Introduction to the Hoshen-Kopelman algorithm and its application to nodal domain statistics

Christian Joas
Institut für theoretische Physik, FU Berlin

Nodal Week, Weizmann Institute, Rehovot (IL)
Tuesday, April 4, 2006

Outline

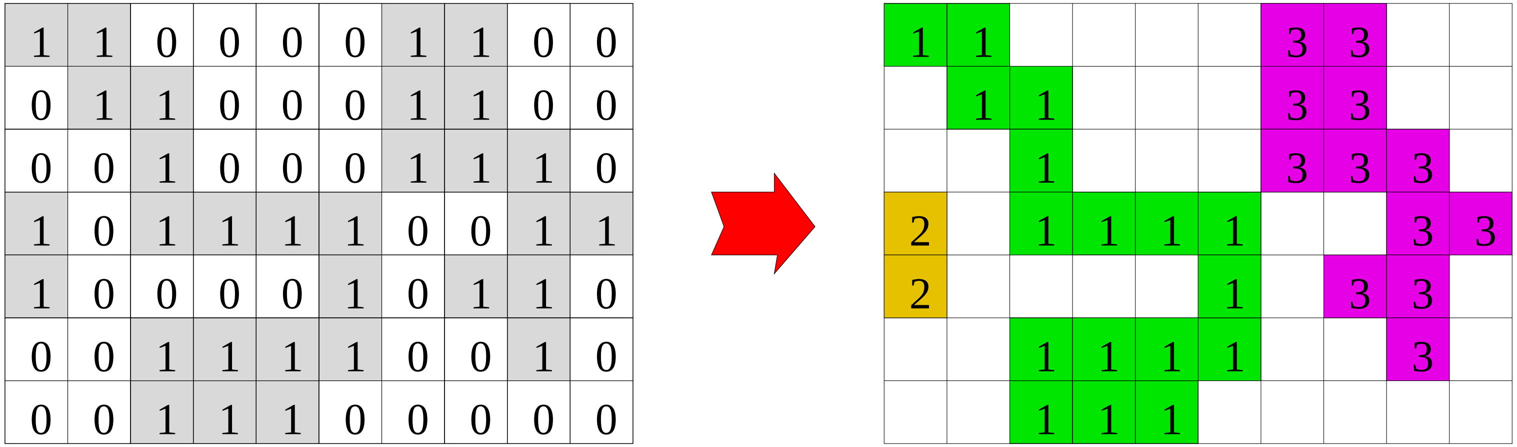
- Hoshen-Kopelman algorithm

HOSHEN (from Wikipedia): The Hoshen (Khosen) was the breastplate "of Judgment" worn by the High Priest in the book of Exodus, covered by 12 stones that represented the 12 tribes of Israel, arranged in a pattern of four rows of three according to Exodus 28:17-21 The priest bore the names of the sons of Israel in the breastplate of judgment, upon his heart, when he would enter the Holy place, to bring them in continual remembrance before the Lord, Exodus 28:29.

- specific aspects for the study of nodal domains
  - treatment of nearly avoided crossings of nodal lines
  - determination of nodal line lengths and domain areas
  - connectivity

Hoshen-Kopelman Algorithm

- simple, efficient algorithm for labeling clusters on a grid



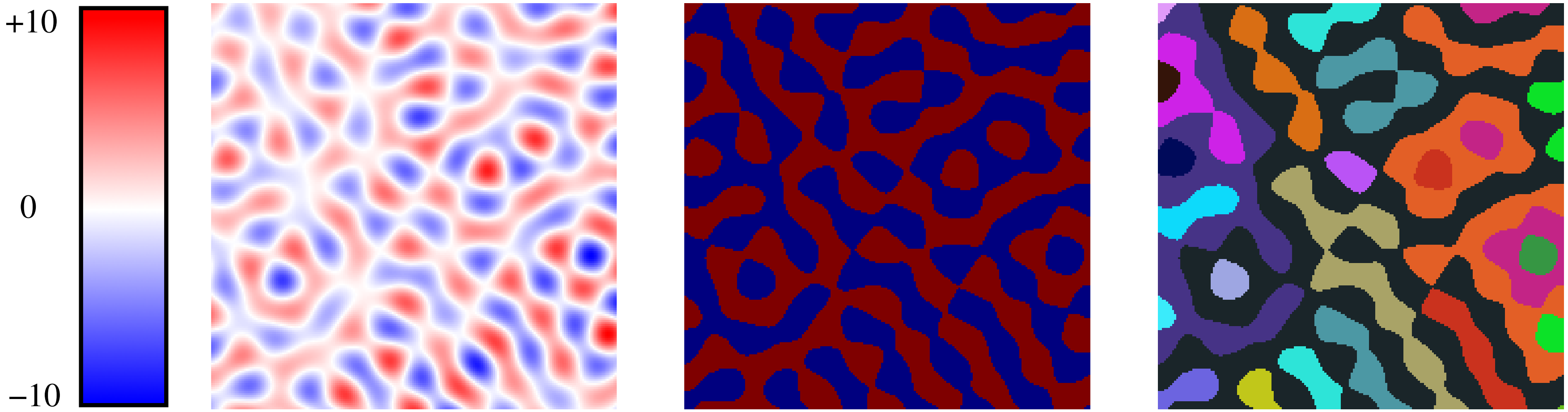
1	1	0	0	0	0	1	1	0	0
0	1	1	0	0	0	1	1	0	0
0	0	1	0	0	0	1	1	1	0
1	0	1	1	1	1	0	0	1	1
1	0	0	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0	1	0
0	0	1	1	1	0	0	0	0	0

1	1					3	3		
	1	1				3	3		
		1				3	3	3	
2		1	1	1	1			3	3
2					1		3	3	
		1	1	1	1			3	
		1	1	1					

- very low computer memory usage and computation time compared to other methods
- first introduced in context of cluster percolation

  [Percolation and Cluster Distribution. I. Cluster multiple labeling technique and critical concentration algorithm, Joseph Hoshen and Raoul Kopelman, PRB **14**, 3438 (1976)]
- special realization of UNION-FIND algorithm for computation of equivalence classes, widely used in computer science

