
C(13)	.061(6)	.043(5)	.039(4)	.003(4)	.037(4)	.001(4)
C(14)	.076(7)	.040(5)	.051(5)	.024(4)	.031(5)	.006(5)
C(15)	.025(4)	.034(4)	.029(4)	.003(3)	.015(3)	-.003(4)
C(16)	.060(6)	.026(4)	.046(4)	.004(3)	.025(5)	-.009(4)
C(17)	.062(6)	.042(5)	.036(4)	-.009(4)	.035(4)	-.016(5)
C(18)	.029(5)	.078(7)	.059(5)	.013(5)	.021(4)	-.007(5)
C(19)	.025(4)	.014(3)	.036(4)	-.002(3)	.006(3)	-.003(3)
C(20)	.023(4)	.022(4)	.023(4)	-.003(3)	.004(3)	-.002(3)
C(21)	.033(5)	.045(5)	.034(4)	-.015(4)	.015(4)	-.018(4)
C(22)	.043(5)	.048(5)	.056(5)	.000(4)	.017(4)	-.025(5)
C(23)	.042(5)	.040(5)	.038(5)	-.008(4)	.001(4)	-.014(4)
C(24)	.059(6)	.051(5)	.027(4)	-.012(4)	.012(4)	-.019(5)
C(25)	.044(5)	.043(5)	.036(4)	-.002(4)	.014(4)	-.020(4)
C(26)	.030(4)	.024(4)	.026(4)	-.006(3)	.007(3)	-.002(3)
C(27)	.018(4)	.023(4)	.020(3)	.000(3)	.003(3)	.005(3)
C(28)	.035(5)	.032(5)	.040(4)	.002(4)	.015(4)	.003(4)
C(29)	.046(5)	.035(5)	.035(4)	.001(4)	.014(4)	.017(4)
C(30)	.041(5)	.030(5)	.029(4)	-.009(4)	-.004(4)	.012(4)
C(31)	.031(5)	.016(4)	.050(5)	.006(3)	-.003(4)	.005(4)
C(32)	.034(5)	.030(4)	.032(4)	.003(3)	.009(4)	.003(4)

Table 7.9: Hydrogen coordinates and isotropic displacement parameters for bis(μ-*tert*-butylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) vanadium^{IV}).

Atom	x	y	z	U_{eq}/Å^2
H(1)	-.0874(10)	.1957(2)	.2221(9)	.051
H(2)	.1765(12)	.2282(2)	.2230(9)	.054
H(3)	.3777(11)	.2325(2)	.4787(10)	.066
H(4)	.2358(13)	.2046(2)	.6353(9)	.060
H(5)	-.0529(12)	.1839(2)	.4791(10)	.061
H(6)	-.1819(11)	.1175(2)	.0385(9)	.064
H(7)	.0363(14)	.1029(3)	-.0733(7)	.075
H(8)	.1561(13)	.0381(4)	.0144(11)	.084
H(9)	.020(2)	.0132(3)	.1701(11)	.085
H(10)	-.1862(12)	.0592(3)	.1866(9)	.069
H(12A)	.5566(10)	.1699(2)	.2949(9)	.070
H(12B)	.4305(10)	.2042(2)	.2475(9)	.070
H(12C)	.5327(10)	.1922(2)	.1544(9)	.070
H(13A)	.3380(11)	.1040(2)	.0273(8)	.063
H(13B)	.5025(11)	.1095(2)	.1608(8)	.063
H(13C)	.4766(11)	.1322(2)	.0208(8)	.063
H(14A)	.1238(12)	.1589(2)	-.0488(8)	.081
H(14B)	.2665(12)	.1855(2)	-.0571(8)	.081
H(14C)	.1643(12)	.1975(2)	.0360(8)	.081
H(16A)	.1738(11)	.0447(2)	.5458(8)	.063
H(16B)	.0053(11)	.0383(2)	.4170(8)	.063
H(16C)	.0068(11)	.0394(2)	.5720(8)	.063
H(17A)	.2255(11)	.1044(2)	.6819(7)	.063
H(17B)	.0578(11)	.0986(2)	.7065(7)	.063
H(17C)	.0916(11)	.1372(2)	.6426(7)	.063
H(18A)	-.2084(10)	.0916(3)	.3444(9)	.081
H(18B)	-.1749(10)	.1293(3)	.4349(9)	.081
H(18C)	-.2088(10)	.0907(3)	.4988(9)	.081
H(19A)	.4423(9)	.1204(2)	.5987(7)	.032
H(19B)	.5265(9)	.1325(2)	.4940(7)	.032
H(21)	.7007(9)	.1881(2)	.5408(7)	.044
H(22)	.8757(11)	.2296(2)	.7108(8)	.059
H(23)	.8354(10)	.2386(2)	.9233(8)	.053
H(24)	.6360(11)	.2041(2)	.9689(7)	.056
H(25)	.4734(10)	.1599(2)	.8065(7)	.050
H(26A)	.4599(9)	.0735(2)	.3506(7)	.033
H(26B)	.3681(9)	.0541(2)	.4386(7)	.033
H(28)	.5385(9)	.0362(2)	.1866(8)	.043
H(29)	.5892(10)	-.0233(2)	.1112(7)	.047
H(30)	.4649(10)	-.0768(2)	.1629(7)	.046
H(31)	.3096(10)	-.0712(2)	.3022(8)	.045
H(32)	.2702(9)	-.0124(2)	.3899(7)	.040

7.1.2 bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}

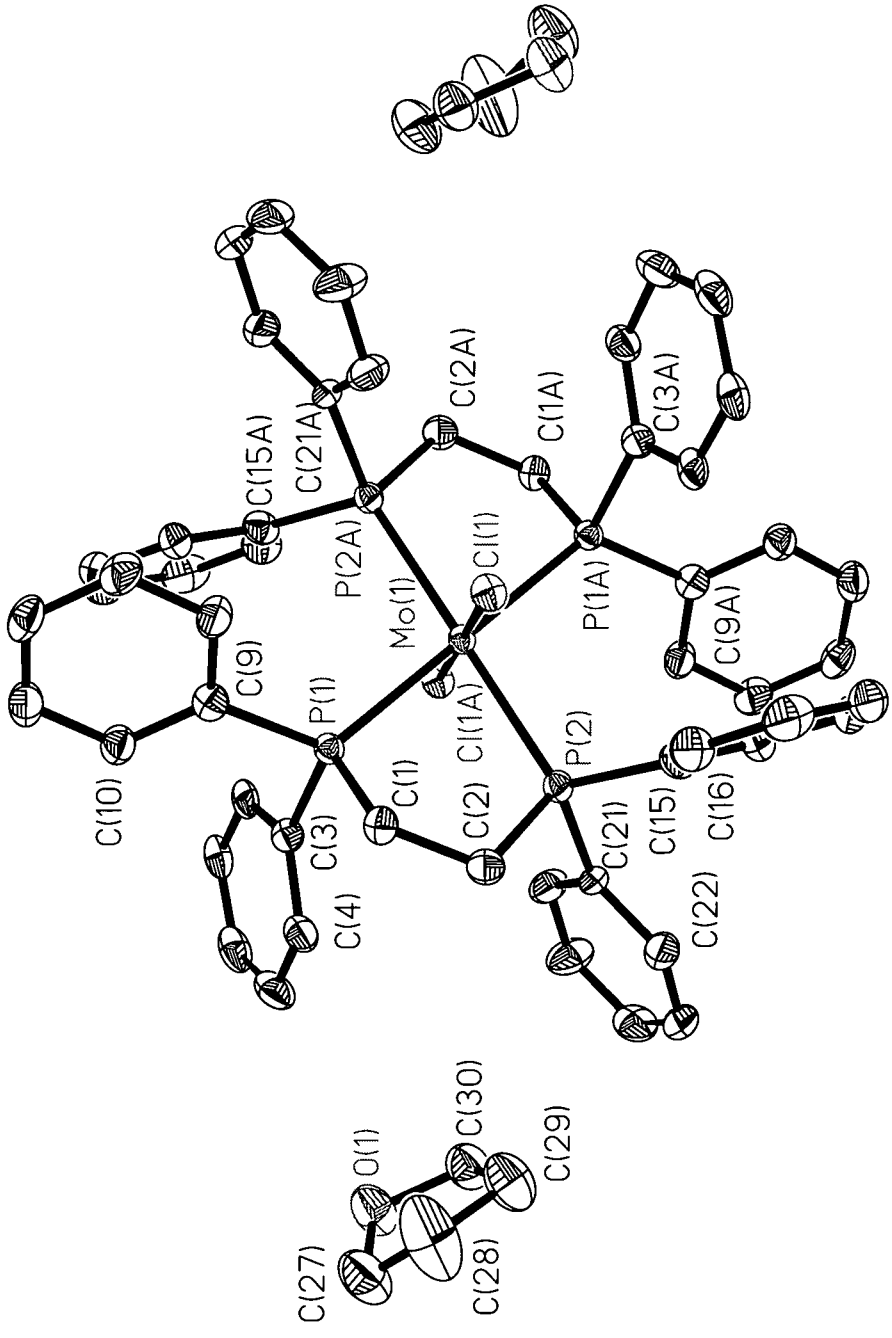
This compound was also thought to be air-sensitive, and was mounted on a glass fibre in perfluoropolyether oil at low temperature. The data were collected on a Siemens P4 diffractometer and processed using Siemens XSCANS software. The solution was by Patterson map in SHELXS and the model was refined using CRYSTALS.

The molybdenum atom lies on a special position in the unit cell, coinciding with a centre of inversion, hence only half of the molecule is actually present in the asymmetric unit, along with one molecule of tetrahydrofuran solvent. Similar structures have been reported, although with different solvents in the crystal or two hydride ligands attached to the molybdenum - there is no evidence for hydride ligands in this case.

Crystal data and structure refinement for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}.

Identification code	95srv041
Empirical formula	Mo Cl_2 P_4 C_{56} H_{56} + 2(C_4 H_8 O)
Structural formula	Mo Cl_2 $(Ph_2PCH_2CH_2PPh_2)_2$, THF solvate
Formula weight	1107.83
Temperature	293(2)K
Wavelength	.71073Å
Crystal system	Monoclinic
Space group	$P2_1/c$

Figure 7.3: Structure diagram for bis(bis(diphenylphosphino)ethane) dichloro molybdenum(II), symmetry-expanded to show whole molecule, hydrogen atoms and some labels omitted for clarity.



Unit cell dimensions	a	=	11.3160(10)Å	α	=	90^o
	b	=	13.4150(10)Å	β	=	96.36^o
	c	=	17.3400(10)Å	γ	=	90^o
	Volume	=	2616.1(3)Å^3			

Z	2
Density (calculated)	$1.406 Mg/m^3$
Absorption coefficient	$0.519 mm^{-1}$
F_{000}	1152
Crystal size	0.2× .15× 0.25 mm
Theta range for data collection	2.36 to 27.49°
Index ranges	$-1 <= h <= 13$
	$-1 <= k <= 17$
	$-20 <= l <= 22$
Reflections collected	5514
Independent reflections	4410 ($R_{int} = 0.0322$)
Refinement method	Full-matrix least-squares on F^2
Data / restraints / parameters	4410/ 0/ 345
Goodness-of-fit on F^2	0.887
Final R indices ($I > 2\sigma(I)$)	
R_1	0.0341
wR_2	0.0729
R indices (all data)	
R_1	0.0534
wR_2	0.0898
Largest diff. peak and hole	0.570 and -0.647$eÅ^{-3}$

Table 7.10: Atomic coordinates and equivalent isotropic displacement parameters for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}. U_{eq} is defined as one third of the trace of the orthogonalized U_{ij} tensor.

Atom	x	y	z	$U_{eq}/Å^2$
Mo(1)	.5000	.0000	.0000	.01505(11)
P(2)	.52811(8)	.11406(7)	.11505(5)	.0187(2)
P(1)	.61390(8)	-.10900(7)	.09747(5)	.0175(2)
Cl(1)	.32789(7)	-.04963(7)	.06361(5)	.0222(2)
C(20)	.3774(3)	.2784(3)	.0936(2)	.0261(8)
C(3)	.7756(3)	-.0929(2)	.1067(2)	.0203(7)
C(7)	.9559(3)	-.1037(3)	.0454(2)	.0294(8)
C(6)	1.0194(3)	-.0603(3)	.1092(3)	.0348(10)
C(22)	.6422(4)	.2798(3)	.1905(2)	.0310(9)
C(14)	.4807(4)	-.2840(3)	.0829(2)	.0285(8)
C(9)	.5947(3)	-.2455(3)	.0998(2)	.0220(7)
C(24)	.8328(4)	.3431(3)	.1669(3)	.0380(10)
C(17)	.2065(4)	.2075(4)	.1817(3)	.0408(11)
C(15)	.3962(3)	.1874(3)	.1321(2)	.0232(8)
C(16)	.3092(4)	.1533(3)	.1762(2)	.0317(9)
C(4)	.8404(3)	-.0529(3)	.1717(2)	.0268(8)
C(10)	.6896(3)	-.3102(3)	.1160(2)	.0264(8)
C(13)	.4632(4)	-.3867(3)	.0817(2)	.0316(9)
C(12)	.5583(4)	-.4507(3)	.0970(2)	.0308(9)
C(26)	.7458(4)	.2051(3)	.0924(2)	.0312(9)
C(19)	.2753(4)	.3335(3)	.0996(3)	.0357(10)
C(1)	.5682(3)	-.0740(3)	.1929(2)	.0214(8)
C(25)	.8384(4)	.2721(4)	.1100(3)	.0400(10)
C(2)	.5638(4)	.0387(3)	.2059(2)	.0256(8)
C(21)	.6464(3)	.2087(3)	.1320(2)	.0220(7)
C(23)	.7351(4)	.3462(3)	.2071(3)	.0356(10)
C(11)	.6713(4)	-.4130(3)	.1147(2)	.0321(9)
C(8)	.8348(3)	-.1181(3)	.0429(2)	.0247(8)
C(18)	.1899(4)	.2981(4)	.1432(3)	.0388(11)
C(5)	.9618(4)	-.0357(3)	.1731(3)	.0345(9)
C(27)	1.0067(5)	.0796(5)	.4154(3)	.058(2)
C(29)	.8470(4)	.1787(4)	.3628(3)	.0512(13)
C(28)	.8802(6)	.0985(6)	.4193(4)	.085(2)
C(30)	.9451(4)	.1770(4)	.3103(3)	.0433(11)
O(1)	1.0282(3)	.1012(3)	.3388(2)	.0477(8)

Table 7.11: Bond lengths for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}.

Atom		Atom	Length/Å
Mo(1)	-	Cl(1)	2.4350(8)
Mo(1)	-	Cl(1)^a	2.4351(8)
Mo(1)	-	P(1)	2.4845(9)
Mo(1)	-	P(1)^a	2.4846(9)
Mo(1)	-	P(2)	2.5064(9)
Mo(1)	-	P(2)^a	2.5064(9)
P(2)	-	C(15)	1.839(4)
P(2)	-	C(21)	1.844(4)
P(2)	-	C(2)	1.878(4)
P(1)	-	C(3)	1.831(4)
P(1)	-	C(9)	1.845(4)
P(1)	-	C(1)	1.849(3)
C(20)	-	C(19)	1.385(6)
C(20)	-	C(15)	1.397(5)
C(3)	-	C(4)	1.384(5)
C(3)	-	C(8)	1.396(5)
C(7)	-	C(8)	1.379(5)
C(7)	-	C(6)	1.379(6)
C(6)	-	C(5)	1.386(6)
C(22)	-	C(23)	1.383(6)
C(22)	-	C(21)	1.397(5)
C(14)	-	C(9)	1.390(6)
C(14)	-	C(13)	1.391(5)
C(9)	-	C(10)	1.385(5)
C(24)	-	C(23)	1.372(7)
C(24)	-	C(25)	1.378(6)
C(17)	-	C(16)	1.383(6)
C(17)	-	C(18)	1.389(7)
C(15)	-	C(16)	1.390(5)
C(4)	-	C(5)	1.390(6)
C(10)	-	C(11)	1.394(5)
C(13)	-	C(12)	1.380(6)
C(12)	-	C(11)	1.378(6)
C(26)	-	C(21)	1.383(6)
C(26)	-	C(25)	1.388(6)
C(19)	-	C(18)	1.376(7)
C(1)	-	C(2)	1.531(5)
C(27)	-	O(1)	1.407(6)
C(27)	-	C(28)	1.463(8)
C(29)	-	C(28)	1.475(8)
C(29)	-	C(30)	1.511(7)
C(30)	-	O(1)	1.435(6)

Symmetry transformations used to generate equivalent atoms:
^{a} -x+1,-y,-z

Table 7.12: Bond angles for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}.

Atom		Atom		Atom	Angle/^{o}
Cl(1)	-	Mo(1)	-	Cl(1)^a	180.0
Cl(1)	-	Mo(1)	-	P(1)	84.93(3)
Cl(1)^a	-	Mo(1)	-	P(1)	95.07(3)
Cl(1)	-	Mo(1)	-	P(1)^a	95.07(3)
Cl(1)^a	-	Mo(1)	-	P(1)^a	84.93(3)
P(1)	-	Mo(1)	-	P(1)^a	180.0
Cl(1)	-	Mo(1)	-	P(2)	81.01(3)
Cl(1)^a	-	Mo(1)	-	P(2)	98.99(3)
P(1)	-	Mo(1)	-	P(2)	78.89(3)
P(1)^a	-	Mo(1)	-	P(2)	101.11(3)
Cl(1)	-	Mo(1)	-	P(2)^a	98.99(3)
Cl(1)^a	-	Mo(1)	-	P(2)^a	81.01(3)
P(1)	-	Mo(1)	-	P(2)^a	101.11(3)
P(1)^a	-	Mo(1)	-	P(2)^a	78.89(3)
P(2)	-	Mo(1)	-	P(2)^a	180.0
C(15)	-	P(2)	-	C(21)	101.2(2)
C(15)	-	P(2)	-	C(2)	104.8(2)
C(21)	-	P(2)	-	C(2)	98.7(2)
C(15)	-	P(2)	-	Mo(1)	114.91(12)
C(21)	-	P(2)	-	Mo(1)	124.96(12)
C(2)	-	P(2)	-	Mo(1)	109.63(12)
C(3)	-	P(1)	-	C(9)	103.5(2)
C(3)	-	P(1)	-	C(1)	105.5(2)
C(9)	-	P(1)	-	C(1)	100.8(2)
C(3)	-	P(1)	-	Mo(1)	115.34(11)
C(9)	-	P(1)	-	Mo(1)	123.07(12)
C(1)	-	P(1)	-	Mo(1)	106.51(12)
C(19)	-	C(20)	-	C(15)	120.9(4)
C(4)	-	C(3)	-	C(8)	118.8(3)
C(4)	-	C(3)	-	P(1)	123.4(3)
C(8)	-	C(3)	-	P(1)	117.7(3)
C(8)	-	C(7)	-	C(6)	120.8(4)
C(7)	-	C(6)	-	C(5)	119.5(4)
C(23)	-	C(22)	-	C(21)	120.3(4)
C(9)	-	C(14)	-	C(13)	119.9(4)
C(10)	-	C(9)	-	C(14)	119.4(3)
C(10)	-	C(9)	-	P(1)	122.4(3)
C(14)	-	C(9)	-	P(1)	118.2(3)
C(23)	-	C(24)	-	C(25)	119.2(4)

Table 7.13: Bond angles for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II} (continued).

