

Upper Bounds on Two-Dimensional Constraint Capacities via Corner Transfer Matrices¹

Yao-ban Chan and Andrew Rechnitzer

Abstract

We study the capacities of a family of two-dimensional constraints, containing the *hard squares*, *non-attacking kings* and *read-write isolated memory* models. Using an assortment of techniques from combinatorics, statistical mechanics and linear algebra, we prove upper bounds on the capacities of these models. Our starting point is Calkin and Wilf’s transfer matrix eigenvalue upper bound. We then use the Collatz-Wielandt formula from linear algebra, together with an approximate eigenvector, to compute an upper bound on the Calkin-Wilf bound. To obtain that approximate eigenvector, we use an ansatz from Baxter’s corner transfer matrix formalism, optimised with Nishino and Okunishi’s corner transfer matrix renormalisation group method. The combination of these methods results in an algorithm for computing upper bounds on the capacities which requires very little memory. Indeed, it is the first approach to this problem whose memory requirements do not grow exponentially. Furthermore, it is very fast and extremely parallelisable, and so allows us to make dramatic improvements to the previous best known upper bounds. In all cases we reduce the gap between the best upper and lower bounds by 4–6 orders of magnitude.

Index Terms

Channel capacity, corner transfer matrices, multi-dimensional constraints.

I. INTRODUCTION

IN THIS PAPER, we study the problem of two-dimensional constraint capacity. A constraint is a rule which induces a model of magnetic spins, each of which can take a small number of spin values (typically just two, denoted by \bullet and \circ representing “up” and “down” magnetic spins). These spins are arranged in an ordered fashion on a two-dimensional plane, typically at the vertices of a regular lattice. For the models we analyse in this paper, we take this lattice to be the square lattice \mathbb{Z}^2 (though the methods we use will generalise to other regular 2d lattices). A constraint then limits the allowed local arrangements of spins on the lattice.

This problem arises in models of information storage on a magnetic hard drive. Due to physical limitations of storage hardware, not all spin configurations may be viable, resulting in a constraint. For example, a one-dimensional spin model may be used as a simple model of 0-1 bits stored on a magnetic tape. As the tape is read from left to right, a \circ spin indicates that the current bit is equal to the previously read bit. A \bullet spin indicates that the current bit is the opposite of the previous bit. To avoid potential *intersymbol interference*, field flips are forbidden from occurring in close succession, *i.e.*, the local configuration $\bullet\bullet$ is forbidden [1]. This constraint results in the *hard squares* model, which in higher dimensions is also used in statistical mechanics as a model of interacting gas particles.

The *capacity* of the model, or the constraint, is defined to be the number of independent bits of information encoded per spin. For example, in a free model of N spins, each spin corresponds directly to one bit, and so its capacity is 1. The capacity of a constraint is directly related to the number of different configurations possible in the corresponding model. Accordingly, we define the *partition function* (a name which arises from statistical mechanics), which is simply the total number of possible configurations:

$$Z_N = \# \text{ of legal configurations of } N \text{ spins.} \quad (1)$$

Manuscript submitted August 5, 2015. An extended abstract of this paper appeared in the proceedings of the *Formal Power Series and Algebraic Combinatorics 2015* conference.

Y. Chan is at the School of Mathematics and Physics, The University of Queensland; yaoban.chan@uq.edu.au

A. Rechnitzer is at the Department of Mathematics, University of British Columbia; andrewr@math.ubc.ca

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

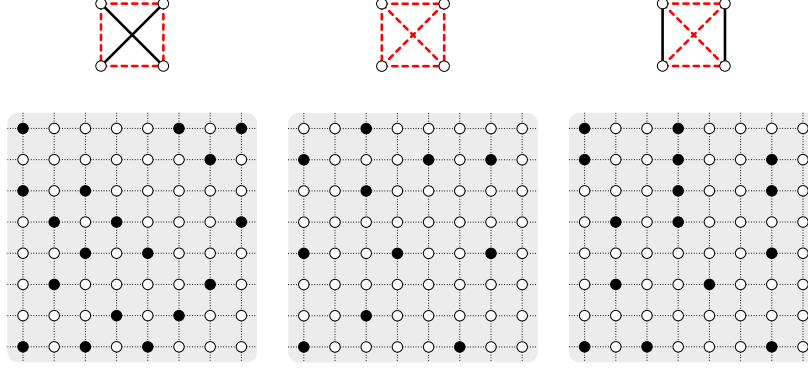


Fig. 1. The models that we study in this paper. The above diagrams show the restrictions on adjacent \bullet spins — dashed red lines are disallowed and solid black lines are allowed. The lower diagrams show example configurations. From left to right: hard squares, non-attacking kings (NAK) and read-write isolated memory (RWIM).

The partition function typically grows exponentially with respect to the number of vertices, and so a more appropriate measure for the number of possible configurations is the *partition function per site* or *growth rate*

$$\kappa = \lim_{N \rightarrow \infty} Z_N^{1/N}. \quad (2)$$

The capacity is directly related to κ by

$$\text{capacity} = \log_2 \kappa. \quad (3)$$

As the problems of finding the growth rate and capacity are virtually identical, in this paper we will restrict our terminology to the former (with the understanding that all results translate immediately to the latter).

This problem is also studied in symbolic dynamics and dynamical systems, with different terminology. Here, the capacity is also known as the (topological) *entropy* of the model, while the spins are *letters* (of an alphabet). Constraints are called *sofic shifts* (or systems), with a slight distinction: in constrained coding, the focus is on configurations of finite systems of arbitrary size, while in symbolic dynamics (and this paper), the focus is on configurations on the entire lattice. The models that we study here are sub-classes of so-called ‘shifts of finite type’ or ‘finite memory systems’.