Application to Nodal Domains



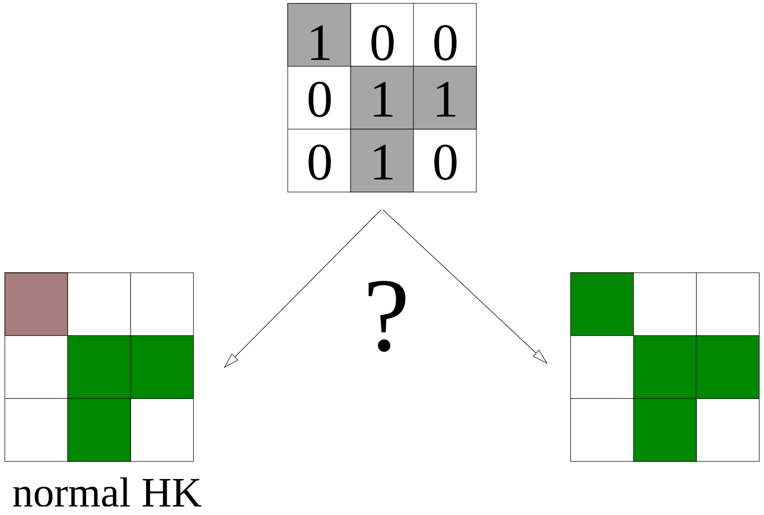
wavefunction value sign of wavefunction domains

objectives:

- label and count positive and negative domains
- determine individual domain areas and domain line lengths
- extract connectivity information

Basics of HK

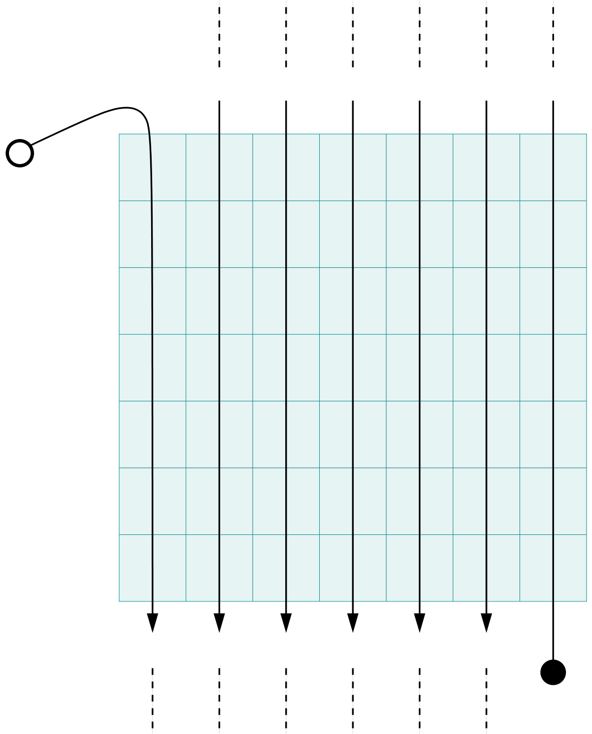
- grid of $M \cdot N$ sites (grid points)
- values at these sites
  - percolation clusters: only 1 (occupied site) and 0 (free site)
  - nodal domains: sign of wavefunction values sgn$[\Psi(\mathbf{r}_{ij})]$
- cluster criterion:



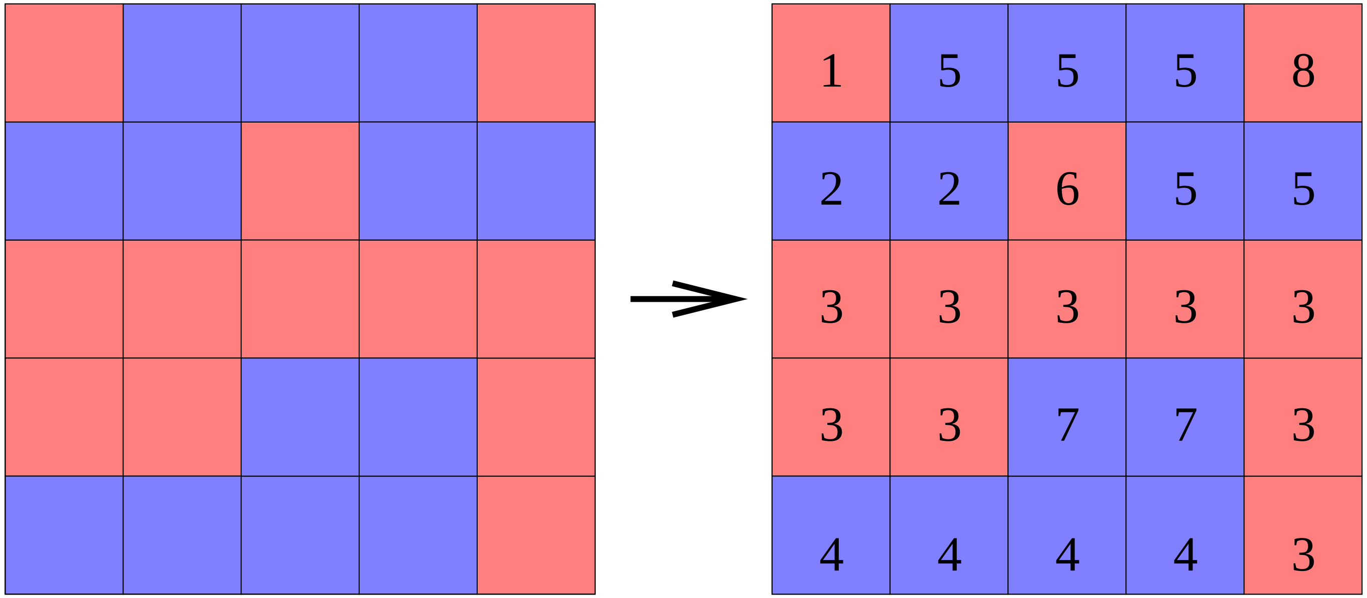
1	0	0
0	1	1
0	1	0

Domain labeling

- HK routine performs column-by-column raster scan through the grid
  ⇒ wavefunction can be stored in a vector



