Powder Indexing of Large Volume Unit Cells (including protein data) using Crysfire

Robin Shirley

School of Human Sciences

University of Surrey

Guildford, UK

Session 1b, Single Crystal and Powder Diffraction Software Workshop, ECM-21, Durban, South Africa

24 August 2003

Summary

- What is powder indexing and why is it needed?
- Crysfire 2003 indexing example (high-quality lab data).
- Indexing with Crysfire and Chekcell.
- Figures of merit - classic and joint-probability.
- Mountaineering in indexing’s solution space.
- Crysfire 2002, Crysfire 2003 (and Industrial Crysfire).
- A “buyer’s guide” for indexing programs.
- How much does cell volume matter? Can one index a protein?
- Crysfire 2003 protein example (Zn Insulin):

  Self-calibration for 2Theta zero error;

  Rescaling and indexing;

  Space-group and higher symmetry: Chekcell.
- Conclusions and acknowledgements.

What is powder indexing and why is it needed?

- Before you can get started on an ab initio powder structure, you need to know the unit cell

  otherwise the intensity from each profile point can’t be located in reciprocal space, and progress is impossible.
- Inferring the unit cell from the limited and partly overlapping and degraded information in a powder pattern is called powder indexing (or auto-indexing).
- This is a complex process of induction that’s far beyond manual methods in all but the easiest cases.

Data Quality

- Perhaps more than for any other aspect of powder diffraction,

  success in powder indexing depends critically on the resolution and accuracy of one’s observed data.
- Although there are tools for correcting data with systematic errors (self-calibration example to follow).

“Indexing? - No problem, we just use program ... [fill in your lab’s preference]”

- “And does that always work?”

  “Well, no - sometimes it doesn’t come out, so we drop that problem and move on.”
- Indexing can be harder than actually solving the structure.
- It can seem an all-or-nothing process, that either comes out painlessly or with much effort, if at all

  (especially if you only have one tool in your toolbox).
- Different types of pattern can need different programs.
- Running Dicvol (or Treor) on a problem that needs Lzon (or Kohl) is like trying to hammer in a nail with a screwdriver.

< A first look at Crysfire 2003 & Chekcell>

Indexing a low-volume monoclinic lab dataset using programs in combination

(Demonstration)

Indexing with Crysfire + Chekcell

- **Crysfire** starts from powder line positions (2Thetas or d-spacings) and returns with a list of plausible trial cells, sorted by number of lines indexed and figure of merit.

  So it should have done the really hard work of pinning down the solution to somewhere specific in solution space.

  Its work is finished when it has provided a list that contains the correct cell in some form (or a derivative cell of it).

  But the solutions may be in unhelpful settings or expressed in too-low symmetry, and not say much about space groups.

- **Chekcell** is a graphical toolkit that can help the user to identify the best crystal system and setting.

  It can narrow the symmetry to a few specific spacegroups for each trial solution, and do an extended Le Page search for derivative sub- and super-cells (for one trial solution at a time).

Indexing with Crysfire + Chekcell (2)

To summarise:

- **Crysfire** does the heavy stuff that you couldn’t think of doing by hand, but doesn’t stop to polish its solutions.
- **Chekcell** provides graphical tools to help with the polishing job, which in principle you could do by hand, but this way is much easier and faster.

Classic Figures of Merit (“M_{20}”, etc)