Growth rates for one-dimensional systems can be readily computed as the dominant eigenvalue of a finite transfer matrix¹ (see [1] for example). The partition function of the hard squares model on a line of length N (containing $N + 1$ spins) can be written as

$$Z_N = \left\langle \mathbf{1} \left| \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^N \right| \mathbf{1} \right\rangle = F_{N+2} \quad (4)$$

where F_N is the N^{th} Fibonacci number². The growth rate is $\kappa = \frac{1+\sqrt{5}}{2}$, being the dominant eigenvalue of the relevant matrix, and the capacity is $\log_2 \kappa = 0.6942419\dots$. Hence each spin can store approximately 0.69 bits of information.

The corresponding problem for two-dimensional systems is typically very hard. Despite considerable effort, exact growth rates are only known for a very small number of systems³. Indeed, it has been shown by Berger [7] that there exist systems for which it is undecidable if there exist any valid configurations at all. Because of this, computing growth rates has become something of a numerical challenge. In this paper, we approach this problem by finding provable upper bounds on the growth rates of several systems. This is a sequel to our previous paper [8], where we proved very precise lower bounds for these same models.

¹This implies that κ for a 1d system is a Perron number. Lind [2] proved that any Perron number is also the growth rate of some 1d system.

²We have used the convention that $F_0 = F_1 = 1$.

³For example, the fully-packed dimer model [3], [4], 3-colourings of the square grid [5], and the ‘odd’ model [6].

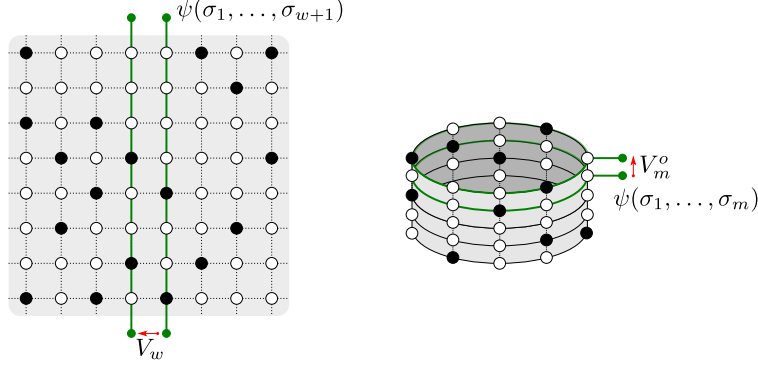


Fig. 2. The transfer matrix V_w extends a system of N columns to a system of $N + 1$ columns by using the state of the N th column $(\sigma_1, \dots, \sigma_{w+1})$. The transfer matrix V_m^o constructs the same system on a cylinder with circumference of length m .

One of the most-studied models in this area is the aforementioned hard squares model [9], [6], [10]. This model is well-known in statistical mechanics⁴, not just for the analysis of its capacity, but also for its macroscopic behaviour as the relative weighting of \bullet and \circ spins changes. We also consider two related models which forbid certain local configurations of \bullet spins (see Fig. 1). The *non-attacking kings* model forbids horizontal, vertical and diagonal adjacencies of \bullet spins — they can be considered as kings on a chessboard which cannot be placed in such a way that one attacks another. The *read-write isolated memory* model [12], [13] forbids two horizontally or diagonally adjacent \bullet spins (vertical adjacency is allowed). If we consider the two-dimensional array of spins as a horizontal line of hard-square spins evolving in time (each row separated by 1 time unit), then the RWIM constraint can be viewed as additionally forbidding the alteration of two adjacent spins in one time step.

It is not at all clear that tractable closed-form expressions for κ exist for these models, and none are known⁵. There is a significant body of work on finding rigorous bounds on κ , most of which are based on the analysis of transfer matrices (see Fig. 2). Rather than analysing the (infinite) transfer matrix for an infinite two-dimensional system, we consider the (finite) transfer matrix for a system on an infinite strip of finite width, say w cells (and $w + 1$ spins). Let V_w be the column transfer matrix associated with this strip, and $\Lambda(w)$ its dominant eigenvalue. Then

$$\kappa = \lim_{w \rightarrow \infty} \Lambda(w)^{1/w}. \quad (5)$$

Lower bounds for κ can then be computed using a clever formula based on Rayleigh quotients due to Engel [15], and Calkin and Wilf [9] (we refer the reader to the latter paper for its proof):

$$\kappa^p \geq \frac{\Lambda(p + 2q)}{\Lambda(2q)}, \quad \text{with } p, q > 0. \quad (6)$$

Calkin and Wilf also proved the following upper bound on κ , which we prove again below:

$$\kappa^{2p} \leq \Lambda^o(2p), \quad \text{with } p > 0 \quad (7)$$

where $\Lambda^o(m)$ is the dominant eigenvalue of a related matrix, V_m^o , the transfer matrix for the same system but with cylindrical boundary conditions, *i.e.*, a cylinder with circumference of length m (see Fig. 2). Note that here, the circumference of the cylinder has only m vertices, not $m + 1$.

Almost all works which compute bounds on the growth rates of the hard squares and related models use the above two inequalities (two recent exceptions being [16] and previous work by the authors [8], both of which use methods from statistical physics). However, the practical application of these inequalities is quickly hampered by the exponential growth of the size of transfer matrix with respect to strip width⁶. There are creative methods to

⁴It is a close relative of the famous hard hexagons model which was solved exactly by Baxter [11].

⁵The authors had hoped that precise numerical estimates of κ might yield such expressions via tools such as the Inverse Symbolic Calculator [14]. Unfortunately, no such expressions were found [8].

⁶For the three models we consider, the matrix dimensions grow as the Fibonacci numbers.

avoid storing the full matrix, perhaps the most successful being a matrix compression method due to Lundow and Markström [17]. Together with Friedland [10], they used this method to compute $\Lambda(28)$ and $\Lambda^o(36)$, which in turn allowed them to exactly determine the first 15 digits of κ for the hard squares model:

$$\kappa_{\text{hard squares}} = 1.503\,048\,082\,475\,33\dots \quad (8)$$

While this compression method greatly decreases the time and memory needed, the requirements still grow exponentially with strip width. A similar but distinct approach appears in the very recent work of Wang, Yong and Golin [18]. They use a recursive description of the transfer matrix for the RWIM model rather than its symmetries to reduce matrix sizes significantly. They determined the first 5 digits κ for the RWIM model:

$$\kappa_{\text{RWIM}} = 1.4489\dots \quad (9)$$

However the time and memory required by this method also grow exponentially with system size.