Example



🟥	🟦	🟦	🟦	🟥
🟦	🟦	🟥	🟦	🟦
🟥	🟥	🟥	🟥	🟥
🟥	🟥	🟦	🟦	🟥
🟦	🟦	🟦	🟦	🟥

1	5	5	5	8
2	2	6	5	5
3	3	3	3	3
3	3	7	7	3
4	4	4	4	3

- need to introduce equivalence classes for domain labels

$$5 \equiv 2\,, \qquad 6 \equiv 3\,, \qquad 7 \equiv 4$$

Implementation of HK

- vector $\mathbf{L}$ of cluster labels (equivalence classes):

$$L_i = j \Rightarrow \text{ labels } i \text{ and } j \text{ equivalent}$$

- $M \cdot N$ grid points $\Rightarrow$ at most $M \cdot N$ equivalence classes $\Rightarrow$ max. of $M \cdot N$ different labels.

$$\text{initialization:} \quad \mathbf{L} = (\,1, 2, 3, 4, 5, 6, \ldots, M \cdot N\,)$$

- consider sign of left and above sites and decide whether current label is member of the left and/or above equivalence class
  - if no, attribute new label
  - if yes, attribute corresponding label ($\Rightarrow$ Find operation) and unite left and above labels, if appropriate ($\Rightarrow$ Union operation)
- collapse labels so that every grid point is labeled by the representative label of its equivalence class (can be done in a second raster scan or simultaneously with the first raster scan)

Union and Find operations

- Union (i, j) makes two labels i and j equivalent by linking their respective aliases

  example: labels 6 and 4 and labels 4 and 2 are equivalent

$$\mathbf{L} = (1, 2, 3, 4, 5, 6) \xmapsto{\textsc{Union}(6,4)} \mathbf{L}' = (1, 2, 3, 4, 5, 4) \xmapsto{\textsc{Union}(4,2)} \mathbf{L}'' = (1, 2, 3, 2, 5, 4)$$

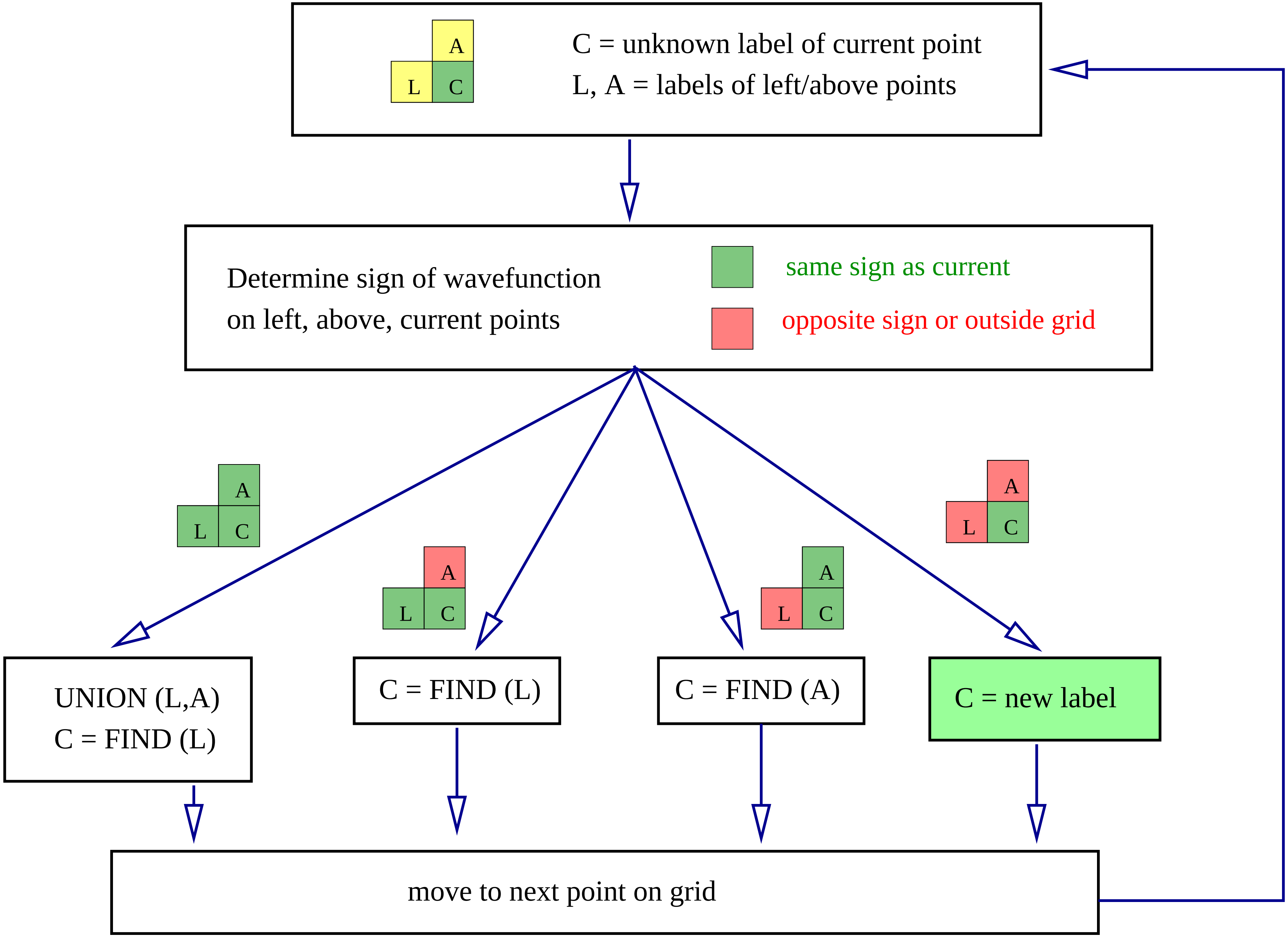
- Find (i) finds representative member of equivalence class (defined by Find $(i) = i$)

  example: for the above $\mathbf{L}''$: $\textsc{Find}(6) = 2$

- improved Find operation: simultaneously collapses labels in $\mathbf{L}$

  example: in the above case, $\textsc{Find}(6)$ returns $\textsc{Find}(6) = 2$ and maps

$$\mathbf{L}'' = (1, 2, 3, 2, 5, 4) \xmapsto{\textsc{Find}(6)} \mathbf{L}''' = (1, 2, 3, 2, 5, 2)$$