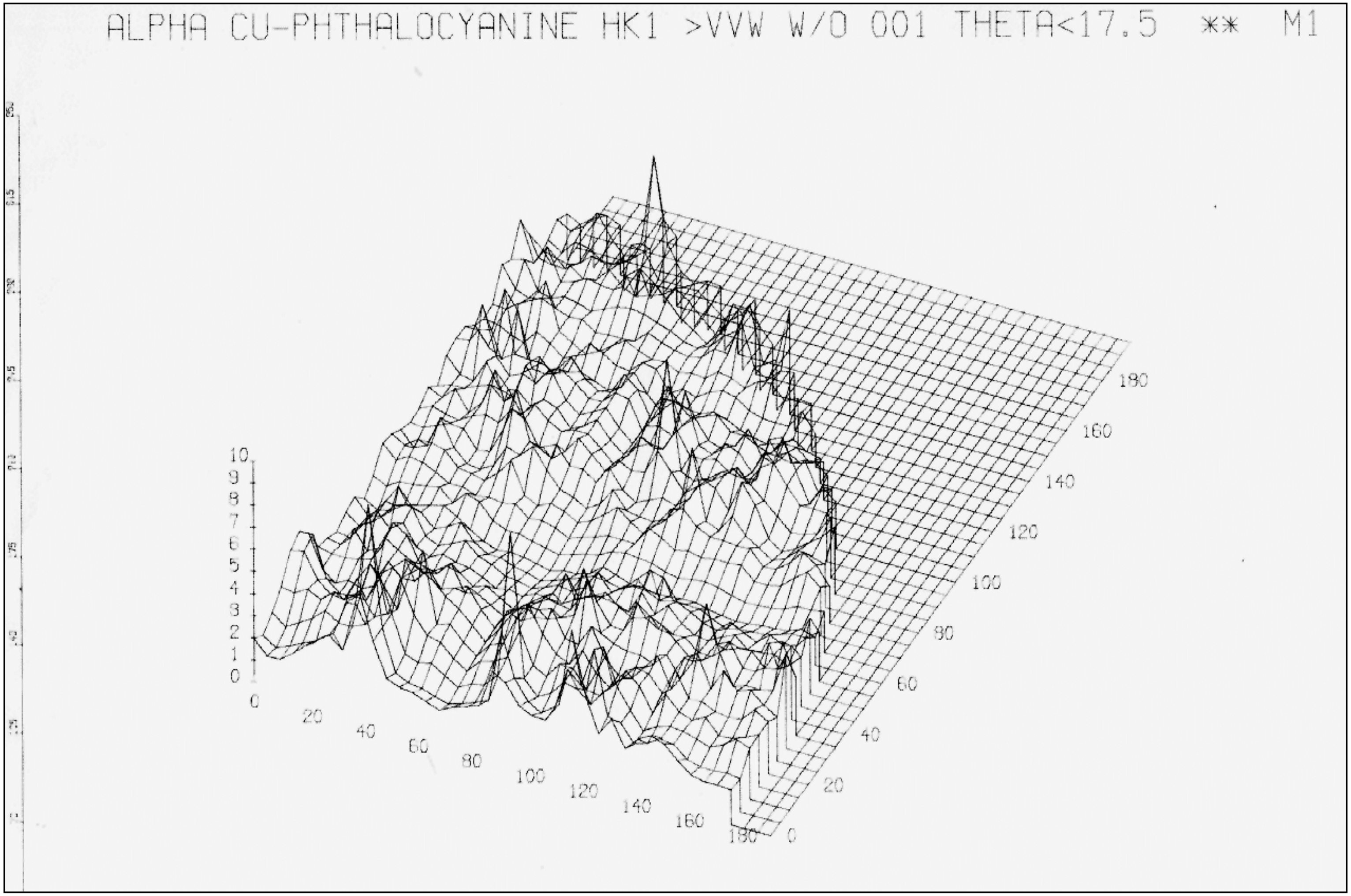
- Indexing figures of merit (M) are roughly the inverse of R-factors, but based on differences in obs and calc Q [=1/dsq] rather than in intensity.
- Different programs define M somewhat differently, but that isn’t usually important - M is too sensitive to tiny changes in cell constant for minor differences to be significant.
- But it’s worth noting that no current program actually uses De Wolff’s original M_{20} definition for its M (though they often call it M_{20}), because its handling of “unindexed” lines would be too cumbersome.
- (IMHO) The best-conditioned and most conservative M is M1, as used by Crysfire’s CA and MM commands.
- Crysfire simply records whatever each program reported.

Joint-Probability Figures of Merit (Pr, Ir)

- The figures of merit discussed so far all depend on the **sums of discrepancies** - the differences between observed and calculated line positions.
- Thus they report the average **misfit** between observations and model, and can be severely degraded by any non-model features such as impurity lines.
- Figures of merit can also be based on the **joint probability** of getting the observed level of fit across the pattern.
- Two new joint-probability figures of merit Pr and Ir are provided in the new Hmap program in Crysfire 2003.
- Unlike M_{20}-style figures of merit, these depend mainly on the regions that **fit** well with that model, and are largely transparent to badly-fitting features like impurity lines.
- For more about joint-probability methods, come to the indexing microsymposium on Thursday morning.

Mountaineering in indexing's solution-space

- If we can specify 6 values for the cell constants (three sides and three angles), we will have fixed the unit cell.
- So in that sense, indexing implies finding a point with a high figure of merit in a 6-dimensional "solution space".
- We could step-scan across this space, making a visual map of its landscape of high and low values of M (or Pr / Ir), which is what Mmap (or Hmap) does.
- The next slide shows such a topographical map for a section through the solution space of an organic-metallic dyestuff, to give some idea of what they can look like.
- Each peak in the map corresponds to a relatively high figure of merit, and hence a possible trial solution.



M1 surface in the Q_D/Q_E (i.e. α^*/β^*) section for α-Cu phthalocyanine (from Powder47)

Where are we now? - Crysfire 2002, Crysfire 2003 and Industrial Crysfire

- The current distribution is still Crysfire 2002, but what I’m demonstrating here is Crysfire 2003, to be released very shortly in 16-bit and 32-bit versions (yes, 32-bit at last, though not yet GUI-based!).
- These are (or will be) available free for non-profit use (and inexpensively for industrial users).
- They’re aimed at making life easier for non-specialists in indexing (with some crystallographic knowledge).
- Industrial Crysfire will be a graphical 32-bit version, more automated and aimed to be usable by technicians, who are not necessarily crystallographers. Hopefully it will be released early next year.