In this work, we compute upper bounds on the growth rates of the models. We start with equation (7), but rather than computing $\Lambda^o(m)$ exactly, we compute an upper bound on it using an approximation of the dominant eigenvector of V_m^o and the Collatz-Wielandt formula [19], [20] (which we include as Lemma 2). To form the approximate eigenvector, we use corner transfer matrix formalism. This is a very powerful approach developed in statistical mechanics by Baxter [21], [22], [23] as a way to estimate the partition function of various lattice models, both numerically and via series expansions [23], [24], [25]. CTMs have indeed been used before to estimate (rather than bound) the growth rate of the hard squares model and its relatives on other 2d lattices [26].

The CTM formalism allows us to express each component of the approximate eigenvector as the trace of a product of auxiliary matrices. This means that we are no longer required to store either the transfer matrix or vector explicitly; we can compute the vector component by component as the Collatz-Wielandt formula requires. Thus the memory requirement is no longer exponential in the size of the system. To optimise the choice of vector (and the associated auxiliary matrices), we use an extension of CTMs known as the corner transfer matrix renormalisation group (CTMRG) method, developed by Nishino and Okunishi [27], [28].

II. UPPER BOUNDS ON UPPER BOUNDS

We start by reproving Calkin and Wilf's upper bound on the growth rate κ .

Lemma 1 ([9]). *Let V_m^o be the column transfer matrix for a system of width m with cylindrical boundary conditions, and let $\Lambda^o(m)$ be its dominant eigenvalue. Then for even m ,*

$$\kappa^m \leq \Lambda^o(m). \quad (10)$$

Proof: We begin by writing κ in terms of the largest eigenvalue of V_w , the column transfer matrix with free boundary conditions:

$$\kappa = \lim_{w \rightarrow \infty} \Lambda(w)^{1/w}. \quad (11)$$

Note that if we now write the dominant eigenvalue $\Lambda(w)$ as the maximum of a Rayleigh quotient, this leads to lower bounds as derived in [15], [9].

For the models considered here, V_w is symmetric, and thus has real eigenvalues. Therefore for even m , all eigenvalues of $(V_w)^m$ must be positive. We then have

$$\kappa^m = \lim_{w \rightarrow \infty} (\Lambda(w)^m)^{1/w} \leq \lim_{w \rightarrow \infty} (\text{Tr}(V_w)^m)^{1/w}. \quad (12)$$

Now consider the quantity $\langle \mathbf{1} | V_w^m | \mathbf{1} \rangle$. This is the partition function for the system on a $w \times m$ rectangle (with $(w+1)(m+1)$ spins) with free boundary conditions. Taking the trace, rather than the product with $\mathbf{1}$ vectors, identifies the first and last column of spins together. Hence $\text{Tr}(V_w)^m$ is the partition function of a cylinder of circumference m and height w , as shown in Fig. 2. However, this can also be written in terms of the transfer matrix around a cylinder of circumference m :

$$\text{Tr}(V_w)^m = \langle \mathbf{1} | (V_m^o)^w | \mathbf{1} \rangle \quad (13)$$

noting that the boundary conditions at the ends of the cylinder are free. Taking the limit $w \rightarrow \infty$, we get

$$\begin{aligned} \kappa^m &\leq \lim_{w \rightarrow \infty} \langle \mathbf{1} | (V_m^o)^w | \mathbf{1} \rangle^{1/w} \\ &= \Lambda^o(m). \end{aligned} \tag{14}$$

Almost all upper bounds in the literature are derived using Lemma 1 by exact calculation of the dominant eigenvalue of V_m^o . As such they are restricted, both in time and memory, by the exponential growth of the transfer matrix size. However we do not actually need to compute $\Lambda^o(m)$ exactly — any good upper bound on this quantity will also give a good upper bound on κ . We proceed by using the Collatz-Wielandt formula [19], [20] to compute upper bounds on $\Lambda^o(m)$. ■

Lemma 2 (Collatz-Wielandt formula). *Let A be an $n \times n$ irreducible square matrix with non-negative entries. Then for any vector $\mathbf{x} > 0$, the largest eigenvalue of A (denoted λ) is real and positive and is bounded by*

$$\min_i \frac{(A\mathbf{x})_i}{x_i} \leq \lambda \leq \max_i \frac{(A\mathbf{x})_i}{x_i}. \tag{15}$$

Proof: Let \mathbf{u} be the left eigenvector of A corresponding to λ . By the Perron-Frobenius theorem, λ is real and positive and \mathbf{u} has strictly positive entries. Then for any $\mathbf{x} > 0$,

$$\langle \mathbf{u} | A | \mathbf{x} \rangle = \lambda \langle \mathbf{u} | \mathbf{x} \rangle \tag{16}$$

and so

$$\langle \mathbf{u} | A\mathbf{x} - \lambda\mathbf{x} \rangle = \sum_{i=1}^n u_i x_i \left(\frac{(A\mathbf{x})_i}{x_i} - \lambda \right) = 0. \tag{17}$$

When \mathbf{x} is a scalar multiple of \mathbf{u} , all the summands will be zero. Otherwise some terms in the sum must be non-negative and some non-positive. Therefore the maximum of the summands is non-negative and the minimum is non-positive. Since $\mathbf{u}, \mathbf{x} > 0$, the result follows. ■

We use these lemmas to find an upper bound on $\Lambda^o(m)$ for even m . In order to calculate a tight bound, we need to choose a vector which is as close as possible to the dominant eigenvector of V_m^o . We do this using Baxter's corner transfer matrix ansatz, which specifies to choose a vector ψ of the form

$$\psi_{\sigma} = \text{Tr} [F(\sigma_1, \sigma_2) F(\sigma_2, \sigma_3) \dots F(\sigma_m, \sigma_1)] \tag{18}$$

where the index $\sigma = (\sigma_1, \dots, \sigma_m) \in \{\bullet, \circ\}^m$ runs over all legal configurations of one row of the cylinder, and $\{F(a, b)\}$ is a set of four matrices of size $n \times n$, indexed by two spin values a and b . Here n is an arbitrary positive integer which need not be related to m (the circumference of the cylinder). The F matrices are calculated from corner transfer matrix methods; the process is quite involved, and we give a brief description in the next section and direct the reader to [8] for more details. We can interpret this ansatz pictorially by thinking of the F matrices as half-row transfer matrices which build up the state ψ_{σ} one row at a time — see Fig. 3.

