The polynomial method for percolation and the Potts model

Christian Scullard
Lawrence Livermore National Laboratory

Collaborators:

Jesper Jacobsen
École Normale Supérieure

Robert Ziff
University of Michigan

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

Introduction

- Method of critical polynomials is used to locate critical points in percolation and the Potts model
- Critical points found as zeros of a graph polynomial
- For many standard problems, the accuracy outstrips Monte Carlo and standard transfer matrix methods
- Provides detailed view of antiferromagnetic regime of the Potts model

Introduction

- Potts model partition function

$$Z = \sum_{\{\sigma\}} \prod_{<ij>} \exp(K\delta_{\sigma_i \sigma_j})$$

- Spins $\sigma_i \in \{0, 1, ..., q-1\}$
- For $d > 1$, critical points are not known exactly, except for a few special cases in $d = 2$
- Numerical techniques needed: Monte Carlo, transfer matrix

Introduction

- Instead of physical representation, we define $v \equiv e^K - 1$ to get random cluster representation

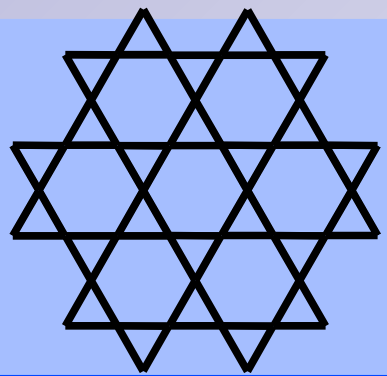
$$Z = \sum_{A \subseteq E} v^{|A|} q^{C(A)}$$

- With $q = 1$ and $v = p/(1-p)$ we get bond percolation: each edge has probability p of being open. Critical probability p_c marks transition to infinite cluster
- Most critical thresholds are not exactly known in percolation

Introduction

- Example: kagome lattice bond percolation

Monte Carlo: $p_c = 0.52440503(3)$



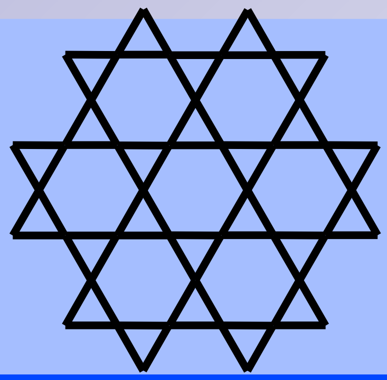
Introduction

- Example: kagome lattice bond percolation

Monte Carlo: $p_c = 0.52440503(3)$

Transfer Matrix: $p_c = 0.52440499(2)$

(Feng, Deng and Blöte, 2008)



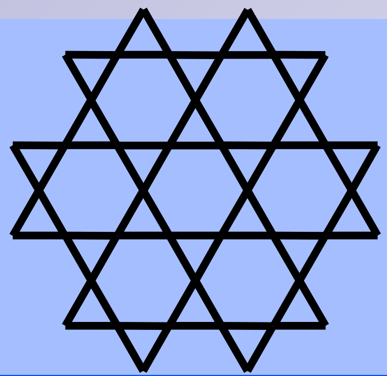
Introduction

- Example: kagome lattice bond percolation

Monte Carlo: $p_c = 0.52440503(3)$ (Feng, Deng and Blöte, 2008)

Transfer Matrix: $p_c = 0.52440499(2)$

Critical Polynomials: $p_c = 0.524404999170(2)$



Introduction

- Example: kagome lattice bond percolation

Monte Carlo: $p_c = 0.52440503(3)$

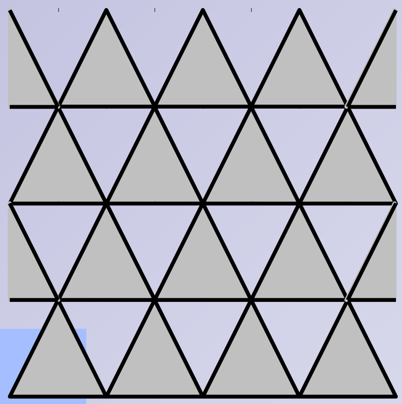
Transfer Matrix: $p_c = 0.52440499(2)$

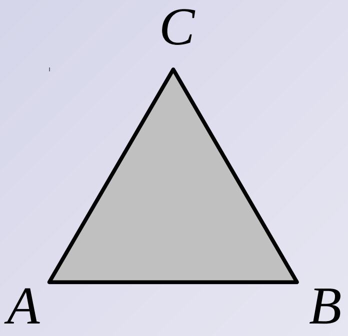
Critical Polynomials: $p_c = 0.524404999170(2)$

- Method can be applied on any periodic lattice and for any q

2D Exact Solutions

- In two dimensions, problems with unit cells contained between three vertices can be solved exactly

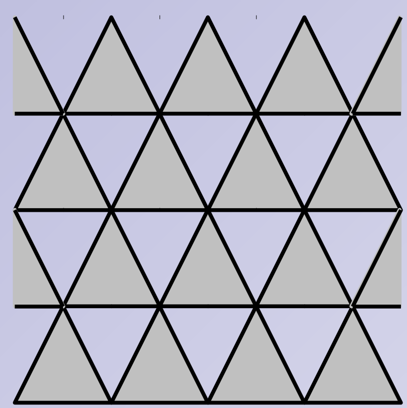


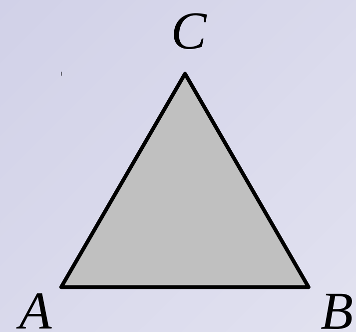


- The critical manifolds are always given by zeros of polynomials in v and q

Critical Polynomials

- Percolation critical criterion





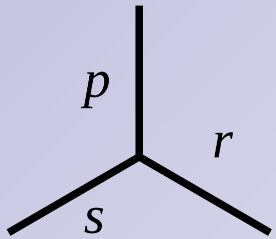
$$P(A,B,C) = P(\bar{A},\bar{B},\bar{C})$$