A “Buyer’s Guide”: Some characteristics of the Indexing Programs now supported by Crysfire

	Tolerates:	Random	
Program	Impurities	Errors	Comments
Ito12	Yes	No	Optimised for low-symmetry
Fjzn6	Yes	No	As for Ito12, but more robust
Dicvol91	No	Slows	Good for screening to orthor./monoc.
Lzon	Slows	Slows	Best for dominant-zone cases
Losh	Slows	Slows	User-guided, faster than Lzon
Treor90	Yes	No	Specialises in v accurate impure data
Kohl [=TMO]	Yes	Yes	Fast, useful for high & low symmetry
Taup [=Powder]	Slows	?Slows	Good for screening down to orthor.
Mmap	No	Yes	Gives visual maps of solution space
Hmap	Yes!	Yes	Gives visual joint-probability maps
McMaille	Yes	?Slows	Likely to need overnight runs

How Much does Cell Volume Matter?
(Can one index a protein?)

- Formally the indexing problem is scale independent: it uses only relative dimensions and is unchanged if the d-spacings and wavelength are both doubled.

  So why do large cells make indexing harder?
- Although indexing is scale-independent, instrument resolution and accuracy are not, so if the d-spacings are 10 times bigger, the data must be 10 times better.

  Also, classic programs like Ito & Treor were optimised for moderate cell volumes (i.e. below c.5000 A^3).
- Bob von Dreele has indexed several protein patterns, at least in high symmetry (and solved their structures).
- Crysfire includes re-scaling to support this, as we’ll see.

< Working with High Volume Datasets >

1) Crysfire: SC command
   Improving data via self-calibration

2) Crysfire: Indexing a Protein

3) Chekcell: LePage / Best Solution
   Seeking a higher-symmetry cell

(Demonstration)

Conclusions

- The powder-indexing problem is now well understood (though not always solved, especially if you don’t use the right tools).
- There are powerful indexing programs available and becoming more widely used (more in the pipeline - watch this space).
- Crysfire offers easy access to 9 of them (from next month, Crysfire 2003 will make that 11).
- Crysfire’s strength lies in finding trial cells, not in establishing their best description and symmetry.
- That job is best done as a separate stage, using the graphical helper program, Chekcell.
- Crysfire and Chekcell are complementary, and if used together are likely to offer the quickest route to success.
- Even protein data can now be indexed (given very good data).

Contributing Authors: Crysfire & Chekcell

- Crysfire: Overall system + Mmap, Hmap, Lzon, Losh, etc

  Robin Shirley

  Contributed crystallographic software

Franz Kohlbeck	Kohl [=TMO]
Armel Le Bail	McMaille
Daniel Louër	Dicvol, Losh, Lzon
Ton Spek & A. Meetsma	Clepage
Daniel Taupin	Taup [=Powder]
Arie van der Lee	Eva2crys
Jan Visser	Ito, Fjzn, Lzon, etc
Per-Eric Werner	Treor

- Chekcell

  Jean Laugier & Bernard Bochu

< Discussion >

Classification of Indexing Methods

Method	Space	Exhaustive	Status	Program(s) available
Zone indexing	Parameter	No	Mature	Yes (2)
Successive dichotomy	Parameter	Yes/(semi)	Mature	Yes (2+)
Grid search	Parameter	Yes	Semi-mature	Yes (1)
Combined heuristic	Parameter	Semi	Mature	Yes (2)
Joint probability	Parameter	Semi	Developing	Yes (1)
Genetic algorithms	Parameter	No	Developing	Yes (1)
Monte Carlo	Parameter	No	Semi-mature	Yes (2)
Simulated annealing	Parameter	No	Not yet tried	No
Diffusion equation	Parameter	No	Not yet tried	No
Scan/covariance	Par. (both)	To monoc.	Semi-mature	Yes (1)
Index heuristics	Index	No	Mature	Yes (2)
Index permutation	Index	Yes	Mature	Yes (1)

(The list is not claimed to be complete. Only symmetry-general methods included)

Methods vs Programs

Space	Method	Programs using this method
Parameter	Zone indexing	ITO12, FJZN6
	Successive dichotomy	DICVOL91, X-Cell,[LZON, LOSH]
	Grid search	SCANIX, Mmap, Hmap, (Powder49)
	Combined heuristic	LZON, LOSH, Mmap, Hmap
	Joint probability	Hmap
	Genetic algorithms	AUTOX, (W.P.G.A.: Harris...)
	Monte Carlo	McMaille, SVD-Index
	Simulated annealing	(not yet implemented)
	Diffusion equation	(not yet implemented)
	Profile-based	McMaille (idealised), (W.P.G.A.)
Par. (both)	Scan/covariance	EFLECH/INDEX
Index	Index heuristics	TREOR90, TMO [=KOHL]
	Index permutation	POWDER [=TAUP]

