

August 25, 2025

Approximate Lattice Matching in Three Dimensions

LAWRENCE C. ANDREWS^{a*} AND HERBERT J. BERNSTEIN^b

^a*Ronin Institute for Independent Scholarship 2.0, USA, and* ^b*Ronin Institute for Independent Scholarship 2.0, USA. E-mail: larry6640995@gmail.com*

lattice; reduction; Niggli; matching; P^3

Abstract

Given two unit cells, the problem of determining the best match of the lattice of one to the lattice of the other can require large numbers of trial transformations. We present a solution that requires only a limited number of trials.

Note: In his later publications, Boris Delaunay used the Russian version of his surname, Delone.

1. Introduction

A relatively common crystallographic problem is to compare two unit cells and determine whether their lattices are the or close to each other. Flor *et al.* (2016) have described a solution for more symmetric crystal families. However, for the general case of triclinic lattices a different method is required.

Andrews *et al.* (2023) described a general solution, using the space \mathbf{S}^6 . While effective, this solution in a space based on projections fails to produce the 3D transformations needed for reindexing reflection data and for examining the relationships of structures.

Attempts to solve the problem by trying a large number of trial transformations can require an unknown and unknowable number of trials of transformation. Here we describe a straight-forward approach that uses a fixed, modest number of trials and that gives multiple possible matches if they are appropriate.

2. Notation

\mathbf{P}^3 : Polar coordinate space. It is a compact and smooth alternative to lengths and angles. \mathbf{P}^3 is useful for comparing unit cells. \mathbf{P}^3 is defined as:

$$\mathbf{P}^3: (|\vec{a}|, \alpha), (|\vec{b}|, \beta), (|\vec{c}|, \gamma)$$

\Rightarrow

$$(|\vec{a}| \cos \alpha, |\vec{a}| \sin \alpha)$$

$$(|\vec{b}| \cos \beta, |\vec{b}| \sin \beta)$$

$$(|\vec{c}| \cos \gamma, |\vec{c}| \sin \gamma)$$

also known as (p_1, p_2, p_3)

3. Lattice matching

The general solution to matching two lattice has been to test the comparison using a large number of likely transformations. Two problems arise: how many trials to make and how to effectively measure the difference between two lattices.

To avoid the need to examine an unknown number of possible transformations, we propose an algorithm with the following stages. One of the input cells will be designated “reference” and the other (to which the transformations in stage three will

be applied) will be designated “mobile”.

1. Convert from both (possibly) centered lattices to primitive. (Apply matrix M_{CR} for reference and M_{CM} for mobile.) (Aroyo, 2016)
2. Niggli reduce both lattices. (Apply matrix M_{NR} for reference and M_{NM} for mobile.)
3. Apply a limited set of transformations to mobile, measuring an agreement factor for each trial, keeping those that meet an acceptable agreement value. This stage creates matrix M_{MR} that transforms the modified mobile to the modified reference. (Niggli, 1928) (Andrews & Bernstein, 2025a)
4. Compare the results of each transformation using the \mathbf{P}^3 Euclidean distance between the reference and the mobile cell. (Andrews & Bernstein, 2025b)

3.1. Finishing

When all of the transformations in stage three have been examined, the group of accepted results needs to be processed. The group might be empty or might contain several possible valid results. We call this the “accepted group”.

In each stage, when a transformation is used, the corresponding matrix is saved for later use. The final transformation chain to convert mobile (in its original lattice centering) to reference (in its original centering) is

$$M_{CM} * M_{NM} * M_{MR} * M_{NR}^{-1} * M_{CR}^{-1}.$$

This matrix chain needs to be computed for each member of the accepted group. Of course, for a given reference/mobile pair, only M_{MR} is different for accepted trials.

The immediate question is “how many transformations will need to be tested in stage three”. We have chosen to use all the unimodular matrices (*i.e.* determinant = 1.0), with elements -1, 0, +1. (3480 total matrices). Each transformation is tested and

acceptable results are accumulated to be “finished”.