C = unknown label of current point

L, A = labels of left/above points

Determine sign of wavefunction on left, above, current points

🟩 same sign as current

🟥 opposite sign or outside grid

UNION (L,A)
C = FIND (L)

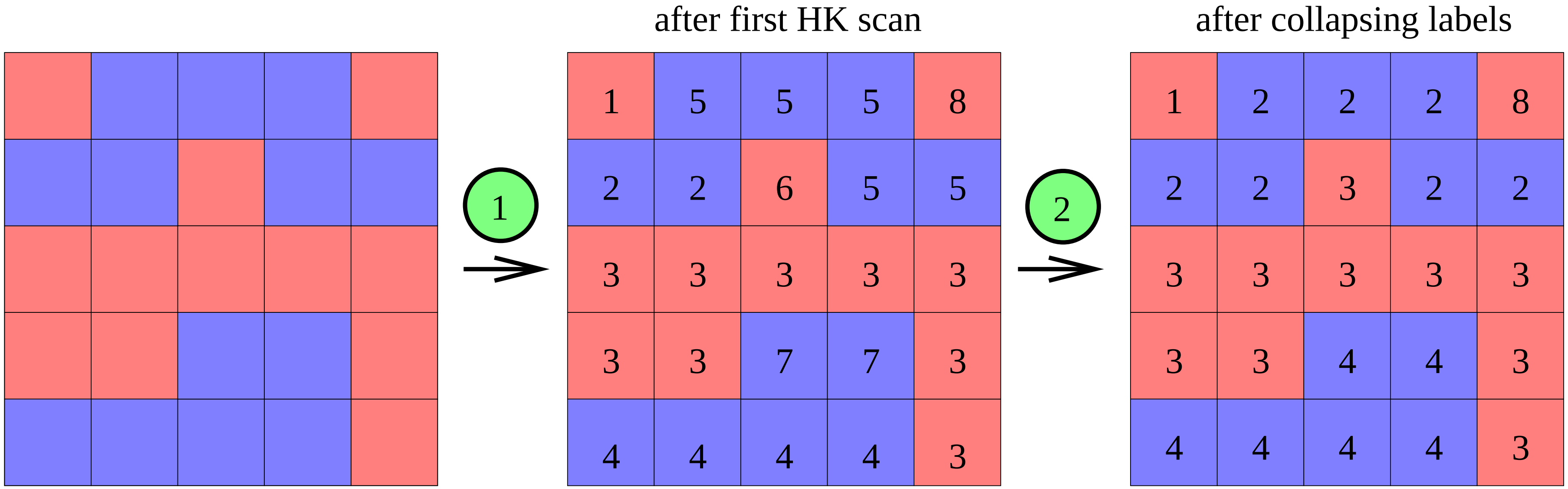
C = FIND (L)

C = FIND (A)

C = new label

move to next point on grid

Example



🟥	🟦	🟦	🟦	🟥
🟦	🟦	🟥	🟦	🟦
🟥	🟥	🟥	🟥	🟥
🟥	🟥	🟦	🟦	🟥
🟦	🟦	🟦	🟦	🟥

after first HK scan

1	5	5	5	8
2	2	6	5	5
3	3	3	3	3
3	3	7	7	3
4	4	4	4	3

after collapsing labels

1	2	2	2	8
2	2	3	2	2
3	3	3	3	3
3	3	4	4	3
4	4	4	4	3

```
Union(2,5)
Union(3,6)
Union(4,7)
```

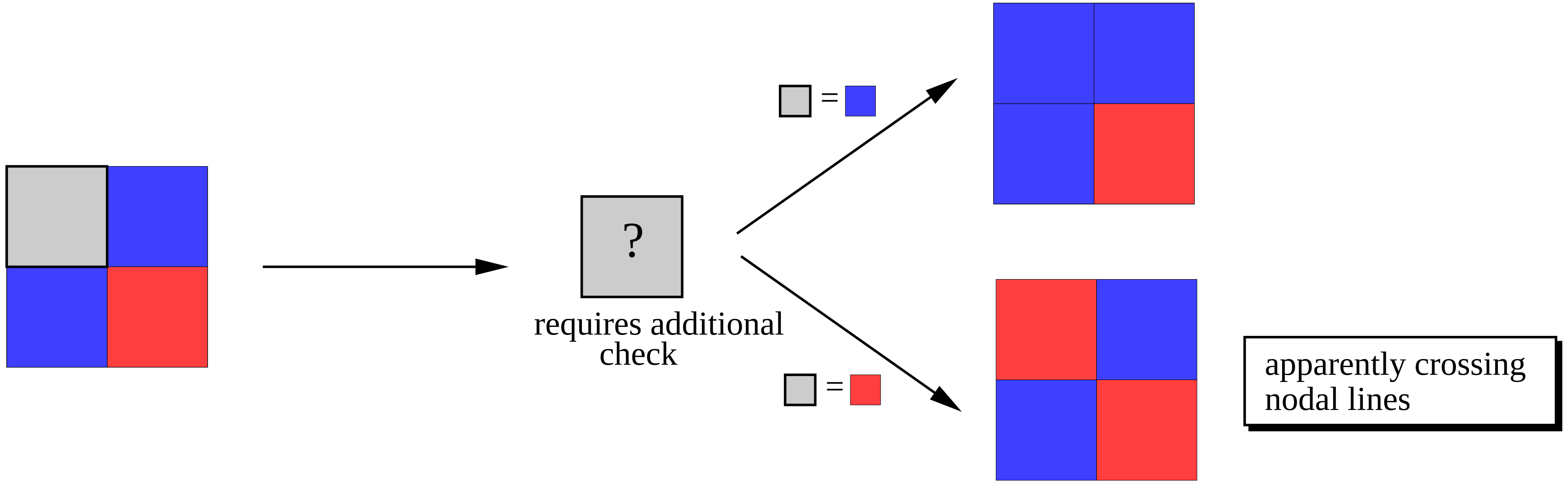
```
Find(5) = 2
Find(6) = 3
Find(7) = 4
```