(Underlined if supported by Crysfire, in round brackets if not generally available)

Indexing Programs: 1) Overview

Program	Author(s)	Method	Space	Exh.	O/S
ITO12	Visser	Zone index	Par	No	DOS +?
FJZN6	Visser & Shirley	Zone index	Par	No	DOS
DICVOL91	Louër	Dichotomy	Par	Mainly	DOS +?
X-Cell	Neumann	Dichotomy	Par	Mainly	Win
SCANIX	Paszkowicz	Grid search	Par	Yes	DOS/ANSI
(Powder49)	Shirley	Grid search	Par	Semi	Mainframe
LZON	Shirley/Louër/Visser	Comb. heur.	Par	Semi	DOS
LOSH	Shirley & Louër	Comb. heur.	Par	Semi	DOS
Mmap, Hmap	Shirley	Comb. heur.	Par	Semi	DOS,Win
AUTOX	Zlokazov	Genetic alg.	Par	No	DOS/extender
McMaille	Le Bail	Monte Carlo	Par	No	Win
SVD-Index	Coelho	Monte Carlo	Par	No	Win
(W.P.G.A.)	Harris…	Genetic alg.	Par	No	Unix
EFLECH/INDEX	Bergmann	Scan/covar.	P/(I)	Mainly	DOS,Win,Lin
TREOR90	Werner	Index heur.	Ind	No	DOS +?
TMO [=KOHL]	Kohlbeck	Index heur.	Ind	No	DOS +?
POWDER [=TAUP]	Taupin	Index perm.	Ind	Yes	DOS +?

(Underlined if supported by Crysfire, in round brackets if not generally available)

Indexing Programs: 2) Performance (1GHz Pentium)

Program	Orthor.	Monocl.	Tricl.	Comments
ITO12	1 sec	1 sec	1 sec	Automatic
FJZN6	2 sec	2 sec	2 sec	Automatic
DICVOL91	<3 sec	1-30 min	mins-hours	Auto., v. vol.-dependent
X-Cell	Variable: minutes-hours			Automatic
SCANIX	c.5 min	(?5 min)	n.a.	User guided in monocl.
(Powder49)	c.5 min in each case			(PC Equiv.), user guided
LZON	30 sec - 5 min in each case			Automatic
LOSH	<30 sec	<30 sec	<30 sec	User guided
Mmap, Hmap	10 sec - 15 min in each case			User guided
AUTOX	?15 sec	?5 min	?30 min	Semi-auto. in practice?
McMaille	Variable: hours in black box mode			Automatic
SVD-Index	Variable: minutes-hours			Automatic
(W.P.G.A.)	Run times said to be lengthy			(Alpha / SGI workstn.)
EFLECH/INDEX	3 min	15 min	5 min	c.Auto, Exh. to monocl.
TREOR90	<30 sec	<30 sec	<30 sec	Automatic
TMO [=KOHL]	<30 sec	<30 sec	<30 sec	Automatic
POWDER [=TAUP]	<5 sec	15 min+	Very long	Automatic

(Typical run times as a rough guide, but may vary considerably with data & settings)

Indexing Programs: 3) Tolerance of impurities, etc

Program	Tolerates: Impurities	Random Errors	Comments
ITO12	Yes	No	Optimised for low-symmetry
FJZN6	Yes	No	As for ITO12, but more robust
DICVOL91	No	Slows	Good for screening to orthor./monoc.
X-Cell	Yes	Slows?	Commercial package (Accelrys)
SCANIX	Yes	Slows?	Under development, user guided
(Powder49)	Slows	Maybe	Superseded by LZON
LZON	Slows	Slows	Best for dominant-zone cases
LOSH	Slows	Slows	User-guided, faster than LZON
Mmap	Slows	Maybe	User-guided, good for dominant zones
Hmap	Yes, very	No	User-guided, good for dominant zones
AUTOX	Slows?	Slows?	Many optional user settings
McMaille	Yes	Slows?	Uses idealised whole profile
SVD-Index	Yes	Slows?	Commercial package (Bruker)
(W.P.G.A.)	Yes	Slows	Uses whole profile, developing
EFLECH/INDEX	Yes	Slows	Uses full peak-fit covariance matrix
TREOR90	Yes	No	Specialises in v accurate impure data
TMO [=KOHL]	Yes	Yes	Fast, useful for high & low symmetry
POWDER [=TAUP]	Slows	Slows?	Good for screening down to orthor.

(Though nominally automatic, in practice AUTOX often needs some user guidance)