Atom		Atom		Atom	Angle/^{o}
C(16)	-	C(17)	-	C(18)	119.8(4)
C(16)	-	C(15)	-	C(20)	118.2(3)
C(16)	-	C(15)	-	P(2)	123.3(3)
C(20)	-	C(15)	-	P(2)	118.3(3)
C(17)	-	C(16)	-	C(15)	121.1(4)
C(3)	-	C(4)	-	C(5)	120.8(4)
C(9)	-	C(10)	-	C(11)	120.4(4)
C(12)	-	C(13)	-	C(14)	120.4(4)
C(11)	-	C(12)	-	C(13)	120.0(4)
C(21)	-	C(26)	-	C(25)	120.3(4)
C(18)	-	C(19)	-	C(20)	120.0(4)
C(2)	-	C(1)	-	P(1)	113.5(3)
C(24)	-	C(25)	-	C(26)	120.8(4)
C(1)	-	C(2)	-	P(2)	114.6(3)
C(26)	-	C(21)	-	C(22)	118.6(3)
C(26)	-	C(21)	-	P(2)	120.9(3)
C(22)	-	C(21)	-	P(2)	120.3(3)
C(24)	-	C(23)	-	C(22)	120.7(4)
C(12)	-	C(11)	-	C(10)	119.9(4)
C(7)	-	C(8)	-	C(3)	120.3(4)
C(19)	-	C(18)	-	C(17)	120.0(4)
C(6)	-	C(5)	-	C(4)	119.9(4)
O(1)	-	C(27)	-	C(28)	106.2(4)
C(28)	-	C(29)	-	C(30)	103.8(4)
C(27)	-	C(28)	-	C(29)	106.0(5)
O(1)	-	C(30)	-	C(29)	107.5(4)
C(27)	-	O(1)	-	C(30)	107.2(4)

Symmetry transformations used to generate equivalent atoms:
[a] -x+1,-y,-z

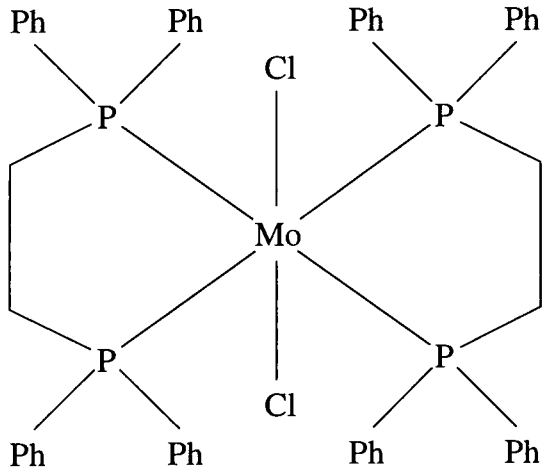
Table 7.14: Anisotropic displacement parameters ($Å^2$) for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}.
The anisotropic displacement factor exponent takes the form:
$-2\pi^2(h^2a*^2U_{11} + ... + 2hka*b*U_{12})$

Atom	$U_{11}/Å^2$	$U_{22}/Å^2$	$U_{33}/Å^2$	$U_{23}/Å^2$	$U_{13}/Å^2$	$U_{12}/Å^2$
Mo(1)	.0139(2)	.0140(2)	.0174(2)	.0005(2)	.00218(14)	.0003(2)
P(2)	.0173(4)	.0182(4)	.0208(4)	-.0011(4)	.0022(3)	.0014(4)
P(1)	.0168(4)	.0161(4)	.0194(4)	.0019(3)	.0017(3)	.0002(3)
Cl(1)	.0185(4)	.0234(4)	.0253(4)	.0013(3)	.0055(3)	-.0007(3)
C(20)	.023(2)	.027(2)	.028(2)	-.003(2)	.002(2)	.003(2)
C(3)	.021(2)	.014(2)	.025(2)	.0023(14)	.0023(14)	-.0004(14)
C(7)	.025(2)	.025(2)	.039(2)	.008(2)	.008(2)	.006(2)
C(6)	.018(2)	.030(2)	.056(3)	.016(2)	.002(2)	.001(2)
C(22)	.027(2)	.027(2)	.039(2)	-.010(2)	.000(2)	.003(2)
C(14)	.026(2)	.024(2)	.036(2)	.001(2)	.005(2)	.000(2)
C(9)	.026(2)	.020(2)	.021(2)	.0028(14)	.005(2)	-.002(2)
C(24)	.032(2)	.028(2)	.051(3)	-.001(2)	-.011(2)	-.013(2)
C(17)	.025(2)	.058(3)	.042(2)	-.018(2)	.013(2)	-.004(2)
C(15)	.020(2)	.024(2)	.025(2)	-.005(2)	.0033(14)	.002(2)
C(16)	.030(2)	.036(2)	.031(2)	-.004(2)	.008(2)	.000(2)
C(4)	.024(2)	.025(2)	.031(2)	.004(2)	.000(2)	.002(2)
C(10)	.027(2)	.020(2)	.032(2)	.000(2)	.002(2)	.000(2)
C(13)	.031(2)	.023(2)	.041(2)	.000(2)	.005(2)	-.009(2)
C(12)	.047(2)	.013(2)	.033(2)	-.003(2)	.007(2)	-.005(2)
C(26)	.029(2)	.032(2)	.032(2)	-.007(2)	.002(2)	-.004(2)
C(19)	.035(2)	.030(2)	.039(2)	-.012(2)	-.007(2)	.011(2)
C(1)	.023(2)	.021(2)	.020(2)	.0038(14)	.0063(14)	.000(2)
C(25)	.030(2)	.046(3)	.044(2)	-.011(2)	.005(2)	-.013(2)
C(2)	.035(2)	.024(2)	.018(2)	-.0001(14)	.000(2)	.004(2)
C(21)	.024(2)	.015(2)	.026(2)	-.0040(14)	-.004(2)	.0020(14)
C(23)	.038(2)	.021(2)	.044(2)	-.011(2)	-.011(2)	.002(2)
C(11)	.033(2)	.022(2)	.041(2)	.002(2)	.002(2)	.005(2)
C(8)	.027(2)	.018(2)	.029(2)	.005(2)	.003(2)	.002(2)
C(18)	.025(2)	.045(3)	.046(3)	-.020(2)	.001(2)	.010(2)
C(5)	.024(2)	.034(2)	.043(2)	.007(2)	-.008(2)	-.002(2)
C(27)	.056(3)	.074(4)	.043(3)	.014(3)	.005(2)	.021(3)
C(29)	.035(3)	.061(3)	.059(3)	.014(3)	.010(2)	.006(2)
C(28)	.077(4)	.100(5)	.086(5)	.055(4)	.046(4)	.041(4)
C(30)	.044(3)	.046(3)	.039(2)	.007(2)	.004(2)	.005(2)
O(1)	.044(2)	.058(2)	.042(2)	.010(2)	.0074(14)	.016(2)

Table 7.15: Hydrogen coordinates and isotropic displacement parameters for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}.

Atom	x	y	z	$U_{eq}/Å^2$
H(20A)	.4343(3)	.3024(3)	.0635(2)	.018(10)
H(7A)	.9951(3)	-.1234(3)	.0036(2)	.036(11)
H(6A)	1.1002(3)	-.0477(3)	.1094(3)	.036(12)
H(22A)	.5766(4)	.2826(3)	.2184(2)	.046(14)
H(14A)	.4163(4)	-.2412(3)	.0725(2)	.048(14)
H(24A)	.8946(4)	.3883(3)	.1779(3)	.044(13)
H(17A)	.1488(4)	.1834(4)	.2111(3)	.05(2)
H(16A)	.3202(4)	.0930(3)	.2025(2)	.045(14)
H(4A)	.8023(3)	-.0373(3)	.2150(2)	.025(10)
H(10A)	.7659(3)	-.2850(3)	.1279(2)	.042(13)
H(13A)	.3869(4)	-.4123(3)	.0704(2)	.043(13)
H(12A)	.5461(4)	-.5193(3)	.0955(2)	.030(11)
H(26A)	.7507(4)	.1577(3)	.0537(2)	.05(2)
H(19A)	.2645(4)	.3944(3)	.0741(3)	.036(12)
H(1A)	.4902(3)	-.1018(3)	.1975(2)	.025(10)
H(1B)	.6234(3)	-.1033(3)	.2334(2)	.024(10)
H(25A)	.9049(4)	.2690(4)	.0830(3)	.07(2)
H(2A)	.6402(4)	.0603(3)	.2314(2)	.06(2)
H(2B)	.5045(4)	.0528(3)	.2408(2)	.037(12)
H(23A)	.7313(4)	.3934(3)	.2460(3)	.040(13)
H(11A)	.7354(4)	-.4561(3)	.1256(2)	.041(13)
H(8A)	.7925(3)	-.1446(3)	-.0014(2)	.013(9)
H(18A)	.1210(4)	.3348(4)	.1469(3)	.06(2)
H(5A)	1.0043(4)	-.0078(3)	.2168(3)	.039(12)
H(27A)	1.0548(5)	.1219(5)	.4518(3)	.10(3)
H(27B)	1.0258(5)	.0105(5)	.4279(3)	.08(2)
H(29A)	.8438(4)	.2426(4)	.3886(3)	.17(4)
H(29B)	.7704(4)	.1653(4)	.3337(3)	.09(2)
H(28A)	.8664(6)	.1192(6)	.4711(4)	.13(3)
H(28B)	.8339(6)	.0389(6)	.4058(4)	.41(13)
H(30A)	.9126(4)	.1623(4)	.2574(3)	.050(14)
H(30B)	.9843(4)	.2414(4)	.3110(3)	.07(2)

Figure 7.4: Two dimensional schematic diagram for bis(bis(diphenylphosphino)ethane) dichloro molybdenum^{II}.



7.1.3 bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}

The sample of this compound was mounted at low temperature on a glass fibre in perfluoropolyether oil. The data were collected using an Siemens SMART CCD area detector on a three-circle diffractometer using molybdenum Kα radiation. The area detector frames were integrated and processed with the SAINT program, the resulting reflection data processed with XPREP, and the structure solved by SHELXS using direct methods, all on a Digital Alpha workstation. The model was refined to it's present condition using SHELXTL-PLUS on a Silicon Graphics Indigo workstation.

This structure is currently considered to be unfinished, due to problems with an extremely intractable solvent molecule disordered about an inversion centre in the unit cell. The solvent, believed to be pentane, is not shown in the diagrams, but has been modelled so far as up to ten partially occupied sites.

Figure 7.5: Partial structure diagram for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane] titanium(IV), hydrogen atoms omitted for clarity.

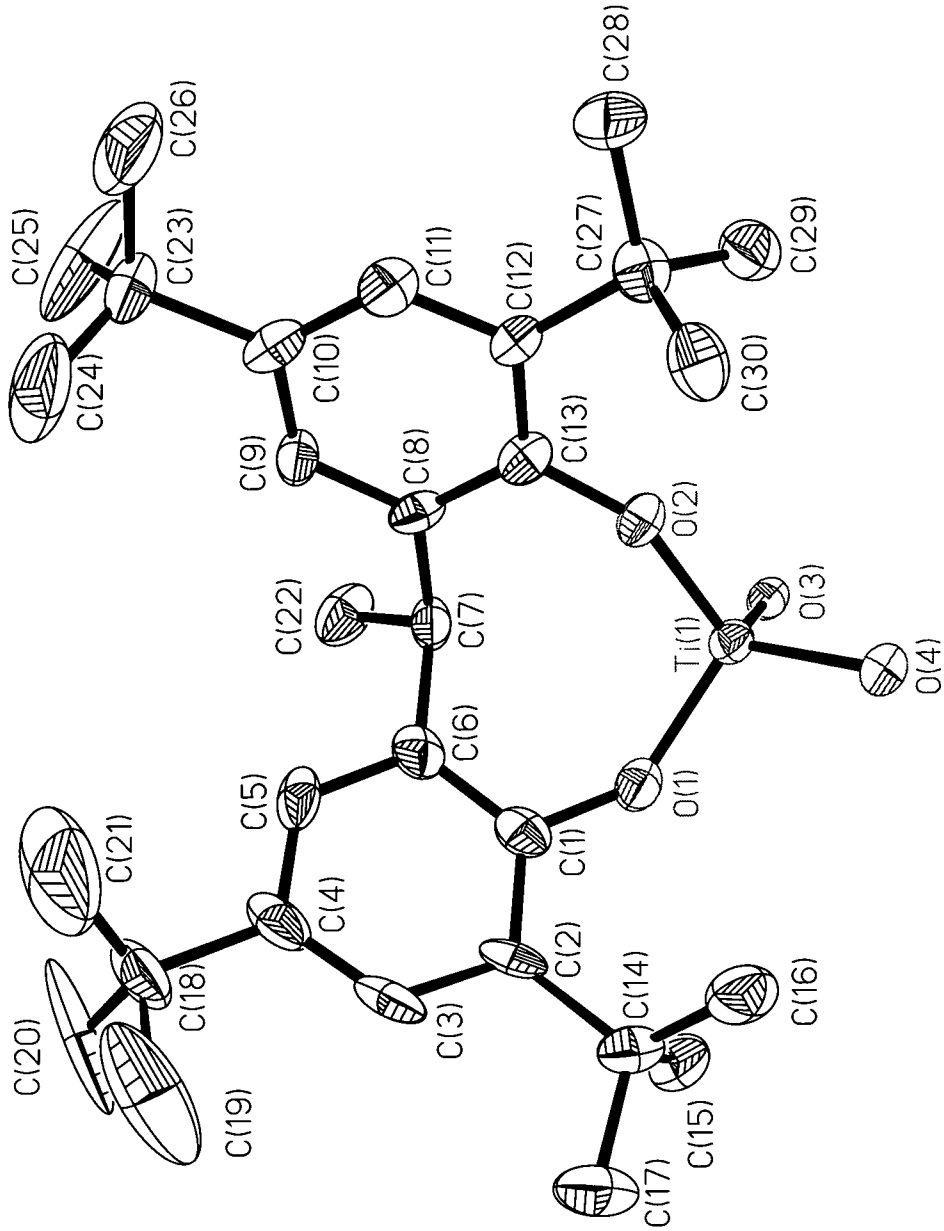


Figure 7.6: Partial structure diagram for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane] titanium(IV), hydrogen atoms omitted for clarity.

