DESY 92 – 126 ISSN 0418 – 9833
September 1992

Neural multigrid for gauge theories and other disordered systems

M. Bäker, T. Kalkreuter, G. Mack and M. Speh
II. Institut für Theoretische Physik, Universität Hamburg,
Luruper Chaussee 149, 2000 Hamburg 50

ABSTRACT

We present evidence that multigrid works for wave equations in disordered systems, e.g. in the presence of gauge fields, no matter how strong the disorder, but one needs to introduce a "neural computations" point of view into large scale simulations: First, the system must learn how to do the simulations efficiently, then do the simulation (fast). The method can also be used to provide smooth interpolation kernels which are needed in multigrid Monte Carlo updates.

Keywords: Multigrid; Neural Networks; Disordered Systems; Gauge Fields; Neural Multigrid.

There is a stochastic multigrid method and a deterministic one. The stochastic version is used to compute high dimensional integrals in Euclidean quantum field theory or statistical mechanics by a Monte Carlo method which uses updates at different length scales [1, 2]. The deterministic version [3, 4] solves discretized partial differential equations. One hopes to use both of them in simulations of lattice QCD, for updating the gauge fields and for computing fermion propagators in given gauge fields. In either case the aim is to beat critical slowing down in nearly critical systems, i.e. to maintain fast convergence when long range correlations appear.

A crucial problem is how to define and exhibit smooth functions in the disordered context, i.e. when translation symmetry is strongly violated. We will present a method how to solve this problem.

We recommend to think of more general disordered systems than gauge theories. This puts the core of the problem into sharper focus, and it opens the way to other possible applications such as low lying states of spin glasses, the shape of a lightning, waves on fractal lattices (with bond percolation), localization of low lying electronic states in amorphous materials, etc.

1. The Multigrid

One starts from a problem on a given "fundamental" lattice Λ^0 of lattice spacing a_0. One introduces a sequence of lattices $\Lambda^1, \Lambda^2, ..., \Lambda^N$ of increasing lattice spacings $a_j = L_b^j a_0$, together with interpolation operators $\mathcal{A}^j$ and restriction operators C^j which map functions on coarser lattices into functions on finer lattices, and vice versa. Let $\mathcal{H}^j$ be the space of functions on lattice Λ^j. Then we need

$$\mathcal{A}^j \ : \ \mathcal{H}^j \mapsto \mathcal{H}^{j-1} : \ \text{interpolation operator} \tag{1.1}$$

$$C^j \ : \ \mathcal{H}^{j-1} \mapsto \mathcal{H}^j : \ \text{restriction operator} \tag{1.2}$$

Given the interpolation operators, one can use them to define restriction operators. The choice $C^j = \mathcal{A}^{j*}$ is made in "variational coarsening" [2].

Typically, we choose $L_b = 2$, and a last layer Λ^N which consists of a single point.

2. The Basic Importance of Smoothness

2.1. *Deterministic multigrid*

One wants to solve a discretized elliptic differential equation on Λ^0

$$D_0 \xi^0 = f^0 \ . \tag{2.1}$$

It might have arisen from an eigenvalue equation $D_0\xi^0 = \varepsilon\xi^0$ by inverse iteration [5]. If D_0 has a small eigenvalue, then local relaxation algorithms suffer from critical slowing down.

Basic observation (in the "ordered case" [3, 4]): After some (damped) relaxation sweeps on Λ^0 one gets an approximate solution $\tilde{\xi}^0$ whose error $e^0 = \xi^0 - \tilde{\xi}^0$ is not necessarily small but is *smooth* (on length scale a_0). The unknown error e^0 satisfies the equation

$$D_0 e^0 = r^0. \tag{2.2}$$

It involves the *residual* $r^0 = f^0 - D_0\tilde{\xi}^0$ which would be zero for an exact solution.

