Numerical protocol of the Tensor Renormalization Group for interacting fields

Manuel Campos, Germán Sierra and Esperanza López

Instituto de Física Teórica UAM/CSIC, C/ Nicolás Cabrera 13-15, Cantoblanco, 28049 Madrid, Spain

Abstract

In this companion text we present the computational details of a code implementing the adapted Tensor Renormalization Group algorithm presented in [1]. A version of the actual program implemented in *Mathematica* can be found at github.com/m-campos/interacting-trg.

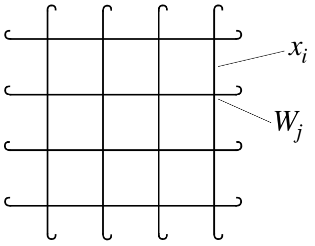
Contents

1 General view of the algorithm

1.1 Set-up of the problem

The adapted TRG protocol computes the partition function of a field theory over a 2-dimensional space discretized on a square lattice. The present code implements this protocol for the scalar $\lambda\phi^4$ theory up to first order in perturbation theory. All considerations in this text apply to this model.

As an example let us consider a lattice of $L = 2^2$ sites per side on a torus:



(1.1)

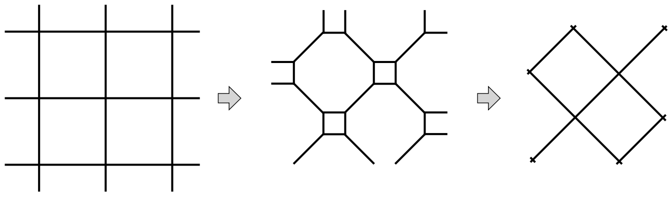
Each lattice link has a field x_i associated to it. Each lattice vertex has a Boltzmann weight W_j associated, that is, a real function over the four fields at the contiguous links. There is an implicit integration over all the fields on the lattice, the result of which is the partition function of the system.

$$Z = Z_0(1+\lambda Z_1) + O(\lambda^2) = \int \prod_i dx_i \prod_j W_j \,. \tag{1.2}$$

We have defined Z_1 in a way that is natural due to the structure of the code. Since the numerical values of the partition function are large, the numerical program will compute the free energy per site.

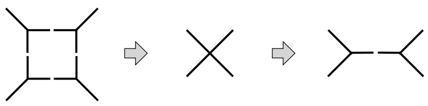
$$f = f_0 + \lambda f_1 + O(\lambda^2) = -\frac{1}{L^2}\left(\log Z_0 + \lambda Z_1\right) + O(\lambda^2)\,. \tag{1.3}$$

The adapted TRG protocol proceeds as the original Tensor Renormalization Group technique [3] by evolving the lattice geometry and the values of the weights. The process is local in the lattice. The application of one step of the protocol to some region of the lattice can be depicted as

 (1.4)

After one step, we end with a lattice tilted by 45 degrees and with a number of nodes and links which is a half of the original. In order to preserve all the information, the number of fields per link will increase unless we truncate some of them (and the adapted TRG protocol provides us with a way of truncating them considering their relevance to the entanglement among neighbour sites). We will call χ or *bond dimension* the number of fields per link. After some number of steps, we end up with a lattice with just one Boltzmann weight. Then the integration of the last fields results in a numeric approximation to the partion function of the system. The bigger the bond dimension we allow, the better the approximation we get.

For convenience reasons that will be clear below, we call an adapted TRG step the operation that goes from a cubic vertex to the next cubic vertex. This operation does not start and end with a square lattice, but with the intermediate lattice in (1.4).

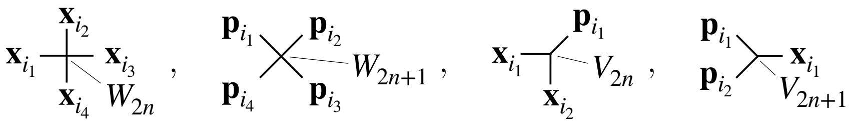
 (1.5)

It is composed of two operations over the Boltzmann weights at the vertices of the lattice, which are the integration of a plaquette (the first arrow) and the operation called (adapted) Singular Value Decomposition (SVD), which splits a Boltzmann weight into two cubic weights (the second arrow).

The conceptual details and physical motivation behind the adapted TRG protocol can be found in [1]. Here we can just mention that the basic idea is to modify the TRG protocol to work at the level of fields. At each SVD we get a hierarchy of fields, ordered by their involvement in local correlations, and we use this information to truncate entire fields. In this way we can make use of ideas and techniques coming from quantum information theory, but preserving the structure of a field theory at all levels of the process.

1.2 Structure of the Boltzmann weights

In (1.5) we saw the two types of weights we will consider in this protocol. We will denote Boltzmann weights W_k the ones associated to quartic vertices in the lattice and weights V_k the ones associated to cubic vertices.



(1.6)

Depending on their orientation in the lattice, there are two kinds of variables. The ones corresponding to vertical or horizontal links $\mathbf{x}$ and the ones corresponding to 45^{o} tilted links $\mathbf{p}$. We will use boldface letters to denote arrays of any number of variables $\mathbf{x} = (x_1, x_2, \dots)$. If there is no confusion, some of the previous diagrams will appear rotated by 45^{o} in order to make the graphical presentation cleaner.