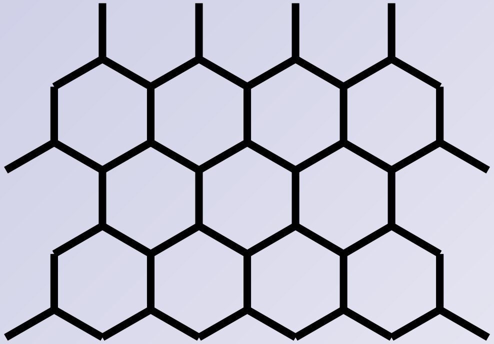
- Grey triangle can be essentially any network of vertices and edges (CS 2006, Ziff 2006, Bollobás and Riordan 2010)

Critical Polynomials

- Examples: $P(A,B,C) - P(\bar{A},\bar{B},\bar{C}) = 0$

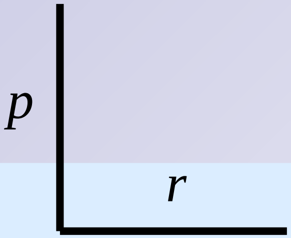
Hexagonal lattice:

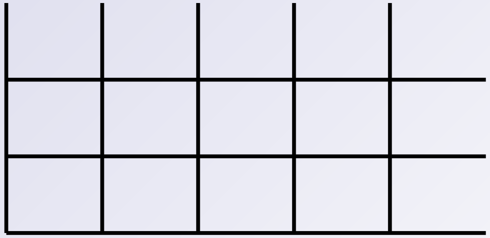




$$H(p,r,s) = prs - pr - ps - rs + 1 = 0$$

Square lattice:





$$S(p,r) = H(p,r,1) = 1 - p - r = 0$$

Critical Polynomials

- All critical surfaces are multi-linear functions in the probabilities
- Setting all probabilities equal gives critical polynomial

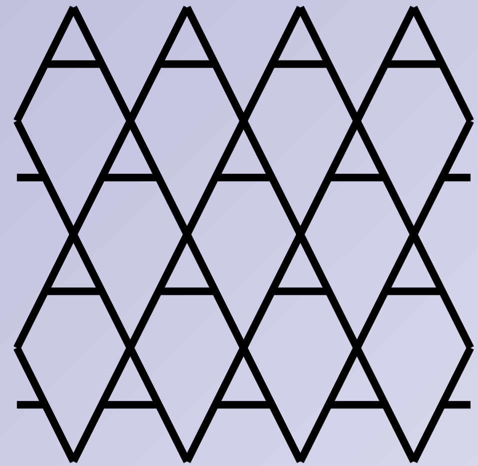
Hexagonal lattice: $H(p,p,p) = p^3 - 3p^2 + 1 = 0$

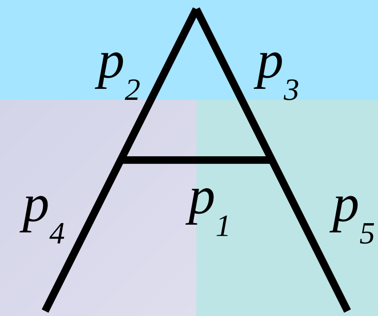
$$p_c = 1 - 2\sin\pi/18 \approx 0.652704$$

- So all exact thresholds are algebraic numbers
- Critical surfaces satisfy deletion-contraction identity

Deletion-Contraction

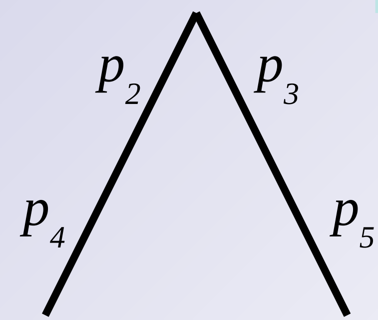
Known: A lattice





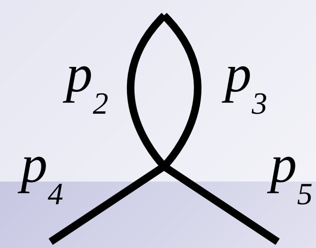
Delete p_1 bond by setting p_1=0:

$$S(p_2 p_4, p_3 p_5)$$



Contract p_1 bond by setting p_1=1:

$$H(1-[1-p_2][1-p_3], p_4, p_5)$$



Deletion-Contraction

Deletion-contraction formula for the A lattice:

$$A(p_1, p_2, p_3, p_4, p_5) =$$

$$p_1 H(1 - [1 - p_2][1 - p_3], p_4, p_5) + (1 - p_1) S(p_2 p_4, p_3 p_5)$$

Expand and set all probabilities equal

$$A(p, p, p, p, p) = 1 - 2p^2 - 3p^3 + 4p^4 - p^5 = 0$$

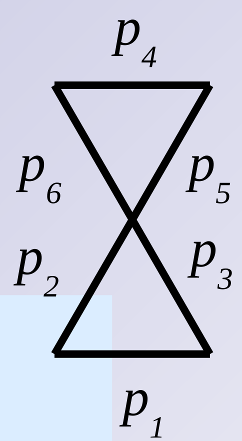
$$p_c \approx 0.625457$$

Consistent with using criticality condition directly

Deletion-Contraction

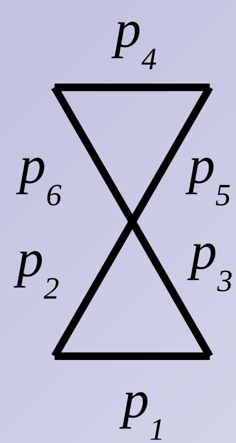
- Definition of critical polynomial can be extended to any periodic lattice using D-C
- Example: the unit cell of the kagome lattice is contained between four vertices and thus it is not exactly solvable