Given that e^0 is smooth, it can be obtained by smooth interpolation of a suitable function e^1 on Λ^1,

$$e^0 = \mathcal{A}^1 e^1 \ . \tag{2.3}$$

That is, $e^0_z = \sum_{x\in\Lambda^1} \mathcal{A}^1_{zx} e^1_x$ with $\mathcal{A}^1_{zx}$ which depends smoothly on z.

Now define a restriction operator C^1 such that interpolation followed by restriction amounts to doing nothing, i.e. $C^1\mathcal{A}^1 = 1$. (For instance $C^1 = (\mathcal{A}^{1*}\mathcal{A}^1)^{-1}\mathcal{A}^{1*}$). Then (2.3) can be inverted, $e^1 = C^1 e^0$. Applying C^1 to both sides of (2.2) yields an equation for e^1,

$$D_1 e^1 = r^1 \ , \tag{2.4}$$

with $r^1 = C^1 r^0$ (restricted residual) and $D_1 = C^1 D_0 \mathcal{A}^1$ (effective differential operator). Given e^1, one obtains e^0 from (2.3), and $\tilde{\xi}^0 + e^0$ is an improved solution of (2.1). Thus, *the problem has been reduced to an equation on the lattice* Λ^1 *which has fewer points.* If necessary, one repeats the procedure, moving to Λ^2 etc. The procedure stops, because an equation on a "lattice" Λ^N with only a single point is easy to solve.

The iterated interpolation $\mathcal{A}^{[0j]} \equiv \mathcal{A}^1\mathcal{A}^2 ... \mathcal{A}^j$ from Λ^j to Λ^0 should yield functions on Λ^0 which are *smooth on length scale* a_j, i.e. which change little over a distance a_j (in the ordered case). For reasons of practicality, one must require that

$$\mathcal{A}^j_{zx} = 0 \text{ unless } z \text{ is near } x \ . \tag{2.5}$$

Example: The optimal choice for the 1-dimensional Laplace equation is [4]

$$\mathcal{A}_{xx} = 1 \quad , \mathcal{A}_{x\pm 1, x} = \frac{1}{2} \quad , \text{ others} = 0 \ . \tag{2.6}$$

We chose to regard Λ^1 as a sublattice of Λ^0 and set $a_0 = 1$.

This interpolation has the property that it maps constant functions on Λ^1 into constant functions on Λ^0. Constant functions are the lowest eigenmodes of the Laplacian with periodic b.c. D_1 comes out proportional to a Laplacian again.

2.2. *Stochastic multigrid*

The successful stochastic multigrid updating method [6, 7] for $O(N)$, CP^N and $SU(N) \times SU(N)$ spin models was described in Wolff's lecture [8]. One uses updates of spins $s(z)$ at sites $z \in \Lambda^0$ of the form

$$s(z) = e^{i\lambda \mathcal{A}^{[0j]}_{zx}} s(z) \tag{2.7}$$

where the matrix λ is a generator of a group of transformations which can act on spins s, and where $\mathcal{A}^{[j]}_{zx}$ vanishes outside a neighborhood of diameter of order a_j of x. One may assume [a] x to take values in a lattice Λ^j. It is important, though, that the supports of $\mathcal{A}^{[0j]}_{zx}$ in z should overlap for adjacent x.

This procedure eliminates critical slowing down in the spin models almost completely, *provided* on chooses $\mathcal{A}^j$ to be smooth in z on length scale j.

For general models, a sufficiently high acceptance rate for nonlocal updates like (2.7) is necessary to eliminate critical slowing down. Pinn and Grabenstein show evidence [9] that smoothness alone is not in general enough to ensure this. Their criterion demands also nonappearance of mass terms. In the spin models this is true. Pure gauge theories have no bare mass parameter. But an effective mass is present in the following "weak coupling" multigrid updating scheme for gauge fields in 4 dimensions, with gauge group G, cp. [9]. Suppose one updates only gauge fields attached to links which point in one selected direction - call it the vertical direction. One can regard these variables as spins, and perform updatings like (2.7), with $\mathcal{A}$ that are smooth in an appropriate sense - cp. later. Only variables residing in the same horizontal 3-dimensional sublattice are actually coupled, therefore one effectively does updatings in 3-dimensional Higgs-models, with action $-\beta s(z)\Delta^{\perp}s(z)$. The covariant Laplacian $\Delta^{\perp}$ depends on a 3-dimensional $G \times G$-gauge field that is determined by the gauge field variables of