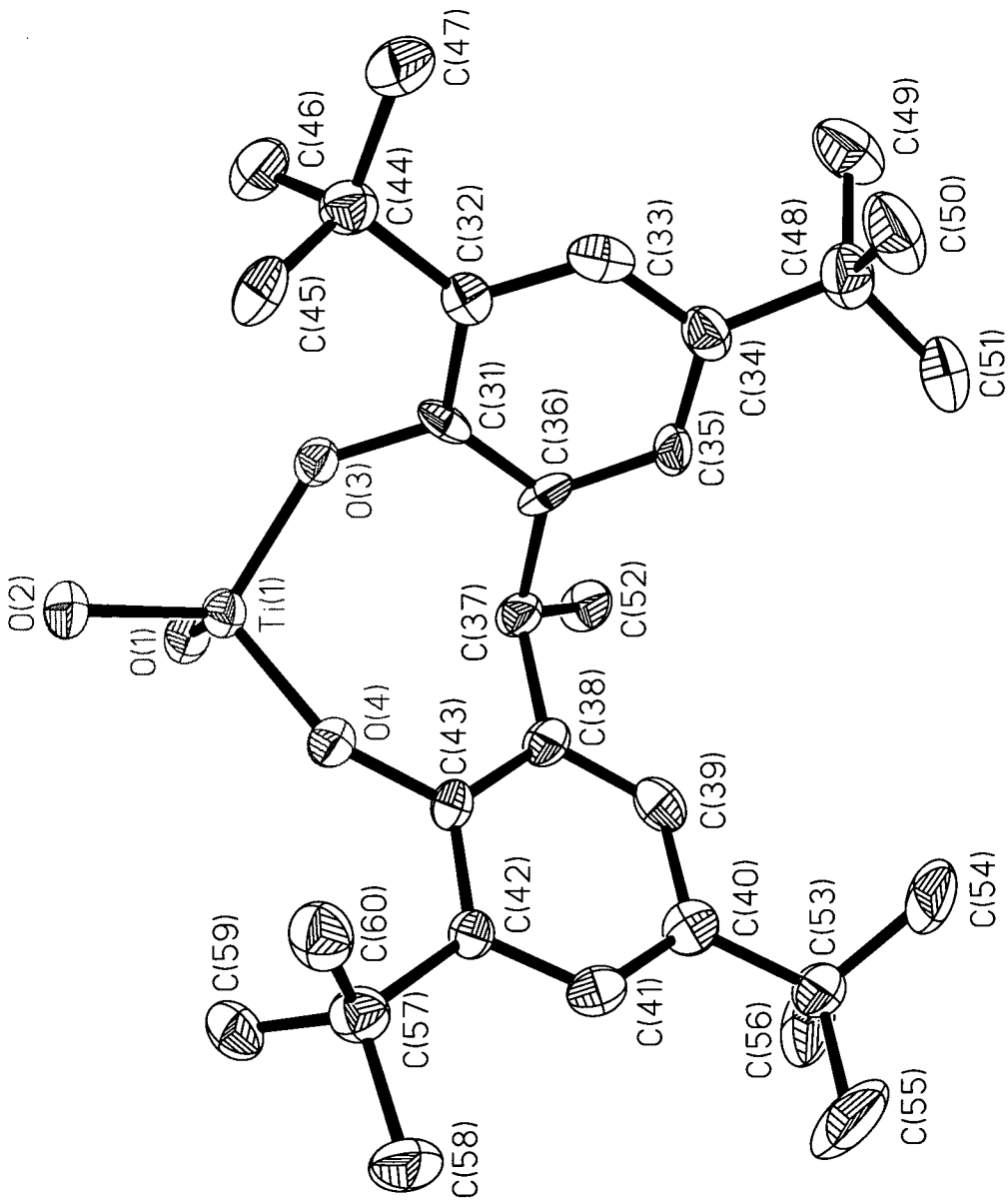
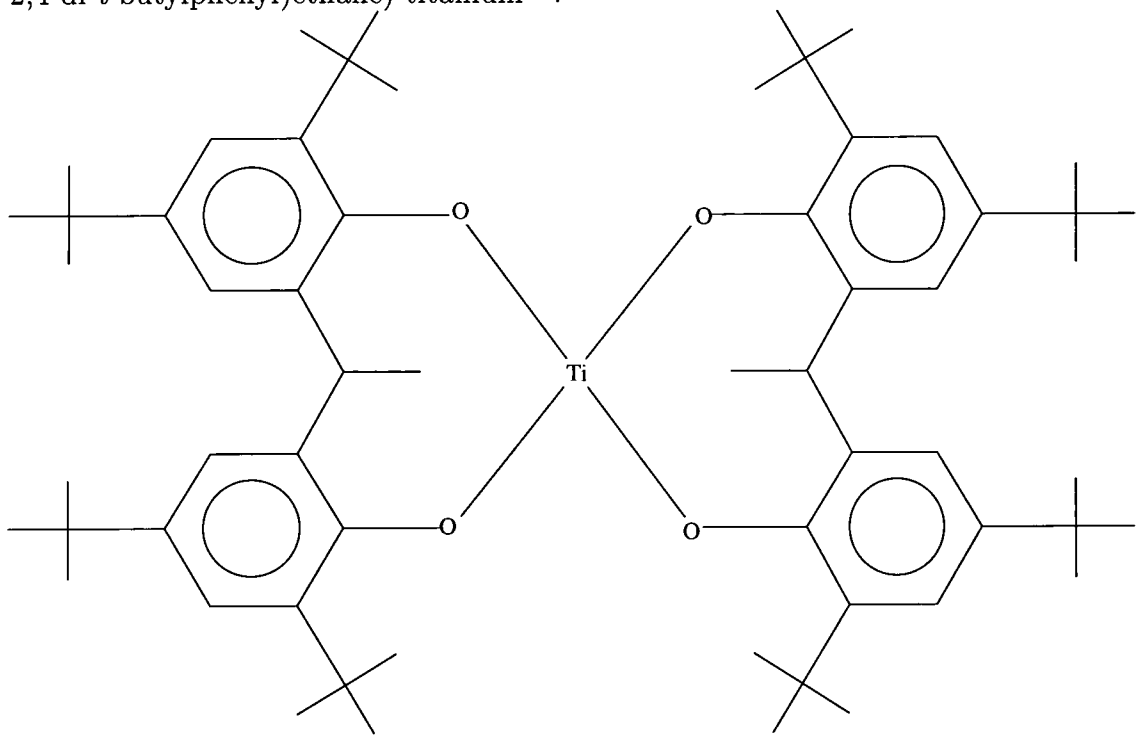


Figure 7.7: Two dimensional schematic diagram for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}.



Crystal data and structure refinement for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}.

Identification code	95srv075
Empirical formula	$C_{59}\ H_{10}\ O_4\ Ti$
Structural formula	$Ti\ ((O(^tBu)_2C_6H_2)CH(CH_3)((^tBu)_2C_6H_2O))_2$
Formula weight	830.57
Temperature	293(2)K
Wavelength	.71073Å
Crystal system	Monoclinic
Space group	$P2_1/c$

Unit cell dimensions	a	=	13.794(5)Å	α	=	90^o
	b	=	23.559(8)Å	β	=	$109.470(7)^o$
	c	=	19.477(6)Å	γ	=	90^o
	Volume	=	5967(3)Å^3			

Z	4
Density (calculated)	$0.925 Mg/m^3$
Absorption coefficient	$0.179 mm^{-1}$
F_{000}	1672
Crystal size	$0.1\times$ $0.15\times$ 0.3mm
Theta range for data collection	1.59 to 23.26^o
Index ranges	$-15 <= h <= 6$
	$-26 <= k <= 25$
	$-21 <= l <= 21$
Reflections collected	20731
Independent reflections	$8256(R_{int} = 0.0814)$
Refinement method	Full-matrix least-squares on F^2
Data / restraints / parameters	8256/ 0/ 714
Goodness-of-fit on F^2	1.323
Final R indices ($I > 2\sigma(I)$)	
R_1	0.1254
wR_2	0.2343
R indices (all data)	
R_1	0.1543
wR_2	0.2478
Largest diff. peak and hole	1.039 and -.398e$Å^{-3}$

Table 7.16: Atomic coordinates and equivalent isotropic displacement parameters for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}. U_{eq} is defined as one third of the trace of the orthogonalized U_{ij} tensor.

Atom	x	y	z	$U_{eq}/Å^2$
Ti(1)	.19725(9)	.23389(5)	.22942(7)	.0243(3)
O(1)	.1870(4)	.1764(2)	.1663(3)	.0321(12)
O(2)	.1485(3)	.2971(2)	.1768(2)	.0252(11)
O(4)	.3289(3)	.2487(2)	.2816(2)	.0275(12)
O(3)	.1262(4)	.2152(2)	.2883(3)	.0302(12)
C(5)	.1858(5)	.3480(3)	.1614(4)	.024(2)
C(6)	.1257(5)	.3970(3)	.1573(4)	.025(2)
C(7)	.0427(5)	.1905(3)	.2998(4)	.031(2)
C(8)	.4248(5)	.2653(3)	.2820(4)	.026(2)
C(9)	.6044(5)	.2752(3)	.3426(4)	.029(2)
C(10)	.4345(5)	.2892(3)	.2187(4)	.027(2)
C(11)	-.0185(5)	.1625(3)	.1658(4)	.030(2)
C(12)	.1653(6)	.4472(3)	.1407(4)	.034(2)
C(13)	.0313(5)	.1927(3)	.3693(4)	.029(2)
C(14)	-.0284(5)	.1618(3)	.2418(4)	.030(2)
C(15)	.2083(6)	.0776(3)	.1426(4)	.032(2)
C(16)	.0478(6)	.1136(3)	.1555(4)	.032(2)
C(17)	.5313(6)	.3053(3)	.2202(4)	.031(2)
C(18)	.6188(5)	.2993(3)	.2819(4)	.031(2)
C(19)	-.0503(5)	.1630(3)	.3779(4)	.034(2)
C(20)	.1475(6)	.1220(3)	.1543(4)	.030(2)
C(21)	.4999(5)	.2292(3)	.4143(4)	.032(2)
C(22)	-.1073(5)	.1331(3)	.2549(4)	.031(2)
C(23)	.2785(5)	.3495(3)	.1482(3)	.026(2)
C(24)	.4322(6)	.2662(4)	.4444(4)	.040(2)
C(25)	.3716(6)	.2905(3)	.0800(4)	.033(2)
C(26)	.5097(5)	.2567(3)	.3455(4)	.024(2)
C(27)	.3416(5)	.2953(3)	.1493(4)	.028(2)
C(28)	-.1206(6)	.1315(3)	.3229(4)	.033(2)
C(29)	.3133(5)	.4021(3)	.1321(4)	.030(2)
C(30)	.7239(6)	.3182(3)	.2787(4)	.037(2)
C(31)	.2584(6)	.4515(3)	.1269(4)	.030(2)
C(32)	.4560(6)	.1692(3)	.3971(4)	.034(2)
C(33)	.0225(5)	.3963(3)	.1723(4)	.031(2)
C(34)	.1071(6)	.2248(3)	.4335(4)	.035(2)
C(35)	.2097(6)	.1925(4)	.4587(4)	.043(2)
C(36)	.3155(6)	.0880(4)	.1379(4)	.040(2)
C(37)	.6038(6)	.2236(4)	.4748(4)	.047(2)
C(38)	.0106(6)	.0583(3)	.1466(4)	.039(2)

Table 7.17: Atomic coordinates for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV} (continued)

C(39)	.2915(6)	.5091(3)	.1063(4)	.038(2)
C(40)	-.0266(6)	.4550(3)	.1649(5)	.043(2)
C(41)	.0403(6)	.3743(4)	.2502(4)	.037(2)
C(42)	.1651(7)	.0245(3)	.1345(4)	.040(2)
C(43)	.1245(7)	.2863(3)	.4141(4)	.041(2)
C(44)	.2147(7)	.5261(4)	.0316(4)	.051(2)
C(45)	-.2087(6)	.0968(4)	.3347(4)	.041(2)
C(46)	.3992(7)	.5071(4)	.1001(6)	.058(3)
C(47)	-.1239(6)	.1635(4)	.1040(4)	.045(2)
C(48)	.0685(7)	.2285(4)	.4989(4)	.049(2)
C(49)	-.0539(6)	.3572(4)	.1163(4)	.042(2)
C(50)	.3073(7)	.1290(4)	.0743(5)	.048(2)
C(51)	.2898(8)	.5540(4)	.1618(5)	.053(3)
C(52)	.0681(8)	.0124(3)	.1375(4)	.047(2)
C(53)	.3875(6)	.1133(4)	.2097(5)	.051(2)
C(54)	.7584(7)	.2767(4)	.2314(6)	.064(3)
C(55)	.8051(7)	.3180(6)	.3545(6)	.082(4)
C(56)	.0280(9)	-.0482(4)	.1303(5)	.060(3)
C(57)	-.0270(14)	-.0634(5)	.0549(6)	.126(7)
C(58)	.3651(8)	.0327(4)	.1248(6)	.060(3)
C(59)	.7188(7)	.3771(4)	.2461(7)	.070(3)
C(60)	-.2133(12)	.1063(8)	.4109(7)	.108(5)
C(61)	-.1922(11)	.0359(4)	.3245(12)	.119(6)
C(62)	-.3100(7)	.1176(7)	.2819(8)	.104(5)
C(63)	.1230(12)	-.0916(5)	.1579(8)	.121(6)
C(64)	-.026(2)	-.0592(7)	.1820(11)	.154(8)

Table 7.18: Bond lengths for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}.

Atom		Atom	Length/Å
Ti(1)	-	O(4)	1.795(4)
Ti(1)	-	O(3)	1.795(5)
Ti(1)	-	O(1)	1.802(5)
Ti(1)	-	O(2)	1.807(5)
O(1)	-	C(20)	1.382(9)
O(2)	-	C(5)	1.376(8)
O(4)	-	C(8)	1.377(8)
O(3)	-	C(7)	1.373(8)
C(5)	-	C(23)	1.385(9)
C(5)	-	C(6)	1.408(9)
C(6)	-	C(12)	1.386(10)
C(6)	-	C(33)	1.545(10)
C(7)	-	C(14)	1.399(10)
C(7)	-	C(13)	1.415(10)
C(8)	-	C(10)	1.401(9)
C(8)	-	C(26)	1.406(9)
C(9)	-	C(18)	1.385(10)
C(9)	-	C(26)	1.396(9)
C(10)	-	C(17)	1.378(10)
C(10)	-	C(27)	1.529(9)
C(11)	-	C(16)	1.524(10)
C(11)	-	C(14)	1.530(10)
C(11)	-	C(47)	1.549(10)
C(12)	-	C(31)	1.402(10)
C(13)	-	C(19)	1.383(10)
C(13)	-	C(34)	1.535(10)
C(14)	-	C(22)	1.376(10)
C(15)	-	C(42)	1.373(11)
C(15)	-	C(20)	1.405(10)
C(15)	-	C(36)	1.532(11)
C(16)	-	C(38)	1.390(10)
C(16)	-	C(20)	1.398(10)
C(17)	-	C(18)	1.398(10)
C(18)	-	C(30)	1.537(10)
C(19)	-	C(28)	1.393(10)
C(21)	-	C(37)	1.527(10)
C(21)	-	C(24)	1.529(10)
C(21)	-	C(32)	1.531(10)
C(21)	-	C(26)	1.534(9)
C(22)	-	C(28)	1.396(10)

Table 7.19: bond lengths for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV} (continued)

C(23)	-	C(29)	1.402(10)
C(23)	-	C(27)	1.541(10)
C(25)	-	C(27)	1.541(9)
C(28)	-	C(45)	1.544(10)
C(29)	-	C(31)	1.373(10)
C(30)	-	C(59)	1.516(12)
C(30)	-	C(54)	1.525(11)
C(30)	-	C(55)	1.528(12)
C(31)	-	C(39)	1.526(10)
C(33)	-	C(40)	1.526(10)
C(33)	-	C(41)	1.543(10)
C(33)	-	C(49)	1.543(10)
C(34)	-	C(35)	1.536(10)
C(34)	-	C(43)	1.537(11)
C(34)	-	C(48)	1.538(10)
C(36)	-	C(58)	1.532(11)
C(36)	-	C(53)	1.538(11)
C(36)	-	C(50)	1.546(11)
C(38)	-	C(52)	1.387(12)
C(39)	-	C(51)	1.520(11)
C(39)	-	C(46)	1.531(12)
C(39)	-	C(44)	1.539(11)
C(42)	-	C(52)	1.388(12)
C(45)	-	C(61)	1.477(13)
C(45)	-	C(62)	1.514(13)
C(45)	-	C(60)	1.523(13)
C(52)	-	C(56)	1.519(11)
C(56)	-	C(57)	1.456(13)
C(56)	-	C(64)	1.46(2)
C(56)	-	C(63)	1.61(2)

Table 7.20: Bond angles for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}.

Atom		Atom		Atom	Angle/^{o}
O(4)	-	Ti(1)	-	O(3)	110.6(2)
O(4)	-	Ti(1)	-	O(1)	111.5(2)
O(3)	-	Ti(1)	-	O(1)	108.4(2)
O(4)	-	Ti(1)	-	O(2)	106.0(2)
O(3)	-	Ti(1)	-	O(2)	112.7(2)
O(1)	-	Ti(1)	-	O(2)	107.6(2)
C(20)	-	O(1)	-	Ti(1)	138.5(4)
C(5)	-	O(2)	-	Ti(1)	138.3(4)
C(8)	-	O(4)	-	Ti(1)	147.9(4)
C(7)	-	O(3)	-	Ti(1)	151.0(5)
O(2)	-	C(5)	-	C(23)	119.9(6)
O(2)	-	C(5)	-	C(6)	118.0(6)
C(23)	-	C(5)	-	C(6)	122.1(6)
C(12)	-	C(6)	-	C(5)	116.2(6)
C(12)	-	C(6)	-	C(33)	120.8(6)
C(5)	-	C(6)	-	C(33)	122.9(6)
O(3)	-	C(7)	-	C(14)	118.2(6)
O(3)	-	C(7)	-	C(13)	120.5(6)
C(14)	-	C(7)	-	C(13)	121.3(7)
O(4)	-	C(8)	-	C(10)	118.9(6)
O(4)	-	C(8)	-	C(26)	118.7(6)
C(10)	-	C(8)	-	C(26)	122.4(6)
C(18)	-	C(9)	-	C(26)	124.4(7)
C(17)	-	C(10)	-	C(8)	117.9(6)
C(17)	-	C(10)	-	C(27)	120.8(6)
C(8)	-	C(10)	-	C(27)	121.2(6)
C(16)	-	C(11)	-	C(14)	111.8(6)
C(16)	-	C(11)	-	C(47)	111.0(6)
C(14)	-	C(11)	-	C(47)	112.9(6)
C(6)	-	C(12)	-	C(31)	124.4(7)
C(19)	-	C(13)	-	C(7)	116.8(7)
C(19)	-	C(13)	-	C(34)	120.8(6)
C(7)	-	C(13)	-	C(34)	122.3(6)
C(22)	-	C(14)	-	C(7)	118.1(7)
C(22)	-	C(14)	-	C(11)	121.0(7)
C(7)	-	C(14)	-	C(11)	120.8(6)
C(42)	-	C(15)	-	C(20)	116.0(7)
C(42)	-	C(15)	-	C(36)	121.9(7)
C(20)	-	C(15)	-	C(36)	122.1(7)

Table 7.21: bond angles for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV} (continued)

C(38)	-	C(16)	-	C(20)	117.1(7)
C(38)	-	C(16)	-	C(11)	120.8(7)
C(20)	-	C(16)	-	C(11)	122.2(7)
C(10)	-	C(17)	-	C(18)	122.9(7)
C(9)	-	C(18)	-	C(17)	116.5(7)
C(9)	-	C(18)	-	C(30)	124.0(7)
C(17)	-	C(18)	-	C(30)	119.5(7)
C(13)	-	C(19)	-	C(28)	124.2(7)
O(1)	-	C(20)	-	C(16)	117.6(7)
O(1)	-	C(20)	-	C(15)	119.6(7)
C(16)	-	C(20)	-	C(15)	122.8(7)
C(37)	-	C(21)	-	C(24)	106.9(6)
C(37)	-	C(21)	-	C(32)	106.8(7)
C(24)	-	C(21)	-	C(32)	111.2(6)
C(37)	-	C(21)	-	C(26)	112.2(6)
C(24)	-	C(21)	-	C(26)	109.8(6)
C(32)	-	C(21)	-	C(26)	110.0(6)
C(14)	-	C(22)	-	C(28)	123.5(7)
C(5)	-	C(23)	-	C(29)	118.1(7)
C(5)	-	C(23)	-	C(27)	121.9(6)
C(29)	-	C(23)	-	C(27)	120.0(6)
C(9)	-	C(26)	-	C(8)	115.8(6)
C(9)	-	C(26)	-	C(21)	121.5(6)
C(8)	-	C(26)	-	C(21)	122.7(6)
C(10)	-	C(27)	-	C(23)	113.2(6)
C(10)	-	C(27)	-	C(25)	112.2(6)
C(23)	-	C(27)	-	C(25)	111.4(6)
C(19)	-	C(28)	-	C(22)	115.9(7)
C(19)	-	C(28)	-	C(45)	123.2(7)
C(22)	-	C(28)	-	C(45)	120.9(7)
C(31)	-	C(29)	-	C(23)	123.0(7)
C(59)	-	C(30)	-	C(54)	108.3(8)
C(59)	-	C(30)	-	C(55)	108.8(9)
C(54)	-	C(30)	-	C(55)	107.8(8)
C(59)	-	C(30)	-	C(18)	111.7(6)
C(54)	-	C(30)	-	C(18)	109.1(7)
C(55)	-	C(30)	-	C(18)	111.0(7)
C(29)	-	C(31)	-	C(12)	116.3(7)
C(29)	-	C(31)	-	C(39)	124.5(7)
C(12)	-	C(31)	-	C(39)	119.2(7)
C(40)	-	C(33)	-	C(41)	108.4(7)