All weights, Boltzmann or cubic, always have the following form

$$W(\mathbf{z}, \partial_{\mathbf{z}}) = e^{-\frac{1}{2}\mathbf{z}M\mathbf{z}} F(\mathbf{z}, \partial_{\mathbf{z}}) e^{-\mathbf{z}D\mathbf{z}} \,, \tag{1.7}$$

where $\mathbf{z}$ is the vector containing all fields associated to the links (that is, all $\mathbf{x}$ and $\mathbf{p}$ fields), M is a (possibly complex) symmetric matrix and D is a diagonal positive real matrix. In the following, we will reserve W and F for Boltzmann weights in the notation and use V and f for cubic weights. The partial derivative operators $\partial_{\mathbf{z}}$ are always considered to be placed at the right of the fields, that is,

$$F(\mathbf{z}, \partial_{\mathbf{z}}) e^{-\mathbf{z}D\mathbf{z}} = [F(\mathbf{z}, \partial_{\mathbf{u}}) e^{-\mathbf{u}D\mathbf{u}}]_{\mathbf{u}=\mathbf{z}} \,. \tag{1.8}$$

We will refer to $e^{-\frac{1}{2}\mathbf{z}M\mathbf{z}-\mathbf{z}D\mathbf{z}}$ as the exponential part of the weight, and to $F(\mathbf{z}, \partial_{\mathbf{z}})$ as the interaction part.

1.3 Initial values

The initial lattice model is the discretization of a model with Lagrangian

$$\mathscr{L} = \frac{1}{2}(\partial x) + \frac{1}{2}m^2 x^2 + \lambda x^4 \,, \tag{1.9}$$

where m is the physical mass. Up to first order in perturbation theory, the initial Boltzmann weights are

$$= W_0(x_i) = \left(1 + \frac{\lambda}{2} \sum_{i=1}^{4} x_i^4 \right) e^{-\Sigma_{i=1}^4 \frac{m^2}{4} x_i^2 - \frac{1}{2} \Sigma_{i=1}^4 (x_i - x_{i+1})^2} \,. \tag{1.10}$$

These Boltzmann weights have the general form shown in (1.7). As it was stated in (1.2), the partition function of the whole system is obtained by integrating over all fields. The adapted TRG performs a renormalization group flow in the space of Boltzmann weights by transforming the weight W_0 into successive “coarse grained” versions of it, W_n.

However, in order to implement the algorithm in the form proposed in the previous section, we have to start with a cubic weight V_1. It is easily constructed from (1.10).

$$\mathbf{x}\begin{matrix}\diagup x_1 \\ \diagdown x_2\end{matrix}\!\!\succ\!\!-\,\mathbf{p} = V_0(\mathbf{x},\mathbf{p}) = \rho_0\left[1+\frac{\lambda}{2}(x_1^4+x_2^4)\right]e^{-\frac{1}{2}\mathbf{x}A_0\mathbf{x}+i\mathbf{x}U_0\mathbf{p}-\frac{m^2}{4}\mathbf{x}^2-\frac{1}{4}\mathbf{p}^2}\ , \qquad (1.11)$$

where

$$A_0 = \begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}\ , \qquad U_0 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\ . \qquad (1.12)$$

Notice that (1.11) matches the general form (1.7). It is easy to check that the original Boltzmann weight is recovered from the new cubic ones when two of them are contracted. In the graphic notation, an integration over all fields associated to contracted links is implied.

$$\mathbf{x}_L \succ\!\underset{\mathbf{p}}{\text{—}}\!\prec\ \mathbf{x}_R = \int d\mathbf{p}\ V_0V_0^\dagger = W_0 = \mathbf{x}_L \times \mathbf{x}_R \qquad (1.13)$$

As we sill see in sec. 2.2, this is a case of application of the SVD operation. Furthermore, the freedom in the definition of U_0 is fixed in order to fit into the general form we will present in sec. 2.

1.4 Outline of the algorithm

The basic structure of the algorithm is a loop over an adapted TRG step until the lattice has collapsed to one vertex. The outline of the numerical code is the following one. In the next sections we will explain the details of the different routines.

```
Adapted TRG protocol main routine
        Set initial values
        Loop over one TRG step until the lattice is reduced
           to one tensor (order-0 regular step routine)
               Compute the contraction of a plaquette
               Diagonalize matrix B to get the SVD
```

```
                Check whether it is necessary to use
                    an approximate diagonalization
                Diagonalize matrix B using the
                   approximation if needed
        Truncate fields if needed
        Update interaction coefficients (order-1
           regular step routine)
                Compute Feynman diagrams matrices
                Contract Feynman diagrams
                Add contributions from different
                   weights
        Store contributions to the free energy
Contract last tensor (order-0 final step routine)
        Compute matrix Q
        Update interaction coefficients (order-1
           final step routine)
                Compute Feynman diagrams matrices
                Contract Feynman diagrams
                Add contributions from different
                   weights
                Obtain the final value of the first
                   correction to the free energy
        Add contributions from all previous steps
        Obtain the final value of the order-0 value
           of the free energy
```

2 Order λ^0 routines

The exponential part of the Boltzmann weights is the same for all orders in λ. In this section we explain how the algorithm compute its evolution. It is the basis upon which the next orders in perturbation theory are constructed.

A cubic weight has always the following form at order-0

$$V_n = \rho_n e^{-\frac{1}{2}\mathbf{x}A_n\mathbf{x}+i\mathbf{x}U_n\mathbf{p}} e^{-\frac{1}{4}\mathbf{x}\mathbb{1}_2\otimes D_{n-1}^{-1}\mathbf{x}-\frac{1}{4}\mathbf{p}D_n^{-1}\mathbf{p}} \ , \tag{2.1}$$

and the Boltzmann weights

$$W_n = \rho_n' e^{-\frac{1}{2}[\mathbf{x}_L A_n \mathbf{x}_L + \mathbf{x}_R A_n \mathbf{x}_R + (\mathbf{x}_L-\mathbf{x}_R)B_n(\mathbf{x}_L-\mathbf{x}_R)]} e^{-\frac{1}{4}\mathbf{x}_L\mathbb{1}_2\otimes D_{n-1}^{-1}\mathbf{x}_L-\frac{1}{4}\mathbf{x}_R\mathbb{1}_2\otimes D_{n-1}^{-1}\mathbf{x}_R} \ . \tag{2.2}$$

For the moment we will forget about the numerical prefactors ρ, they will be take into account in sec. 2.4. The matrices U_n and D_n achieve the diagonalization of $B_n = U_n D_n U_n^T$. These matrices have the following block-structure, which is preserved under the adapted TRG iterations [2].