[a] In most simulations, it was actually chosen at random.

the model which are attached to horizontal links. The lowest eigenvalue of $\Delta^{\perp}$ has dimension mass squared, and is strictly positive for generic gauge fields. It is determined by variables that are not updated and is therefore like a parameter.

Experience with ϕ^4-theory suggests that such a "weak coupling" approach might nevertheless lead to a substantial acceleration in practise. This is under investigation.

3. Smoothness in Disordered Systems

From section 2 we learn that a successful multigrid scheme, whether deterministic or stochastic, needs smooth interpolation kernels $\mathcal{A}$. This raises the basic

Question: *What is a smooth function in the disordered situation, for instance in an external gauge field?*

Naive smoothness of a function ξ means that $\sum_\mu (\nabla_\mu \xi, \nabla_\mu \xi) \ll (\xi, \xi)$, where ∇_μ are discretized ordinary derivatives. But this is not gauge covariant. A tentative remedy would be to take the covariant derivative ∇_μ. But

$$\sum_\mu (\nabla_\mu \xi, \nabla_\mu \xi) = (\xi, -\Delta \xi) \geq \varepsilon_0 (\xi, \xi) \ . \tag{3.1}$$

The lowest eigenvalue ε_0 of the negative covariant Laplacian $-\Delta$ is a measure for the disorder of the gauge field. (It is positive and vanishes only for pure gauges.) By definition, it is not small for disordered gauge fields. Therefore there are no smooth functions in this case.

Nevertheless there is an answer to the question, assuming a fundamental differential operator D_0 is specified by the problem. In the stochastic case, the Hamiltonian often provides D_0.

Answer: *A function ξ on Λ^0 is smooth on length scale a when*

$$\|D_0 \xi\|^2 \ll \|\xi\|^2 \ \text{ in units } a = 1 \ . \tag{3.2}$$

We found that a deterministic multigrid which employs interpolation kernels $\mathcal{A}^{[0j]}$ from Λ^j to the fundamental lattice Λ^0 which are smooth in this sense, works for arbitrarily disordered gauge fields - see later.

When there are no smooth functions in this sense at length scale a_0, then D_0 has no low eigenvalue, and there is no critical slowing down and no need for a multigrid.

The above answer appears natural, and the "projective multigrid" of [10, 11] is in its spirit. But there are subtleties, and there is the question of how to obtain kernels $\mathcal{A}^{[0j]}_{zx}$ which are smooth *on length scale a_j*.

Problem: *One needs approximate solutions of eigenvalue equations*

$$D_0 \mathcal{A}^{[0j]}_{zx} = \varepsilon_0(x) \mathcal{A}^{[0j]}_{zx} \ . \tag{3.3}$$

x is fixed, and D_0 acts on z. Since $\mathcal{A}^{[0j]}$ is required to vanish for z outside a neighbourhood of x, the problem involves Dirichlet boundary conditions.

For large j, $\mathcal{A}^{[0j]}$ will have a large support. If there is no degeneracy in the lowest eigenvalue, one can use inverse iteration combined with standard relaxation algorithms for the resulting inhomogeneous equation[b]. But this and other standard methods will suffer from critical slowing down again.

Moreover, in the standard multigrid setup, one uses basic interpolation kernels $\mathcal{A}^j$ which interpolate from one grid Λ^j to the next finer one. In this case

$$\mathcal{A}^{[0j]} = \mathcal{A}^1 \mathcal{A}^2 ... \mathcal{A}^j \ , \tag{3.4}$$

and (3.3) becomes a very complicated set of nonlinear conditions.