Table 7.22: bond angles for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV} (continued)

C(40)	-	C(33)	-	C(49)	107.2(6)
C(41)	-	C(33)	-	C(49)	110.0(6)
C(40)	-	C(33)	-	C(6)	112.4(6)
C(41)	-	C(33)	-	C(6)	109.6(6)
C(49)	-	C(33)	-	C(6)	109.1(6)
C(13)	-	C(34)	-	C(35)	108.7(6)
C(13)	-	C(34)	-	C(43)	112.5(6)
C(35)	-	C(34)	-	C(43)	110.0(7)
C(13)	-	C(34)	-	C(48)	111.9(6)
C(35)	-	C(34)	-	C(48)	107.5(7)
C(43)	-	C(34)	-	C(48)	106.0(7)
C(58)	-	C(36)	-	C(15)	111.6(7)
C(58)	-	C(36)	-	C(53)	107.0(7)
C(15)	-	C(36)	-	C(53)	110.5(6)
C(58)	-	C(36)	-	C(50)	108.3(7)
C(15)	-	C(36)	-	C(50)	109.5(6)
C(53)	-	C(36)	-	C(50)	109.8(7)
C(52)	-	C(38)	-	C(16)	122.9(8)
C(51)	-	C(39)	-	C(31)	110.8(7)
C(51)	-	C(39)	-	C(46)	108.8(7)
C(31)	-	C(39)	-	C(46)	112.1(7)
C(51)	-	C(39)	-	C(44)	108.7(7)
C(31)	-	C(39)	-	C(44)	107.9(6)
C(46)	-	C(39)	-	C(44)	108.3(7)
C(15)	-	C(42)	-	C(52)	124.7(8)
C(61)	-	C(45)	-	C(62)	111.8(12)
C(61)	-	C(45)	-	C(60)	109.4(12)
C(62)	-	C(45)	-	C(60)	106.6(11)
C(61)	-	C(45)	-	C(28)	109.3(8)
C(62)	-	C(45)	-	C(28)	108.8(7)
C(60)	-	C(45)	-	C(28)	110.9(7)
C(38)	-	C(52)	-	C(42)	116.5(7)
C(38)	-	C(52)	-	C(56)	122.5(8)
C(42)	-	C(52)	-	C(56)	121.0(8)
C(57)	-	C(56)	-	C(64)	116.1(13)
C(57)	-	C(56)	-	C(52)	112.1(8)
C(64)	-	C(56)	-	C(52)	110.9(9)
C(57)	-	C(56)	-	C(63)	105.8(11)
C(64)	-	C(56)	-	C(63)	101.5(13)
C(52)	-	C(56)	-	C(63)	109.6(9)

Table 7.23: Anisotropic displacement parameters ($Å^2$) for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}.
The anisotropic displacement factor exponent takes the form:
$-2\pi^2(h^2a*^2 U_{11} + ... + 2hka*b* U_{12})$

Atom	$U_{11}/Å^2$	$U_{22}/Å^2$	$U_{33}/Å^2$	$U_{23}/Å^2$	$U_{13}/Å^2$	$U_{12}/Å^2$
Ti(1)	.0239(7)	.0256(7)	.0251(7)	.0009(6)	.0106(5)	-.0007(6)
O(1)	.028(3)	.034(3)	.037(3)	.000(2)	.014(2)	-.003(2)
O(2)	.022(2)	.026(3)	.030(3)	.000(2)	.012(2)	-.001(2)
O(4)	.023(3)	.031(3)	.029(3)	.003(2)	.009(2)	-.001(2)
O(3)	.029(3)	.033(3)	.032(3)	.004(2)	.015(2)	-.004(2)
C(5)	.036(4)	.010(4)	.021(4)	.000(3)	.005(3)	.004(3)
C(6)	.029(4)	.025(4)	.020(4)	-.002(3)	.008(3)	-.001(3)
C(7)	.025(4)	.037(5)	.031(4)	.004(4)	.011(3)	.003(3)
C(8)	.021(4)	.030(4)	.031(4)	.004(3)	.013(3)	-.001(3)
C(9)	.027(4)	.033(5)	.026(4)	-.004(3)	.006(3)	.004(3)
C(10)	.020(4)	.032(4)	.032(4)	.002(3)	.014(3)	.002(3)
C(11)	.027(4)	.026(4)	.036(4)	-.002(3)	.009(3)	-.008(3)
C(12)	.036(5)	.030(5)	.030(4)	.000(4)	.003(4)	.010(4)
C(13)	.025(4)	.033(4)	.031(4)	.002(3)	.014(3)	.000(3)
C(14)	.030(4)	.040(5)	.026(4)	.001(3)	.016(3)	.003(4)
C(15)	.053(5)	.028(5)	.016(4)	-.006(3)	.014(4)	.010(4)
C(16)	.037(5)	.037(5)	.018(4)	-.006(3)	.005(3)	-.006(4)
C(17)	.042(5)	.028(4)	.025(4)	.004(3)	.015(4)	.002(4)
C(18)	.030(4)	.034(5)	.033(4)	-.003(4)	.016(3)	.004(3)
C(19)	.029(4)	.044(5)	.027(4)	.003(4)	.007(3)	.002(4)
C(20)	.041(5)	.032(5)	.017(4)	-.004(3)	.009(3)	-.001(4)
C(21)	.032(4)	.037(5)	.028(4)	.000(4)	.009(3)	.002(4)
C(22)	.025(4)	.029(4)	.040(5)	-.002(4)	.012(4)	-.008(3)
C(23)	.012(3)	.047(5)	.017(4)	.000(3)	.003(3)	.003(3)
C(24)	.045(5)	.044(5)	.030(4)	.002(4)	.012(4)	-.005(4)
C(25)	.028(4)	.049(5)	.025(4)	.003(4)	.014(3)	.002(4)
C(26)	.022(4)	.028(4)	.025(4)	.003(3)	.013(3)	.002(3)
C(27)	.020(4)	.039(5)	.028(4)	.008(3)	.010(3)	.003(3)
C(28)	.031(4)	.038(5)	.038(5)	.012(4)	.020(4)	.009(4)
C(29)	.025(4)	.036(5)	.026(4)	.012(3)	.006(3)	-.008(3)
C(30)	.026(4)	.043(5)	.044(5)	.007(4)	.015(4)	.003(4)
C(31)	.036(4)	.028(4)	.026(4)	.005(3)	.009(3)	-.002(4)
C(32)	.044(5)	.036(5)	.026(4)	.007(4)	.016(4)	.002(4)
C(33)	.029(4)	.031(4)	.032(4)	.000(4)	.006(3)	.001(3)
C(34)	.036(4)	.035(5)	.031(4)	.002(4)	.007(4)	-.001(4)
C(35)	.041(5)	.039(5)	.039(5)	.000(4)	.000(4)	-.004(4)
C(36)	.042(5)	.047(5)	.039(5)	-.003(4)	.022(4)	.008(4)
C(37)	.034(5)	.065(7)	.033(5)	.011(5)	-.002(4)	.001(5)

Table 7.24: Anisotropic displacement parameters for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV} (continued)

C(38)	.045(5)	.051(6)	.027(4)	-.008(4)	.021(4)	-.026(4)
C(39)	.043(5)	.034(5)	.034(4)	.000(4)	.009(4)	-.009(4)
C(40)	.032(5)	.045(5)	.058(6)	.001(5)	.021(4)	.008(4)
C(41)	.026(4)	.050(6)	.042(5)	-.003(4)	.020(4)	-.005(4)
C(42)	.075(6)	.018(4)	.035(5)	-.004(3)	.029(4)	.007(4)
C(43)	.045(5)	.041(5)	.037(5)	-.004(4)	.014(4)	-.003(4)
C(44)	.068(7)	.036(6)	.037(5)	.007(4)	.003(5)	.003(5)
C(45)	.030(4)	.049(5)	.045(5)	.005(4)	.014(4)	-.012(4)
C(46)	.057(6)	.042(6)	.077(7)	.017(6)	.026(6)	-.013(5)
C(47)	.030(4)	.069(7)	.034(5)	.004(4)	.009(4)	-.005(5)
C(48)	.057(6)	.066(7)	.035(5)	-.007(5)	.027(4)	-.009(5)
C(49)	.027(4)	.051(6)	.047(5)	-.001(4)	.013(4)	-.003(4)
C(50)	.044(5)	.059(6)	.051(6)	-.005(5)	.028(5)	-.006(5)
C(51)	.071(7)	.039(6)	.040(5)	.012(4)	.007(5)	-.014(5)
C(52)	.086(7)	.030(5)	.037(5)	-.014(4)	.035(5)	-.014(5)
C(53)	.034(5)	.062(7)	.054(6)	-.008(5)	.011(4)	.012(4)
C(54)	.045(6)	.061(7)	.107(10)	-.006(6)	.054(7)	-.008(5)
C(55)	.026(5)	.144(12)	.071(7)	.012(9)	.008(5)	-.023(6)
C(56)	.108(8)	.025(5)	.066(6)	-.015(5)	.054(6)	-.024(5)
C(57)	.20(2)	.050(8)	.065(8)	.010(6)	-.043(10)	-.082(9)
C(58)	.063(7)	.054(6)	.068(7)	-.011(6)	.030(6)	.020(5)
C(59)	.028(5)	.058(7)	.129(10)	.017(7)	.032(7)	-.006(5)
C(60)	.123(12)	.15(2)	.085(9)	-.034(9)	.074(9)	-.086(10)
C(61)	.086(10)	.034(7)	.28(2)	.003(9)	.112(12)	-.008(6)
C(62)	.031(6)	.17(2)	.104(11)	.066(10)	.019(6)	-.017(7)
C(63)	.17(2)	.028(7)	.107(11)	.020(6)	-.028(10)	-.008(8)
C(64)	.26(2)	.091(12)	.15(2)	-.026(11)	.14(2)	-.10(2)

Table 7.25: Hydrogen coordinates and isotropic displacement parameters for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV}.

Atom	x	y	z	U_{eq}/Å^2
H(9A)	.6634(5)	.2709(3)	.3857(4)	.03(2)
H(11A)	.0163(5)	.1970(3)	.1619(4)	.01(2)
H(12A)	.1266(6)	.4814(3)	.1387(4)	.01(2)
H(17A)	.5392(6)	.3208(3)	.1768(4)	.001(14)
H(19A)	-.0597(5)	.1645(3)	.4245(4)	.03(2)
H(22A)	-.1560(5)	.1134(3)	.2151(4)	.04(2)
H(24A)	.4610(6)	.3036(4)	.4549(4)	.03(2)
H(24B)	.3646(6)	.2686(4)	.4088(4)	.03(2)
H(24C)	.4278(6)	.2494(4)	.4882(4)	.05(3)
H(25A)	.4104(6)	.2564(3)	.0818(4)	.07(3)
H(25B)	.3107(6)	.2897(3)	.0378(4)	.06(3)
H(25C)	.4127(6)	.3228(3)	.0774(4)	.04(2)
H(27A)	.2971(5)	.2638(3)	.1483(4)	.06(3)
H(29A)	.3788(5)	.4038(3)	.1249(4)	.04(2)
H(32A)	.5001(6)	.1469(3)	.3786(4)	.05(3)
H(32B)	.4518(6)	.1519(3)	.4407(4)	.05(2)
H(32C)	.3886(6)	.1711(3)	.3613(4)	.03(2)
H(35A)	.2362(6)	.1898(4)	.4190(4)	.06(3)
H(35B)	.2583(6)	.2122(4)	.4986(4)	.002(14)
H(35C)	.1986(6)	.1550(4)	.4740(4)	.06(3)
H(37A)	.6343(6)	.2604(4)	.4874(4)	.03(2)
H(37B)	.5943(6)	.2065(4)	.5168(4)	.05(2)
H(37C)	.6481(6)	.2001(4)	.4580(4)	.04(2)
H(38A)	-.0581(6)	.0513(3)	.1465(4)	.02(2)
H(40A)	.0189(6)	.4804(3)	.1993(5)	.06(3)
H(40B)	-.0387(6)	.4688(3)	.1164(5)	.07(3)
H(40C)	-.0907(6)	.4527(3)	.1741(5)	.04(2)
H(41A)	.0876(6)	.3991(4)	.2844(4)	.05(2)
H(41B)	-.0236(6)	.3735(4)	.2597(4)	.02(2)
H(41C)	.0687(6)	.3367(4)	.2549(4)	.08(3)
H(42A)	.2049(7)	-.0067(3)	.1266(4)	.03(2)
H(43A)	.0602(7)	.3063(3)	.3986(4)	.08(3)
H(43B)	.1713(7)	.3048(3)	.4560(4)	.06(3)
H(43C)	.1532(7)	.2861(3)	.3754(4)	.05(3)
H(44A)	.1468(7)	.5280(4)	.0348(4)	.10(4)
H(44B)	.2329(7)	.5624(4)	.0170(4)	.09(4)
H(44C)	.2163(7)	.4979(4)	-.0035(4)	.05(3)
H(46A)	.4482(7)	.4971(4)	.1465(6)	.05(3)
H(46B)	.4002(7)	.4790(4)	.0647(6)	.10(4)
H(46C)	.4168(7)	.5434(4)	.0852(6)	.10(4)
H(47A)	-.1140(6)	.1644(4)	.0575(4)	.06(3)
H(47B)	-.1612(6)	.1299(4)	.1075(4)	.06(3)
H(47C)	-.1622(6)	.1964(4)	.1089(4)	.04(2)

Table 7.26: Hydrogen atom coordinates for bis(O,O'-1,1-di(1-oxo-2,4-di-*t*-butylphenyl)ethane) titanium^{IV} (continued)

H(48A)	.0045(7)	.2487(4)	.4851(4)	.08(4)
H(48B)	.0585(7)	.1907(4)	.5139(4)	.05(3)
H(48C)	.1182(7)	.2479(4)	.5385(4)	.07(3)
H(49A)	-.0648(6)	.3714(4)	.0681(4)	.11(4)
H(49B)	-.0259(6)	.3196(4)	.1205(4)	.04(2)
H(49C)	-.1182(6)	.3563(4)	.1253(4)	.04(2)
H(50A)	.2763(7)	.1640(4)	.0815(5)	.08(4)
H(50B)	.3747(7)	.1366(4)	.0721(5)	.04(2)
H(50C)	.2656(7)	.1119(4)	.0295(5)	.04(2)
H(51A)	.3380(8)	.5439(4)	.2084(5)	.06(3)
H(51B)	.3079(8)	.5904(4)	.1473(5)	.04(2)
H(51C)	.2219(8)	.5559(4)	.1650(5)	.06(3)
H(53A)	.3916(6)	.0872(4)	.2484(5)	.05(3)
H(53B)	.4548(6)	.1188(4)	.2065(5)	.05(3)
H(53C)	.3609(6)	.1489(4)	.2193(5)	.06(3)
H(54A)	.7085(7)	.2762(4)	.1833(6)	.09(4)
H(54B)	.8240(7)	.2880(4)	.2287(6)	.06(3)
H(54C)	.7640(7)	.2394(4)	.2523(6)	.07(3)
H(55A)	.7853(7)	.3442(6)	.3851(6)	.12(5)
H(55B)	.8105(7)	.2806(6)	.3748(6)	.10(5)
H(55C)	.8704(7)	.3292(6)	.3512(6)	.05(3)
H(57A)	-.0860(14)	-.0397(5)	.0340(6)	.4(2)
H(57B)	.0204(14)	-.0583(5)	.0288(6)	.05(3)
H(57C)	-.0482(14)	-.1024(5)	.0517(6)	.11(4)
H(58A)	.3712(8)	.0068(4)	.1640(6)	.14(6)
H(58B)	.3231(8)	.0160(4)	.0797(6)	.09(4)
H(58C)	.4321(8)	.0407(4)	.1224(6)	.12(5)
H(59A)	.6977(7)	.4037(4)	.2757(7)	.09(4)
H(59B)	.7852(7)	.3877(4)	.2444(7)	.06(3)
H(59C)	.6698(7)	.3771(4)	.1977(7)	.10(4)
H(60A)	-.2683(12)	.0845(8)	.4176(7)	.17(6)
H(60B)	-.1494(12)	.0955(8)	.4470(7)	.4(2)
H(60C)	-.2255(12)	.1460(8)	.4158(7)	.18(8)
H(61A)	-.2466(11)	.0138(4)	.3316(12)	.19(7)
H(61B)	-.1918(11)	.0305(4)	.2757(12)	.32(14)
H(61C)	-.1275(11)	.0240(4)	.3585(12)	.13(6)
H(62A)	-.3656(7)	.0964(7)	.2886(8)	.08(3)
H(62B)	-.3180(7)	.1571(7)	.2911(8)	.4(2)
H(62C)	-.3105(7)	.1129(7)	.2329(8)	.19(8)
H(63A)	.0983(12)	-.1300(5)	.1540(8)	.13(5)
H(63B)	.1655(12)	-.0869(5)	.1280(8)	2(2)
H(63C)	.1626(12)	-.0836(5)	.2077(8)	.31(13)
H(64A)	-.086(2)	-.0358(7)	.1722(11)	.6(4)
H(64B)	-.045(2)	-.0984(7)	.1794(11)	.13(5)
H(64C)	.021(2)	-.0509(7)	.2298(11)	.16(7)

7.1.4 bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV})

Published in J. Organomet. Chem. [42].

The sample of this compound was mounted at low temperature on a glass fibre in perfluoropolyether oil. The data were collected using an Siemens SMART CCD area detector on a three-circle diffractometer using molybdenum Kα radiation. The area detector frames were integrated and processed with the SAINT program, the resulting reflection data processed with XPREP, and the structure solved by SHELXS using direct methods, all on a Digital Alpha workstation. The model was refined to convergence using SHELXTL-PLUS on a Silicon Graphics Indigo workstation.