$$A_n = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \otimes a_n \ , \quad B_n = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes b_{n,a} + \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \otimes b_{n,b} \ . \tag{2.3}$$

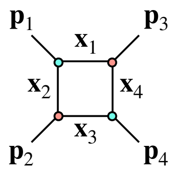
a_n, $b_{n,a}$ and $b_{n,b}$, are symmetric and positive semi-definite matrices. The diagonalization of B_n is given by

$$U_n = \frac{1}{\sqrt{2}} \begin{pmatrix} u_{n,a} & u_{n,b} \\ u_{n,a} & -u_{n,b} \end{pmatrix} \ , \quad D_n = \begin{pmatrix} d_{n,a} & 0 \\ 0 & d_{n,b} \end{pmatrix} \ , \quad b_{n,{}^{a}_{b}} = u_{n,{}^{a}_{b}} d_{n,{}^{a}_{b}} u^T_{n,{}^{a}_{b}} \tag{2.4}$$

The main goal of the `order-0 regular step` routine is to take the values of a_n, $b_{n,a}$, $b_{n,b}$ and d_{n-1} at the step n and compute their coarse grained versions at the step $n+1$. This is done by first integrating four cubic weights in a plaquette and then performing the SVD to get the new cubic weights. We shall next explain those two operations.

2.1 Integration of a plaquette

When integrating four cubic weights, we have to take into account that there are two types of cubic weights in our lattice: left-type $V_{n,L}$ and right-type $V_{n,R}$. The origin of this duplication can be found in the SVD operation explained in 2.2. One is the complex conjugate of the other $V_{n,R}(\mathbf{x}) = V_{n,L}(\mathbf{x})^*$. We shall take as reference the left-type, i.e., $V_{n,L} = V_n$ in 2.1. From now on we shall decorate the left-type weights in the graphical representation with a pink dot and the right-type with a cyan dot in order to differentiate them. The expression for the plaquette before integration is



$$= W_{n+1} = \rho_n^4 \int d\mathbf{x} \, e^{-\frac{1}{2}\mathbf{x}Q_n\mathbf{x} - i\mathbf{x}_L C_{n,L}\mathbf{p} - i\mathbf{x}_R C_{n,R}\mathbf{p}} \ , \tag{2.5}$$

where those matrices are constructed from the blocks in (2.3) and (2.4)

$$Q_n = S_n + K_n \ , \quad S_n = \mathbb{1}_4 \otimes D_{n-1}^{-1} \ , \quad \begin{aligned} C_{n,L} &= \begin{pmatrix} U_n & 0 \\ 0 & 0 \end{pmatrix} - U_{\text{shift}} \begin{pmatrix} 0 & 0 \\ 0 & U_n' \end{pmatrix} \ , \\ C_{n,R} &= \begin{pmatrix} 0 & 0 \\ 0 & U_n \end{pmatrix} - U_{\text{shift}} \begin{pmatrix} U_n' & 0 \\ 0 & 0 \end{pmatrix} \ . \end{aligned} \tag{2.6}$$

In the last expression K_n is

$$K_n = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix} \otimes a_n \ , \tag{2.7}$$

U_{shift} is the matrix that shifts $\mathbf{x}_i$ to $\mathbf{x}_{i-1}$ and U'_n is

$$U'_n = \begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix} U_n \ . \tag{2.8}$$

Matrix Q_n defined in (2.6) has a simple block diagonalization, which we present now for future use.

$$Q_n = U_Q \begin{pmatrix} q_{n,a} & 0 & 0 & 0 \\ 0 & q_{n,a} & 0 & 0 \\ 0 & 0 & q_{n,b} & 0 \\ 0 & 0 & 0 & q_{n,c} \end{pmatrix} U_Q^T \ , \quad \begin{aligned} q_{n,a} &= D_{n-1}^{-1} + 2a_n \ , \\ q_{n,b} &= D_{n-1}^{-1} \ , \\ q_{n,c} &= D_{n-1}^{-1} + 4a_n \ , \end{aligned} \tag{2.9}$$

where

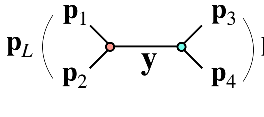
$$U_Q = \begin{pmatrix} 1/\sqrt{2} & 0 & 1/2 & 1/2 \\ 0 & 1/\sqrt{2} & 1/2 & -1/2 \\ -1/\sqrt{2} & 0 & 1/2 & 1/2 \\ 0 & -1/\sqrt{2} & 1/2 & -1/2 \end{pmatrix} \otimes \mathbb{1} \ . \tag{2.10}$$

The result of the integration in (2.5) gives the expression for A and B at the next step

$$A_{n+1} + B_{n+1} = C_{n,L}^T Q_n^{-1} C_{n,L} = C_{n,R}^T Q_n^{-1} C_{n,R} \ , \quad B_{n+1} = C_{n,L}^T Q_n^{-1} C_{n,R}^T \ , \tag{2.11}$$

2.2 Gaussian SVD

The SVD operation takes the Boltzmann weight W_n and splits it into two cubic weights $V_{n+1,L}$ and $V_{n+1,R}$. The basic idea behind it is to use a simple Fourier transform to introduce the new variables, in such a way that the new variables are ordered by their relevance to the numerical result. We will call these new variables $\mathbf{y}$ in order to differentiate them from variables $\mathbf{x}$ which we have already integrated when both of them appear in the same expression. This ordering is inherited by the order of the eigenvalues of matrix B at the exponent, which is the matrix that mixes fields at both sides of the weight. A detailed analysis and motivation of this operation can be found in [1]. We already know from (2.1) and (2.2) the form of the weights involved in the operation. They are given by



$$= W_n(\mathbf{p}_L, \mathbf{p}_R) = \int d\mathbf{y} \ V_{n,L}(\mathbf{p}_L, \mathbf{y}) V_{n,R}(\mathbf{p}_R, \mathbf{y}) \ . \tag{2.12}$$