Possible solutions are

---

[b] Basically one computes an approximation to $D_0^{-n} \mathcal{A}^{[0j]}_{start}$.

(i) Replace (3.3) by minimality of a cost functional (cp. later). Use neural algorithms to find kernels $\mathcal{A}^j$ which minimize it. This is still under study.

(ii) Give up factorization (3.4) and determine independent kernels $\mathcal{A}^{[0j]}$ as solutions of (3.3) by multigrid iteration. This is done successively for $j = 1, 2, \ldots$ One uses already determined kernels $\mathcal{A}^{[0k]}$ with $k < j$ for updating $\mathcal{A}^{[0j]}$. We found that this "unigrid"-method works very well - cp. later.

Method (ii) is not quite as efficient as standard multigrid methods (MG) for ordered systems because of the overhead for storing and computing the kernels. Assuming convergence as expected, algorithms compare as follows for a lattice Λ^0 of L^d sites in d dimensions. (The overhead is included):

Algorithm	work (flops)	storage space
local relaxation	$L^{d+z}(z \approx 2)$	L^d
MG ["ordered"]	L^d	L^d
MG ["disordered", (i)]	???	L^d
MG ["disordered", (ii)]	$L^d \ln^2 L$	$L^d \ln L$

4. Criteria for Optimality

We consider iterative solution of a discretized partial differential equation (2.1). Any iteration amounts to updating steps of the form

$$\tilde{\xi}^0 \mapsto \tilde{\xi}^{0\prime} = \varrho\ \tilde{\xi}^0 + \sigma\ f^0\ . \tag{4.1}$$

ϱ is called the iteration matrix, and $\sigma = (1 - \varrho)D_0^{-1}$. The convergence is governed by the norm $\|\varrho\|$ of the iteration matrix. The iteration converges if $\|\varrho\| < 1$, and its relaxation time is

$$\tau \leq -\frac{1}{\ln \|\varrho\|}\ . \tag{4.2}$$

Parameters in the algorithm - such as interpolation kernels $\mathcal{A}^j_{zx}$, restriction operators C^j_{xz} and effective differential operators D_j - are optimal if the cost functional $E = \|\varrho\|^2$ is at its minimum.

Example: Consider a twogrid iteration in which a standard relaxation sweep on Λ^0 with iteration matrix ϱ_0 is followed by exact solution of the coarse grid equation (2.4). The second step leads to an updating with some iteration matrix ϱ_1, and $\varrho = \varrho_0\varrho_1$. Therefore one may estimate $E \leq \|D_0\ \varrho_0\|^2\ E_1$ with $E_1 = \|D_0^{-1}\ \varrho_1\|^2$. This form of the estimate [4] is motivated by the fact that the fine grid relaxation smoothens the error, but does not converge fast. Therefore $\|D_0\ \varrho_0\|$ is suppressed, whereas $\|\varrho_0\|$ is not much smaller than 1.

Only E_1 depends on the interpolation kernels etc. Therefore one can try to optimize these parameters by minimizing E_1.

Using the trace norm, $\|\varrho\|^2 = \mathrm{tr}\ \varrho\varrho^*$, one finds

$$E_1 = \mathrm{Volume}^{-1} \sum_{z,w\in\Lambda^0} |\Gamma_{zw}|^2 \quad \text{with} \quad \Gamma = D_0^{-1} - \mathcal{A}^1\ D_1^{-1}\ C^1\ . \tag{4.3}$$

Prescribing C^1, and determining D_1 and $\mathcal{A}^1$ by minimizing E_1 yields what we call the "ideal interpolation kernel" $\mathcal{A}^1$ for a given restriction map C^1. Kalkreuter did twogrid iterations for (2.1)

Figure 5.1: A feed-forward neural multigrid architecture (selected connections).

in 4 dimensions with D_0 as shown in section 6 below, using ideal interpolation kernels [12]. He found absence of critical slowing down for arbitrarily disordered $SU(2)$-gauge fields. This showed for the first time that multigrid could work in principle for arbitrarily strong disorder. The ideal kernel $\mathcal{A}^1_{zx}$ is impractical for production runs, though. This is because it has exponential tails instead of vanishing for z outside a neighbourhood of x.