Unknown: kagome lattice

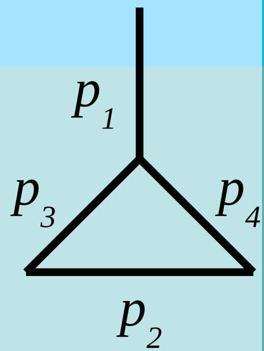


Deletion-Contraction

Unknown: kagome lattice



Known: B lattice

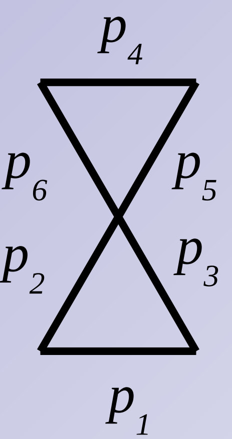


Delete and contract on p_4 bond:

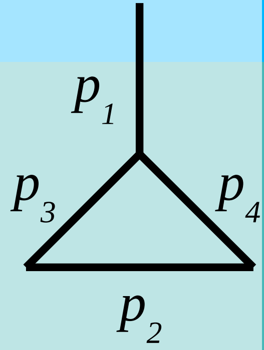
$$K(p_1,p_2,p_3,p_4,p_5,p_6) = \; p_4 B(1-[1-p_5][1-p_6],p_1,p_2,p_3)$$
$$+(1-p_4)A(p_1,p_2,p_3,p_5,p_6) = 0$$

Deletion-Contraction

Unknown: kagome lattice



Known: B lattice



Delete and contract on p_4 bond:

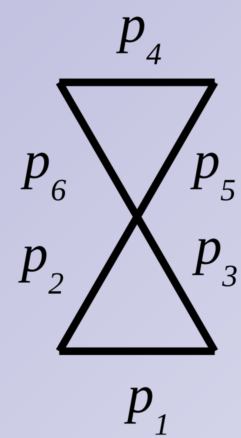
$$K(p_1,p_2,p_3,p_4,p_5,p_6) = p_4 B(1-[1-p_5][1-p_6],p_1,p_2,p_3) + (1-p_4)A(p_1,p_2,p_3,p_5,p_6) = 0$$

$$K(p,p,p,p,p,p) = 1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6 = 0$$

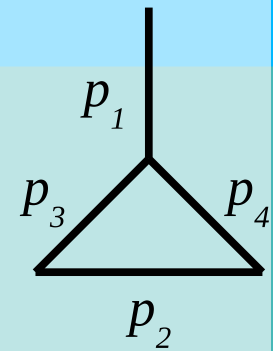
$$p_c \approx 0.52442971$$

Deletion-Contraction

Unknown: kagome lattice



Known: B lattice



Delete and contract on p_4 bond:

$$K(p_1,p_2,p_3,p_4,p_5,p_6) = \ p_4 B(1-[1-p_5][1-p_6],p_1,p_2,p_3)$$
$$+(1-p_4)A(p_1,p_2,p_3,p_5,p_6) = 0$$

$$K(p,p,p,p,p,p) = 1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6 = 0$$

(CS and R.M. Ziff 2006) $p_c \approx 0.52442971$

Not exact, but close!

Transfer Matrix: $p_c = 0.52440499(2)$

(Feng, Deng and Blöte, 2008)

Critical Polynomials: Properties

- Can be defined on any periodic lattice

(CS and R.M. Ziff 2008,2010)

Critical Polynomials: Properties

- Can be defined on any periodic lattice
  (CS and R.M. Ziff 2008,2010)
- Polynomial depends on the subgraph used to calculate it, called the basis
- Simplest case: basis is a unit cell

Critical Polynomials: Properties

- Can be defined on any periodic lattice

(CS and R.M. Ziff 2008,2010)

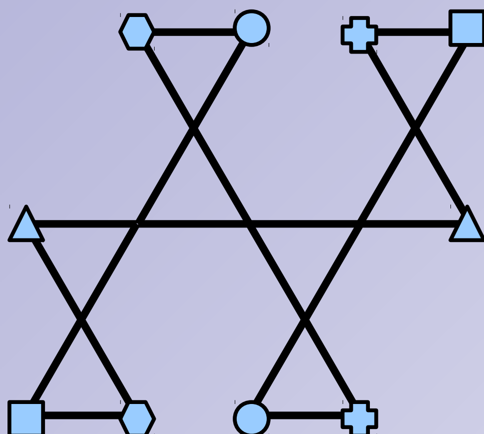
- Polynomial depends on the subgraph used to calculate it, called the basis
- Simplest case: basis is a unit cell
- By appropriately choosing larger bases, we can get better approximations

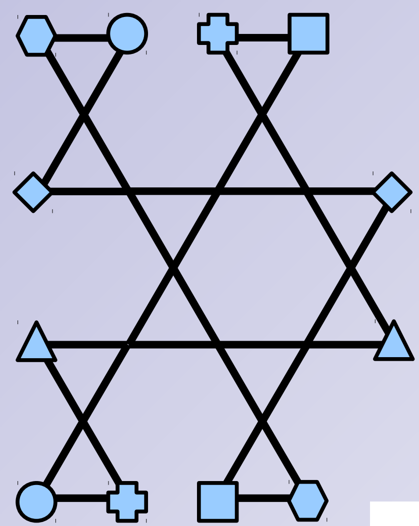
Critical Polynomials: Properties

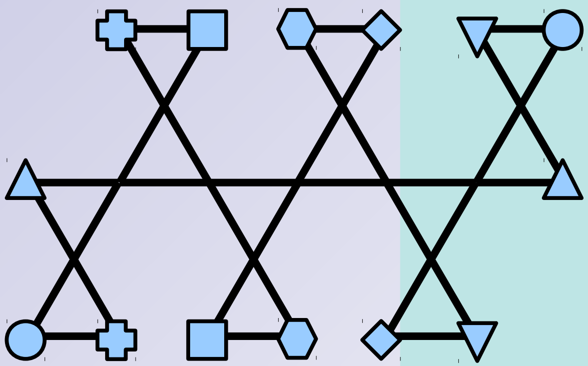
- Can be defined on any periodic lattice (CS and R.M. Ziff 2008,2010)
- Polynomial depends on the subgraph used to calculate it, called the basis
- Simplest case: basis is a unit cell
- By appropriately choosing larger bases, we can get better approximations
- Can be computed by a recursive computer algorithm

