Acta Cryst A

FOUNDATIONS
ADVANCES

Volume 72 (2016)

Supporting information for article:

A topological coordinate system for the diamond cubic grid

Lidija Čomić and Benedek Nagy

We present algorithms that compute various closures of a given set A of cells in the diamond cubic grid. First, an algorithm computing 2-closure is shown:

ALGORITHM 2-CLOSURE (A):

1. **for** every voxel (p, q, r, s) in A **do**
2. add the tri-faces $(p + 3, q - 3, r, s)$,$(p - 3, q + 3, r, s)$,$(p + 3, q, r - 3, s)$, $(p - 3, q, r + 3, s)$, $(p + 3, q, r, s - 3)$,$(p - 3, q, r, s + 3)$,$(p, q + 3, r - 3, s)$,$(p, q - 3, r + 3, s)$, $(p, q + 3, r, s - 3)$, $(p, q - 3, r, s + 3)$,$(p, q, r + 3, s - 3)$,$(p, q, r - 3, s + 3)$ to A
3. **if** $p + q + r + s = 0$ **then** add the hex-faces $(p + 3, q, r, s)$,$(p, q + 3, r, s)$,$(p, q, r + 3, s)$, $(p, q, r, s + 3)$ to A
4. **else** add the hex-faces $(p - 3, q, r, s)$,$(p, q - 3, r, s)$,$(p, q, r - 3, s)$,$(p, q, r, s - 3)$ to A
5. **end for**

The previous algorithm adds all the missing faces of voxels included in the set A.

ALGORITHM 1-CLOSURE (A):

1. **for** every hex-face (p, q, r, s) in A **do**
2. **if** $p \equiv 3 \pmod 6$ **then** add up-hex-edges $(p - 1, q + 2, r, s)$, $(p - 1, q, r + 2, s)$, $(p - 1, q, r, s + 2)$
3. and down-hex-edges $(p + 1, q - 2, r, s)$, $(p + 1, q, r - 2, s)$, $(p + 1, q, r, s - 2)$ to A
4. **if** $q \equiv 3 \pmod 6$ **then** add up-hex-edges $(p + 2, q - 1, r, s)$, $(p, q - 1, r + 2, s)$, $(p, q - 1, r, s + 2)$
5. and down-hex-edges $(p - 2, q + 1, r, s)$, $(p, q + 1, r - 2, s)$, $(p, q + 1, r, s - 2)$ to A
6. **if** $r \equiv 3 \pmod 6$ **then** add up-hex-edges $(p + 2, q, r - 1, s)$, $(p, q + 2, r - 1, s)$, $(p, q, r - 1, s + 2)$
7. and down-hex-edges $(p - 2, q, r + 1, s)$, $(p, q - 2, r + 1, s)$, $(p, q, r + 1, s - 2)$ to A
8. **if** $s \equiv 3 \pmod 6$ **then** add up-hex-edges $(p + 2, q, r, s - 1)$, $(p, q + 2, r, s - 1)$, $(p, q, r + 2, s - 1)$
9. and down-hex-edges $(p - 2, q, r, s + 1)$, $(p, q - 2, r, s + 1)$, $(p, q, r - 2, s + 1)$ to A
10. **end for**
11. **for** every up-tri-face (p, q, r, s) in A **do**
12. **if** $p, q \equiv 3 \pmod 6$ **then** add up-hex-edges $(p - 1, q - 1, r + 2, s)$, $(p - 1, q - 1, r, s + 2)$
13. and down-hex-edge $(p + 1, q + 1, r, s)$ to A
14. **if** $p, r \equiv 3 \pmod 6$ **then** add up-hex-edges $(p - 1, q + 2, r - 1, s)$, $(p - 1, q, r - 1, s + 2)$
15. and down-hex-edge $(p + 1, q, r + 1, s)$ to A
16. **if** $p, s \equiv 3 \pmod 6$ **then** add up-hex-edges $(p - 1, q + 2, r, s - 1)$, $(p - 1, q, r + 2, s - 1)$
17. and down-hex-edge $(p + 1, q, r, s + 1)$ to A
18. **if** $q, r \equiv 3 \pmod 6$ **then** add up-hex-edges $(p + 2, q - 1, r - 1, s)$, $(p, q - 1, r - 1, s + 2)$
19. and down-hex-edge $(p, q + 1, r + 1, s)$ to A
20. **if** $q, s \equiv 3 \pmod 6$ **then** add up-hex-edges $(p + 2, q - 1, r, s - 1)$, $(p, q - 1, r + 2, s - 1)$
21. and down-hex-edge $(p, q + 1, r, s + 1)$ to A
22. **if** $r, s \equiv 3 \pmod 6$ **then** add up-hex-edges $(p + 2, q, r - 1, s - 1)$, $(p, q + 2, r - 1, s - 1)$
23. and down-hex-edge $(p, q, r + 1, s + 1)$ to A
24. **end for**
25. **for** every down-tri-face (p, q, r, s) in A **do**
26. **if** $p, q \equiv 3 \pmod 6$ **then** add down-hex-edges $(p + 1, q + 1, r - 2, s)$, $(p + 1, q + 1, r, s - 2)$
27. and up-hex-edge $(p - 1, q - 1, r, s)$ to A
28. **if** $p, r \equiv 3 \pmod 6$ **then** add down-hex-edges $(p + 1, q - 2, r + 1, s)$, $(p + 1, q, r + 1, s - 2)$
29. and up-hex-edge $(p - 1, q, r - 1, s)$ to A
30. **if** $p, s \equiv 3 \pmod 6$ **then** add down-hex-edges $(p + 1, q - 2, r, s + 1)$, $(p + 1, q, r - 2, s + 1)$
31. and up-hex-edge $(p - 1, q, r, s - 1)$ to A
32. **if** $q, r \equiv 3 \pmod 6$ **then** add down-hex-edges $(p - 2, q + 1, r + 1, s)$, $(p, q + 1, r + 1, s - 2)$
33. and up-hex-edge $(p, q - 1, r - 1, s)$ to A
34. **if** $q, s \equiv 3 \pmod 6$ **then** add down-hex-edges $(p - 2, q + 1, r, s + 1)$, $(p, q + 1, r - 2, s + 1)$
35. and up-hex-edge $(p, q - 1, r, s - 1)$ to A
36. **if** $r, s \equiv 3 \pmod 6$ **then** add down-hex-edges $(p - 2, q, r + 1, s + 1)$, $(p, q - 2, r + 1, s + 1)$
37. and up-hex-edge $(p, q, r - 1, s - 1)$ to A
38. **end for**