In the analytic multigrid approach to Euclidean quantum field theory [1], Γ is known as the fluctuation field or high frequency field propagator. It has an infrared cutoff which makes it decay exponentially with distance $|z-w|$. Typically, the stronger the decay, the smaller E_1.

5. Neural Multigrid

A feed-forward neural network [13] can perform the computations to solve (2.2) by multigrid relaxation. The nodes of the network ("neurons") are identified with points of the multigrid. There are two copies of the multigrid, except that the last layer Λ^N is not duplicated. In the standard multigrid approach, the basic interpolation kernels $\mathcal{A}^j$ interpolate from one layer Λ^j of the multigrid to the preceding one, Λ^{j-1}. In this case the network looks like in fig. 5.1. Each node is connected to some of the nodes in the preceding layer in the neural network. In the upper half, the connection strength from $x \in \Lambda^j$ to $z \in \Lambda^{j-1}$ is $\mathcal{A}^j_{zx}$. In the lower half, node $z \in \Lambda^{j-1}$ is connected to $x \in \Lambda^j$ with strength R^j_{xz}. In addition there is a connection of strength $\omega_j d^{-1}_{j,x}$ ($d_{j,x} \equiv (D_j)_{xx}$, $\omega_j \equiv$ damping parameter on Λ^j) between the two nodes which represent the same point z in Λ^j, $j < N$. These connections model the synapses in a brain. According to Hebb's hypothesis of synaptical learning, the brain learns by adjusting the strength of its synaptical connections.

The network receives as input an approximate solution ξ of (2.1), from which the residual $r^0 = f^0 - D_0\xi$ is then determined. It computes as output an improved solution $O = \xi + \delta\xi$. The desired output ("target") is $\zeta = D_0^{-1} f^0$. $\delta\xi$ is a linear function of r^0.

Except on the bottom layer, each neuron receives as input a weighted sum of the output of those neurons below it in the diagram to which it is connected. The weights are given by the connection strengths. Our neurons are linear because our problem is linear. The output of each neuron is a linear function of the input. (One may take output = input).

The result of the computation is

$$\delta\xi = (\omega_0\ d_0^{-1} + \sum_{k\geq 1} \mathcal{A}^{[0k]}\ \omega_k\ d_k^{-1}\ R^{[k0]})\ r^0 \tag{5.1}$$

where $R^{[k0]} = R^k ... R^2 R^1$ and $\mathcal{A}^{[0k]} = \mathcal{A}^1 \mathcal{A}^2 ... \mathcal{A}^k$.

In principle, R^j is determined by the restriction operators and by the effective differential operator, $R^j = C^j(1 - \omega_{j-1} D_{j-1} d_{j-1}^{-1})$. But actually, the restriction kernels C^j, effective differential operators D_j $(j > 0)$ and their diagonal part d_j, and the damping parameters ω_j $(j > 0)$ enter only in the combination R^j (assuming $\omega_j d_{j,x}^{-1}$ can be scaled to 1). They are therefore not needed separately.

The fundamental differential operator D_0 and its diagonal part d_0 are furnished as part of the problem. The connection strengths ("synaptical strengths") $\mathcal{A}^j_{zx}$, R^j_{xz} (and possibly the damping factor ω_0) need to be found by a learning process in such a way that the actual output is as close as possible to the desired output.

In supervised learning of a neural network [13], a sequence of pairs (ξ^μ, ζ^μ) is presented to the network. Given input ξ^μ, the actual output O^μ is compared to the target ζ^μ, and the connection strengths are adjusted in such a way that the cost functional

$$E = \sum_\mu \|O^\mu - \zeta^\mu\|^2$$

gets minimized. If an iterative procedure to achieve this minimization is specified, one calls this a *learning rule.*

Because of linearity, it suffices to consider the equation $D_0\xi = f^0$ in the limit of small f^0. In the limit $f^0 \mapsto 0$, the target $\zeta^\mu = 0$ for any input, and $O^\mu = \varrho\ \xi^\mu$ by (4.1), with $\varrho =$ iteration matrix.