Critical Polynomials: Properties

Kagome lattice


2x2


3x2


2x3

base	polynomial	error
1x1	0.52442971...	2.5×10^{-5}
2x2	0.52440672...	1.7×10^{-6}
3x2	0.52440607...	1.1×10^{-6}
2x3	0.52440572...	6.9×10^{-7}

$$\begin{aligned}
&1 - 6p^4 - 24p^5 - 14p^6 + 36p^7 + 39p^8 - 100p^9 \\
&- 462p^{10} + 780p^{11} + 4583p^{12} + 4812p^{13} \\
&- 9276p^{14} - 71600p^{15} - 85626p^{16} \\
&+ 312336p^{17} + 1091146p^{18} - 509340p^{19} \\
&- 9675936p^{20} + 5297340p^{21} + 66607704p^{22} \\
&- 151097304p^{23} - 5319734p^{24} + 610494828p^{25} \\
&- 1461237180p^{26} + 2022998000p^{27} \\
&- 1949295060p^{28} + 1387593528p^{29} \\
&- 745850356p^{30} + 303533928p^{31} \\
&- 92388675p^{32} + 20427736p^{33} - 3103578p^{34} \\
&+ 290052p^{35} - 12579p^{36} = 0,
\end{aligned}$$

Critical Polynomials: Properties

- Polynomial is unique for a given basis (i.e., it does not depend on bond order in DC algorithm) (CS 2012)

Critical Polynomials: Properties

- Polynomial is unique for a given basis (i.e., it does not depend on bond order in DC algorithm) (CS 2012)
- Central Conjecture: An m x n basis will approach the exact solution as m and n go to infinity

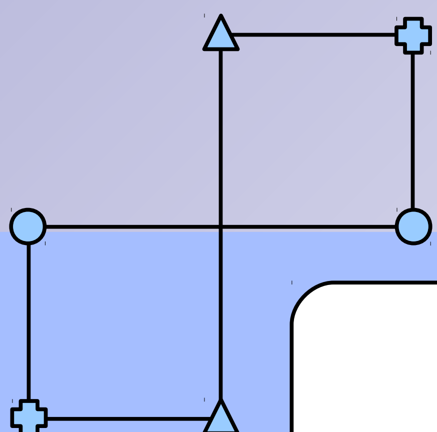
Critical Polynomials: Properties

- Polynomial is unique for a given basis (i.e., it does not depend on bond order in DC algorithm) (CS 2012)
- Central Conjecture: An m x n basis will approach the exact solution as m and n go to infinity
- If a threshold is exactly solved, a basis of any number of unit cells will give the exact answer

Critical Polynomials: Properties

- If a threshold is exactly solved, a basis of any number of unit cells will give the exact answer

Square lattice



exact:

$$1 - 2p = 0$$

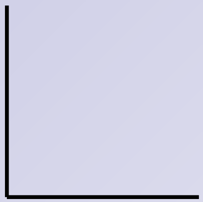
$$1 - 4p^2 - 10p^4 + 32p^5 - 28p^6 + 8p^7$$

Critical Polynomials: Properties

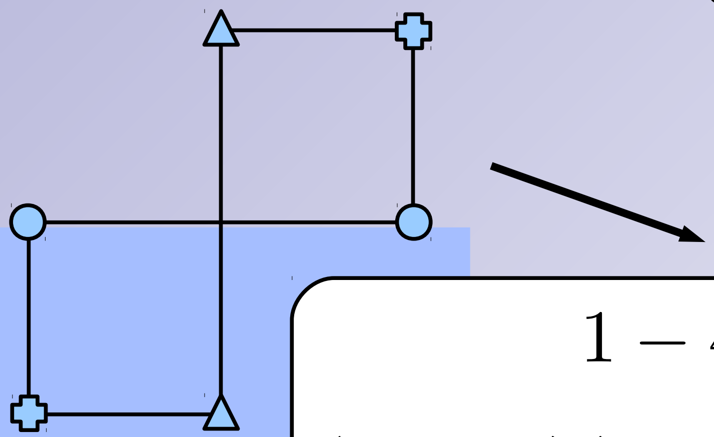
- If a threshold is exactly solved, a basis of any number of unit cells will give the exact answer

Square lattice

exact:



$$1 - 2p = 0$$



$$1 - 4p^2 - 10p^4 + 32p^5 - 28p^6 + 8p^7 =$$
$$(1-2p)(1+2p-2p^2)(1+2p^2-4p^3+2p^4) = 0$$

Critical Polynomials: Properties

- Polynomial is unique for a given basis (i.e., it does not depend on bond order in DC algorithm) (CS 2012)
- Central Conjecture: An m x n basis will approach the exact solution as m and n go to infinity
- If a threshold is exactly solved, a basis of any number of unit cells will give the exact answer