L = (1,2,3,4,5,6,7,8,...) → L = (1,2,3,4,2,3,4,8) → L = (1,2,3,4,0,0,0,8)

Extensions for Nodal Domain Statistics

- HK routine can easily be extended to extract further information about nodal domains:
  - treatment of apparently crossing nodal lines
  - determination of nodal domain area and nodal line lengths
  - retrieval of connectivity information

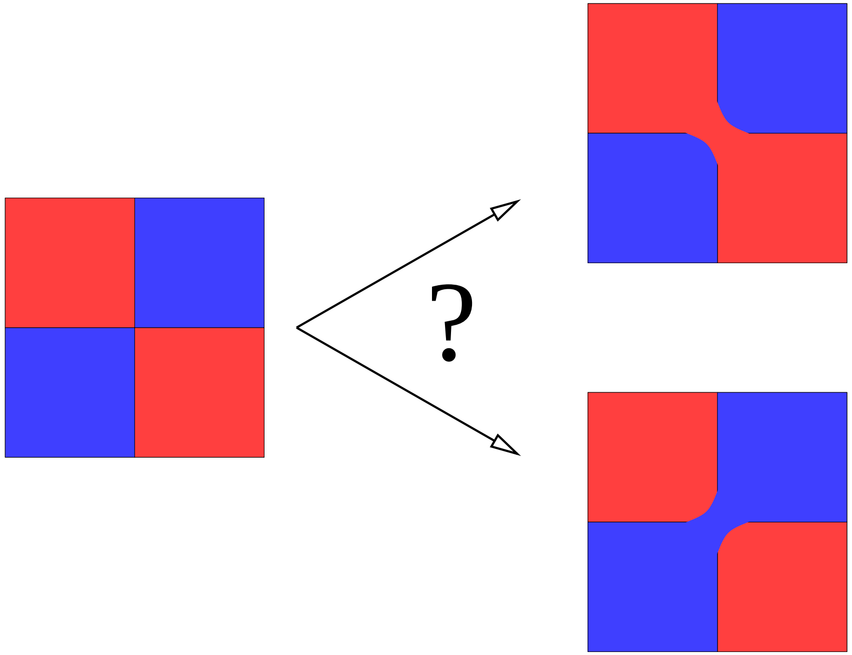
Apparently Crossing Nodal Lines



- problem of crossing not present in cluster percolation (no intertwining clusters)

- during HK scan, additional query has to be made about sign (or label) of left above cluster

