Journal of Applied Crystallography

ISSN 1600-5767

# The Follower algorithm and a program using it to explore spaces

**Lawrence C. Andrews[a]* and Herbert J. Bernstein[b]**

[a]Ronin Institute, 9515 NE 137th St, Kirkland, WA, 98034-1820 USA, and [b]Ronin Institute, c/o NSLS-II, Brookhaven National Laboratory, Upton, NY, 11973 USA. Correspondence e-mail: lawrence.andrews@ronininstitute.org

The Follower algorithm was developed to examine the properties of continuous functions and to verify their computer implementations. In particular, we studied measures for distances between unit cells.

© 0000 International Union of Crystallography

## 1. Introduction

We have created several methods for determining "distances" between unit cells (or lattices): $\mathbf{V^7}$ (Andrews *et al.*, 1980), $\mathbf{G^6}$ (Andrews & Bernstein, 1988), $\mathbf{D^7}$ (Andrews & Bernstein, 2014), and $\mathbf{S^6}$ (Andrews *et al.*, 2019*b*). They have been used for the creation of databases (Andrews *et al.* (1980) and Andrews *et al.* (2019*a*)), for identification of Bravais lattice types (Andrews & Bernstein, 1988), and for implementing clustering for serial crystallography (Bernstein *et al.*, 2020) (Nguyen *et al.*, 2020).

### 1.1. Issues to be solved

Implementation of computer methods for distance calculations in spaces $\mathbf{V^7}$, $\mathbf{G^6}$, $\mathbf{D^7}$, and $\mathbf{S^6}$ begins by reduction to standard form (Niggli, 1928) (Delaunay, 1932). The implementations are complicated by the need for iterative examination of many alternative unit cells that may be close to matching the reduced cell (Andrews *et al.*, 1980), (Andrews & Bernstein, 1988), (Andrews & Bernstein, 2014). There is no presently-known method to evaluate how deeply these iterations through alternatives should be taken, resulting in complex computer code that is difficult to verify.

These being studies where the correct answers cannot be foreseen, a method to at least test for consistency of the results was needed. Two desirable properties of a valid measure are that the distance between two points would vary continuously with displacement and that two non-coincident points would not have a distance measure of zero.

Because we are working with maps on manifolds (locally flat Euclidean tangents to curved surfaces) rather than simple global Euclidean spaces, we cannot extrapolate linearly from what we see in the small, which is sufficient for databases and clustering, to the longer distances needed for a clear understanding of the topology of these spaces.

## 2. Creating Follower

Our solution to the above issues was to develop the Follower algorithm. In its simplest form, the algorithm is initialized with two points. One is chosen as the fixed point and the other becomes the mobile point. Then (in the chosen space) the mobile point is moved along a line to the fixed point, computing the distance between them at each step. Figure 1 shows one kind of result that might be expected. In the figure, a mobile point has been chosen, and the Niggli-reduced cell will be the fixed point. The graph displays the $\mathbf{G^6}$ (Andrews & Bernstein, 2014) distance from the mobile point to the fixed point.