Critical Polynomials: Properties

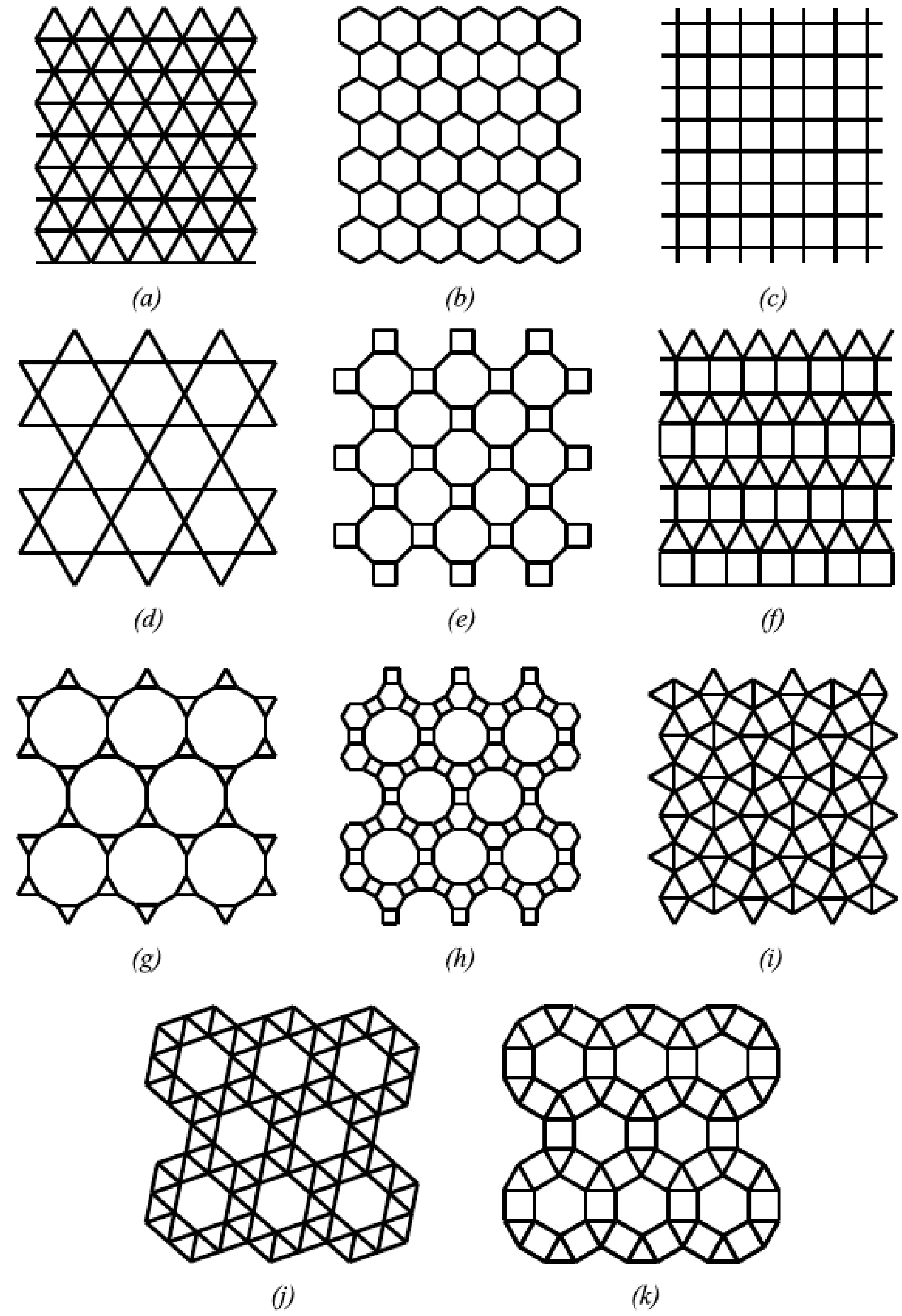
- Polynomial is unique for a given basis (i.e., it does not depend on bond order in DC algorithm) (CS 2012)
- Central Conjecture: An m x n basis will approach the exact solution as m and n go to infinity
- If a threshold is exactly solved, a basis of any number of unit cells will give the exact answer
- If the unit cell polynomial does not give the exact answer, then **no finite basis will either**

Archimedean Lattices

11 lattices for which all vertices are equivalent

Only 3 of these have exactly known bond thresholds

Grunbaum-Shepard naming convention; e) $(4,8^2)$, f) $(3^3,4^2)$, etc.



Lattices	Polynomial	Numerical	Difference
kagome	0.52440572...	0.52440499(2)	6.9×10^{-7}
$(4,8^2)$	0.67680215...	0.67680232(63)	--
$(3^3,4^2)$	0.41964531...	0.41964191(43)	3.6×10^{-7}
$(3,12^2)$	0.74042099...	0.74042077(2)	2.2×10^{-7}
$(4,6,12)$	0.69375829...	0.69373383(72)	1.6×10^{-5}
$(3^2,4,3,4)$	0.41412438...	0.41413743(46)	1.3×10^{-5}
$(3^4,6)$	0.43435240...	0.4343282(2)	2.4×10^{-5}
$(3,4,6,4)$	0.52483166...	0.52483258(53)	9.1×10^{-7}

Numerics: kagome: Feng, Deng and Blöte PRE 78 031136 2008,
(3,12^2): Ding et. al PRE 81 061111 (2010),
(3^4,6): R. M. Ziff,
the rest: Parviainen JPA 40 (2007) 9253

Polynomials: CS, J Stat Mech 2012

Part 2

(work with Jesper Jacobsen)

Critical Polynomials

- Critical polynomial easily generalized to the Potts model
- Deletion-contraction in random cluster representation (Sokal 2005)

$$P_B(q,\{v\}) = v_e P_{B/e}(q,\{v\}_{\neq e}) + P_{B\setminus e}(q,\{v\}_{\neq e})$$

- Computer implementation requires minor modifications (J.L. Jacobsen and CS, 2012)
- Deletion-contraction still limited to ~ 36 edges