In order to use (15), we must also be able to compute $(V_m^o \psi)_{\sigma}$, and to do this we define the *face weight* ω . The weight of a face (a single cell of the lattice) is 1 when the spins around it form a legal configuration and 0 otherwise. The element of V_m^o which maps a column in state τ to a column in state σ is then given by the product

$$(V_m^o)_{\sigma, \tau} = \prod_{i=1}^m \omega \begin{pmatrix} \sigma_{i+1} & \tau_{i+1} \\ \sigma_i & \tau_i \end{pmatrix}, \quad \text{where} \quad \begin{aligned} \sigma_{m+1} &\equiv \sigma_1, \text{ and} \\ \tau_{m+1} &\equiv \tau_1. \end{aligned} \tag{19}$$

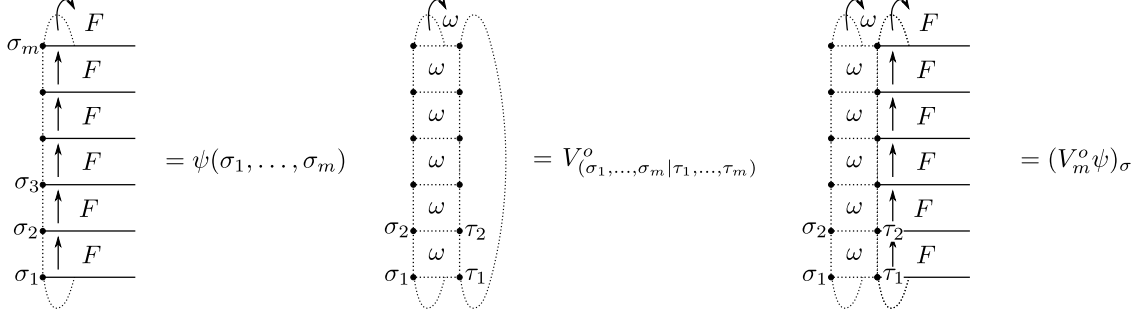


Fig. 3. Pictorial interpretation of (18), (19) and (20).

This is shown pictorially in Fig. 3. We can then write the action of V_m^o on ψ as

$$\begin{aligned}
(V_m^o \psi)_\sigma &= \sum_{\tau} (V_m^o)_{\sigma, \tau} \psi_\tau \\
&= \sum_{\tau} \left[\prod_{i=1}^m \omega \begin{pmatrix} \sigma_{i+1} & \tau_{i+1} \\ \sigma_i & \tau_i \end{pmatrix} \right] \text{Tr} [F(\tau_1, \tau_2) F(\tau_2, \tau_3) \dots F(\tau_m, \tau_1)] \\
&= \text{Tr} \sum_{\tau} \prod_{i=1}^m \omega \begin{pmatrix} \sigma_{i+1} & \tau_{i+1} \\ \sigma_i & \tau_i \end{pmatrix} F(\tau_i, \tau_{i+1}) = \text{Tr} [F_l(\sigma_1, \sigma_2) F_l(\sigma_2, \sigma_3) \dots F_l(\sigma_m, \sigma_1)] \quad (20)
\end{aligned}$$

where F_l is a $2n \times 2n$ matrix defined by the block matrix equation (shown pictorially in Fig. 5)

$$F_l(c, a) \Big|_{d, b} = \omega \begin{pmatrix} a & b \\ c & d \end{pmatrix} F(d, b). \quad (21)$$

We are now able to combine the above expression with (10) and (15) to derive

$$\begin{aligned}
\kappa^m &\leq \Lambda^o(m) \leq \max_{\sigma} \frac{(V_m^o \psi)_\sigma}{\psi_\sigma} \\
&= \max_{\sigma} \left\{ \frac{\text{Tr} [F_l(\sigma_1, \sigma_2) F_l(\sigma_2, \sigma_3) \dots F_l(\sigma_m, \sigma_1)]}{\text{Tr} [F(\sigma_1, \sigma_2) F(\sigma_2, \sigma_3) \dots F(\sigma_m, \sigma_1)]} \right\}. \quad (22)
\end{aligned}$$

This is the upper bound that we use for κ , and it is valid for any F , n , and even m , as long as we satisfy the following conditions:

- V_m^o is non-negative. In fact V_m^o is a 0-1 matrix for the considered models, so this follows immediately.
- V_m^o is irreducible. This is equivalent to showing that every set of states σ on a cut can be reached from any other. This is easy to show for our models, as there is no restriction on adjacency to \circ spins — so any σ can be adjacent to $\{\circ, \circ, \dots, \circ\}$, and thus can reach any other set of states.
- ψ_σ is positive. This does not immediately follow from (18). We must show this by explicitly computing (18) for every σ and verifying that it is positive. However, this does not result in any extra work since we already need these values to compute (22).

Putting all of this together, we find an upper bound as follows:

- 1) Select $m, n > 0$ with m even.
- 2) Calculate a set of $n \times n$ matrices $\{F(a, b)\}$ as specified in the next section.
- 3) For all possible cut states σ of m spins:
 - a) Verify $\psi_\sigma > 0$, where ψ_σ is given by (18).
 - b) Calculate $\left(\frac{(V_m^o \psi)_\sigma}{\psi_\sigma}\right)^{1/m}$.
- 4) The upper bound for κ is the maximum of all values calculated in step 3b.