A straightforward computation gives the expressions for the new block matrices (2.3) and (2.4) at step $n+1$

$$a_{n+1} = \frac{1}{2}\begin{pmatrix} {u_{n,a}}^T \\ -{u_{n,b}}^T \end{pmatrix} \left(D_{n-1}^{-1} + 2a_n\right)^{-1} \begin{pmatrix} u_{n,a} & -u_{n,b} \end{pmatrix} , \tag{2.13}$$

$$b_{n+1,a} = u_{n+1,a} d_{n+1,a} {u_{n+1,a}}^T \tag{2.14}$$

$$= \frac{1}{2}\begin{pmatrix} {u_{n,a}}^T \\ {u_{n,b}}^T \end{pmatrix} \left(D_{n-1}^{-1} + 2a_n\right)^{-1} \begin{pmatrix} u_{n,a} & u_{n,b} \end{pmatrix} , \tag{2.15}$$

$$b_{n+1,b} = u_{n+1,b} d_{n+1,b} {u_{n+1,b}}^T \tag{2.16}$$

$$\begin{aligned} = &-\frac{1}{2}\begin{pmatrix} {u_{n,a}}^T \\ -{u_{n,b}}^T \end{pmatrix} \left(D_{n-1}^{-1} + 2a_n\right)^{-1} \begin{pmatrix} u_{n,a} & -u_{n,b} \end{pmatrix} \\ &+ \begin{pmatrix} {u_{n,a}}^T \\ 0 \end{pmatrix} D_{n-1}^{-1} \begin{pmatrix} u_{n,a} & 0 \end{pmatrix} \\ &+ \begin{pmatrix} 0 \\ {u_{n,b}}^T \end{pmatrix} \left(D_{n-1}^{-1} + 4a_n\right)^{-1} \begin{pmatrix} 0 & u_{n,b} \end{pmatrix} \end{aligned} \tag{2.17}$$

$$D_{n+1} = \begin{pmatrix} d_{n+1,a} & 0 \\ 0 & d_{n+1,b} \end{pmatrix} \tag{2.18}$$

We shall make two remarks about the SVD operation. Firstly, for small values of the mass m one of the eigenvalues of B can be several orders of magnitude larger than the others. This could lead to a significant numerical error in the diagonalization of B. A solution to this problem is to employ an approximate diagonalization by expanding the eigenvalues and eigenvectors of B in a power series of the inverse of the largest eigenvalue. The details of this approximation are explained in sec. 4

The other remark has to do with the truncation of fields. When some of the eigenvalues of B are very small, the associated exponentials in (2.1) can be treated as Dirac delta functions with small effect on the numerical error.

$$\rho_n e^{-\frac{1}{4}\mathbf{p}D_n^{-1}\mathbf{p}} \approx \tilde{\rho}_n e^{-\frac{1}{4}\mathbf{p}'\tilde{D}_n^{-1}\mathbf{p}'} \delta^k(\mathbf{p}'') , \tag{2.19}$$

where $\mathbf{p} = (\mathbf{p}', \mathbf{p}'')$, $\mathbf{p}'$ being the set of $\chi_n - k$ fields associated with larger eigenvalues of B and $\mathbf{p}''$ the set of k fields associated with eigenvalues smaller than a certain threshold ε_{trunc}, which is a parameter of the algorithm. $\tilde{D}_n$ is just the diagonal block of D_n corresponding to its larger entries (the set of large eigenvalues of B_n). When replacement (2.19) is performed, those less-relevant fields can be integrated out and they stop playing any role in the numerical computation. As we will show in sec. 2.4, the smaller are the eigenvalues we are willing to truncate the smaller is the numerical error we introduce.

2.3 Summary

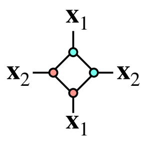
The code for the exponential part of the algorithm has the following structure

```
Order-0 regular step routine
        Integrate a plaquette
                Compute matrices q
                Compute matrices a and b
        Diagonalize b
                Check whether it is necessary to use an
                   approximate diagonalization
                Diagonalize b using the approximation if
                   needed
        Truncate fields if needed
        Update interaction coefficients using the already
           computed values of the exponent matrices (order
           -1 regular step routine)
        Store eigenvalues of matrices Q and B, they are the
           contribution of this step to the free energy
```

The `order-1 regular step` subroutine is included inside the `order-0 regular step` routine for computational reasons, although it is conceptually a different task that will be explained in later sections.

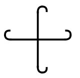
2.4 Last step

In the last step of the adapted TRG protocol we have to contract four cubic weights following this disposition



(2.20)

From (2.3), (2.4) and (2.9) it is an simple computation to construct the exponent of the plaquette. After the integration of the internal variables we get the following expression for the path integral at order λ^0.

$$Z_0 = \text{} = \rho'_n \int d\mathbf{x}\, e^{-\frac{1}{2}\mathbf{x}\widetilde{Q}_n\mathbf{x}} \,, \tag{2.21}$$

where

$$\widetilde{Q}_n = \widetilde{S}_n + 2\begin{pmatrix} \tilde{q}_a & 0 & 0 & 0 \\ 0 & \tilde{q}_c & 0 & \tilde{q}_c \\ 0 & 0 & \tilde{q}_a & 0 \\ 0 & \tilde{q}_c & 0 & \tilde{q}_c \end{pmatrix} \,, \quad \begin{aligned} \widetilde{S}_n &= \mathbb{1}_2 \otimes D_n^{-1} \,, \\ \tilde{q}_a &= u_{n,a}^T q_{n,a} u_{n,a} \,, \\ \tilde{q}_c &= u_{n,c}^T q_{n,c} u_{n,c} \,. \end{aligned} \tag{2.22}$$