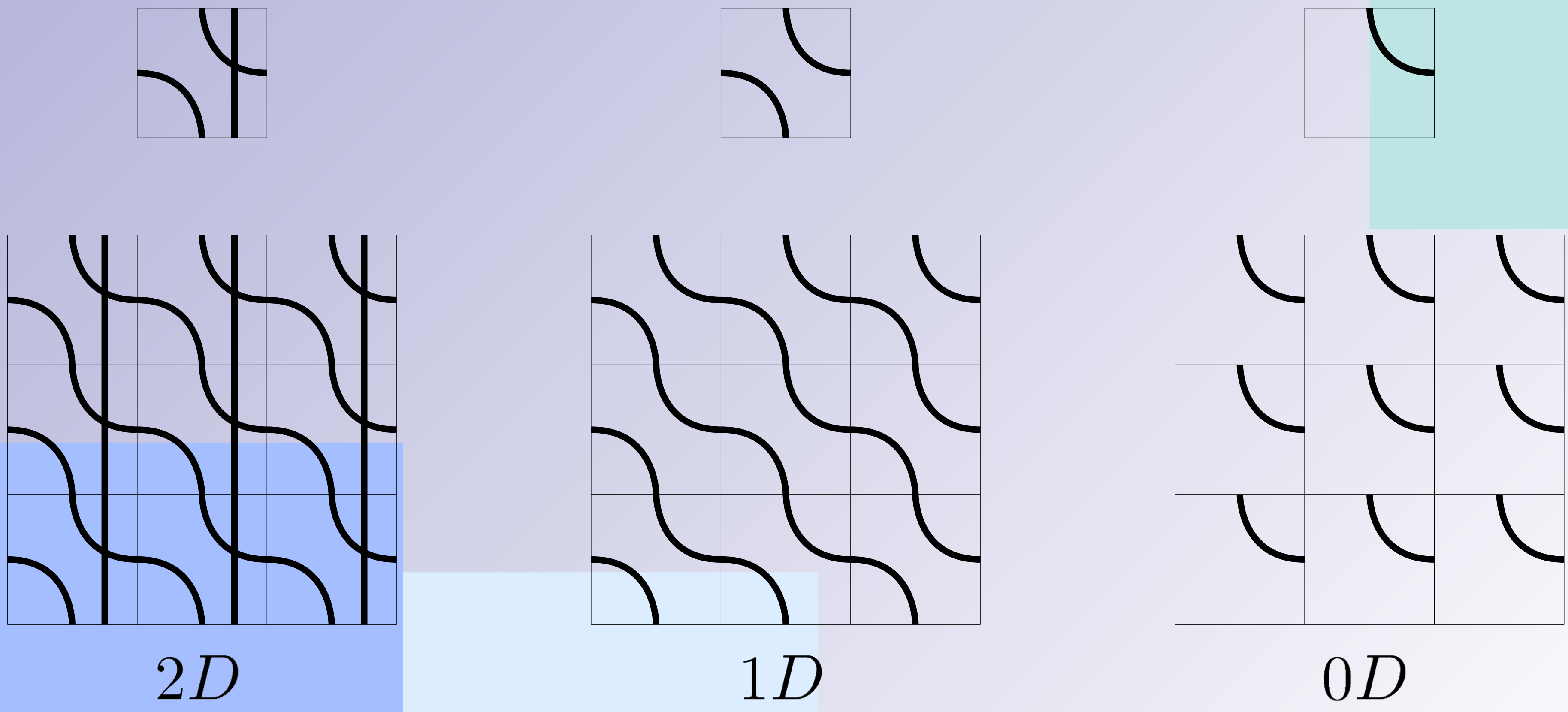
Polynomial Redefinition

- Random cluster representation has another advantage
- Gives a clear picture of the actual events underlying the polynomial
- By inspection of a few cases, we can easily infer an alternate definition of the critical polynomial:

$$P(2D) = P(0D)$$

P(2D)=P(0D)

- Three global events on a basis



P(2D)=P(0D)

- Equivalent with the exact criticality condition for solved lattices

$P(2D)=P(0D)$

- Equivalent with the exact criticality condition for solved lattices
- We can easily prove that the new definition is equivalent with the old

P(2D)=P(0D)

- Equivalent with the exact criticality condition for solved lattices
- We can easily prove that the new definition is equivalent with the old
- Both satisfy deletion-contraction and both have the same solved cases

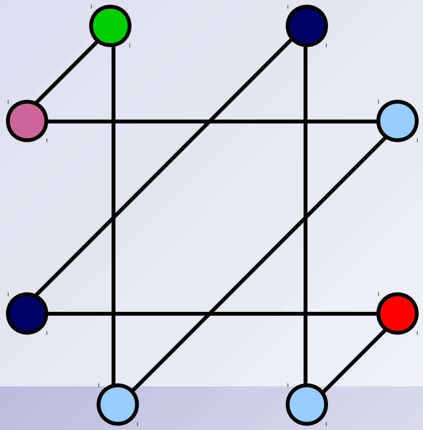
P(2D)=P(0D)

- Equivalent with the exact criticality condition for solved lattices
- We can easily prove that the new definition is equivalent with the old
- Both satisfy deletion-contraction and both have the same solved cases
- Potts generalization: $P(2D) = qP(0D)$

P(2D)=q P(0D)

- New definition allows the use of the transfer matrix
- The strategy is to compute the weight of every boundary connectivity state, then pick out the 2D and 0D states and set their weights equal (and multiply by q)

boundary connectivity:
planar partitions



Transfer Matrix

- Build complete basis edge by edge
- Compute the weight (restricted partition function) for each planar partition
- Put weights into a vector
- Addition of an edge corresponds to the action of a sparse matrix on the weight vector
- Transfer matrix gives weights of completed basis

$$\mathbf{Z_f} = T\mathbf{Z_i}$$

Transfer Matrix

- When we have vector of weights, we pick out those corresponding to 0D and 2D and set their weights equal

Transfer Matrix

- When we have vector of weights, we pick out those corresponding to 0D and 2D and set their weights equal
- Far more efficient than deletion-contraction

Transfer Matrix

- When we have vector of weights, we pick out those corresponding to 0D and 2D and set their weights equal
- Far more efficient than deletion-contraction
- With “naïve” method we can compute polynomials on 4x4 bases

(CS and J. L. Jacobsen J Phys A 45 (2012) 494004)
(J.L. Jacobsen and CS, JPA 46 (2013) 075001)

Transfer Matrix

- When we have vector of weights, we pick out those corresponding to 0D and 2D and set their weights equal
- Far more efficient than deletion-contraction
- With “naïve” method we can compute polynomials on 4x4 bases

  (CS and J. L. Jacobsen J Phys A 45 (2012) 494004)
  (J.L. Jacobsen and CS, JPA 46 (2013) 075001)
- Improved method reduces states and allows much larger calculations

  (J.L. Jacobsen, J Phys A 47 (2014) 135001)

Results

Kagome Lattice

n	threshold
1	0.524429717521274793546880
2	0.524406723188231819143234
3	0.524405172713769972706130
4	0.524405027427414720699076
5	0.524405005980616347838693
6	0.524405001306581048813495

(CS and J. L. Jacobsen 2012)

(J. L. Jacobsen 2014)

$$p_c(n) = p_c(\infty) + An^{-w}$$
$$w \approx 6.35$$

Results

Kagome Lattice

n	threshold
1	0.52442971752127479354688 0
2	0.52440672318823181914323 4
3	0.52440517271376997270613 0
4	0.52440502742741472069907 6
5	0.52440500598061634783869 3
6	0.52440500130658104881349 5

(CS and J. L. Jacobsen 2012)

(J. L. Jacobsen 2014)