This molecule appears as two half molecules in the asymmetric unit, with a centre of inversion occuring close to each titanium centre providing the full molecules.

Figure 7.8: Partial structure diagram for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}), showing first symmetry-expanded molecule, hydrogen atoms omitted for clarity.

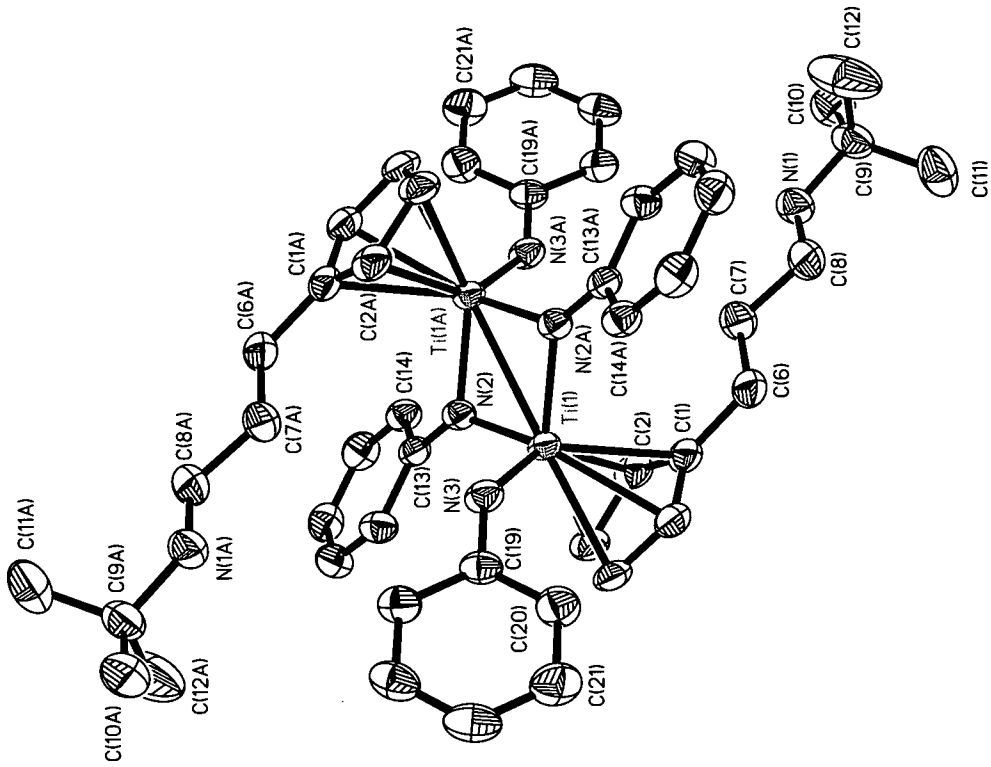


Figure 7.9: Partial structure diagram for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}), showing second symmetry-expanded molecule, hydrogen atoms omitted for clarity.

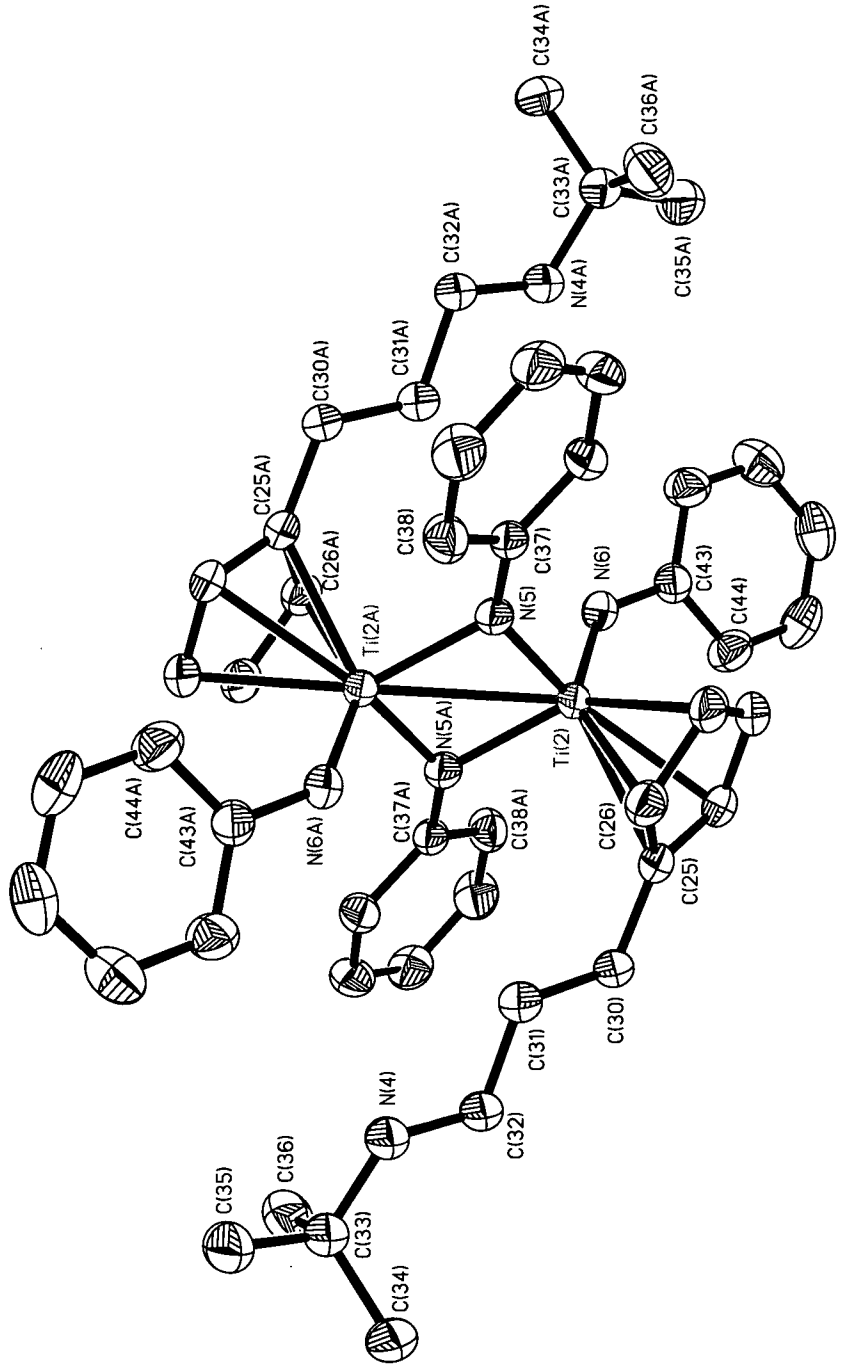


Figure 7.10: Unit cell diagram for , showing relationship between the two molecules in the asymmetric unit, hydrogen atoms omitted for clarity.

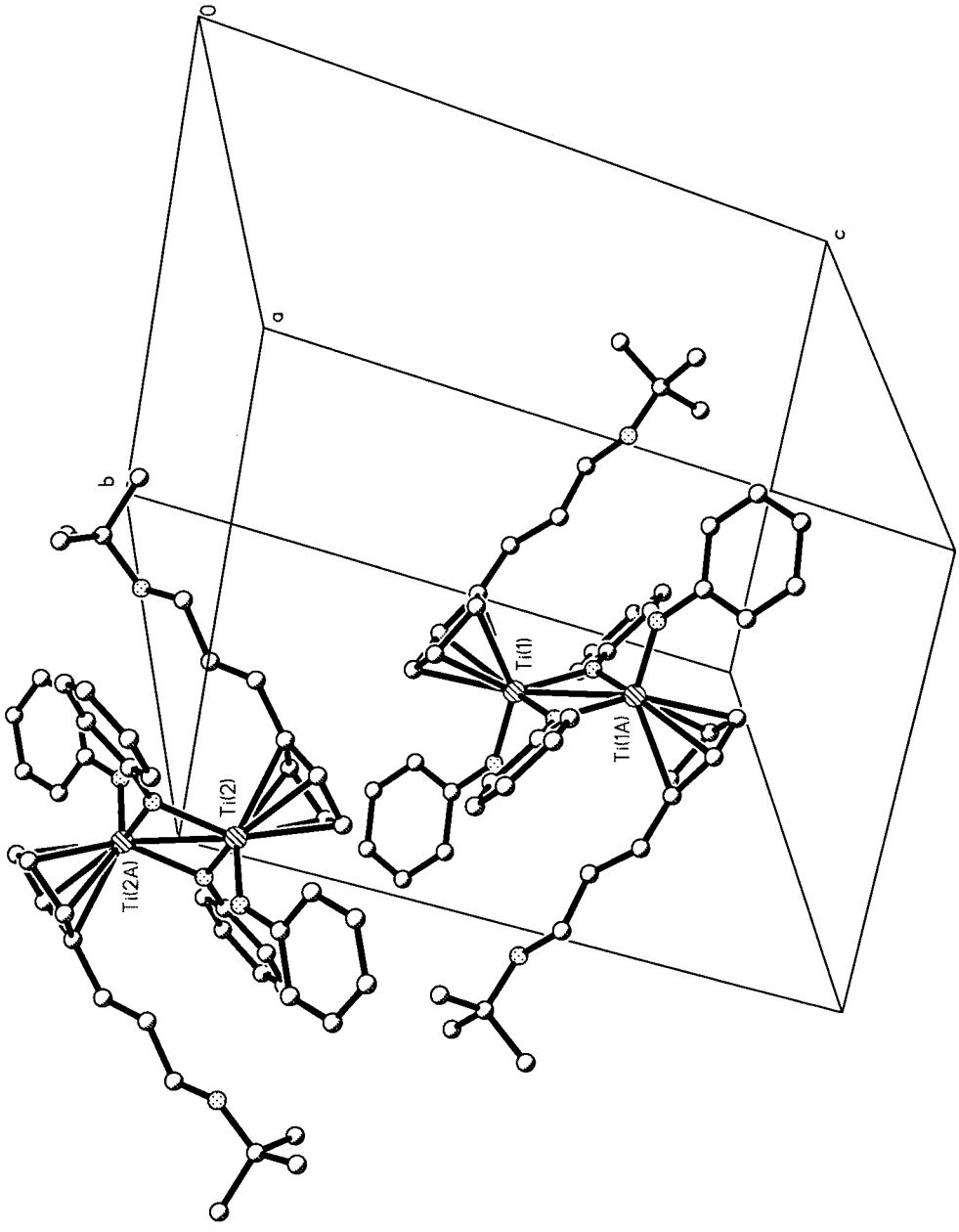
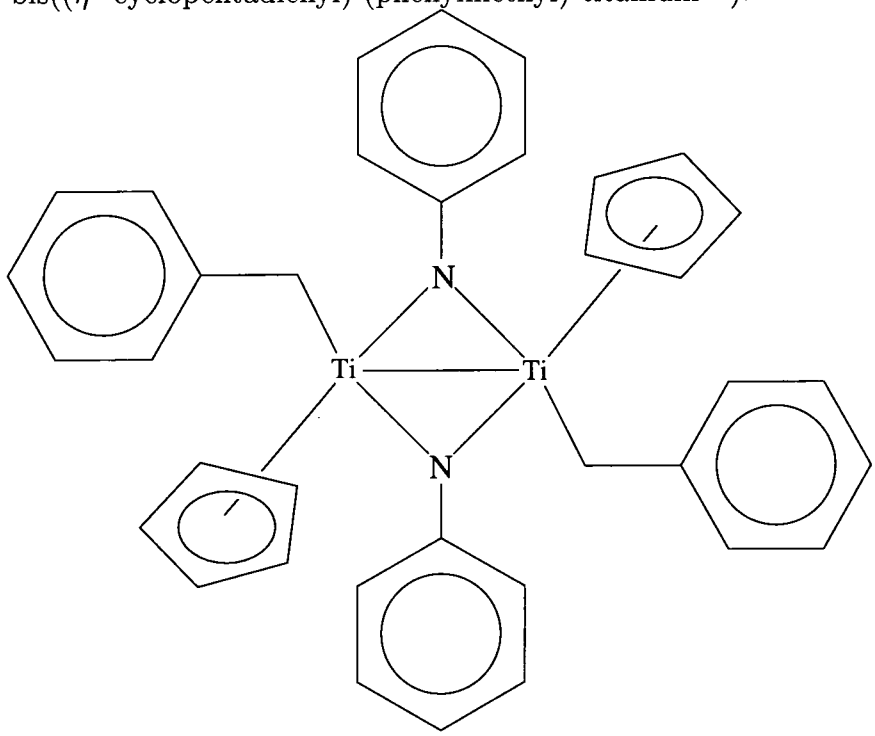


Figure 7.11: Two dimensional schematic diagram for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).



Crystal data and structure refinement for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).

Identification code	96srv023
Empirical formula	C_{48} H_{62} N_6 Ti_2
Structural formula	(η^5-Cp) ($PhCH_2$) Ti (μ-NPh)$_2$ Ti ($PhCH_2$) (η^5-Cp)
Formula weight	818.84
Temperature	293(2)K
Wavelength	.71073Å
Crystal system	Triclinic
Space group	$P\bar{1}$

Unit cell dimensions						
	a	=	11.57050(10)Å	α	=	78.3580(10)^o
	b	=	12.0315(2)Å	β	=	83.0390(10)^o
	c	=	17.2718(3)Å	γ	=	67.9640(10)^o
	Volume	=	2180.18(6)Å^3			

Z	2
Density (calculated)	$1.247 Mg/m^3$
Absorption coefficient	$0.406 mm^{-1}$
F_{000}	872
Crystal size	0.2× 0.1× 0.1 mm
Theta range for data collection	3.46 to 34.56°
Index ranges	$-18 <= h <= 11$
	$-18 <= k <= 13$
	$-22 <= l <= 24$
Reflections collected	16296
Independent reflections	11067(R_{int} = 0.0877)
Refinement method	Full-matrix least-squares on F^2
Data / restraints / parameters	10912/ 0/ 575
Goodness-of-fit on F^2	1.116
Final R indices ($I > 2\sigma(I)$)	
R_1	0.0647
wR_2	0.1615
R indices (all data)	
R_1	0.0880
wR_2	0.2183
Largest diff. peak and hole	.677 and -.929e$Å^{-3}$

Table 7.27: Atomic coordinates and equivalent isotropic displacement parameters for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).
U_{eq} is defined as one third of the trace of the orthogonalized U_{ij} tensor.

Atom	x	y	z	$U_{eq}/Å^2$
Ti(1)	.93051(5)	.57534(4)	.43314(3)	.02661(13)
Ti(2)	1.07147(4)	.91446(4)	.06285(3)	.02623(13)
N(5)	.9724(2)	1.0822(2)	.04682(13)	.0284(5)
N(3)	.9032(3)	.7474(2)	.4315(2)	.0345(5)
N(2)	1.1020(2)	.4926(2)	.45493(13)	.0295(5)
N(6)	1.2420(2)	.9159(2)	.0534(2)	.0346(5)
N(4)	.6889(3)	.7409(2)	-.0553(2)	.0372(5)
C(37)	.9176(3)	1.1764(2)	.0913(2)	.0274(5)
C(43)	1.3578(3)	.8586(3)	.0890(2)	.0325(6)
C(13)	1.2166(3)	.4629(2)	.4113(2)	.0292(5)
C(18)	1.2424(3)	.5533(3)	.3536(2)	.0362(6)
C(19)	.8488(3)	.8665(2)	.3917(2)	.0329(6)
C(26)	.9175(3)	.8590(3)	.1460(2)	.0345(6)
C(31)	.8424(3)	.7435(3)	.0282(2)	.0335(6)
C(29)	1.1164(3)	.7197(2)	.1478(2)	.0335(6)
C(42)	.9881(3)	1.1930(3)	.1461(2)	.0333(6)
C(30)	.9620(3)	.6742(3)	.0727(2)	.0345(6)
C(25)	.9973(3)	.7500(2)	.1187(2)	.0315(6)
N(1)	.7794(3)	.1081(3)	.5559(2)	.0436(6)
C(14)	1.3093(3)	.3457(3)	.4235(2)	.0349(6)
C(3)	.9657(3)	.5238(3)	.3056(2)	.0395(7)
C(4)	.8636(3)	.6330(3)	.3013(2)	.0402(7)
C(38)	.7939(3)	1.2582(3)	.0818(2)	.0367(6)
C(5)	.7643(3)	.6108(3)	.3500(2)	.0399(7)
C(32)	.8043(3)	.6648(3)	-.0146(2)	.0376(6)
C(34)	.6208(3)	.5713(3)	-.0716(2)	.0438(7)
C(41)	.9369(3)	1.2897(3)	.1877(2)	.0400(7)
C(24)	.9082(3)	.9510(3)	.3854(2)	.0379(6)

Table 7.28: Atomic coordinates for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

C(33)	.6475(3)	.6866(3)	-.1111(2)	.0385(7)
C(27)	.9875(3)	.8925(3)	.1933(2)	.0387(7)
C(40)	.8159(3)	1.3706(3)	.1771(2)	.0419(7)
C(23)	.8551(4)	1.0689(3)	.3445(2)	.0437(8)
C(28)	1.1098(3)	.8068(3)	.1945(2)	.0391(7)
C(1)	.8048(3)	.4859(3)	.3845(2)	.0387(7)
C(7)	.7918(3)	.2975(3)	.4793(2)	.0439(7)
C(20)	.7342(3)	.9063(3)	.3568(2)	.0426(7)
C(2)	.9300(3)	.4322(3)	.3578(2)	.0392(7)
C(22)	.7438(4)	1.1062(3)	.3088(2)	.0481(8)
C(17)	1.3568(3)	.5268(3)	.3118(2)	.0426(7)
C(16)	1.4476(3)	.4112(3)	.3243(2)	.0424(7)
C(44)	1.4089(3)	.7319(3)	.1107(2)	.0416(7)
C(48)	1.4248(3)	.9270(3)	.1040(2)	.0412(7)
C(15)	1.4227(3)	.3204(3)	.3803(2)	.0431(7)
C(6)	.7215(3)	.4244(3)	.4351(2)	.0460(8)
C(8)	.7052(4)	.2345(3)	.5232(2)	.0479(8)
C(39)	.7448(3)	1.3537(3)	.1241(2)	.0447(7)
C(9)	.7111(4)	.0339(3)	.6043(2)	.0483(8)
C(47)	1.5367(3)	.8700(4)	.1419(2)	.0500(8)
C(35)	.5259(4)	.7843(3)	-.1436(3)	.0542(9)
C(45)	1.5204(3)	.6768(3)	.1496(2)	.0471(8)
C(46)	1.5844(3)	.7436(4)	.1658(2)	.0494(8)
C(21)	.6829(3)	1.0242(3)	.3148(2)	.0498(8)
C(10)	.8070(5)	-.0960(4)	.6210(3)	.0635(11)
C(11)	.6053(4)	.0310(4)	.5594(3)	.0636(11)
C(36)	.7456(4)	.6571(4)	-.1789(2)	.0533(9)
C(12)	.6610(8)	.0806(5)	.6812(3)	.103(2)

Table 7.29: Bond lengths for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).