This method uses very little memory since we do not need to keep the entire ψ vector in memory. Each component of ψ and $V_m^o \psi$ can be computed independently and discarded after the ratio of traces in (22) has been computed. This

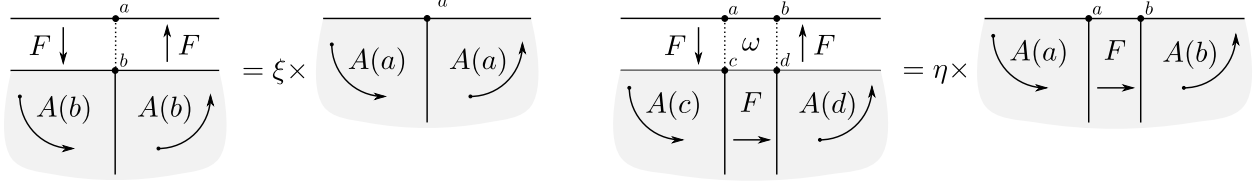


Fig. 4. Pictorial interpretation of the CTM equations.

computation requires only that we store the F, F_l matrices and the particular product corresponding the component we are computing. These matrices are tiny compared to ψ and V_m^o , so this is a very modest requirement. Furthermore, the calculation in step 3 for any particular component does not depend on any other component, so can be done in parallel (and ratios compared afterwards). This step is by far the most time-consuming part of the process, as the number of traces required grows exponentially with m . Hence parallelisation creates a huge real-time saving. Details of our implementation and some optimisations are given in Section IV below.

III. APPROXIMATE EIGENVECTORS FROM CORNER TRANSFER MATRICES

So far, we have not described how to calculate the F matrices, and this is clearly critical in order to obtain a good upper bound. (Note however that careful selection of the F matrices is not required to obtain a valid bound — any F will suffice for this purpose — merely a *good* bound.) We need to choose the matrices in such a way that the approximate eigenvector ψ given by (18) is close to the true eigenvector of V_m^o . We accomplish this through the use of corner transfer matrix formalism, which we outline briefly here; we direct the reader to a more detailed discussion in [8].

The expression (18) is the starting point of Baxter’s corner transfer matrix formalism. It can be represented pictorially by considering $F(a, b)$ in the limit $n \rightarrow \infty$ as a half-row transfer matrix; it takes a half-infinite row of spins ending with spin a , and transfers it along one row to a half-infinite row of spins ending with spin b . Using this representation, we can see that the expression for ψ represents a half-plane partition function (normalised in some way), which is unchanged under the action of V_m^o save for a constant factor — *i.e.*, it is an eigenvector of V_m^o as required. This representation is shown in Fig. 3.

These infinite-dimensional half-row transfer matrices can be shown [22], [8] to satisfy the *CTM equations*

$$\xi A^2(a) = \sum_b F(a, b) A^2(b) F(b, a) \quad (23)$$

$$\eta A(a) F(a, b) A(b) = \sum_{c, d} \omega \begin{pmatrix} a & b \\ c & d \end{pmatrix} F(a, c) A(c) F(c, d) A(d) F(d, b) \quad (24)$$

where $A(a)$ are matrices of the same size as the F matrices, and are indexed by a single spin value. The A matrices are the eponymous *corner transfer matrices*; they transfer a half-infinite row of spins ending with spin a around an angle of $\pi/2$, adding the weight of a quarter of the lattice. The CTM equations can then be interpreted as equating half-plane transfer matrices (see Fig. 4). The matrices on either side of the equality differ only by an extra row, which results in the scalar multipliers ξ and η .

In the infinite-dimensional limit, the CTM equations directly yield the solution of the model via $\kappa = \eta/\xi$. We can approximate the true solution by taking F matrices of finite size $n \times n$. The CTM equations can be solved for finite-dimensional matrices, and we take these finite sized solutions for our F matrices in our bound finding procedure.

Solving the CTM equations at finite size is quite non-trivial. For this purpose, we use the corner transfer matrix renormalisation group method developed by Nishino and Okunishi [27], [28]. In this method, we start with some trial values for $A(a)$ and $F(a, b)$ which are then “polished” into solutions. We do this by expanding A and F into the $2n \times 2n$ matrices A_l and F_l , using the block matrix equations (21) and

$$A_l(c) \Big|_{d, a} = \sum_b \omega \begin{pmatrix} a & b \\ c & d \end{pmatrix} F(d, b) A(b) F(b, a). \quad (25)$$

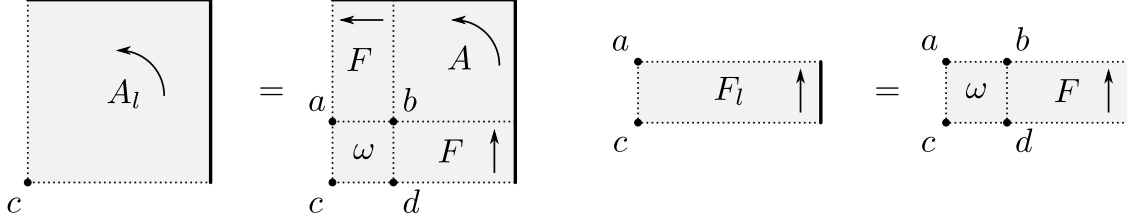


Fig. 5. Pictorial interpretation of A_l and F_l .

These can be interpreted as “larger” versions of the corner and half-row transfer matrices (see Fig. 5).

We then reduce the large matrices to generate new iterations of A and F . To do this, we diagonalise $A_l(a)$, producing orthogonal diagonalising matrices $P(a)$. We take the transformations

$$A(a) \rightarrow P^T(a)A(a)P(a) \quad F(a, b) \rightarrow P^T(a)F(a, b)P(b) \quad (26)$$

which leave the CTM equations invariant. We then truncate the matrices to the original $n \times n$ size by keeping the n largest eigenvalues of A_l and performing a consistent transformation on F_l . This has the effect of making the matrices intuitively “cover as large an area as possible”, so that they are close to the infinite-size solution of the CTM equations.