Bulirsch-Stoer extrapolation: 0.52440499919(4)

Conventional numerical result: 0.52440499(2)

Results

Kagome Lattice

n	threshold
1	0.52442971752127479354688O
2	0.524406723188231819143234
3	0.524405172713769972706130
4	0.524405027427414720699076
5	0.524405005980616347838693
6	0.524405001306581048813495
7	0.524404999973208900495364

(CS and J. L. Jacobsen 2012)

(J. L. Jacobsen 2014)

Bulirsch-Stoer extrapolation: 0.524404999174(7)

Conventional numerical result: 0.52440499(2)

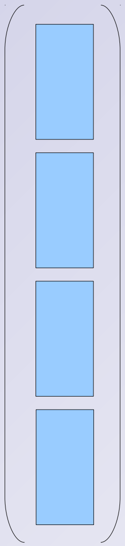
Parallel Implementation

- Transfer matrix algorithm can be made to run in parallel for a large-scale computation

Parallel Implementation

- Transfer matrix algorithm can be made to run in parallel for a large-scale computation
- Strategy is to distribute weight vector among processors

z =



Parallel Implementation

- Transfer matrix algorithm can be made to run in parallel for a large-scale computation
- Strategy is to distribute weight vector among processors
- Allows us to compute thresholds for bases of size 8x8

Results

Kagome Lattice

n	threshold
1	0.52442971752127479354688 0
2	0.52440672318823181914323 4
3	0.52440517271376997270613 0
4	0.52440502742741472069907 6
5	0.52440500598061634783869 3
6	0.52440500130658104881349 5
7	0.52440499997320890049536 4
8	0.52440499951414108517718 2

(CS and J. L. Jacobsen 2012)

(J. L. Jacobsen 2014)

(Jacobsen and CS 2014)

Bulirsch-Stoer extrapolation: 0.524404999170(2)

Conventional numerical result: 0.52440499(2)

Results

Kagome Lattice

$q = 3$

Polynomial: $v_c = 1.8764595734(3)$ (Jacobsen 2014)

Conventional: $v_c = 1.876458(3)$ (Ding, Fu, Guo, Wu 2010)

$q = 4$

Polynomial: $v_c = 2.1562545798(8)$ (Jacobsen 2014)

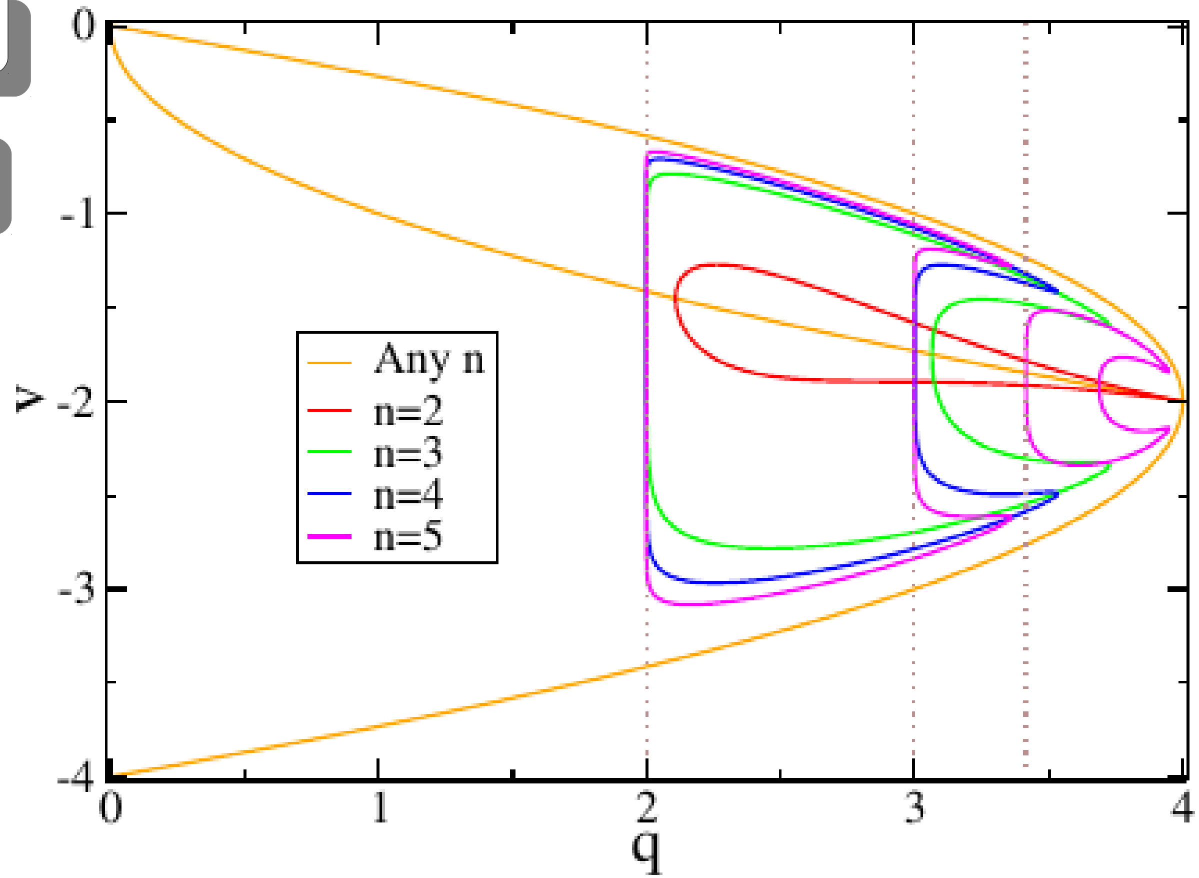
Conventional: $v_c = 2.15620(5)$ (Ding, Fu, Guo, Wu 2010)