Atom		Atom	Length/Å
Ti(1)	-	N(2)	1.902(2)
Ti(1)	-	N(3)	1.969(2)
Ti(1)	-	N(2)^a	1.991(2)
Ti(1)	-	C(3)	2.356(3)
Ti(1)	-	C(2)	2.361(3)
Ti(1)	-	C(4)	2.377(3)
Ti(1)	-	C(5)	2.398(3)
Ti(1)	-	C(1)	2.412(3)
Ti(1)	-	Ti(1)^a	2.8688(9)
Ti(2)	-	N(5)	1.891(2)
Ti(2)	-	N(6)	1.967(3)
Ti(2)	-	N(5)^b	2.008(2)
Ti(2)	-	C(27)	2.348(3)
Ti(2)	-	C(26)	2.359(3)
Ti(2)	-	C(28)	2.381(3)
Ti(2)	-	C(29)	2.408(3)
Ti(2)	-	C(25)	2.426(3)
Ti(2)	-	Ti(2)^b	2.8633(9)
N(5)	-	C(37)	1.405(3)
N(5)	-	Ti(2)^b	2.008(2)
N(3)	-	C(19)	1.397(3)
N(2)	-	C(13)	1.402(4)
N(2)	-	Ti(1)^a	1.991(2)
N(6)	-	C(43)	1.408(4)
N(4)	-	C(32)	1.471(4)
N(4)	-	C(33)	1.480(4)
C(37)	-	C(38)	1.407(4)
C(37)	-	C(42)	1.410(4)
C(43)	-	C(48)	1.399(4)
C(43)	-	C(44)	1.401(4)
C(13)	-	C(14)	1.408(4)
C(13)	-	C(18)	1.414(4)
C(18)	-	C(17)	1.385(4)
C(19)	-	C(20)	1.398(5)
C(19)	-	C(24)	1.407(4)
C(26)	-	C(27)	1.422(4)
C(26)	-	C(25)	1.425(4)
C(31)	-	C(32)	1.520(4)
C(31)	-	C(30)	1.528(4)

Table 7.30: Bond lengths for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

C(29)	-	C(25)	1.416(4)
C(29)	-	C(28)	1.420(4)
C(42)	-	C(41)	1.391(4)
C(30)	-	C(25)	1.510(4)
N(1)	-	C(8)	1.470(4)
N(1)	-	C(9)	1.480(4)
C(14)	-	C(15)	1.388(5)
C(3)	-	C(4)	1.395(5)
C(3)	-	C(2)	1.431(4)
C(4)	-	C(5)	1.416(5)
C(38)	-	C(39)	1.388(4)
C(5)	-	C(1)	1.418(4)
C(34)	-	C(33)	1.540(4)
C(41)	-	C(40)	1.381(5)
C(24)	-	C(23)	1.391(4)
C(33)	-	C(36)	1.526(5)
C(33)	-	C(35)	1.536(5)
C(27)	-	C(28)	1.404(5)
C(40)	-	C(39)	1.392(5)
C(23)	-	C(22)	1.374(6)
C(1)	-	C(2)	1.408(5)
C(1)	-	C(6)	1.520(4)
C(7)	-	C(8)	1.520(4)
C(7)	-	C(6)	1.524(5)
C(20)	-	C(21)	1.395(4)
C(22)	-	C(21)	1.394(5)
C(17)	-	C(16)	1.387(5)
C(16)	-	C(15)	1.398(5)
C(44)	-	C(45)	1.394(5)
C(48)	-	C(47)	1.391(5)
C(9)	-	C(12)	1.513(6)
C(9)	-	C(10)	1.533(6)
C(9)	-	C(11)	1.540(6)
C(47)	-	C(46)	1.400(5)
C(45)	-	C(46)	1.365(5)

Symmetry transformations used to generate equivalent atoms:
[a] -x+2,-y+1,-z+1
[b] -x+2,-y+2,-z

Table 7.31: Bond angles for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).

Atom		Atom		Atom	Angle/^{o}
N(2)	-	Ti(1)	-	N(3)	103.89(10)
N(2)	-	Ti(1)	-	N(2)^a	85.08(10)
N(3)	-	Ti(1)	-	N(2)^a	106.06(10)
N(2)	-	Ti(1)	-	C(3)	91.70(11)
N(3)	-	Ti(1)	-	C(3)	112.76(11)
N(2)^a	-	Ti(1)	-	C(3)	140.64(10)
N(2)	-	Ti(1)	-	C(2)	93.17(11)
N(3)	-	Ti(1)	-	C(2)	145.09(11)
N(2)^a	-	Ti(1)	-	C(2)	105.57(10)
C(3)	-	Ti(1)	-	C(2)	35.33(11)
N(2)	-	Ti(1)	-	C(4)	121.33(11)
N(3)	-	Ti(1)	-	C(4)	87.57(11)
N(2)^a	-	Ti(1)	-	C(4)	147.02(11)
C(3)	-	Ti(1)	-	C(4)	34.28(12)
C(2)	-	Ti(1)	-	C(4)	57.65(11)
N(2)	-	Ti(1)	-	C(5)	147.75(11)
N(3)	-	Ti(1)	-	C(5)	96.74(11)
N(2)^a	-	Ti(1)	-	C(5)	112.86(10)
C(3)	-	Ti(1)	-	C(5)	57.18(12)
C(2)	-	Ti(1)	-	C(5)	57.05(12)
C(4)	-	Ti(1)	-	C(5)	34.49(11)
N(2)	-	Ti(1)	-	C(1)	124.06(11)
N(3)	-	Ti(1)	-	C(1)	130.05(11)
N(2)^a	-	Ti(1)	-	C(1)	92.21(10)
C(3)	-	Ti(1)	-	C(1)	57.75(11)
C(2)	-	Ti(1)	-	C(1)	34.31(12)
C(4)	-	Ti(1)	-	C(1)	57.47(10)
C(5)	-	Ti(1)	-	C(1)	34.29(10)
N(2)	-	Ti(1)	-	Ti(1)^a	43.74(7)
N(3)	-	Ti(1)	-	Ti(1)^a	110.55(8)
N(2)^a	-	Ti(1)	-	Ti(1)^a	41.34(7)
C(3)	-	Ti(1)	-	Ti(1)^a	123.80(9)
C(2)	-	Ti(1)	-	Ti(1)^a	102.88(8)
C(4)	-	Ti(1)	-	Ti(1)^a	157.94(9)
C(5)	-	Ti(1)	-	Ti(1)^a	146.12(8)
C(1)	-	Ti(1)	-	Ti(1)^a	113.46(7)
N(5)	-	Ti(2)	-	N(6)	102.53(10)

Table 7.32: Bond angles for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

N(5)	-	Ti(2)	-	N(5)^b	85.57(10)
N(6)	-	Ti(2)	-	N(5)^b	106.41(10)
N(5)	-	Ti(2)	-	C(27)	90.17(10)
N(6)	-	Ti(2)	-	C(27)	114.54(11)
N(5)^b	-	Ti(2)	-	C(27)	138.75(11)
N(5)	-	Ti(2)	-	C(26)	93.61(10)
N(6)	-	Ti(2)	-	C(26)	146.33(10)
N(5)^b	-	Ti(2)	-	C(26)	104.12(10)
C(27)	-	Ti(2)	-	C(26)	35.17(11)
N(5)	-	Ti(2)	-	C(28)	119.13(11)
N(6)	-	Ti(2)	-	C(28)	88.63(11)
N(5)^b	-	Ti(2)	-	C(28)	148.08(10)
C(27)	-	Ti(2)	-	C(28)	34.54(12)
C(26)	-	Ti(2)	-	C(28)	57.74(11)
N(5)	-	Ti(2)	-	C(29)	147.05(10)
N(6)	-	Ti(2)	-	C(29)	96.81(10)
N(5)^b	-	Ti(2)	-	C(29)	114.27(9)
C(27)	-	Ti(2)	-	C(29)	57.36(11)
C(26)	-	Ti(2)	-	C(29)	57.20(11)
C(28)	-	Ti(2)	-	C(29)	34.50(10)
N(5)	-	Ti(2)	-	C(25)	125.67(10)
N(6)	-	Ti(2)	-	C(25)	129.58(10)
N(5)^b	-	Ti(2)	-	C(25)	92.42(9)
C(27)	-	Ti(2)	-	C(25)	57.77(10)
C(26)	-	Ti(2)	-	C(25)	34.62(10)
C(28)	-	Ti(2)	-	C(25)	57.33(10)
C(29)	-	Ti(2)	-	C(25)	34.06(9)
N(5)	-	Ti(2)	-	Ti(2)^b	44.37(7)
N(6)	-	Ti(2)	-	Ti(2)^b	109.97(7)
N(5)^b	-	Ti(2)	-	Ti(2)^b	41.19(6)
C(27)	-	Ti(2)	-	Ti(2)^b	121.96(9)
C(26)	-	Ti(2)	-	Ti(2)^b	102.28(8)
C(28)	-	Ti(2)	-	Ti(2)^b	156.48(9)
C(29)	-	Ti(2)	-	Ti(2)^b	147.42(7)
C(25)	-	Ti(2)	-	Ti(2)^b	114.51(7)
C(37)	-	N(5)	-	Ti(2)	138.8(2)
C(37)	-	N(5)	-	Ti(2)^b	125.6(2)
Ti(2)	-	N(5)	-	Ti(2)^b	94.43(10)
C(19)	-	N(3)	-	Ti(1)	145.2(2)
C(13)	-	N(2)	-	Ti(1)	136.9(2)

Table 7.33: Bond angles for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

C(13)	-	N(2)	-	Ti(1)^a	127.3(2)
Ti(1)	-	N(2)	-	Ti(1)^a	94.92(10)
C(43)	-	N(6)	-	Ti(2)	141.7(2)
C(32)	-	N(4)	-	C(33)	117.2(2)
N(5)	-	C(37)	-	C(38)	122.0(2)
N(5)	-	C(37)	-	C(42)	119.9(2)
C(38)	-	C(37)	-	C(42)	118.1(3)
C(48)	-	C(43)	-	C(44)	118.1(3)
C(48)	-	C(43)	-	N(6)	120.8(3)
C(44)	-	C(43)	-	N(6)	121.1(3)
N(2)	-	C(13)	-	C(14)	122.4(2)
N(2)	-	C(13)	-	C(18)	119.6(3)
C(14)	-	C(13)	-	C(18)	118.0(3)
C(17)	-	C(18)	-	C(13)	120.4(3)
N(3)	-	C(19)	-	C(20)	121.6(3)
N(3)	-	C(19)	-	C(24)	120.8(3)
C(20)	-	C(19)	-	C(24)	117.5(3)
C(27)	-	C(26)	-	C(25)	108.3(3)
C(27)	-	C(26)	-	Ti(2)	72.0(2)
C(25)	-	C(26)	-	Ti(2)	75.3(2)
C(32)	-	C(31)	-	C(30)	113.6(2)
C(25)	-	C(29)	-	C(28)	108.8(3)
C(25)	-	C(29)	-	Ti(2)	73.7(2)
C(28)	-	C(29)	-	Ti(2)	71.7(2)
C(41)	-	C(42)	-	C(37)	120.4(3)
C(25)	-	C(30)	-	C(31)	114.1(2)
C(29)	-	C(25)	-	C(26)	106.9(3)
C(29)	-	C(25)	-	C(30)	125.4(3)
C(26)	-	C(25)	-	C(30)	127.5(3)
C(29)	-	C(25)	-	Ti(2)	72.3(2)
C(26)	-	C(25)	-	Ti(2)	70.1(2)
C(30)	-	C(25)	-	Ti(2)	126.1(2)
C(8)	-	N(1)	-	C(9)	117.0(3)
C(15)	-	C(14)	-	C(13)	120.7(3)
C(4)	-	C(3)	-	C(2)	107.8(3)
C(4)	-	C(3)	-	Ti(1)	73.7(2)
C(2)	-	C(3)	-	Ti(1)	72.5(2)
C(3)	-	C(4)	-	C(5)	108.1(3)
C(3)	-	C(4)	-	Ti(1)	72.0(2)
C(5)	-	C(4)	-	Ti(1)	73.6(2)

Table 7.34: Bond angles for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

C(39)	-	C(38)	-	C(37)	120.4(3)
C(4)	-	C(5)	-	C(1)	108.7(3)
C(4)	-	C(5)	-	Ti(1)	71.9(2)
C(1)	-	C(5)	-	Ti(1)	73.4(2)
N(4)	-	C(32)	-	C(31)	108.7(2)
C(40)	-	C(41)	-	C(42)	121.1(3)
C(23)	-	C(24)	-	C(19)	120.8(3)
N(4)	-	C(33)	-	C(36)	109.7(3)
N(4)	-	C(33)	-	C(35)	105.9(3)
C(36)	-	C(33)	-	C(35)	109.6(3)
N(4)	-	C(33)	-	C(34)	113.1(3)
C(36)	-	C(33)	-	C(34)	109.7(3)
C(35)	-	C(33)	-	C(34)	108.8(3)
C(28)	-	C(27)	-	C(26)	108.2(3)
C(28)	-	C(27)	-	Ti(2)	74.0(2)
C(26)	-	C(27)	-	Ti(2)	72.8(2)
C(41)	-	C(40)	-	C(39)	118.8(3)
C(22)	-	C(23)	-	C(24)	121.2(3)
C(27)	-	C(28)	-	C(29)	107.8(3)
C(27)	-	C(28)	-	Ti(2)	71.4(2)
C(29)	-	C(28)	-	Ti(2)	73.8(2)
C(2)	-	C(1)	-	C(5)	107.1(3)
C(2)	-	C(1)	-	C(6)	128.3(3)
C(5)	-	C(1)	-	C(6)	124.5(3)
C(2)	-	C(1)	-	Ti(1)	70.9(2)
C(5)	-	C(1)	-	Ti(1)	72.3(2)
C(6)	-	C(1)	-	Ti(1)	125.6(2)
C(8)	-	C(7)	-	C(6)	112.7(3)
C(21)	-	C(20)	-	C(19)	121.1(3)
C(1)	-	C(2)	-	C(3)	108.4(3)
C(1)	-	C(2)	-	Ti(1)	74.8(2)
C(3)	-	C(2)	-	Ti(1)	72.1(2)
C(23)	-	C(22)	-	C(21)	119.0(3)
C(18)	-	C(17)	-	C(16)	121.1(3)
C(17)	-	C(16)	-	C(15)	119.1(3)
C(45)	-	C(44)	-	C(43)	120.5(3)
C(47)	-	C(48)	-	C(43)	120.4(3)
C(14)	-	C(15)	-	C(16)	120.6(3)
C(1)	-	C(6)	-	C(7)	114.3(3)
N(1)	-	C(8)	-	C(7)	109.3(3)

Table 7.35: Bond angles for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

C(38)	-	C(39)	-	C(40)	121.2(3)
N(1)	-	C(9)	-	C(12)	109.9(3)
N(1)	-	C(9)	-	C(10)	105.9(3)
C(12)	-	C(9)	-	C(10)	110.0(4)
N(1)	-	C(9)	-	C(11)	112.1(3)
C(12)	-	C(9)	-	C(11)	110.8(4)
C(10)	-	C(9)	-	C(11)	108.0(3)
C(48)	-	C(47)	-	C(46)	120.8(3)
C(46)	-	C(45)	-	C(44)	121.4(3)
C(45)	-	C(46)	-	C(47)	118.7(3)
C(22)	-	C(21)	-	C(20)	120.4(3)

Symmetry transformations used to generate equivalent atoms:
[a] -x+2,-y+1,-z+1
[b] -x+2,-y+2,-z

Table 7.36: Anisotropic displacement parameters ($Å^2$) for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).
The anisotropic displacement factor exponent takes the form:
$-2\pi^2(h^2a*^2 U_{11} + ... + 2hka*b* U_{12})$