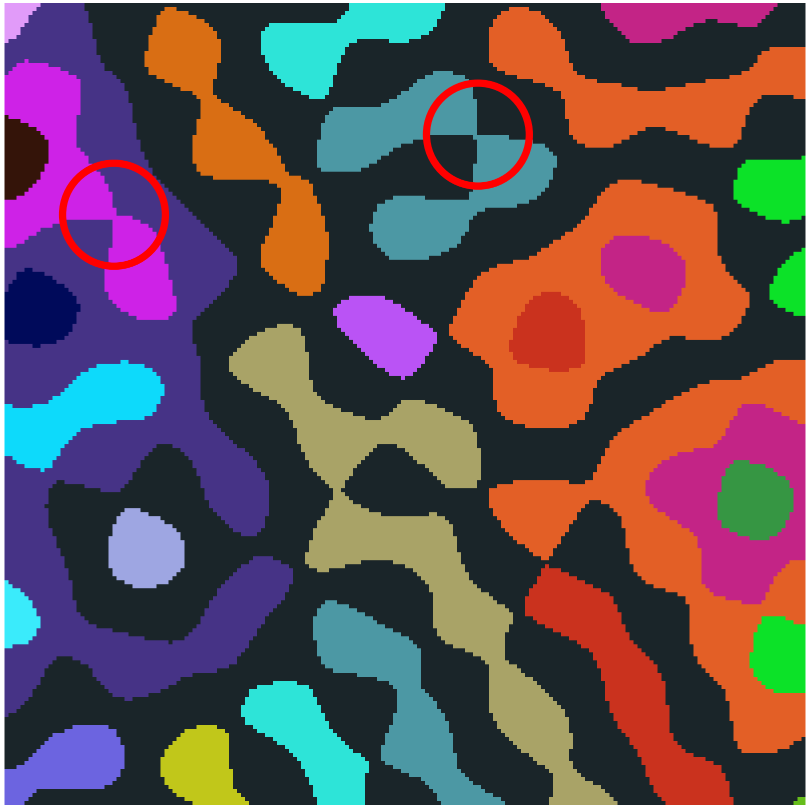
Solution



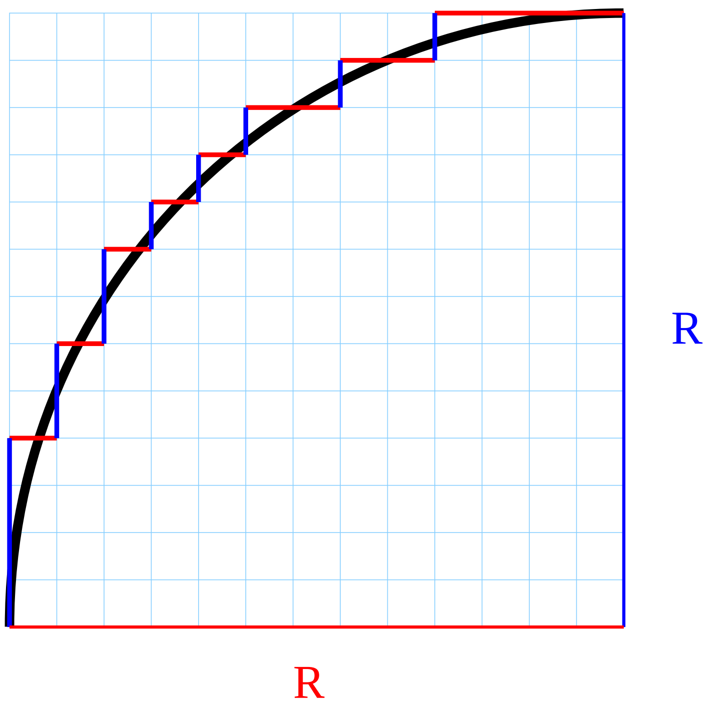
Solution: calculate sum S of wavefunction values on the four grid points

- if $S > 0$, join domains with positive signs ($\textsc{Union}$ operation)
- if $S \leq 0$, join domains with negative signs ($\textsc{Union}$ operation)

Example



Nodal Line Length Determination



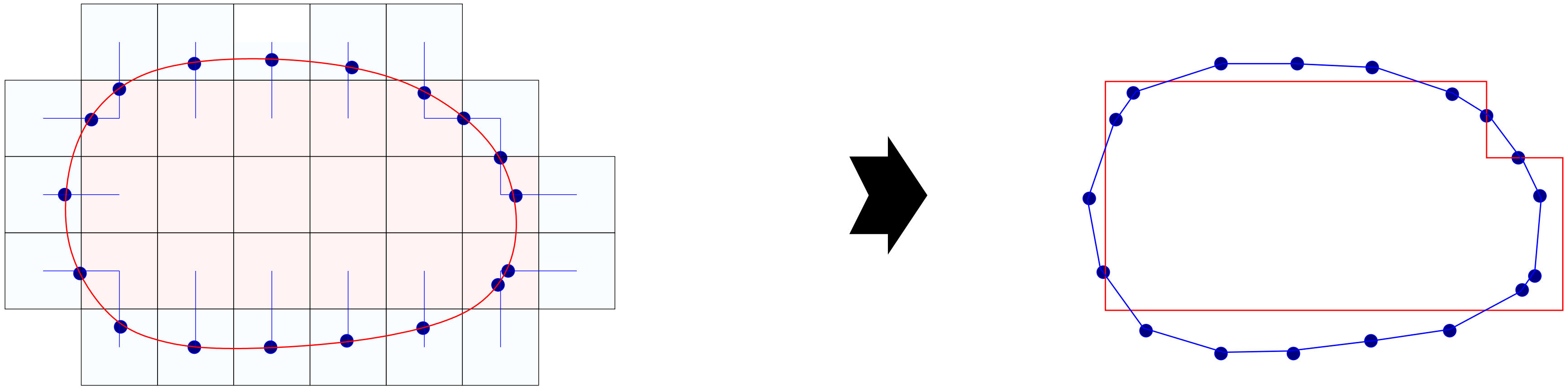
$$\frac{\pi}{2}R \neq R + R = 2R$$

⇒ alternative algorithm needed for determination of line length

Nodal Line Lengths

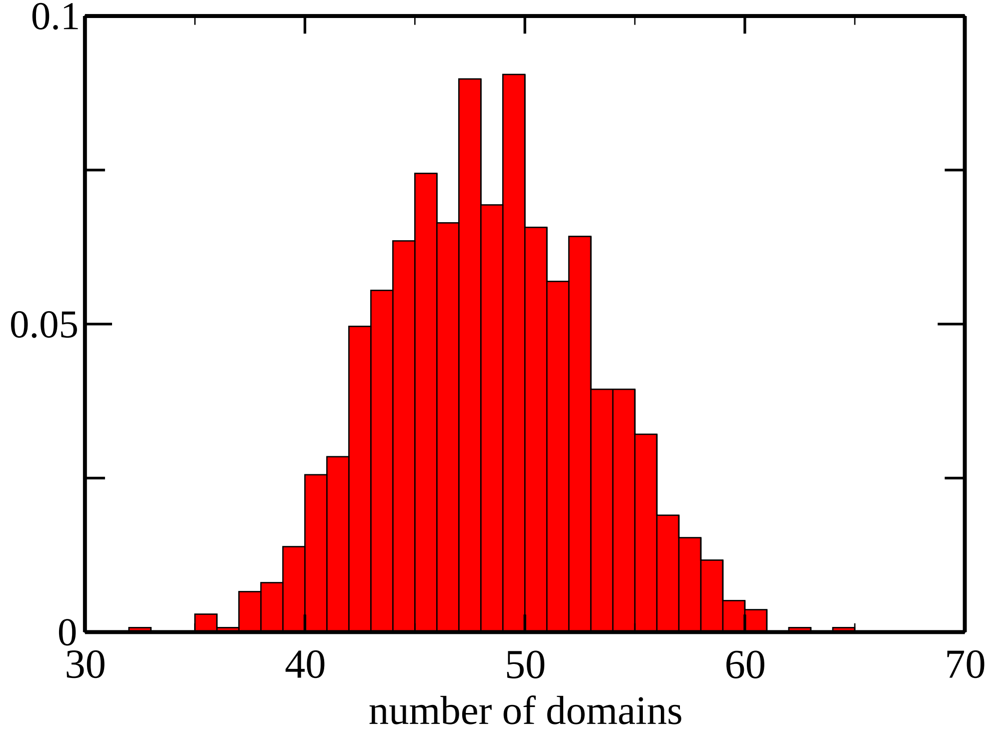
- extract information about points on nodal line during HK scan
- determine approximate position of points on nodal line by interpolating with respect to neighboring grid point of opposite sign
- after HK, travel along nodal line and join points by line segments