4. Examples

4.1. example 1

personal communication, Aroyo (2025)

More than one acceptable match was found

Test input

p 5.17 3.18 7.74 90. 104.5 90. (reference)

p 30.9616 3.1800 8.1608 90.00 171.18 90.00 (mobile)

Result of matching

--- Match 1 ---

Quality: EXCELLENT P3 Distance 0.000)

Transformation Matrix:

[1.0000, 0.0000, 4.0000]

[0.0000, 1.0000, 0.0000]

[-1.0000, 0.0000, -3.0000]

Matrix determinant: 1.0000

Reference cell: P 5.170 3.180 7.740 90.000 104.500 90.000

Transformed cell: P 5.170 3.180 7.740 90.000 104.500 90.000

*** SUCCESS: Excellent match within P3-relative threshold ***

--- Match 2 ---

Quality: EXCELLENT (P3 Distance 0.000)

Transformation Matrix:

[-1.0000, 0.0000, -4.0000]

[0.0000, 1.0000, 0.0000]

[1.0000, 0.0000, 3.0000]

Matrix determinant: 1.0000

Reference cell: P 5.170 3.180 7.740 90.000 104.500 90.000

Transformed cell: P 5.170 3.180 7.740 90.000 104.500 90.000

*** SUCCESS: Excellent match within P3-relative threshold ***

4.2. example 2

Test input:

C 12.770 21.235 14.411 136.017 84.071 111.795 (reference)

F 33.151 18.241 20.218 83.054 144.781 120.639 (mobile)

```

=== LATTICE MATCHING RESULTS === =====
Found 1 excellent match
Success threshold: 1.43e-01 (Fixed strict)

--- Match 1 ---
Quality: EXCELLENT (0.000 )
P3 Distance: 0.000
S6 Angle: 0.00
Transformation Matrix:
[ 0.0000, -0.5000, 0.5000]
[ 1.0000, 0.5000, 0.5000]
[ 0.0000, 0.5000, 0.5000]
Matrix determinant: 0.5000
Reference cell:  C   12.770   21.235   14.411   136.017   84.071   111.795
Transformed cell: C   12.770   21.235   14.411   136.017   84.071   111.795
*** SUCCESS: Excellent match within P3-relative threshold ***

```

4.3. example 3

personal communication (Simmons, 2025)

A case where more than one accepted match was found.

Input:

p 10.25 10.74 21.08 87.72 75.97 61.53 (reference)

p 10.25 10.74 21.08 78.96 75.97 61.49 (mobile)

```

--- Match 1 ---
Quality: EXCELLENT (0.010 )
P3 Distance: 0.010
S6 Angle: 0.01
Transformation Matrix:
[ -1.0000, 0.0000, 0.0000]
[ -1.0000, 1.0000, 0.0000]
[ 0.0000, 0.0000, -1.0000]
Matrix determinant: 1.0000
Reference cell  p   10.25   10.74   21.08   87.72   75.97   61.53 (reference)\\
Transformed cell: P   10.250   10.739   21.080   87.714   75.970   61.502
*** SUCCESS: Excellent match within P3-relative threshold ***

```

```

--- Match 2 ---
Quality: EXCELLENT (0.023 )
P3 Distance: 0.023
S6 Angle: 0.03
Transformation Matrix:
[ -1.0000, 0.0000, 0.0000]
[ 0.0000, -1.0000, 0.0000]
[ -1.0000, 0.0000, 1.0000]

```

```

Matrix determinant: 1.0000
  Reference cell    p    10.25    10.74    21.08    87.72    75.97    61.53 (reference)\\
  Transformed cell: P    10.250    10.740    21.087    87.674    75.893    61.490
*** SUCCESS: Excellent match within P3-relative threshold ***

```

4.4. example 4

20 slight variations on a primitive cubic cell were created, with $a=10.0$, each Niggli reduced, and lattice-matched.

Input:

```

P  7.579  7.579  7.579  90.000  90.000  90.000 (reference)
P  7.570  7.573  7.574  89.957  89.937  89.958
P  7.576  7.580  7.581  89.989  89.957  89.964
P  7.573  7.574  7.576  89.950  89.946  89.994
P  7.571  7.578  7.583  90.058  90.029  90.062
P  7.577  7.578  7.578  90.051  90.033  90.051
P  7.572  7.576  7.578  90.033  90.034  90.084
P  7.573  7.576  7.580  90.028  90.060  90.020
P  7.570  7.573  7.580  89.993  89.977  89.931
P  7.572  7.572  7.573  89.982  89.934  89.936
P  7.577  7.580  7.588  89.958  89.970  89.996
P  7.571  7.581  7.581  90.061  90.028  90.059
P  7.573  7.574  7.584  90.005  90.023  90.065
P  7.580  7.581  7.581  89.933  89.996  89.968
P  7.576  7.581  7.586  89.913  89.990  89.956
P  7.573  7.574  7.581  89.979  89.983  89.974
P  7.570  7.575  7.57  90.025  90.063  90.066
P  7.571  7.579  7.581  89.988  89.993  89.942
P  7.578  7.581  7.588  90.049  90.023  90.002
P  7.580  7.580  7.585  89.932  89.989  89.988
P  7.571  7.576  7.577  89.970  89.993  89.943