Atom	$U_{11}/Å^2$	$U_{22}/Å^2$	$U_{33}/Å^2$	$U_{23}/Å^2$	$U_{13}/Å^2$	$U_{12}/Å^2$
Ti(1)	.0346(3)	.0225(2)	.0224(2)	-.0005(2)	-.0066(2)	-.0101(2)
Ti(2)	.0341(3)	.0216(2)	.0230(2)	-.0021(2)	-.0064(2)	-.0094(2)
N(5)	.0377(12)	.0232(10)	.0256(11)	-.0047(8)	-.0035(9)	-.0114(9)
N(3)	.0457(14)	.0260(11)	.0308(12)	.0001(9)	-.0104(10)	-.0117(11)
N(2)	.0385(12)	.0223(10)	.0280(11)	-.0018(8)	-.0055(9)	-.0116(9)
N(6)	.0396(13)	.0275(12)	.0361(13)	.0023(9)	-.0148(10)	-.0115(10)
N(4)	.0448(14)	.0299(12)	.0399(14)	-.0032(10)	-.0124(11)	-.0150(11)
C(37)	.0359(14)	.0228(12)	.0240(12)	-.0039(9)	-.0033(10)	-.0104(11)
C(43)	.0360(14)	.0301(13)	.0304(14)	-.0047(10)	-.0047(11)	-.0102(12)
C(13)	.0353(14)	.0277(13)	.0253(12)	-.0028(9)	-.0059(10)	-.0117(11)
C(18)	.044(2)	.0304(14)	.0313(14)	-.0011(10)	-.0029(12)	-.0126(13)
C(19)	.042(2)	.0236(12)	.0281(13)	-.0024(10)	-.0007(11)	-.0078(12)
C(26)	.041(2)	.0330(14)	.0294(14)	-.0031(10)	-.0001(11)	-.0149(12)
C(31)	.039(2)	.0300(14)	.0339(14)	-.0053(11)	-.0040(11)	-.0148(12)
C(29)	.041(2)	.0243(13)	.0326(14)	.0044(10)	-.0117(11)	-.0112(12)
C(42)	.040(2)	.0303(14)	.0303(14)	-.0075(10)	-.0046(11)	-.0120(12)
C(30)	.047(2)	.0262(13)	.0343(14)	-.0021(10)	-.0106(12)	-.0165(12)
C(25)	.043(2)	.0253(12)	.0272(13)	.0014(9)	-.0087(11)	-.0138(11)
N(1)	.057(2)	.0385(14)	.041(2)	-.0039(11)	-.0027(13)	-.0260(13)
C(14)	.040(2)	.0284(14)	.0337(14)	-.0021(11)	-.0066(12)	-.0093(12)
C(3)	.058(2)	.042(2)	.0233(13)	-.0069(11)	-.0049(12)	-.021(2)
C(4)	.056(2)	.043(2)	.0247(13)	.0027(11)	-.0153(13)	-.022(2)
C(38)	.039(2)	.0329(14)	.035(2)	-.0058(11)	-.0079(12)	-.0076(12)
C(5)	.047(2)	.041(2)	.034(2)	.0027(12)	-.0158(13)	-.0186(14)
C(32)	.049(2)	.0286(14)	.040(2)	-.0026(11)	-.0130(13)	-.0177(13)
C(34)	.052(2)	.032(2)	.052(2)	-.0045(13)	-.010(2)	-.0199(14)
C(41)	.055(2)	.039(2)	.034(2)	-.0128(12)	-.0028(13)	-.0222(14)
C(24)	.053(2)	.0288(14)	.033(2)	-.0068(11)	-.0022(13)	-.0151(13)
C(33)	.046(2)	.033(2)	.040(2)	-.0028(12)	-.0111(13)	-.0175(13)
C(27)	.061(2)	.035(2)	.0230(13)	-.0026(10)	-.0017(12)	-.0210(14)
C(40)	.055(2)	.0282(14)	.040(2)	-.0124(12)	.0016(14)	-.0101(14)
C(23)	.066(2)	.0282(14)	.037(2)	-.0079(11)	.007(2)	-.020(2)
C(28)	.059(2)	.036(2)	.0270(14)	.0031(11)	-.0143(13)	-.0218(14)
C(1)	.054(2)	.039(2)	.033(2)	-.0022(11)	-.0138(13)	-.0263(14)
C(7)	.057(2)	.037(2)	.046(2)	-.0048(13)	-.006(2)	-.026(2)
C(20)	.042(2)	.033(2)	.048(2)	.0004(13)	-.0061(14)	-.0093(13)
C(2)	.058(2)	.034(2)	.0308(14)	-.0089(11)	-.0089(13)	-.0191(14)

Table 7.37: Anisotropic parameters for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

C(22)	.060(2)	.026(2)	.042(2)	.0010(12)	.002(2)	-.0027(14)
C(17)	.049(2)	.046(2)	.036(2)	-.0024(13)	.0013(13)	-.024(2)
C(16)	.035(2)	.053(2)	.041(2)	-.0141(14)	.0021(13)	-.0154(14)
C(44)	.044(2)	.0290(14)	.049(2)	-.0029(12)	-.0047(14)	-.0116(13)
C(48)	.041(2)	.031(2)	.052(2)	-.0077(13)	-.0084(14)	-.0116(13)
C(15)	.041(2)	.039(2)	.046(2)	-.0090(13)	-.0067(14)	-.0080(14)
C(6)	.054(2)	.045(2)	.047(2)	.0033(14)	-.017(2)	-.029(2)
C(8)	.057(2)	.042(2)	.053(2)	.000(2)	-.010(2)	-.029(2)
C(39)	.048(2)	.032(2)	.045(2)	-.0115(13)	-.0047(14)	-.0011(14)
C(9)	.070(2)	.045(2)	.040(2)	-.0063(13)	.005(2)	-.034(2)
C(47)	.041(2)	.057(2)	.055(2)	-.020(2)	-.009(2)	-.015(2)
C(35)	.059(2)	.039(2)	.067(2)	.001(2)	-.029(2)	-.017(2)
C(45)	.042(2)	.034(2)	.049(2)	.0077(13)	-.0009(14)	-.0042(14)
C(46)	.035(2)	.063(2)	.038(2)	-.004(2)	-.0090(13)	-.004(2)
C(21)	.045(2)	.038(2)	.051(2)	.0026(14)	-.008(2)	-.002(2)
C(10)	.085(3)	.049(2)	.062(3)	.007(2)	-.014(2)	-.037(2)
C(11)	.057(2)	.051(2)	.088(3)	-.003(2)	-.002(2)	-.030(2)
C(36)	.064(2)	.067(2)	.038(2)	-.010(2)	-.007(2)	-.032(2)
C(12)	.192(7)	.086(4)	.054(3)	-.029(3)	.042(4)	-.084(5)

Table 7.38: Hydrogen coordinates and isotropic displacement parameters for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}).

Atom	x	y	z	U_{eq}/Å^2
H(3N)	.965(4)	.741(3)	.458(2)	.041
H(6N)	1.236(3)	.989(4)	.029(2)	.042
H(4N)	.627(4)	.760(3)	-.016(2)	.045
H(18A)	1.1823(3)	.6311(3)	.3437(2)	.038(8)
H(26A)	.8272(3)	.8973(3)	.1399(2)	.051(10)
H(31A)	.7751(3)	.7785(3)	.0655(2)	.039(9)
H(31B)	.8540(3)	.8099(3)	-.0102(2)	.027(7)
H(29A)	1.1891(3)	.6462(2)	.1415(2)	.042(9)
H(42A)	1.0695(3)	1.1390(3)	.1546(2)	.028(7)
H(30A)	.9517(3)	.6053(3)	.1090(2)	.030(7)
H(30B)	1.0300(3)	.6426(3)	.0349(2)	.033(8)
H(1N)	.822(4)	.071(4)	.515(3)	.052
H(14A)	1.2943(3)	.2846(3)	.4610(2)	.048(10)
H(3A)	1.0437(3)	.5099(3)	.2730(2)	.064(13)
H(4A)	.8583(3)	.7093(3)	.2659(2)	.063(12)
H(38A)	.7447(3)	1.2482(3)	.0469(2)	.036(8)
H(5A)	.6792(3)	.6698(3)	.3546(2)	.037(8)
H(32A)	.8702(3)	.6299(3)	-.0528(2)	.033(8)
H(32B)	.7905(3)	.5988(3)	.0231(2)	.022(6)
H(34A)	.5590(3)	.5902(3)	-.0290(2)	.053(11)
H(34B)	.6963(3)	.5095(3)	-.0511(2)	.053(11)
H(34C)	.5906(3)	.5424(3)	-.1099(2)	.054(11)
H(41A)	.9849(3)	1.3000(3)	.2232(2)	.048(10)
H(24A)	.9838(3)	.9278(3)	.4089(2)	.050(10)
H(27A)	.9533(3)	.9570(3)	.2256(2)	.071(14)
H(40A)	.7825(3)	1.4353(3)	.2049(2)	.065(13)
H(23A)	.8957(4)	1.1235(3)	.3412(2)	.053(11)
H(28A)	1.1759(3)	.8023(3)	.2268(2)	.043(9)
H(7A)	.8460(3)	.2483(3)	.4418(2)	.035(8)
H(7B)	.8441(3)	.3037(3)	.5169(2)	.042(9)
H(20A)	.6913(3)	.8533(3)	.3616(2)	.041(9)
H(2A)	.9800(3)	.3450(3)	.3673(2)	.056(11)
H(22A)	.7096(4)	1.1849(3)	.2810(2)	.054(11)
H(17A)	1.3729(3)	.5876(3)	.2746(2)	.064(12)
H(16A)	1.5240(3)	.3943(3)	.2958(2)	.039(9)
H(44A)	1.3682(3)	.6841(3)	.0990(2)	.046(10)
H(48A)	1.3943(3)	1.0110(3)	.0884(2)	.058(12)
H(15A)	1.4827(3)	.2424(3)	.3888(2)	.052(10)

Table 7.39: Hydrogen coordinates for bis(μ-phenylamine) bis((η^5-cyclopentadienyl) (phenylmethyl) titanium^{IV}) (continued)

H(6A)	.6634(3)	.4187(3)	.4014(2)	.065(13)
H(6B)	.6731(3)	.4752(3)	.4734(2)	.08(2)
H(8A)	.6584(4)	.2771(3)	.5657(2)	.060(12)
H(8B)	.6463(4)	.2357(3)	.4873(2)	.051(10)
H(39A)	.6630(3)	1.4073(3)	.1170(2)	.063(12)
H(47A)	1.5804(3)	.9165(4)	.1515(2)	.055(11)
H(35A)	.4643(4)	.8028(3)	-.1008(3)	.040(9)
H(35B)	.4959(4)	.7547(3)	-.1816(3)	.068(13)
H(35C)	.5415(4)	.8566(3)	-.1685(3)	.076(14)
H(45A)	1.5518(3)	.5927(3)	.1648(2)	.059(12)
H(46A)	1.6583(3)	.7060(4)	.1923(2)	.08(2)
H(21A)	.6076(3)	1.0482(3)	.2908(2)	.054(11)
H(10A)	.8383(5)	-.1250(4)	.5719(3)	.09(2)
H(10B)	.7680(5)	-.1482(4)	.6535(3)	.061(13)
H(10C)	.8749(5)	-.0962(4)	.6481(3)	.08(2)
H(11A)	.6391(4)	.0009(4)	.5110(3)	.12(2)
H(11B)	.5445(4)	.1118(4)	.5477(3)	.08(2)
H(11C)	.5663(4)	-.0214(4)	.5916(3)	.09(2)
H(36A)	.8216(4)	.5962(4)	-.1585(2)	.07(2)
H(36B)	.7612(4)	.7294(4)	-.2039(2)	.073(14)
H(36C)	.7159(4)	.6273(4)	-.2169(2)	.074(14)
H(12A)	.6011(8)	.1619(5)	.6703(3)	.09(2)
H(12B)	.7285(8)	.0807(5)	.7085(3)	.09(2)
H(12C)	.6216(8)	.0288(5)	.7138(3)	.10(2)

Appendix A

Programs for Bonding Survey

This Appendix contains listings of the C-shell scripts and FORTRAN source codes for the programs used in the Main Group Elements Bonding Survey work (Chapter 2).

start (C-shell script)

```
#!/bin/csh
#
# Controlling script for extraction of Metal-Atom distances from Cambridge
# Crystallographic Database

# Initialise executable path and local setup
        source /usr/local/csds/csdsetup
        setenv PATH ${PATH}:${HOME}/CSD
        rehash

# Get metal types for search from command line list
set LIST = $*
if ( $LIST == "" ) then
        exit 1
endif

# Loop over each metal type specified
foreach Job ( $LIST )
        echo "Starting search for ${Job}-AA systems"
        set job = `echo $Job | awk '{print tolower($0)}'`
        set JOB = `echo $Job | awk '{print toupper($0)}'`
```

```
# Construct a QUEST instuction file to search for all structures
# containing the metal type in question
      echo "Creating QUEST instruction file start.qu"

# Make a search fragment consisting only the metal atom
      echo "T1 *CONN" > start.qu
      echo "NFRAG  1" >> start.qu
      echo "AT1 $JOB 0" >> start.qu                                    30
      echo "END" >> start.qu

# Setting screen 153 ensures that atomic coordinates are present
      echo "SCREEN 153" >> start.qu

# Saving refcodes creates a virtual database of structures containing the
# metal type
      echo "SAVE REFCODE" >> start.qu

# The STOP limit is set higher than the number of entries in the database   40
# to ensure it is not reached
      echo "STOP 200000" >> start.qu
      echo "QUEST T1" >> start.qu

# Run the QUEST search to create the virtual database of just one metal type
      echo "Running initial QUEST for ${Job}"
      quest -j $job -v all_IA_IIA.gcd -if start.qu > /dev/null

      mkdir $Job
      mv $job.gcd $Job                                                  50

# Set up a Quest instruction file to search this new database
# Templates are used with the metal type substituted using awk
      echo "Creating QUEST file for ${Job}-AA search"
      awk -v J=$job '{sub(/m/,J); print $0}' tmpl.in > $Job/${job}2az.in
      awk -v J=$JOB '{sub(/m/,J); print $0}' tmpl.qu > $Job/${job}2az.qu

      chmod 644 $Job/${job}2az.in
      chmod 644 $Job/${job}2az.qu
      chmod 644 $Job/${job}.gcd                                         60

      rm ${job}.*

      echo "Moving to directory $Job"
      cd $Job
# The C-Shell script "getdist" is called to extract distance information
```

```
# from the virtual database
        echo "Calling getdist on jobname ${job}2az"
        getdist ${job}2az
                                                                    70
# Complete the loop
end
exit
```

---

getdist (C-shell script)

```
#!/bin/csh
#
# Wrapper for getdist program: runs getdist.x then sorts output

# Check for jobname on command line
if ($1 != "") then
        set JOB = $1
# Otherwise use default name
else
        set JOB = getdist                                           10
endif

# Set up the variables for commands, files, etc.
set QUEST = "quest -j $JOB"
set AWK = $HOME/CSD/getdist.awk
set AWK2 = $HOME/CSD/getcoord.awk
set EXE = $HOME/CSD/getdist.x

# Look for any virtual database files in the current directory, and use them
# if present                                                        20
set GCD = `ls *.gcd`
if ( "$GCD" == "" ) then
        echo "No virtual database found in this directory - using full database"
else
        foreach FILE ( $GCD )
                set QUEST = "$QUEST -v $FILE"
        end
endif

# Check for a commands file to run QUEST in batch mode              30
if ( -f $JOB.qu ) then
```

```
        set QUEST = "$QUEST -if $JOB.qu"
else
        echo "No QUEST commands file $JOB.qu found - running interactively"
endif

# Run the quest job
echo "Running QUEST job on $JOB.qu"
$QUEST > /dev/null
                                                                        40
# Check for an input file for GETDIST
if ( -f $JOB.in ) then
        echo "Running GETDIST with $JOB.in"
        $EXE < $JOB.in | sort -r | grep -v Input | tee $JOB.prog | grep -v DNF > $JOB.out
else
        echo "Running GETDIST interactively"
        $EXE | sort -r | grep -v Input | grep -v DNF > $JOB.out
endif

# Process output into $JOB.out and $JOB.n files                         50
rm -f $JOB.final
set NUM = 0
while ( $NUM < 20 )
        @ NUM = $NUM + 1
        echo "Processing GETDIST output for coordination number $NUM"
        awk -v C=$NUM -v F=$JOB -f $AWK $JOB.out
        if ( -f $JOB.$NUM) then
#               echo "bond  cn n  mean    sd" >> $JOB.final
                sort -r $JOB.$NUM | awk -v C=$NUM -f $AWK2 >> $JOB.final
        endif                                                           60
end

# Pass the jobname on to the tabulation program, table.x
echo $JOB | table.x

echo "Removing temporary files"
rm -f $JOB.fgd $JOB.fgn $JOB.u38 $JOB.tmp

echo "Remaining files"
ls $JOB.*                                                               70

exit
```