To obtain the partition function we multiply all numerical contributions from the previous steps. They are codified in the numerical prefactors ρ_n. Each SVD operation contributes with a $(2\pi)^{-1/2}|D_n|^{-1/2}$ factor per Boltzmann weight to the final result, and each integration of a plaquette with a $(2\pi)^{1/2}|Q_n|^{-1/2}$ factor per plaquette. Hence the final expression for the partition function is

$$Z_0 = (2\pi)^{N/2}|\widetilde{Q}_n|^{-1/2}\prod_{k=1}^{2o-1}|D_n|^{-2^{2o-k-1}}|Q_n|^{-2^{2o-k-2}} \tag{2.23}$$

Summary The code for the last step has the following structure

```
Order-0 last step routine
        Integrate a plaquette
                Compute matrices q
        Compute matrix M
        Add contributions from all the previous steps to the
            free energy at order 0
        Compute the order 1 contribution to the free energy
           (order-1 final step routine)
        Return results
```

The `order-1 final step` subroutine is included inside the `order-0 final step` routine for computational reasons, although it is conceptually a different task. It is explained below.

3 Order λ^1 routines

The order-1 routines proceed in a similar way to the order-0 ones. Each adapted TRG step starts with a cubic weight V_n and evolves it to a coarse grained version V_{n+1} as depicted in (1.5). The last step takes four cubic weights and contract them to get the final result of the partition function.

There are, however, a couple of structural differences. Firstly, the order-1 factors of the cubic weight are updated in a single operation which combines the integration of a plaquette and the SVD operations. This operation is summarized at the end of sec. 3.2. Secondly, we have to introduce new formal variables which duplicate the number of original fields. As we will see, only one of the two independent linear combination of fields appear in the free part of the weights, so the numerical cost associated to the order-0 computations does not increase. See [1] for a more detailed explanation of those issues.

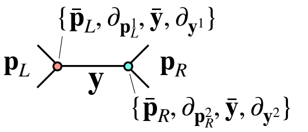
We call $\mathbf{x}^1$ and $\mathbf{x}^2$ the two sets of fields. Their even and odd combinations are

$$\bar{\mathbf{x}} = \mathbf{x}^1 + \mathbf{x}^2\ , \quad \hat{\mathbf{x}} = \mathbf{x}^1 - \mathbf{x}^2\ . \tag{3.1}$$

The general form of a cubic weight is

$$V_n = \rho_n e^{-\frac{1}{2}\bar{\mathbf{x}}A_n\bar{\mathbf{x}}+i\bar{\mathbf{x}}U_n\bar{\mathbf{p}}}[1+\lambda g_n(\partial_{\mathbf{p}^1},\bar{\mathbf{x}},\partial_{\mathbf{x}^1})]e^{-\frac{1}{4}\mathbf{x}^1\mathbb{1}_2\otimes D_{n-1}^{-1}\mathbf{x}^1-\frac{1}{4}\mathbf{p}^1D_n^{-1}\mathbf{p}^1}\ . \tag{3.2}$$

There are two kinds of lattices, which were depicted in (1.4): the ones composed with cubic weights and the ones composed with Boltzmann weights. In both cases each link connects two weights and has two sets of fields, $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$ (which can be either $\mathbf{x}$ or $\mathbf{p}$ fields), associated to it. The interaction part of the weights generically depends on the partial derivatives of fields. However, the algorithm is designed in such a way that, for two weights linked by link i, the interaction part of one depends only on $\partial_{\mathbf{z}_i^1}$ and the interaction part of the other on $\partial_{\mathbf{z}_i^2}$. As an example, after the SVD (2.12) we have a link associated to $\mathbf{y}$ variables with the following dependences at the interaction part of the weights.


$\{\bar{\mathbf{p}}_L, \partial_{\mathbf{p}_L^1}, \bar{\mathbf{y}}, \partial_{\mathbf{y}^1}\}$
$\mathbf{p}_L$
$\mathbf{y}$
$\mathbf{p}_R$
$\{\bar{\mathbf{p}}_R, \partial_{\mathbf{p}_R^2}, \bar{\mathbf{y}}, \partial_{\mathbf{y}^2}\}$
(3.3)

Since we are working with a $\lambda\phi^4$ theory, the function g_n in (3.2) is a order-4 polynomial with all monomials of odd order equal to zero. Equation (3.2) may be compared to (1.7) and (2.1). This duplication of fields is not necessary until the second TRG step. However, we want to use the same numerical routine at all steps from the start. This can be accomplished by identifying the original order-0 fields of the initial lattice with the even fields $\bar{\mathbf{x}}$, and introducing odd fields $\hat{\mathbf{x}}$ from the beginning. Since odd and even fields are decoupled at the initial lattice, the odd part does not contribute to the partition function. The introduction of those fields is the price to pay for keeping the operations local in terms of the weights when there are interactions. A detailed account of this procedure can be found in [1] (however, in that reference the new fields are not introduced from the start, but only when necessary).

The function g_n carries all the information of the interaction. We saw in (1.11) that at the initial cubic weights it is

$$g_0 = \frac{1}{2}[(\bar{x}_1)^4 + (\bar{x}_2)^4\ ]\ , \tag{3.4}$$

where the subindex indicate the different individual fields $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2)$.