ADVANTAGE: segments of approximate nodal line need not be horizontal or vertical
$\Longrightarrow$ leads to good convergence with increasing grid finesse



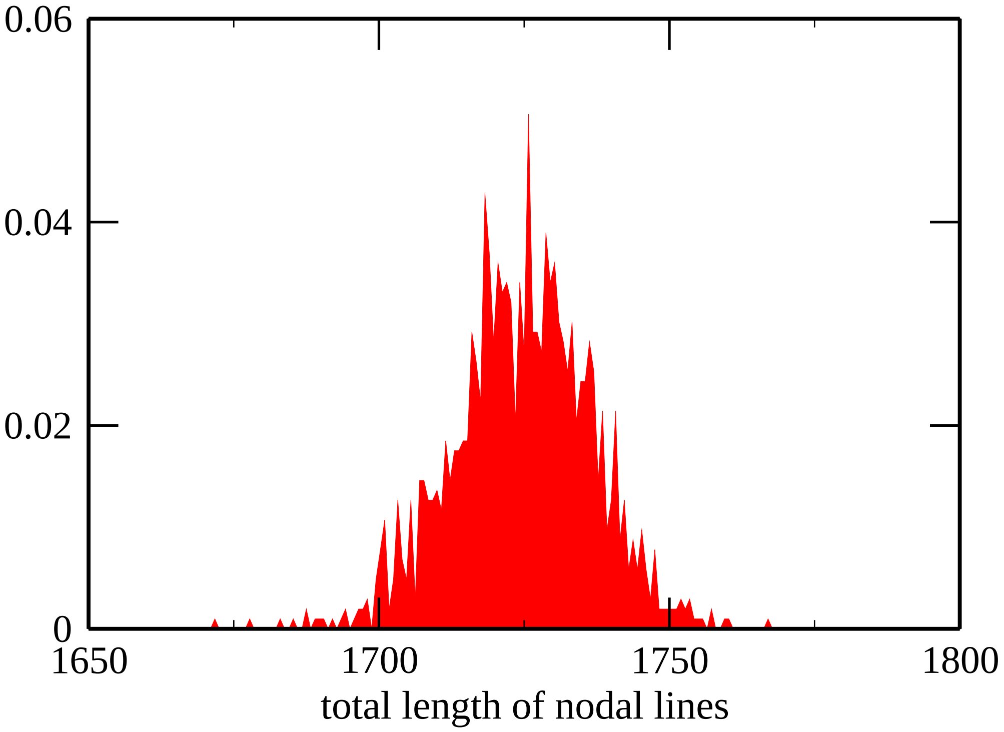
Number of nodal domains

(1370 wavefunctions)



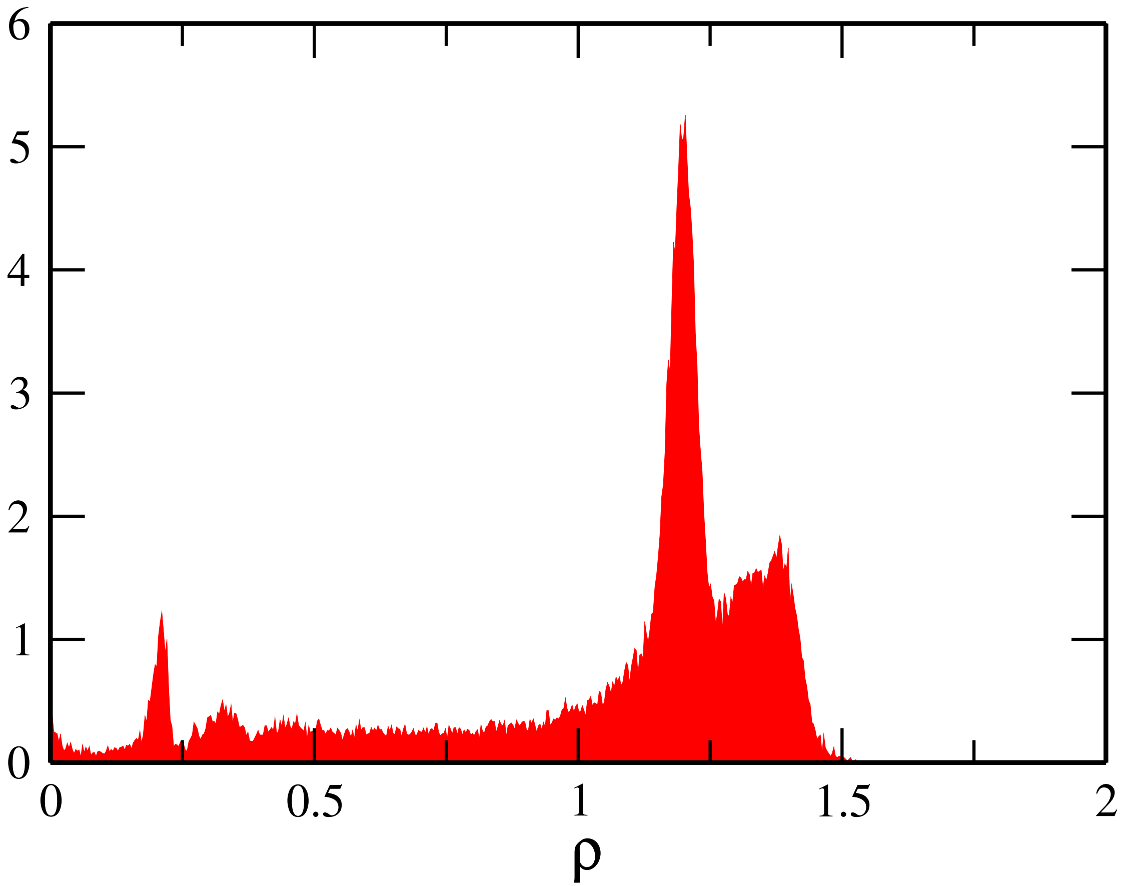
Total length of nodal lines (without bundary)

(1370 wavefunctions)