The expanding and reducing procedures are iterated, and it is observed that the matrices eventually converge to the finite-size solution of the CTM equations. The initial values for the A and F matrices can be taken to be of some small size, typically 1×1 or 2×2 . They can then be “built up” to the desired $n \times n$ size by sometimes keeping extra eigenvalues at the reduction step, resulting in larger matrices until the desired size is reached. We again direct the reader to [8] for a more detailed description of this process.

IV. RESULTS

The method was implemented in 2 distinct steps: computing the F matrices, and then computing the trace ratios for all legal states σ . Both parts were implemented in C++ using the Eigen numerical linear algebra library [29]. We used Eigen since it readily supports multiple precision computation through the MPFR library [30]. High precision is necessary in the first step because the eigenvalues of corner transfer matrices range over many orders of magnitude, and in the second step so that the trace ratios are also of high precision — though less precision is needed. The first step requires only modest computing resources, and all F matrices were generated within a few hours on a modest Linux laptop using only a few megabytes of memory.

The second step — computing the trace ratios — can be implemented quite naïvely and still give good results. However we use a number of simple and very effective optimisations based on properties of traces. The simplest implementation will compute the trace ratio of every legal σ in the set $\{\circ, \bullet\}^m$. However since traces of products are invariant under cyclic permutations, we need only consider the legal words in $\{\circ, \bullet\}^m$ modulo cyclic permutations. Since traces are invariant under transpositions and the F matrices satisfy $F(a, b) = F^T(b, a)$ (as do the F_l matrices), we can further reduce the set of words by also removing reflection symmetry. Words modulo both cyclic permutations and reflections are known in the combinatorial literature as *bracelets*.

Thus we need only consider bracelets which are legal for the system which we can generate using the methods in [31]. This reduces the number of computations by a factor of approximately $2m$. Additionally, we can gain a little more speed by using similarity transforms to diagonalise one of the F matrices. Finally, we split the set of bracelets into batches of equal size across about 300 cores on the WestGrid computing cluster. Each job (with $n = 30, m = 50$) took approximately 1 day of CPU time. Note that final upper bounds were computed at high precision and then rounded up.

Our results are given in Table I. For each of the three models, we computed the bound using F matrices of size 30×30 on cylindrical systems of circumference 50 vertices. In this table, we also indicate the previous best upper bounds. In Table II, we combine these new upper bounds with the best known lower bounds to show the digits of the growth rates which we now know rigorously. We extend the number of rigorously known digits from previous work by between 4 and 6 digits for our models.

Model	m	n	Upper bound
Hard squares	50	30	<u>1.503 048 082 475 332 264 519</u>
NAK	50	30	<u>1.342 643 951 124 602 238</u>
RWIM	50	30	<u>1.448 957 372 102</u>

TABLE I
UPPER BOUNDS ON THE GROWTH RATES OF HARD SQUARES, NAK AND RWIM. THE PREVIOUS BEST KNOWN BOUNDS [10], [6] ARE UNDERLINED.

Model	Lower bounds [8]	Exact digits
Hard squares	<u>1.503 048 082 475 332 264 322 066 329 475 553 689 385 781 038 610 305 062 028</u> 101 735 933 850 396 923 440 380 463 299 47	19
NAK	<u>1.342 643 951 124 601 297 851 730 161 875 740 395 719 438 196 938 393 943 434</u> 885 455 0	15
RWIM	<u>1.448 957 371 775 608 489 872 231 406 108 136 686 434 371</u>	9

TABLE II
LOWER BOUNDS ON THE GROWTH RATES OF HARD SQUARES, NAK AND RWIM FROM [8]; COMBINING THESE LOWER BOUNDS WITH THE NEW UPPER BOUNDS, THE UNDERLINED DIGITS ARE KNOWN RIGOROUSLY. NOTE THAT THE LOWER BOUNDS AGREE WITH THE BEST ESTIMATES [8] IN ALL BUT THEIR LAST COUPLE OF DIGITS.

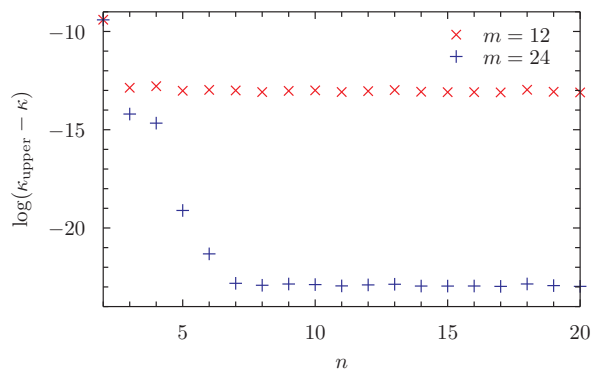
We note here that the use of floating point precision for the trace-ratio computations made our code about 50–100 times slower than if we had been able to use hardware floating point. Unfortunately most commercially available processors, including those we had access to, support 80-bit double-extended precision rather than true 128-bit or higher precision. If such high precision hardware floating point were available, then we could have increased m by about 8, which would perhaps give an extra 2 or 3 digits exactly. The previous best upper bounds [10], [6] used machine floating point.

Due to a lack of rigorous results on the theoretical convergence of the CTMRG, it is difficult to predict how good our bounds will be for given m (the circumference of the cylinder) and n (the size of the F matrices). Thus we must observe their empirical behaviour for various m and n . We first note that m is the computational bottleneck parameter, as the time required is exponential in m (while the time and memory requirements depend polynomially on n). Thus we wish to increase m to the limit of our computational power, and then select an optimal n for that value of m . We show the results in the context of the hard squares model; other models behave in much the same way.