3.1 Integration of a plaquette

An adapted TRG step starts with four cubic weights (3.2) and contracts them in a plaquette as shown in (2.5)

$$W_{n+1} = \rho_n^4 \int d\mathbf{x}^1 d\mathbf{x}^2\ e^{-\frac{1}{2}\bar{\mathbf{x}}K_n\bar{\mathbf{x}} - i\bar{\mathbf{x}}C_n\bar{\mathbf{p}}}[1 + \lambda \sum_{k=1}^{4} g_{n,k}]e^{-\frac{1}{4}\sum_{l=1}^{2}\mathbf{x}^l S_{n-1}\mathbf{x}^l - \frac{1}{4}\mathbf{p}^1 D_n^{-1}\mathbf{p}^1}\ , \tag{3.5}$$

where we have relabelled $\{1 \leftrightarrow 2\}$ some of the external $\mathbf{p}^l$ fields so that all the diagonal terms at the exponent concerns $\mathbf{p}^1$. Functions $g_{n,k}$ come from the four cubic vertices. They can be $g_{n,k}(\partial_{\mathbf{p}^1}, \bar{\mathbf{x}}, \partial_{\mathbf{x}^l})$, with $l = 1,2$ depending on the vertex. We can simplify (3.5) by integrating by parts the $\mathbf{x}^l$ variables to change from $\partial_{\mathbf{x}^l}$ to $-\partial_{\bar{\mathbf{x}}}$ in the $g_{n,k}$ functions. Then the dependence on the odd variables $\hat{\mathbf{x}}$ factorizes and can be integrated out. Using new variables $\mathbf{u}$ and $\mathbf{q}$ to clarify the action of the partial derivatives, the simplified expression of the Boltzmann weight is

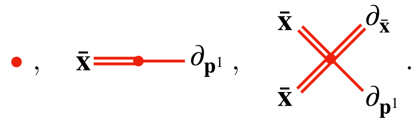
$$W_{n+1} \propto \int d\bar{\mathbf{x}} \;\; [1 + \lambda g'_n(\partial_{\mathbf{q}^1}, \bar{\mathbf{x}}, \partial_{\bar{\mathbf{u}}})] e^{-\frac{1}{2}\bar{\mathbf{u}}K_n\bar{\mathbf{u}} - i\bar{\mathbf{u}}C_n\bar{\mathbf{p}}} e^{-\frac{1}{4}\bar{\mathbf{x}}S_{n-1}\bar{\mathbf{x}} - \frac{1}{4}\mathbf{q}^1 S_n \mathbf{q}^1} \Big|_{\bar{\mathbf{u}}=\bar{\mathbf{x}}, \mathbf{q}^1=\mathbf{p}^1} , \qquad (3.6)$$

where

$$g'_n(\partial_{\mathbf{p}^1}, \bar{\mathbf{x}}, \partial_{\bar{\mathbf{x}}}) = \sum_{k=1}^{4} g_{n,k}(\partial_{\mathbf{p}^1}, \bar{\mathbf{x}}, -\partial_{\bar{\mathbf{x}}}) \, . \qquad (3.7)$$

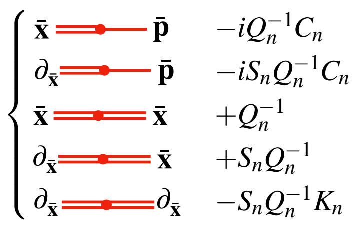
Since the constant prefactors ρ are already explained in sec. 2.4, here we shall forget about them an focus on the g functions, which encode all the contributions up to the first order correction to the partition function Z_1 in (1.2).

The operation for going from four cubic weights to one Boltzmann weight can be represented in terms of Feynman diagrams. g'_n is an order-4 polynomial which only contains monomials of even powers. Therefore the coefficients in g'_n can be organized in Feynman diagrams with 2 and 4 legs, which will be depicted in red in order to avoid confusion with lattice weights. There are 3 kinds of legs, corresponding to the variables $\partial_{\mathbf{p}^1}$, $\bar{\mathbf{x}}$ and $\partial_{\bar{\mathbf{x}}}$. We will represent the variables wich are going to be integrated in the present operation by double legs and the external variables by single legs. Some examples of diagrams, with orders 0 (the constant term of g'_n), 2 and 4 are:


,
$\bar{\mathbf{x}}$
$\partial_{\mathbf{p}^1}$,
$\bar{\mathbf{x}}$
$\partial_{\bar{\mathbf{x}}}$
$\bar{\mathbf{x}}$
$\partial_{\mathbf{p}^1}$
.

(3.8)

Diagrams with 2 legs codify the coefficients of a matrix, diagrams with 4 legs codify a tensor of rank 4. The action of the integral (3.6) on the diagrams results in a new set of diagrams for the external fields, that is, variables $\partial_{\mathbf{p}^1}$ and $\bar{\mathbf{p}}$. The corresponding Feynman rules for the different propagators and new legs are


$\bar{\mathbf{x}}$ $\bar{\mathbf{p}}$ $-iQ_n^{-1}C_n$
$\partial_{\bar{\mathbf{x}}}$ $\bar{\mathbf{p}}$ $-iS_nQ_n^{-1}C_n$
$\bar{\mathbf{x}}$ $\bar{\mathbf{x}}$ $+Q_n^{-1}$
$\partial_{\bar{\mathbf{x}}}$ $\bar{\mathbf{x}}$ $+S_nQ_n^{-1}$
$\partial_{\bar{\mathbf{x}}}$ $\partial_{\bar{\mathbf{x}}}$ $-S_nQ_n^{-1}K_n$

(3.9)

The resulting diagrams after the integration are obtained by contracting all $\bar{\mathbf{x}}$ and $\partial_{\bar{\mathbf{x}}}$ legs of the diagrams in (3.7) with the corresponding legs of the diagrams in (3.9) in all possible ways. They codify the interaction part $g''_n(\bar{\mathbf{p}}, \partial_{\mathbf{p}^1})$ of the new Boltzmann weight

$$W_{n+1} \propto e^{-\frac{1}{2}\bar{\mathbf{p}}_L(A_{n+1}+B_{n+1})\bar{\mathbf{p}}_L - \frac{1}{2}\bar{\mathbf{p}}_R(A_{n+1}+B_{n+1})\bar{\mathbf{p}}_R + \bar{\mathbf{p}}_L B_{n+1}\bar{\mathbf{p}}_R}[1+\lambda g''_{n+1}(\bar{\mathbf{p}}, \partial_{\mathbf{p}^1})]e^{-\frac{1}{4}\mathbf{p}^1 S_n \mathbf{p}^1}\ . \quad (3.10)$$