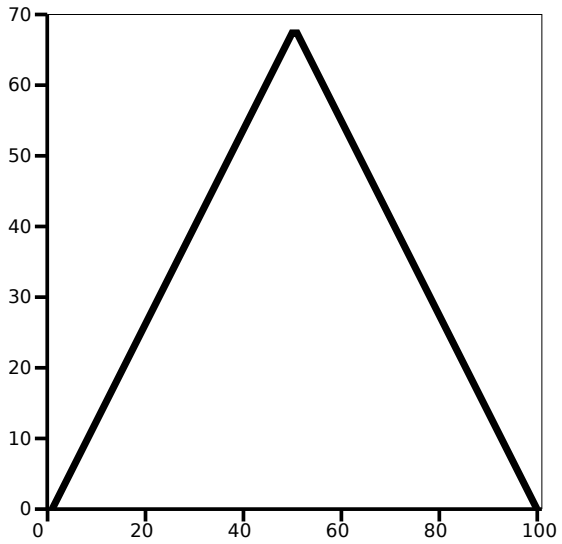


**Figure 1**
Simple Follower plot where Euclidean distance is adequate. Here the initial mobile point is the unreduced unit cell 14.329, 10.216, 10.346, 90.11, 91.04, 133.20 ($a, b.c.\alpha, \beta, \gamma$), and the fixed point is the corresponding Niggli-reduced cell 10.453, 10.216, 10.346, 89.89, 91.53, 92.23. 101 points are sampled; the abscissa is the $\mathbf{G^6}$ distance between them in units of Å$^2$. The graph ends with zero distance because the mobile point takes the same value as the fixed point. The initial point has zero distance because it is a sample from the same lattice as the reduced cell (the final point).

One crucial use of Follower was for visualizing software errors during development of the above mentioned distance measures for lattices. Figure 2 shows the effect of an intentionally introduced software error. In this case, the search for nearest points was truncated before it had reached adequate depth.

## 3. Analogy

To anticipate the complexity that Follower might show us, consider a sodium chloride crystal. Inject a proton into the crystal and determine the distance to the nearest sodium ion. Then move the proton through the crystal in a straight line. Barring

a head-on collision, as the proton continues through the crystal the distance to the nearest sodium will first decrease and then increase, but eventually another sodium ion will be closer, and the distance to the newly-nearest sodium ion will decrease and then later increase. This fall, rise, fall, and rise of the distance will continue as the proton movement continues through the crystal.

In the same fashion, in Follower the mobile point moves through the multitude of representations of the crystal lattice. We will expect that sometimes the shortest distance will increase and sometimes decrease. This might happen several times.

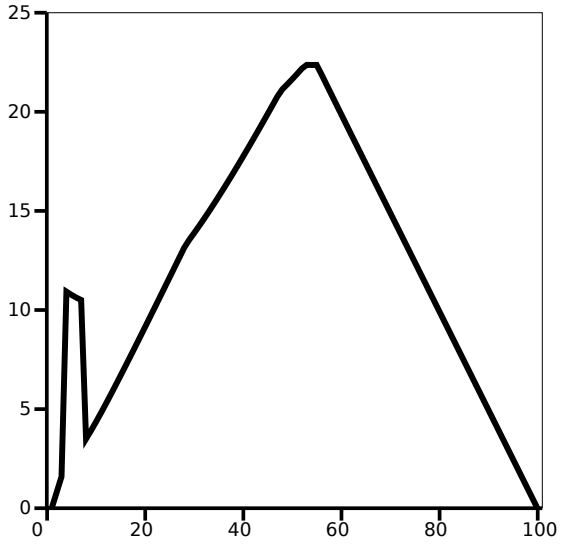


**Figure 2**
An intentionally introduced software error shows up as a discontinuity in the graphs.

### 4. Alternate path choices

In our implementation of Follower, there are five different choices of measurements.

**POINT:** a single unit cell is input. The generated path goes from the input point to its reduced cell. See Figure 3.

**LINE:** two points are input. The first point becomes the mobile point, and the other point is used as the fixed point. See Figure 4.

**CHORD:** two points are input. Each point follows a line to its reduced cell. The reported distance is between the two mobile points as they move jointly on their separate lines, each at the same percentage to their respective distances from the starting point to the final point. See Figure 5.

**CHORD3:** three points are input. The first and second are mobile points as in **CHORD**, but they maintain their initial separation. The midpoint between the two mobile points moves from the initial midpoint towards the third point. The reported distance is between the two mobile points. See Figure 6.

**TRIANGLE:** three points are input. The third point is fixed. The other two move toward the fixed point. The distance between the mobile points is reported. See Figure 7.