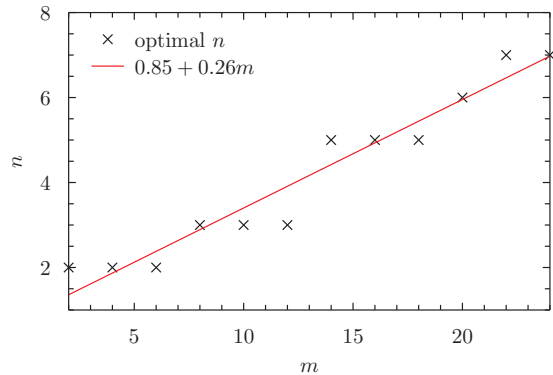
For any fixed m , the bound given by (22) decreases as n increases, but only up to a small limit, after which it stays level. Therefore, given a value of m , we can be assured of calculating the best bound for that m as long as n is sufficiently large. This is shown in Fig. 6(a). The optimal n (*i.e.*, the value of n after which the bound does not decrease further) for any given m appears to follow a linear relationship with m . In Fig. 6(b), we show this value for small m . For our final bound computation, we were able to run the method for $m = 50$. We estimate by extrapolating the fitted linear relationship that for this value of m , the optimal value of n is approximately 14. We have taken $n = 30$, and so are confident that this value is sufficient to achieve the best bound for our value of m .

Next, we look at the performance of the bound in relation to m . Because our bound is itself an upper bound on $\Lambda^o(m)^{1/m}$, we know that it will be inferior to an explicit evaluation of (10) for the same m . However, for sufficiently large n , the difference between (22) and (10) is very small. This is shown in Fig. 7, where there is almost no discernible difference between the two bounds. Thus we expect that the accuracy of our bound will behave similarly to that of (10), which is conjectured to decrease to κ exponentially with respect to m . This behaviour can also be observed in Fig. 7. This means that the algorithm, which is exponential-time in m , will also be exponential-time in the number of digits found. This is worse than the observed performance for the lower bound found in [8], although this is not unexpected — previous bound-finding methods have always managed to produce much tighter lower bounds than upper bounds.

As previously mentioned, (10) is the formula used to generate all previous best known upper bounds. The advantage of our method is that (22) is almost as accurate and much faster to compute, even though both methods are exponential in m . For example, in [10] the eigenvalue $\Lambda^o(36)$ was computed using over 200 gigabytes of memory and 40 days of cpu-time (and real-time). We were able to push to $m = 50$ using only a few megabytes



(a) Log error in the upper bounds generated by increasing n for $m = 12, 24$. Here we have used the best estimate of κ from [8].



(b) Optimal n for various m . There appears to be a linear relationship with m .

Fig. 6. Choosing an n value for a given m .

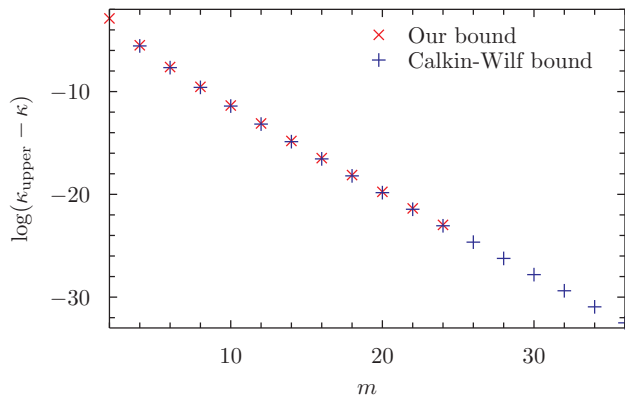


Fig. 7. Log error in the upper bounds generated by (22) (red) and (10) (blue) for various m . Here we used $n = 20$ which always produces the best possible bound for these values of m . Again we have used the best estimate of κ from [8].

of memory and about 300 days of cpu-time but only 1 day of real-time due to parallelisation. Thus we are able to achieve much better bounds.

V. CONCLUSION

In this paper, we have used transfer matrix analysis, the Collatz-Wielandt formula, and corner transfer matrix formalism to derive upper bounds on the growth rates and capacities of three lattice models motivated by an information storage problem. Our bounds are a significant improvement on all previously known upper bounds, and together with the corresponding lower bounds from [8], fully determine the growth rates to a generous number of digits.

Our method is not limited to the models we have studied; nearly any lattice model which can be written in terms of local face weights and satisfies some elementary symmetry properties can be analysed in this way. It can be applied directly to colouring models, and it appears possible to exploit symmetries in this case to make it even more efficient. On the other hand, some other models lack irreducible transfer matrices, for example the “even” model in [8]. We are extending our method to account for these difficulties.

One weakness of our method is that it appears to be exponential-time in the number of digits required, due to the need to evaluate all the ratios in (22) (modulo symmetries). It seems intuitive that much of this is wasted work, since in the limit $n \rightarrow \infty$ it can be shown that the solution of the CTM equations gives the exact eigenvector in (18). As such, all of the ratios should be very close to each other, and it should be unnecessary to calculate all 2^m of them.

We observed, for example, that the traces were always positive but we have been unable to prove this. Similarly, we observed that the maximum or minimum trace ratio was frequently given by $\sigma = \circ^m$ or $\sigma = (\circ\bullet)^{m/2}$, but, again, we have been unable to prove this. Such results would allow us to compute bounds much more efficiently and we are currently working to prove them.

ACKNOWLEDGMENT

The authors would like to thank Ian Enting and Brian Marcus for many helpful and interesting discussions, and WestGrid for providing access to their computer cluster.