3.2 Gaussian SVD

The splitting of the Boltzmann weights (3.10) into two cubic weights (3.2) is given by

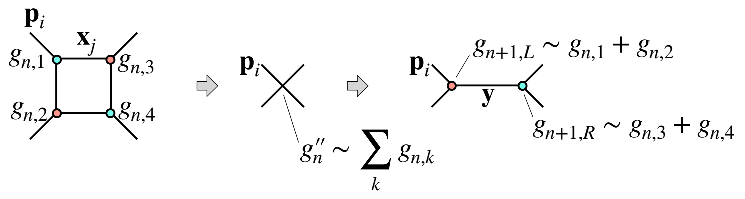
$$W_{n+1}(\bar{\mathbf{p}}, \mathbf{p}^1) = \int d\mathbf{y}^1 d\mathbf{y}^2\ V_{n+1,L}(\bar{\mathbf{y}}, \mathbf{y}^1, \bar{\mathbf{p}}, \mathbf{p}^1) V_{n+1,R}(\bar{\mathbf{y}}, \mathbf{y}^2, \bar{\mathbf{p}}, \mathbf{p}^1)\ . \quad (3.11)$$

We have introduce a new set of fields $\mathbf{y}$ associated with the new links in the lattice. They are the fields analogous to $\mathbf{x}$ at the next coarse grained level.

The exponential part is fixed by the order-0 SVD. Now we have to choose the splitting of the interaction part of W_{n+1} into two pieces, one which will go to the left cubic weight $V_{n+1,L}$ and one to the right $V_{n+1,R}$.

$$1+\lambda g''_{n+1} = (1+\lambda g''_{n+1,L})(1+\lambda g''_{n+1,R}) + O(\lambda^2) \quad (3.12)$$

The easiest choice comes from (3.7) and can be pictured as follows.



(3.13)

We send the Feynman diagrams from $g_{n,1}$ and $g_{n,2}$ to the left and the ones from $g_{n,3}$ and $g_{n,4}$ to the right. Operations are the same at both sides under certain symmetries and give the same results. In the following we are just going to consider the operations on the left side.

To sum up, we start the adapted TRG step with two functions $g_{n,1}$ and $g_{n,2}$ coming from two cubic weights (3.2). Then we sum them and apply the Feynman rules (3.9) to get $g''_{n+1,L}$, which is the left piece of the interaction part of the Boltzmann weight (3.10). Finally we apply the SVD to get the interaction part g_{n+1} of the new cubic weight $V_{n+1,L}$ in (3.11).

In order to satisfy (3.11), the relation of g_{n+1} with $g''_{n+1,L}$ is just the change of variables

$$\bar{\mathbf{p}}_R \to \bar{\mathbf{p}}_L - iU_{n+1}\partial_{\bar{\mathbf{z}}} \quad (3.14)$$

This change of variables can be included in the Feynman rules (3.9) resulting in the following set of rules

$$\begin{cases} \bar{\mathbf{x}} \Relbar\!\!\bullet\!\!\relbar \bar{\mathbf{p}} & -iQ_n^{-1}(C_{n,L}+C_{n,R}) \\ \bar{\mathbf{x}} \Relbar\!\!\bullet\!\!\relbar \partial_{\bar{\mathbf{y}}} & -Q_n^{-1}C_{n,R}U_{n+1} \\ \partial_{\mathbf{x}^1} \Relbar\!\!\bullet\!\!\relbar \bar{\mathbf{p}} & +iS_nQ_n^{-1}(C_{n,L}+C_{n,R}) \\ \partial_{\mathbf{x}^1} \Relbar\!\!\bullet\!\!\relbar \partial_{\bar{\mathbf{y}}} & +S_nQ_n^{-1}C_{n,R}U_{n+1} \\ \bar{\mathbf{x}} \Relbar\!\!\bullet\!\!\Relbar \bar{\mathbf{x}} & +Q_n^{-1} \\ \partial_{\mathbf{x}^1} \Relbar\!\!\bullet\!\!\Relbar \bar{\mathbf{x}} & -S_nQ_n^{-1} \\ \partial_{\mathbf{x}^1} \Relbar\!\!\bullet\!\!\Relbar \partial_{\mathbf{x}^1} & -S_nQ_n^{-1}K_n \end{cases} \tag{3.15}$$

Joining all the pieces, we get the final form of the operation that performs the whole adapted TRG step at order-1. We start with the Feynman diagrams of the polynomial $g_{n,1}+g_{n,2}$ and contract all $\bar{\mathbf{x}}$ and $\partial_{\mathbf{x}^1}$ legs with the diagrams (3.15) in all possible ways to obtain the Feynman diagrams of g_{n+1}.

Summary The code for one regular step at order-1 has the following simple structure.

```
Order-1 regular step routine
          Compute combinations of matrices needed for the
              Feynman rules
          Apply Feynman rules to the Feynman diagrams
```

3.3 Final step

At the last step we perform the integration of the last plaquette, similarly to what we did in (3.5) but with the disposition of weights shown in (2.20). The integration of the internal variables of the last plaquette can be performed using the Feynman rules for the integration (3.9), with no posterior SVD. The integration of the remaining fields can be performed using the last three rules of (3.9), which are the ones that integrate fields, but using the exponential matrices of the last step (2.22).

$$\begin{cases} \bar{\mathbf{x}} \Relbar\!\!\bullet\!\!\Relbar \bar{\mathbf{x}} & +\widetilde{Q}_n^{-1} \\ \partial_{\mathbf{x}^1} \Relbar\!\!\bullet\!\!\Relbar \bar{\mathbf{x}} & -\widetilde{S}_n\widetilde{Q}_n^{-1} \\ \partial_{\mathbf{x}^1} \Relbar\!\!\bullet\!\!\Relbar \partial_{\mathbf{x}^1} & -\widetilde{S}_n\widetilde{Q}_n^{-1}\widetilde{K}_n \end{cases} \tag{3.16}$$

where $\widetilde{K}_n = \widetilde{Q}_n - \widetilde{S}_n$.