```

and here is the matching results:

```

Most common transformation matrices:
[1,0,0,0,1,0,0,0,1] : 9 mobiles (P3 agreement: 0.016)
[1,0,0,0,-1,0,0,0,-1] : 3 mobiles (P3 agreement: 0.015)
[0,1,0,0,0,-1,-1,0,0] : 2 mobiles (P3 agreement: 0.015)
[1,0,0,0,0,-1,0,1,0] : 1 mobiles (P3 agreement: 0.011)
[0,0,1,1,0,0,0,1,0] : 1 mobiles (P3 agreement:, 0.012)
[0,0,1,0,-1,0,1,0,0] : 1 mobiles (P3 agreement: 0.014)
[0,0,-1,0,1,0,1,0,0] : 1 mobiles (P3 agreement: 0.012)
[0,-1,0,-1,0,0,0,0,-1] : 1 mobiles (P3 agreement: 0.009)
[-1,0,0,0,0,-1,0,-1,0] : 1 mobiles (P3 agreement: 0.017)

```

```

=== BEST MATCH DETAILS ===
Mobile 2: 8.42e-03 (EXCELLENT)
Matrix: [1 0 0 0 1 0 0 0 1]

```

Reference	P	7.579	7.579	7.579	90.000	90.000	90.000
Transformed:		7.576	7.580	7.581	89.989	89.957	89.964

Mobile 15: 9.36e-03 (EXCELLENT)

Matrix: [0 -1 0 -1 0 0 0 0 -1]

Reference:	P	7.579	7.579	7.579	90.000	90.000	90.000
Transformed:		7.574	7.573	7.581	89.983	89.979	89.974

Mobile 13: 1.03e-02 (EXCELLENT)

Matrix: [1 0 0 0 1 0 0 0 1]

Reference:	P	7.579	7.579	7.579	90.000	90.000	90.000
Transformed		7.580	7.581	7.581	89.933	89.996	89.968

5. Limitations

The group of all unimodular matrices is an infinite group. Clearly, by limiting this algorithm to matrices with elements $-1/0/+1$, a complete search cannot be done. However as the examples show, in common cases, the algorithm is effective. One limitation is obvious: if one axial length of the reduced cell is longer than the sum of the other two axial lengths, then $-1/0/+1$ cannot explore some regions that are available for shorter axial lengths. For instance, if one axis is much longer than the others, then in a matrix for that long axis, the column elements for that axis cannot have more than one element $-1/+1$. Similar arguments apply to the case of a very short axis. In order to treat these more extreme cases using this algorithm, a larger range of element values would need to be examined. The consequence is that the number of matrices and thus the number of trials grows rapidly.

6. Availability of code

The code for Lattice Matching and related software tools are available in github.com, in <https://github.com/duck10/LatticeRepLib.git>. The program CmdLMP3 uses the required files.

Acknowledgements

Careful copy-editing and corrections by Frances C. Bernstein are gratefully acknowledged.

Funding information

Funding for this research was provided in part by: US Department of Energy Offices of Biological and Environmental Research and of Basic Energy Sciences (grant No. DE-AC02-98CH10886; grant No. E-SC0012704); U.S. National Institutes of Health (grant No. P41RR012408; grant No. P41GM103473; grant No. P41GM111244; grant No. R01GM117126, grant No. 1R21GM129570); Dectris, Ltd.

7. Synopsis

A solution is proposed for the problem of best matching the lattices of unit cells that are described in different presentations.

References

- Andrews, L. C. & Bernstein, H. J., (2025*a*). Niggli reduction: matrices for 3-d lattice transformations. Fill in cite once paper is accepted.
- Andrews, L. C. & Bernstein, H. J., (2025*b*). P^3 , linearizing unit cell parameters. Fill in cite once paper is accepted.
- Andrews, L. C., Bernstein, H. J. & Sauter, N. K. (2023). *Foundations of Crystallography*, **79**(5), 480–484.
- Aroyo, M., (2025). Personal communication. Private correspondence with L. C. Andrews, August 2025.
- Aroyo, M. I. (2016). In *International Tables for Crystallography, Volume A: Space-group symmetry*, edited by M. I. Aroyo, chap. 3.1, pp. 698–700. International Union of Crystallography.
- Flor, G., Orobengoa, D., Tasci, E., Perez-Mato, J. M. & Aroyo, M. I. (2016). *Applied Crystallography*, **49**(2), 653–664.
- Niggli, P., (1928). Krystallographische und Strukturtheoretische Grundbegriffe, Handbuch der Experimentalphysik, Vol. 7, part 1. Akademische Verlagsgesellschaft, Leipzig.
- Simmons, C. J., (2025). Personal communication. Private correspondence with L. C. Andrews, August 2025.

Synopsis