Phase Diagrams

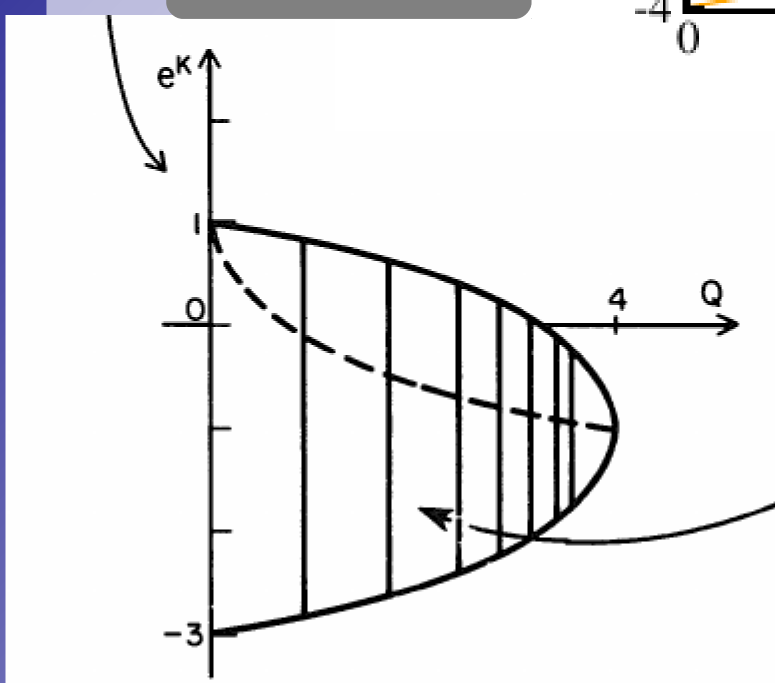
- Method allows us to find phase diagrams in the (q,v) plane
- Maximum basis size of 6x6 using supercomputer
- Phase diagrams reveal many new and mysterious features in the antiferromagnetic regime

Square Lattice

Jacobsen 2014:



Saleur 1991:



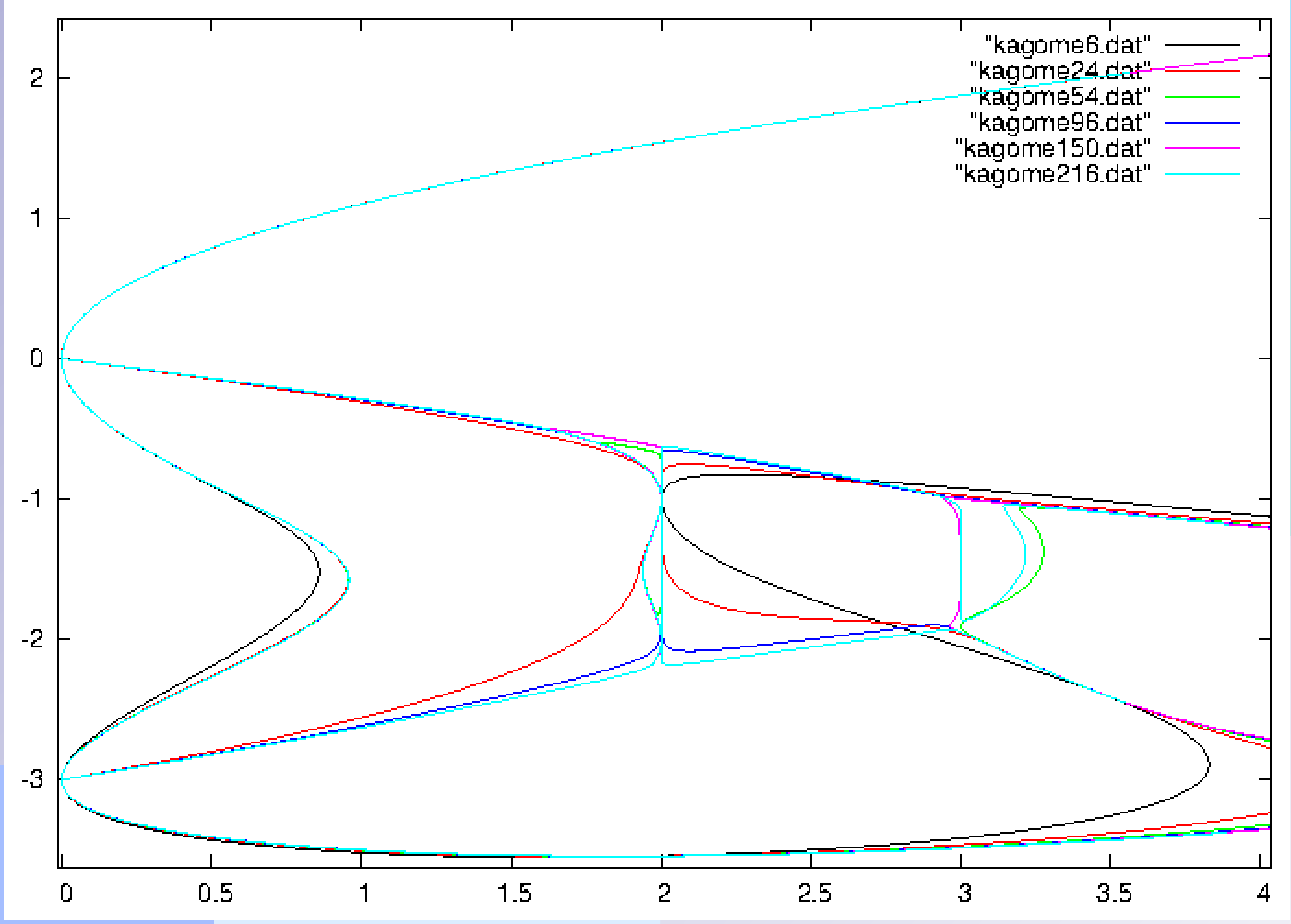
Clearly visible Berker-Kadanoff phase

Vertical rays occur at Beraha numbers:

$$B_k = 4\cos^2 \pi/k$$

Polynomial: only B_{2k}

Kagome Lattice



Questions

- Why does it work? (Universality)
- Why is the critical polynomial better than other polynomials?
- What is the exponent w?
- What is going on the phase diagrams?
- Can it be extended to 3D?
- What other models can be handled with this approach?

Conclusion

- We introduced the critical polynomial
- The zeros of the critical polynomial make accurate predictions of the critical points for the Potts model
- Early days for the method: we hope many discoveries await!