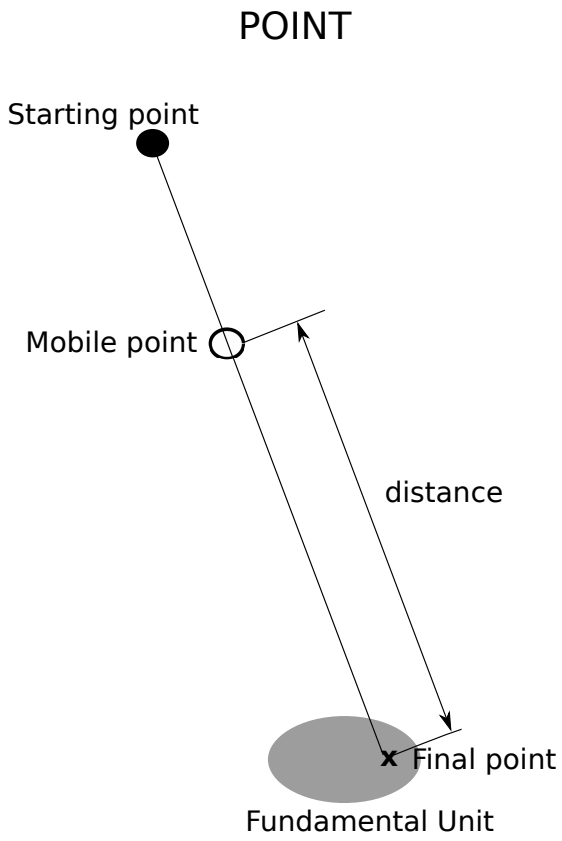


**Figure 3**
Follower POINT Method

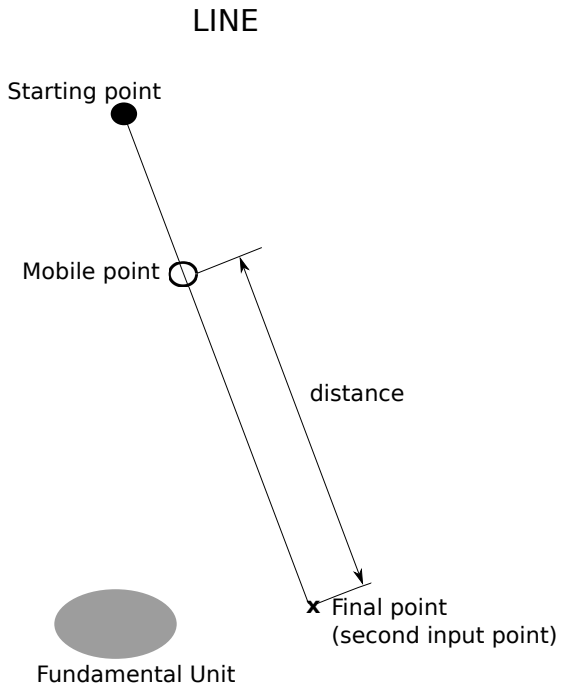


**Figure 4**
Follower LINE Method

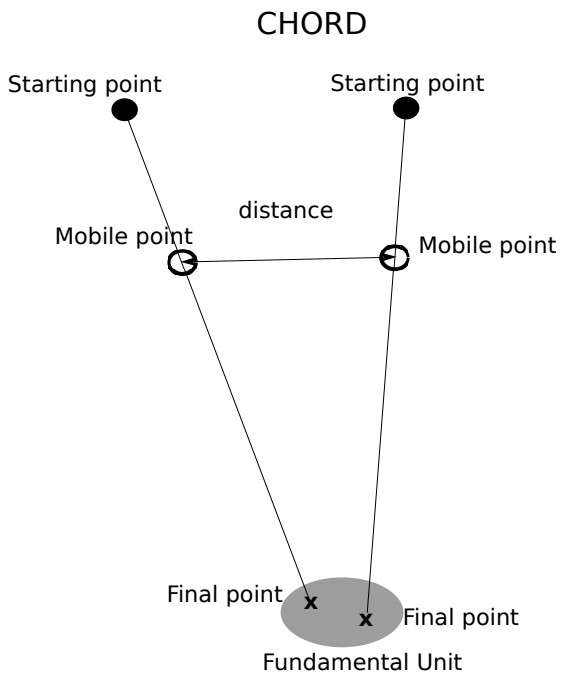


**Figure 5**
Follower CHORD Method

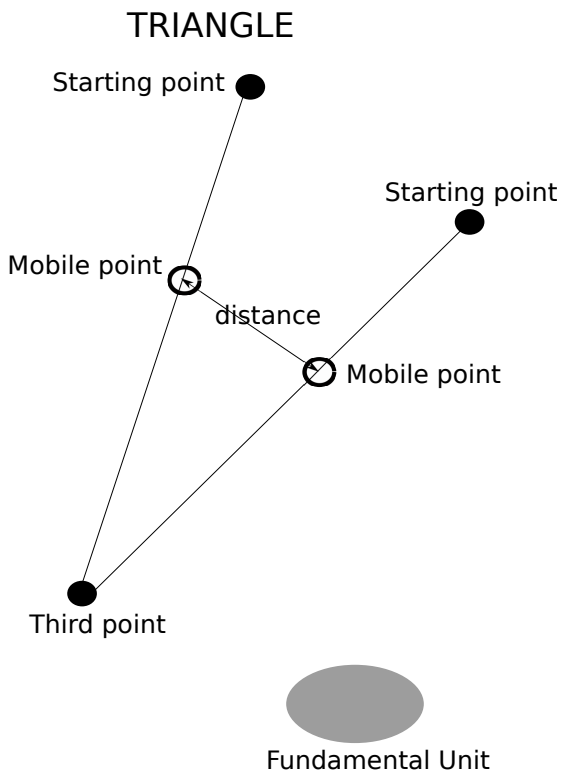


**Figure 7**
Follower TRIANGLE Method

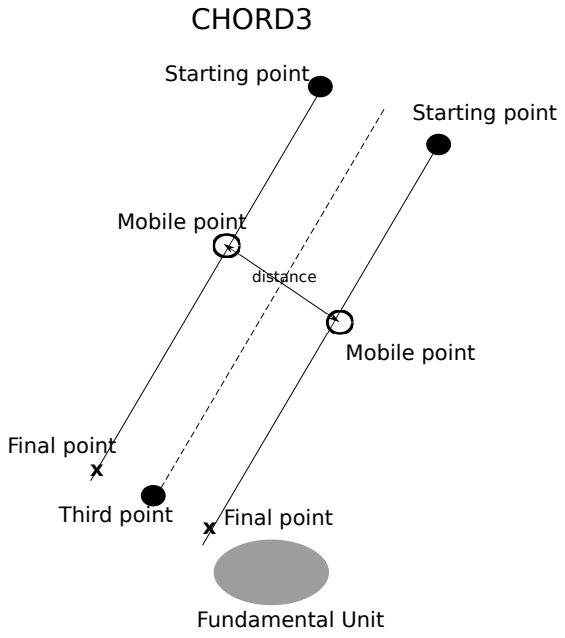


**Figure 6**
Follower CHORD3 Method

The calculations of the points' positions along the line are done in the space $\mathbf{S}^6$ (Andrews *et al.*, 2019*a*). However, the distances can be calculated in whatever space the user chooses.