getdist.f (FORTRAN source)

```
      program getdist

      parameter (dat=1, tab=2, maxatom=1000, maxsymm=192)
      real c(6), axyzr(maxatom,5), smat(maxsymm,4,4)
      logical bflag(maxatom,maxatom)
      integer nsymm, natom, ntype
      character aname(maxatom)*8
      character*80 jobname, filename, inline
      character*8 refcode, oldref, parname
      character elname*2, t1, t2*6, mt1, mt2*6                          10
      integer jnlen, pnlen, enlen, nrec, rrec, drec, reclen, num, i
      integer ol, rl, dummy
      real value

      integer toupper

      common /mol/ c, smat, aname, axyzr, nsymm, natom, ntype, bflag

10    continue
            write (*,'(A,$)') 'Input:jobname: '                         20
            read (*,'(A)') jobname
            jnlen = len2(jobname)
            if (jnlen .eq. 0) goto 10
            write (*,'(A)') jobname(1:jnlen)
            filename = jobname(1:jnlen) // '.dat'
            open (dat, file=filename, status='old', err=10)
            filename = jobname(1:jnlen) // '.tab'
            open (tab, file=filename, status='old', err=10)
15    continue
                                                                        30
20    continue
            write (*,'(A,$)') 'Input:name of parameter required: '
            read (*,'(A)') parname
            pnlen = toupper(parname)
            if (pnlen .eq. 0) goto 20
            write (*,'(A)') parname(1:pnlen)
25    continue

30    continue
            write (*,'(A,$)') 'Input:name of element: '                 40
            read (*,'(A)') elname
            enlen = toupper(elname)
```

```
            dummy = tolower(elname(2:2))
            write (*,'(A)') elname(1:enlen)
            if (enlen .eq. 0) goto 20
35      continue

70      continue
            write (*,'(A,$)') 'Input:tolerance for search: '
            read (*,'(F)') tol                                          50
            if (tol .lt. 0.0001) tol = 0.0001
            write (*,'(F6.4)') tol
        i = 0
40      continue
            read (tab,1000,err=999) t1, t2
            if (t1 .eq. '#') then
                mt1 = t1
                mt2 = t2
            else if (t1 .eq. 'I') then
                nrec = i                                                60
            else if (t1 .eq. 'L') then
                rrec = i
            else if (t1 .eq. 'D') then
                if (t2 .eq. parname) drec = i
            else if (t1 .eq. '!') then
                goto 45
            end if
            i = i + 1
            goto 40
45      continue                                                        70

        reclen = i

        refcode = ' '
        i=0
50      continue
            read (tab,1010,end=60,err=999) inline
            if (inline(1:8) .eq. mt1 // ' ' //mt2) then
                if (i .eq. 0) then
                    i=1                                                 80
                else
                    call process(refcode(1:rl),num,value,elname,tol)
                    i=1
                end if
            else if (i .eq. nrec) then
                read (inline,*,end=60,err=999) num
                i = i + 1
```

```
            else if (i .eq. drec) then
                  read (inline,*,end=60,err=999) value
                  i = i + 1                                          90
            else if (i .eq. rrec) then
                  oldref = refcode
                  refcode = inline(1:8)
                  ol = len2(oldref)
                  rl = len2(refcode)
                  if (refcode .ne. oldref) then
                        call coord(oldref(1:ol))
                        call readfdat(refcode(1:rl))
                  endif
                  if (refcode .eq. 'ERROR') goto 999                 100
                  i = i + 1
            end if
            goto 50
60    continue
            call coord(refcode(1:rl))

999   continue
1000  format(A1,X,A6)
1010  format(A80)
      end                                                            110

c

      subroutine readfdat(refcode)

      character refcode*(*), atype(20)*2, symrep*15, test*9
      character inline*80, tt*2, nums*10 /'0123456789'/
      integer idata(11), iflag(19), incl(6), incm(6), ntext, i, j,d
      real rd, arad(20), x, y, z, xo(3,3)
      parameter (pi = 3.1415926536)                                  120
      integer tolower

      parameter (dat=1, tab=2, maxatom=1000, maxsymm=192)
      real c(6), axyzr(maxatom,5), smat(maxsymm,4,4)
      integer nsymm, natom, ntype
      character aname(maxatom)*8
      logical bflag(maxatom,maxatom)
      common /mol/ c, smat, aname, axyzr, nsymm, natom, ntype, bflag
      common /cart/ xo
                                                                     130
      test = '#' // refcode // '  '
10    continue
```

```
            read (dat,1000,end=900,err=999) inline
            if (inline(1:9) .ne. test) goto 10
15      continue

        read (inline,1010) idata, iflag

        read (dat,1020) incl, incm
        do i=1,6                                                    140
            c(i) = incl(i) / 10.0 ** incm(i)
            if (i .gt. 3) c(i) = c(i) * pi / 180.0
        end do

        call cart1(c(1),c(2),c(3),c(4),c(5),c(6))

        ntext = idata(2) + idata(3) + idata(4) + idata(5)
20      continue
            if (ntext .le. 0) goto 25
            read (dat,1000) inline                                  150
            ntext = ntext - 80
            goto 20
25      continue

        read (dat,1000) inline
        do nsymm=1,idata(6)
            if (inline(1:1) .eq. ' ') read(dat,1000) inline
            symrep = inline(1:15)
            inline = inline(16:80) // '                '
            call rep2mat(symrep)                                    160
        end do
        if (iflag(4) .ne. 0) then
            nsymm = idata(6) + 1
            call rep2mat('011 0101 0110 0')
        end if

        read (dat,1000) inline
        ntype = idata(7)
        do i=1,ntype
            atype(i) = inline(1:2)                                  170
            read (inline(3:5),'(F3.0)') rd
            arad(i) = rd / 100.0
            inline = inline(6:80) // '      '
        end do

        natom = idata(8) + idata(9)
        read (dat,1000) inline
```

```
      do i=1,natom
            if (inline(1:2) .eq. '  ') read (dat,1000) inline
            test = inline(1:5) // '    '                                    180
            if (iflag(3) .eq. 1) then
                  read (inline(6:20),'(3F5.0)') x, y, z
                  inline = inline(21:80) // '                    '
            else
                  read (inline(6:27),'(3F7.0)') x, y, z
                  inline = inline(28:80) // '                         '
            end if
            tt(1:1) = test(1:1)
            test = test(2:9) // ' '
            if (index(nums,test(1:1)) .eq. 0) then                         190
                  tt(2:2) = test(1:1)
                  d = tolower(tt(2:2))
                  test = test(2:9) // ' '
            else
                  tt(2:2) = ' '
            end if
            aname(i) = tt // ' ' // test
            do j=1,ntype
                  if (tt .eq. atype(j)) then
                        axyzr(i,4) = arad(j)                               200
                  end if
            end do
            x = x / 100000.0
            y = y / 100000.0
            z = z / 100000.0
            call cart2(x,y,z,axyzr(i,1),axyzr(i,2),axyzr(i,3))
            axyzr(i,5) = 0.0
            do j=1,natom
                  bflag(i,j) = .false.
            end do                                                         210
      end do

      return

900   continue
      write (*,1030) refcode
      return

999   continue
      refcode = 'ERROR'                                                    220
      return
```

```
1000  format (A80)
1010  format (23X,11I3,19I1)
1020  format (6F6.0,6F1.0)
1030  format (A,':Not found in FDAT file')
      end

c
                                                                      230
      subroutine rep2mat(symrep)

      character symrep*(*)
      integer i, j, is(4)

      parameter (dat=1, tab=2, maxatom=1000, maxsymm=192)
      real c(6), axyzr(maxatom,5), smat(maxsymm,4,4)
      integer nsymm, natom, ntype
      character aname(maxatom)*8
      logical bflag(maxatom,maxatom)                                  240
      common /mol/ c, smat, aname, axyzr, nsymm, natom, ntype, bflag

      do i=1,3
            read (symrep,'(3I1,I2)') is
            symrep = symrep(6:15) // '      '
            do j=1,3
                  smat(nsymm,i,j) = real (is(j)) - 1.0
            end do
            smat(nsymm,i,4) = real (is(4)) / 12.0
      end do                                                          250
      smat(nsymm,4,4) = 1.0

      return
      end

c

      subroutine process(refcode,num,value,elnam,tol)

      character *(*) refcode, elnam                                   260
      integer num
      real value, tol

      integer i, j, il, jl, flag
      real r2, d(8)

      parameter (dat=1, tab=2, maxatom=1000, maxsymm=192)
```

```
      real c(6), axyzr(maxatom,5), smat(maxsymm,4,4)
      integer nsymm, natom, ntype
      logical bflag(maxatom,maxatom)                                    270
      character aname(maxatom)*8
      common /mol/ c, smat, aname, axyzr, nsymm, natom, ntype, bflag

      flag = 0
      do i=1,natom
            if (aname(i)(1:2) .eq. elnam) then
                  il = len2(aname(i))
                  do j=1,natom
                        jl = len2(aname(j))
                        r2 = (axyzr(i,1) - axyzr(j,1)) ** 2             280
                        r2 = r2 + (axyzr(i,2) - axyzr(j,2)) ** 2
                        r2 = r2 + (axyzr(i,3) - axyzr(j,3)) ** 2
                        r2 = sqrt(r2)
                        r2 = real (nint (r2 * 1000.0)) / 1000.0
                        value = real (nint (value * 1000.0)) / 1000.0
                        if (abs(r2 - value) .lt. tol .and. flag .eq. 0) then
                              if (bflag(i,j) .eq. .FALSE.) then
c                                   write (*,1010) refcode, num, value
                                    write (*,1010) refcode
                                    write (*,1000) aname(i)(1:il),aname(j)(1:jl),r2 290
                                    bflag(i,j) = .TRUE.
                                    flag = 1
                                    axyzr(i,5) = axyzr(i,5) + 1.0
                              endif
                        endif
                  enddo
            endif
      enddo
      if (flag .eq. 0) then
c           write (*,1010) refcode, num, value                          300
            write (*,1010) refcode
            write (*,'(a)') ' : #DNF# : Distance not found'
      endif
      d(1) = value
      d(2) = value - c(1)
      d(3) = value - c(2)
      d(4) = value - c(3)
      d(5) = value - sqrt(c(1) * c(1) + c(2) * c(2))
      d(6) = value - sqrt(c(1) * c(1) + c(3) * c(3))
      d(7) = value - sqrt(c(2) * c(2) + c(3) * c(3))                    310
      d(8) = value - sqrt(c(1) * c(1) + c(2) * c(2) + c(3) * c(3))
      value = min(d(1),d(2),d(3),d(4),d(5),d(6),d(7),d(8))
```

```
      return
1000  format (A,' - ',A,':',F6.3)
c1010 format (A,':',I4,F6.3,':'$)
1010  format (A,': ',$)
      end

c                                                                    320

      subroutine coord(refcode)

      character *(*) refcode
      integer i, il

      parameter (dat=1, tab=2, maxatom=1000, maxsymm=192)
      real c(6), axyzr(maxatom,5), smat(maxsymm,4,4)
      integer nsymm, natom, ntype
      logical bflag(maxatom,maxatom)                                 330
      character aname(maxatom)*8
      common /mol/ c, smat, aname, axyzr, nsymm, natom, ntype, bflag

      do i=1,natom
            if (axyzr(i,5) .gt. 0.0) then
                  il = len2(aname(i))
                  write (*,1000) refcode, aname(i)(1:il), nint(axyzr(i,5))
            endif
      enddo
      return                                                         340

1000  format (A,': ',A,':Coordination number:',I3)
      end
```

```
      integer function len2(string)

C Returns length of string as position of last non-whitespace character

      character string*(*)
      integer i

      len2=0
      i=len(string)
10    continue                                                       10
            if (string(i:i) .ne. ' ') then
```

```
                len2=i
        else
                i = i - 1
                if (i .gt. 0) goto 10
        end if
15      continue

        return
        end                                                          20

c

        integer function toupper(string)

c Converts string to uppercase, returns length of string from len2

        character string*(*)
        character lower*26 /'abcdefghijklmnopqrstuvwxyz'/
        character upper*26 /'ABCDEFGHIJKLMNOPQRSTUVWXYZ'/            30
        integer i, j ,sl

        sl = len2(string)
        do i=1,sl
                j = index(lower,string(i:i))
                if (j .ne. 0) then
                        string(i:i) = upper(j:j)
                endif
        enddo
                                                                     40
        toupper = sl
        return
        end

c

        integer function tolower(string)

c Converts string to lowercase, returns length of string from len2
                                                                     50
        character string*(*)
        character lower*26 /'abcdefghijklmnopqrstuvwxyz'/
        character upper*26 /'ABCDEFGHIJKLMNOPQRSTUVWXYZ'/
        integer i, j ,sl

        sl = len2(string)
```

```
      do i=1,sl
            j = index(upper,string(i:i))
            if (j .ne. 0) then
                  string(i:i) = lower(j:j)                    60
            endif
      enddo

      tolower = sl
      return
      end
```

```
      subroutine cart1(a, b, c, al, be, ga)

C Take cell parameters and form a matrix to convert from fractional coords
C to cartesian coords

      real a, b, c, al, be, ga
      real xo(3,3), ra, rb, rc, ral, rbe, rga, vol
      real sa, sb, sg, ca, cb, cg
      common /cart/ xo
                                                              10
      ca = cos(al)
      cb = cos(be)
      cg = cos(ga)
      sa = sin(al)
      sb = sin(be)
      sg = sin(ga)
      vol = a*b*c*sqrt(1.0-ca*ca-cb*cb-cg*cg+2.0*ca*cb*cg)
      ra = b*c*sa/vol
      rb = a*c*sb/vol
      rc = a*b*sg/vol                                         20
      ral = (cb*cg-ca)/(sb*sg)
      rbe = (ca*cg-cb)/(sa*sg)
      rga = (ca*cb-cg)/(sa*sb)
      xo(1,1) = a
      xo(1,2) = b*cg
      xo(1,3) = c*cb
      xo(2,1) = 0.0
      xo(2,2) = b*sg
      xo(2,3) = -c*sb*ral
      xo(3,1) = 0.0                                           30
      xo(3,2) = 0.0
      xo(3,3) = 1.0/rc
```

```
      return
      end

      subroutine cart2(x,y,z,rx,ry,rz)

C Use the matrix created by cart1 to transform input fractional coords to
C cartesian coordinates                                                    40

      real x,y,z,rx,ry,rz
      real xo(3,3)
      common /cart/ xo

      rx = xo(1,1) * x + xo(1,2) * y + xo(1,3) * z
      ry = xo(2,1) * x + xo(2,2) * y + xo(2,3) * z
      rz = xo(3,1) * x + xo(3,2) * y + xo(3,3) * z

      return                                                               50
      end
```

Bibliography

[1] *The Royal Society Scientific Information Conference Report*, London, U.K., 1948.

[2] F.H. Allen, O. Kennard, D.G. Watson, L. Brammer, A.G. Orpen, and R. Taylor. Tables of Bond Lengths determined by X-Ray and Neutron Diffraction. Part 1. Bond Lengths in Organic Compounds. *J. Chem. Soc. Perkin Trans. II*, (12):S1–S19, 1987.

[3] A.G. Orpen, L. Brammer, F.H. Allen, O. Kennard, D.G. Watson, and R. Taylor. Tables of Bond Lengths determined by X-ray and Neutron Diffraction. Part 2. Organometallic Compounds and Co-ordination complexes of the d- and f-Block Metals. *J. Chem. Soc. Dalton Trans.*, (12):S1–S83, 1989.

[4] A.V. Aho, B.W. Kernighan, and P.J. Weinberger. *The AWK Programming Language*. Addison-Wesley, 1988. See also awk(1) in UNIX online manual pages.

[5] Frank H. Allen et. al. *Cambridge Structural Database System.* 12 Union Road, Cambridge, CB2 1EZ, United Kingdom, 1992.

[6] Theo Hahn, editor. *International Tables for Crystallography*, volume A. IUCr / D. Reidel Publishing Co., Dordrecht: Holland / Boston: U.S.A., 1983.

[7] C. Giaccovazzo, H.L. Monaco, D. Viterbo, F. Scordari, G. Gilli, G. Zanotti, and M. Catti. *Fundamentals of Crystallography.* IUCr / Oxford University Press, Oxford: U.K., 1992.

[8] J.D. Foot and E.A.Colbourn. Electrostatic potentials for surface of inorganic and molecular crystals. *J. Mol. Graphics*, 6:93–99, 1988.

[9] D.J. Watkin, J.R. Carruthers, and P.W. Betteridge. *Crystals User Manual.* Chemical Crystallography Laboratory, University of Oxford, 1985.

[10] G. Sheldrick. *SHELXL 93. Program for the Refinement of Crystal Structures.* University of Gottingen, Germany, 1993.

[11] A. Markovits, A. Fahmi, and C. Minot. A theoretical study of CO_2 adsorption on TiO_2. *Theochem - Journal of Molecular Structure*, 371:219–235, 1996.

[12] R.V. Kasowski. *Phys. Rev. B*, 14:3398, 1976.

[13] O.K. Anderson and R.V. Kasowski. *Phys. Rev. B*, 4:1064, 1971.

[14] R.G. Gordon and Y.S. Kim. *J. Chem. Phys.*, 56:3122, 1972.

[15] M. Waldman and R.G. Gordon. *J. Chem. Phys.*, 71:1325, 1979.

[16] C. Mulhausen and R.G. Gordon. *Phys. Rev. B*, 22:900, 1981.

[17] R. Dovesi, V.R. Saunders, and C. Roetti. *Crystal92, An ab-initio Hartree-Fock LCAO program for periodic systems.* Theoretical Chemistry Group, University of Turin, Italy, 1993.

[18] P. Durand and J.C. Barthelat. *Chem. Phys. Lett.*, 27(191), 1974.

[19] P. Durand and J.C. Barthelat. *Theor. Chim. Acta*, 38(283), 1975.

[20] J.C. Barthelat and P. Durand. *Gazz. Chim. Ital.*, 108(225), 1978.

[21] J.C. Barthelat, P. Durand, and A. Serafini. *Molec. Phys.*, 33(159), 1977.

[22] T. Restori, D. Schwarzenbach, and J.R. Schneider. Charge Density in Rutile, TiO_2. *Acta Cryst. B*, 43:251, 1987.

[23] J.P. Perdew. In P. Ziesche and H. Eschrig, editors, *Electronic Structure of Solids.* Akademie Verlag, Berlin, 1991.

[24] R. Colle and O. Salvetti. *Theor. Chim. Acta*, 37:329, 1979.

[25] R. Colle and O. Salvetti. *J. Chem. Phys.*, 79:1404, 1983.

[26] C.R.A. Catlow and R. James. *Proc. R. Soc. Lond. A*, 384:157, 1983.

[27] C.R.A. Catlow, R. James, W.C. Mackrodt, and R.F. Stewart. *Phys. Rev. B*, 25:1006, 1982.

[28] M. Mostoller and J.C. Wang. *Phys. Rev. B*, 32:6773, 1985.

[29] C. Pisani and R. Dovesi. *International Journal of Quantum Chemistry*, 17:501, 1980.

[30] W.H. Press, S.A. Teukolsky, W.H. Vetterling, and B.P. Flannery. *Numerical Recipes in FORTRAN.* Cambridge University Press, Cambridge: U.K., second edition, 1992.

[31] B. Silvi, N. Fourati, R. Nada, and C.R.A. Catlow. Pseudopotential periodic Hartree-Fock study of rutile TiO_2. *J. Phys. Chem. Solids*, 52(8):1005–1009, 1991.

[32] P. Jones and J.A. Hockey. Infra-Red studies of Rutile surfaces. Part 2. Hydroxylation and structure of Rutile surfaces. *Trans. Faraday Soc.*, 67:2679, 1971.

[33] M.J. Jaycock and J.C.R. Wildsax. Calculation of adsorption potentials for water on Rutile. *J. Chem. Soc., Faraday Trans. 1*, 70:1501, 1974.

[34] J. Goniakowski abd C. Noguera. Relaxation and rumpling mechanisms on oxide surfaces. *Surface Science*, 323:129–141, 1995.

[35] D. Vogtenhuber, R. Podloucky, A. Neckel, S.G. Steinmann, and A.J. Freeman. Electronic structure and relaxed geometry of the TiO_2 Rutile (110) surface. *Phys. Rev. B*, 49(3):2099, 1994.

[36] K. Jug, G. Geudtner, and T. Bredow. Theoretical investigations on adsorbtion at ion crystal surfaces. *Journal of Molecular Catalysis*, 82:171–194, 1993.

[37] E.R. Davidson and S.J. Chakravorty. A possible definition of basis-set superposition error. *Chemical Physics Letters*, 217(1-2):48–54, 1994.

[38] A. Allouche. Quantum ab initio study of acetylene adsorption on NaCl(100). 1. Topology and adsorption energy. *Surface Science*, 374(1-3):117–124, 1997.

[39] J. Rasure and S. Kubica. The Khoros Application Development Environment. In H.I Christensen and J.L Crowley, editors, *Experimental Environments for Computer Vision and Image Processing*. World Scientific, 1994.

[40] G. Sheldrick. *SHELXTL-PLUS. Release 4.1.* Siemens Analytical X-ray Instruments Inc., Madison, Wisconsin, U.S.A., 1991.

[41] H.T. Thimbleby, S. Inglis, and I.H. Witten. Displaying 3d images - algorithms for single-image random-dot stereograms. *Computer*, 27(10):38, 1994.

[42] A.K. Hughes, S.M.B. Marsh, J.A.K. Howard, and P.S. Ford. Lewis base functionalised cyclopentadienyl complexes of titanium. *Journal of Organometallic Chemistry*, 528:195–198, 1997.