Summary The code for the last step at order-1 has the following structure.

```
Order-1 final step routine
        Compute combinations of matrices needed for the
           Feynman rules
        Apply Feynman rules to the Feynman diagrams of the
           internal variables
        Compute combinations of matrices needed for the
           Feynman rules of the last variables
        Apply Feynman rules to the Feynman diagrams of the
           last variables
```

4 Approximate diagonalization

In this section we explain the subroutine used to diagonalize a matrix B when it has one very large eigenvalue. The starting data are the first approximation to the large eigenvalue $\tilde{\varepsilon}^{-1}$ and its corresponding eigenvector $\tilde{v}^0$. That is, we have a matrix B with $N+1$ eigenvalues, which can be decomposed as

$$B = \frac{1}{\tilde{\varepsilon}}\tilde{v}^0\tilde{v}^{0^T} + \tilde{B} \; , \qquad \tilde{v}^{0^T}\tilde{v}^0 = 1. \tag{4.1}$$

The projection onto the space orthogonal to $\tilde{v}^0$ is given by $P = \mathbb{1} - \tilde{v}^0\tilde{v}^{0^T}$. Firstly we define

$$\frac{1}{\hat{\varepsilon}} = \frac{1}{\tilde{\varepsilon}} + \tilde{v}^{0^T}\tilde{B}\tilde{v}^0 \; , \quad \hat{v}^0 = \tilde{v}^0 + \hat{\varepsilon}P\tilde{B}v^0 \; , \tag{4.2}$$

$$\bar{B} = P\Big(\tilde{B} - \hat{\varepsilon}\tilde{B}\tilde{v}^0\tilde{v}^{0^T}\tilde{B}\Big)P \; , \quad \varepsilon = \frac{\hat{\varepsilon}}{\hat{v}^{0^T}\hat{v}^0} \; , \quad v^0 = \frac{\hat{v}^0}{\sqrt{\hat{v}^{0^T}\hat{v}^0}}, \tag{4.3}$$

which now is an orthogonal splitting

$$\frac{1}{\varepsilon}v^0 v^{0^T} + \bar{B} = B \; . \tag{4.4}$$

Our goal is to expand the diagonalization of B in powers of ε. The diagonalization of $P\bar{B}P$ can be computed by a standard numerical routine since it does no have divergent terms.

$$P\bar{B}P = \sum_k^N b_k^0 v_k^0 v_k^{0^T} \; , \qquad v_k^{0^T} v_l^0 = \delta_{kl} \; . \tag{4.5}$$

The eigenvectors v_k^0 are normalized. On the other hand, let us consider the exact (unknown) diagonalization of B, also with normalized eigenvectors

$$B = bvv^T + \sum_k^N b_k v_k v_k{}^T \, . \tag{4.6}$$

One can now expand everything in powers of ε

$$b = \frac{1}{\varepsilon} + b^0 + \varepsilon b^1 + \mathscr{O}\big(\varepsilon^2\big) \ , \quad v = v^0 + \varepsilon v^1 + \varepsilon^2 v^2 + \mathscr{O}\big(\varepsilon^3\big) \ , \tag{4.7}$$

$$b_k = b_k^0 + \varepsilon b_k^1 + \varepsilon^2 b_k^2 + \mathscr{O}\big(\varepsilon^3\big) \ , \quad v_k = v_k^0 + \varepsilon v_k^1 + \varepsilon^2 v_k^2 + \mathscr{O}\big(\varepsilon^3\big) \ . \tag{4.8}$$

Expanding the diagonalization equations

$$Bv = bv \ , \quad Bv_k = b_k v_k \ , \quad v^T v = 1 \ , \quad v_j^T v_k = \delta_{jk} \ , \quad v^T v_k = 0 \ , \tag{4.9}$$

and projecting onto v^0 and v_l^0 it is a straightforward computation to get the first terms in the expansion of the eigenvalues and eigenvectors of B. Let us define

$$c = {v^0}^T \bar{B} v^0 \ , \quad c_k = {v_k^0}^T \bar{B} v^0 \ , \quad c_{lk} = \frac{c_l c_k}{b_l^0 - b_k^0} \ , \tag{4.10}$$

$$d_k = c_k\big(c - b_k^0\big) - \sum_{l \neq k}^{N} c_{lk} c_l \ , \quad d_{lk} = \frac{c_k^2 c_{lk} + c_l d_k}{b_l^0 - b_k^0} \ . \tag{4.11}$$

Then the first terms in the expansion can be written as

$$b^0 = c \ , \quad b^1 = \sum_{l}^{N} c_l^2 \ , \tag{4.12}$$

$$b_k^1 = -c_k^2 \ , \quad b_k^2 = c_k d_k \ , \tag{4.13}$$

$$v^1 = \sum_{l}^{N} c_l v_l^0 \ , \quad v^2 = -\frac{1}{2}\left(\sum_{l}^{N} c_l^2\right) v^0 + \sum_{l}^{N} \big(b_l^0 - c\big) c_l v_l^0 \ , \tag{4.14}$$

$$v_k^1 = -c_k v^0 + \sum_{l \neq k}^{N} c_{lk} v_l^0 \ , \quad v_k^2 = d_k v^0 - \frac{1}{2}\left(c_k^2 + \sum_{l \neq k}^{N} c_{lk}^2\right) v_k^0 - \sum_{l \neq k}^{N} d_{lk} v_l^0 \ . \tag{4.15}$$

References

[1] M. Campos, G. Sierra, E. López, Tensor Renormalization Group for interacting quantum fields,

[2] M. Campos, G. Sierra, E. López, Tensor renormalization group in bosonic field theory, Phys. Rev. B **100**, 195106 (2019).

[3] M. Levin and C. P. Nave, “Tensor Renormalization Group Approach to Two-Dimensional Classical Lattice Models”, Phys. Rev. Lett. **99**, 120601 (2007).