## 5. Results

Follower is a program with many options. In Figures 8 through 12 we present a few typical graphs of some of the options one might choose.

## 6. Summary

The Follower algorithm for examining continuous functions is described.

## 7. Availability

C++ source code and descriptions of input and how to build Follower are available at

https://github.com/duck10/LatticeRepLib/releases/tag/v1.0

The file README_FOLLOWER.pdf accompanies this article as supplementary material.

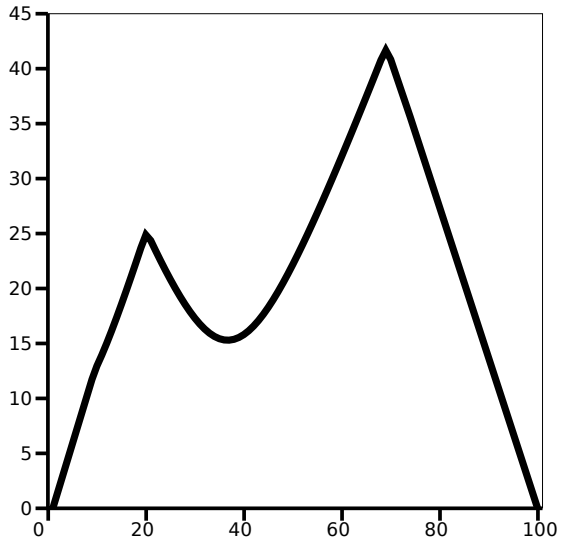


**Figure 8**
**POINT** method with initial mobile point 8.667, 10.510, 10.581, 125.904, 113.666, 99.246 to reduced point 8.667, 14.657, 10.581, 100.55, 113.67, 134.95.