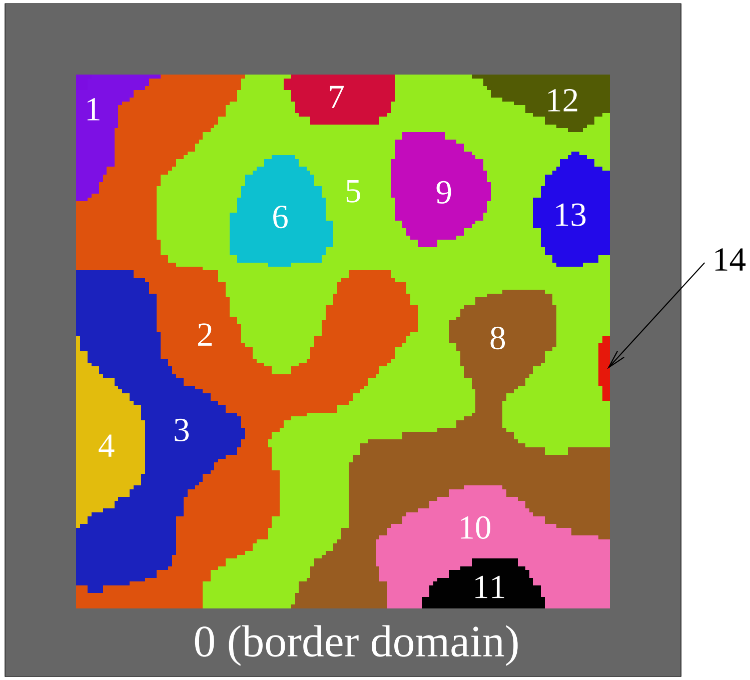
Distribution of $\rho = A/L$ for individual domains

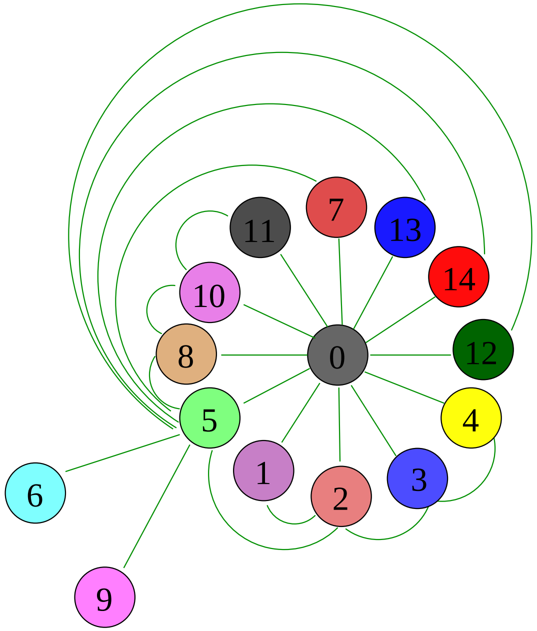
(66886 domains from random superposition of plane waves)



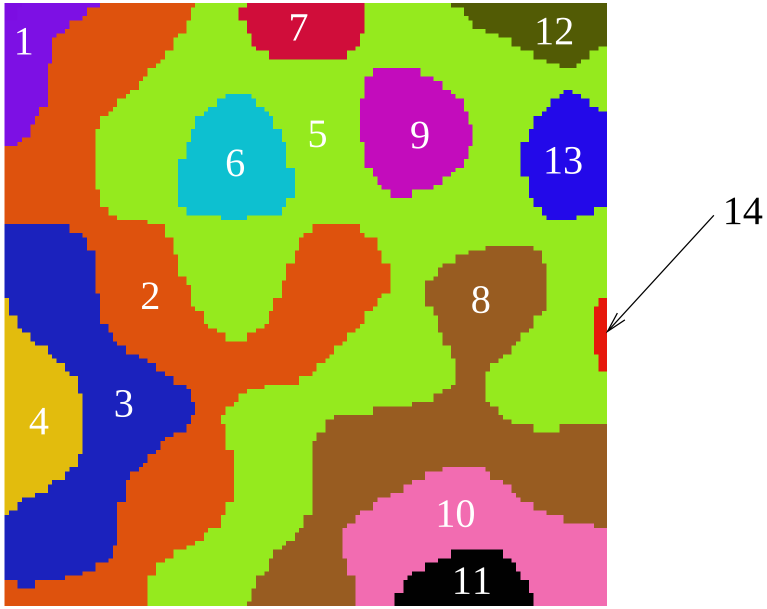
Connectivity

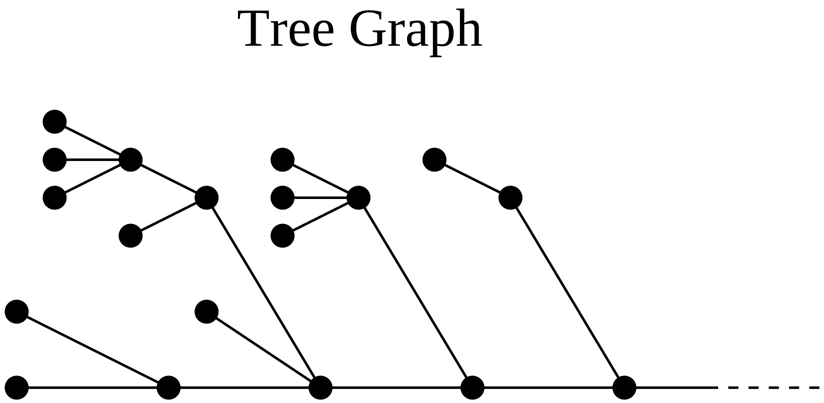
- connectivity information can easily be retrieved during the HK scan and be used for a more efficient calculation of line lengths.
- introduce border domain labeled '0'





Connectivity





- What is the statistics of the number of neighboring domains?

Summary

- Hoshen-Kopelman algorithm easily applicable to study of nodal domains
- attention has to be paid to apparently crossing nodal lines
- allows extraction of
  - nodal line areas
  - nodal line lengths
  - connectivity information
  - ...