REFERENCES

- [1] D. Lind and B. Marcus, *An introduction to symbolic dynamics and coding*. Cambridge University Press, 1995.
- [2] D. Lind, “The entropies of topological Markov shifts and a related class of algebraic integers,” *Ergod. Theor. Dyn. Syst.*, vol. 4, pp. 283–300, 6 1984. [Online]. Available: http://journals.cambridge.org/article_S0143385700002443
- [3] P. W. Kasteleyn, “The statistics of dimers on a lattice: I. the number of dimer arrangements on a quadratic lattice,” *Physica*, vol. 27, no. 12, pp. 1209–1225, 1961.
- [4] H. Temperley and M. E. Fisher, “Dimer problem in statistical mechanics — an exact result,” *Philos. Mag.*, vol. 6, no. 68, pp. 1061–1063, 1961.
- [5] E. H. Lieb, “Exact solution of the problem of the entropy of two-dimensional ice,” *Phys. Rev. Lett.*, vol. 18, pp. 692–694, Apr 1967. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevLett.18.692>
- [6] E. Loidor and B. Marcus, “Improved lower bounds on capacities of symmetric 2D constraints using Rayleigh quotients,” *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1624–1639, 2010.
- [7] R. Berger, *The undecidability of the domino problem*, ser. Memoirs of the AMS. American Mathematical Society, 1966, no. 66.
- [8] Y. Chan and A. Rechnitzer, “Accurate lower bounds on 2-D constraint capacities from corner transfer matrices,” *IEEE Trans. Info. Theory*, vol. 60, no. 7, pp. 3845–3858, 2014.
- [9] N. Calkin and H. Wilf, “The number of independent sets in a grid graph,” *SIAM J. Disc. Math.*, vol. 11, no. 1, pp. 54–60, 1998.
- [10] S. Friedland, P. H. Lundow, and K. Markström, “The 1-vertex transfer matrix and accurate estimation of channel capacity,” *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3692–3699, 2010.
- [11] R. J. Baxter, “Hard hexagons: exact solution,” *J. Phys. A: Math. Gen.*, vol. 13, p. L61, 1980.
- [12] M. Cohn, “On the channel capacity of read/write isolated memory,” *Disc. Appl. Math.*, vol. 56, no. 1, pp. 1–8, 1995.
- [13] M. Golin, X. Yong, Y. Zhang, and L. Sheng, “New upper and lower bounds on the channel capacity of read/write isolated memory,” in *Information Theory, 2000. Proceedings. IEEE International Symposium on*. IEEE, 2000, p. 280.
- [14] J. Borwein, P. Borwein, and S. Plouffe, “Inverse symbolic calculator,” <http://isc.carma.newcastle.edu.au/>.
- [15] K. Engel, “On the Fibonacci number of an $M \times N$ lattice,” *Fibonacci Quart.*, vol. 28, no. 1, pp. 72–78, 1990.
- [16] D. Gamarnik and D. Katz, “Sequential cavity method for computing free energy and surface pressure,” *J. Stat. Phys.*, vol. 137, no. 2, pp. 205–232, 2009.
- [17] P. H. Lundow and K. Markström, “Exact and approximate compression of transfer matrices for graph homomorphisms,” *LMS J. Comput. Math.*, vol. 11, pp. 1–14, 2008.
- [18] C. Wang, X. Yong, and M. Golin, “The channel capacity of read/write isolated memory,” *Discrete Applied Mathematics*, 2015.
- [19] L. Collatz, “Einschließungssatz für die charakteristischen Zahlen von Matrizen,” *Math. Z.*, vol. 48, no. 1, pp. 221–226, 1942. [Online]. Available: <http://dx.doi.org/10.1007/BF01180013>
- [20] H. Wielandt, “Unzerlegbare, nicht negative matrizen,” *Math. Z.*, vol. 52, no. 1, pp. 642–648, 1950. [Online]. Available: <http://dx.doi.org/10.1007/BF02230720>
- [21] R. J. Baxter, “Dimers on a rectangular lattice,” *J. Math. Phys.*, vol. 9, p. 650, 1968.
- [22] —, “Variational approximations for square lattice models in statistical mechanics,” *J. Stat. Phys.*, vol. 19, no. 5, pp. 461–478, 1978.
- [23] R. J. Baxter, I. G. Enting, and S. K. Tsang, “Hard-square lattice gas,” *J. Stat. Phys.*, vol. 22, no. 4, pp. 465–489, 1980.
- [24] Y. Chan, “Series expansions from the corner transfer matrix renormalization group method: the hard squares model,” *J. Phys. A: Math. Theor.*, vol. 45, p. 085001, 2012.
- [25] —, “Series expansions from the corner transfer matrix renormalization group method: II. Asymmetry and high-density hard squares,” *J. Phys. A: Math. Theor.*, vol. 46, no. 12, p. 125009, 2013.
- [26] R. J. Baxter, “Planar lattice gases with nearest-neighbor exclusion,” *Ann. Comb.*, vol. 3, no. 2, pp. 191–203, 1999.
- [27] T. Nishino and K. Okunishi, “Corner transfer matrix renormalization group method,” *J. Phys. Soc. Jpn.*, vol. 65, no. 4, pp. 891–894, 1996.
- [28] —, “Corner transfer matrix algorithm for classical renormalization group,” *J. Phys. Soc. Jpn.*, vol. 66, no. 10, pp. 3040–3047, 1997.
- [29] G. Guennebaud, B. Jacob *et al.*, “Eigen v3,” <http://eigen.tuxfamily.org>, 2010.
- [30] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicissier, and P. Zimmermann, “MPFR: A multiple-precision binary floating-point library with correct rounding,” *ACM Trans. Math. Software*, vol. 33, no. 2, pp. 13:1–13:15, Jun. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1236463.1236468>
- [31] F. Ruskey and J. Sawada, “Generating necklaces and strings with forbidden substrings,” in *Computing and Combinatorics*. Springer, 2000, pp. 330–339.

Yao-ban Chan received his B.Inf.Sc. degree from Massey University in 2001, then continued with Honours and a Ph.D. in Mathematics at the University of Melbourne, awarded in 2006. He spent some years in postdoctoral positions in Melbourne, Bordeaux, Vienna and Montpellier, before joining the University of Queensland as a Lecturer in 2013. His research interests span mathematical biology, combinatorics and statistical mechanics, and statistics.

Andrew Rechnitzer received a B.Sc. degree from the University of Melbourne in 1996 and then combined study at the University of Melbourne and the University of Bordeaux to receive a Ph.D. in Mathematics in 2001. After postdoctoral work in Toronto and Melbourne, he joined the Department of Mathematics at the University of British Columbia in 2006, first as an Assistant Professor and presently as an Associate Professor. His research interests are in the application of methods from combinatorics and statistical physics to problems in mathematics, physics and computer science.