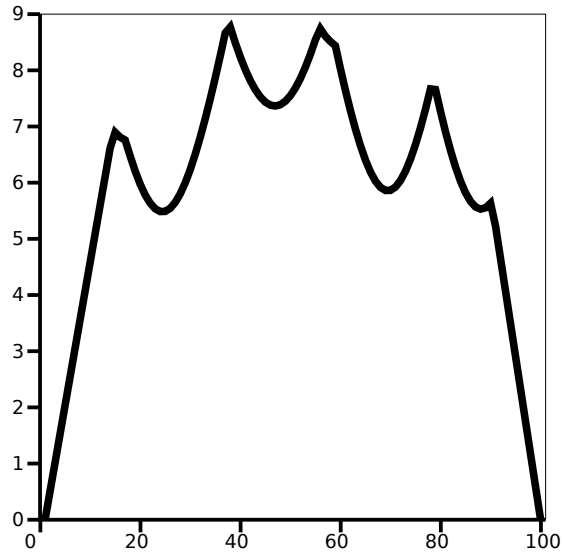


**Figure 10**
**POINT** method with initial mobile point 9.509 9.353, 9.388, 119.74, 117.25, 112.56 to reduced point 9.840, 11.157, 9.509, 103.87, 121.98, 127.22.

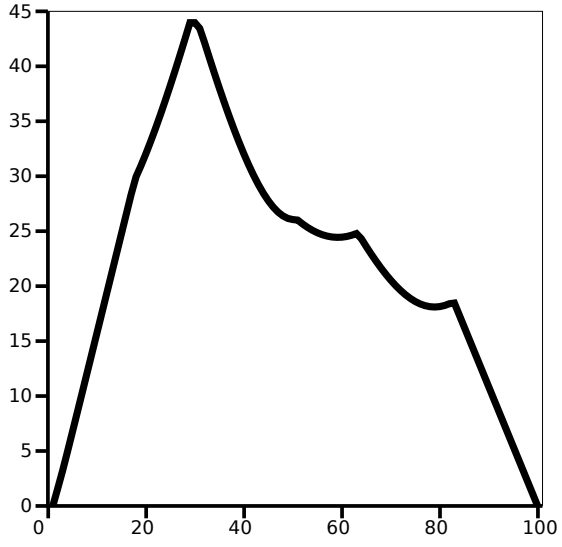


**Figure 9**
**POINT** method with initial mobile point 12.075, 9.894, 7.832, 102.47, 105.51, 132.43 to reduced point 9.083, 12.075, 12.601, 105.51, 120.31, 126.48.

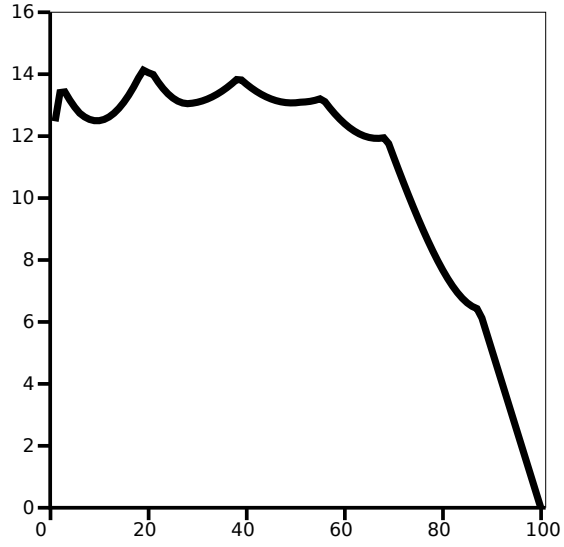


**Figure 11**
**LINE** method with mobile point: Line from 6.852, 7.316, 8.79,3 136.43, 86.78, 130.18 to 5.319, 5.125, 7.778, 113.60, 134.40, 90.60.