The previous two algorithms add first all the missing faces, and then all the missing edges bordering any of the faces already included in the set A. Finally, the next algorithm adds all the missing vertices incident to the edges in A.

```
ALGORITHM 0-CLOSURE (A):
for every up-hex-edge (p, q, r, s) in A do
    if p, q ≡ 2( mod 6) then add the hex-vertices (p + 1, q + 1, r − 3, s), (p + 1, q + 1, r, s − 3) to A
    if p, r ≡ 2( mod 6) then add the hex-vertices (p + 1, q − 3, r + 1, s), (p + 1, q, r + 1, s − 3) to A
    if p, s ≡ 2( mod 6) then add the hex-vertices (p + 1, q − 3, r, s + 1), (p + 1, q, r − 3, s + 1) to A
    if q, r ≡ 2( mod 6) then add the hex-vertices (p − 3, q + 1, r + 1, s), (p, q + 1, r + 1, s − 3) to A
    if q, s ≡ 2( mod 6) then add the hex-vertices (p − 3, q + 1, r, s + 1), (p, q + 1, r − 3, s + 1) to A
    if r, s ≡ 2( mod 6) then add the hex-vertices (p − 3, q, r + 1, s + 1), (p, q − 3, r + 1, s + 1) to A
end for
for every down-hex-edge (p, q, r, s) in A do
    if p, q ≡ 4( mod 6) then add the hex-vertices (p − 1, q − 1, r + 3, s), (p − 1, q − 1, r, s + 3) to A
    if p, r ≡ 4( mod 6) then add the hex-vertices (p − 1, q + 3, r − 1, s), (p − 1, q, r − 1, s + 3) to A
    if p, s ≡ 4( mod 6) then add the hex-vertices (p − 1, q + 3, r, s − 1), (p − 1, q, r + 3, s − 1) to A
    if q, r ≡ 4( mod 6) then add the hex-vertices (p + 3, q − 1, r − 1, s), (p, q − 1, r − 1, s + 3) to A
    if q, s ≡ 4( mod 6) then add the hex-vertices (p + 3, q − 1, r, s − 1), (p, q − 1, r + 3, s − 1) to A
    if r, s ≡ 4( mod 6) then add the hex-vertices (p + 3, q, r − 1, s − 1), (p, q + 3, r − 1, s − 1) to A
end for
for every up-tri-edge (p, q, r, s) in A do
    if p ≡ 0( mod 6) then add the vertices (p + 2, q, r, s), (p, q + 1, r + 1, s + 1) to A
    if q ≡ 0( mod 6) then add the vertices (p, q + 2, r, s), (p + 1, q, r + 1, s + 1) to A
    if r ≡ 0( mod 6) then add the vertices (p, q, r + 2, s), (p + 1, q + 1, r, s + 1) to A
    if s ≡ 0( mod 6) then add the vertices (p, q, r, s + 2), (p + 1, q + 1, r + 1, s) to A
end for
for every down-tri-edge (p, q, r, s) in A do
    if p ≡ 0( mod 6) then add the vertices (p − 2, q, r, s), (p, q − 1, r − 1, s − 1) to A
    if q ≡ 0( mod 6) then add the vertices (p, q − 2, r, s), (p − 1, q, r − 1, s − 1) to A
    if r ≡ 0( mod 6) then add the vertices (p, q, r − 2, s), (p − 1, q − 1, r, s − 1) to A
    if s ≡ 0( mod 6) then add the vertices (p, q, r, s − 2), (p − 1, q − 1, r − 1, s) to A
end for
```

The closure can be computed by executing the above algorithms in a consecutive way, as follows: 0-CLOSURE(1-CLOSURE(2-CLOSURE(A))).