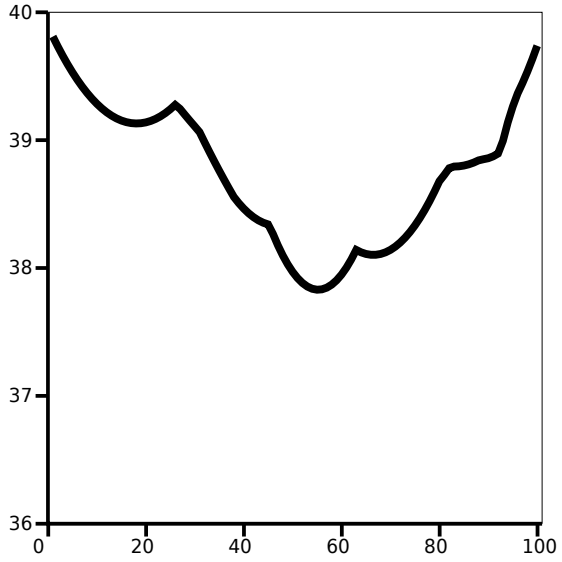


**Figure 12**
**CHORD** method with mobile points: First line from 8.96,2 3.180, 8.746, 113.93, 120.78, 76.08 to 9.629, 3.180, 8.902, 119.86, 112.2,7 100.43. Second line from 7.910, 5.863, 5.126, 105.72, 132.56, 111.11 to 8.659, 5.86,3 5.39,5 110.28, 115.10, 124.98.

**Acknowledgements:** Outstanding copy-editing and corrections by Frances C. Bernstein are gratefully acknowledged. Our thanks to Jean Jakoncic and Alexei Soares for helpful conversations and access to data and facilities at Brookhaven National Laboratory.

**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.

Work funded in part by funding from Brookhaven National Laboratory Center for BioMolecular Structure (CBMS), which is primarily supported by the National Institutes of Health, National Institute of General Medical Sciences (NIGMS) through a Center Core P30 Grant (P30GM133893), and by the DOE Office of Biological and Environmental Research (KP1607011). As part of NSLS-II, a national user facility at Brookhaven National Laboratory, work performed at the CBMS is supported in part by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences Program under contract number and DE-SC0012704.

## References

Andrews, L. C. & Bernstein, H. J. (1988). *Acta Cryst.* A**44**(6), 1009 – 1018.

Andrews, L. C. & Bernstein, H. J. (2014). *J. Appl. Cryst.* **47**(1), 346 – 359.

Andrews, L. C., Bernstein, H. J. & Pelletier, G. A. (1980). *Acta Cryst.* A**36**, 248 – 252.

Andrews, L. C., Bernstein, H. J. & Sauter, N. K. (2019*a*). *Acta Crys.* A**75**(1), 115–120.

Andrews, L. C., Bernstein, H. J. & Sauter, N. K. (2019*b*). *Acta Cryst.* A**75**(3), 593–599.

Bernstein, H. J., Andrews, L. C., Diaz Jr, J. A., Jakoncic, J., Nguyen, T., Sauter, N. K., Soares, A. S., Wei, J. Y., Wlodek, M. R. & Xerri, M. A. (2020). *Struct. Dyn.* **7**(1), 014302.

Delaunay, B. N. (1932). *Z. Krist.* **84**, 109 – 149.

Nguyen, T., Phan, K. L., Kreitler, D. F., Andrews, L. C., Gabelli, S. B., Kozakov, D., Jakoncic, J., Sweet, R. M., Soares, A. S. & Bernstein, H. J. (2020). *bioRxiv*.

Niggli, P., (1928). Krystallographische und Strukturtheoretische Grundbegriffe, Handbuch der Experimentalphysik, Vol. 7, part 1. Akademische Verlagsgesellschaft, Leipzig.