Taking for the sequence ξ^μ a complete orthonormal system of functions on Λ^0,

$$E = \sum_\mu \|\varrho\ \xi^\mu\|^2 = \mathrm{tr}\ \varrho\varrho_* \equiv \|\varrho\|^2\ .$$

$E =$ minimum is the previous optimality condition of section 4 for multigrid relaxation.

Conclusion: *Optimizing the kernels $\mathcal{A}$ and R is a standard learning problem for a feed forward neural net.*

6. Learning Rule Performance

The second variant, for which a learning rule was already described in section 3, involves a slightly different neural net. Instead of the connections between neighbouring layers of the multigrid, we have now connections from Λ^0 to Λ^k with connection strength $C^{[k0]}_{xz}$, and from Λ^j to Λ^0 with connection strength $\mathcal{A}^{[0j]}_{zx}$. We adopt variational coarsening, $C^{[k0]} = \mathcal{A}^{[0k]*}$. Then all connection strengths are determined by interpolation kernels $\mathcal{A}^{[0k]}$ which have to be learned. The damping factors ω_k were set to 1, and d_k is the diagonal part of D_k as before, with

$$D_k = \mathcal{A}^{[0k]*} D_0 \mathcal{A}^{[0k]}\ . \tag{6.1}$$

The learning rule of section 3 requires a process of "hard thinking" by the neural net. Neurons which have learned their lesson already - i.e. which have their synaptical strengths fixed - are used to instruct the rest of the neural net, adjusting the synaptical strengths of the next multigrid layer of neurons.

A checkerboard variant of this algorithm was tested in 2 dimensions, using SU(2)-gauge fields which were equilibrated with standard Wilson action at various values of β, and $D_0 = -\Delta - \varepsilon_0 + \delta m^2$. ε_0 is the lowest eigenvalue of the covariant Laplacian $-\Delta$, and $\delta m^2 > 0$.

Figure 6.1: Correlation time τ as function of the lowest eigenvalue δm^2 in a representative gauge field configuration equilibrated at $\beta = 1.0$. For the 64^2 lattice, the correlation time fluctutates very little with the gauge field configuration.

Conventional relaxation algorithms for solving (2.1) suffer from critical slowing down for such D_0, for any volume and small δm^2.

It turned out that it was not necessary to find accurate solutions of the eigenvalue equation for the interpolation kernels $\mathcal{A}^{[0j]}$. An approximation $\mathcal{A}^{[0j]}_{zx}$ to $(-\Delta)^{-n}\delta_{zx}$ was computed. A total of four relaxation sweeps through each multigrid layer below j ($n = 2$ inverse iteration steps à one V-cycle each) was enough. The convergence rate of the following ξ-iteration is shown in fig. 6.1 for $\beta = 1.0$. The correlation time τ is in units of MG-iterations. One MG-iteration involved one sweep through each multigrid layer, starting with $j = 0$. Updating ξ at $x \in \Lambda^j$ changes ξ by

$$\delta\xi_z = \mathcal{A}^{[0j]}_{zx} d^{-1}_{j,x} r^j_x \ , \quad r^j = \mathcal{A}^{[0j]*} r^0 .$$

Sweeps were actually performed in checkerboard fashion.

Acknowledgements

We thank the HLRZ Jülich for computer time and assistance.

Support by Deutsche Forschungsgemeinschaft is gratefully acknowledged.

References

[1] G. Mack and M. Pordt, *Commun. Math. Phys.* **97** (1985) 267
G. Mack in *Nonperturbative quantum field theory*, G. 't Hooft et al (eds) (Plenum, New York 1989)

[2] J. Goodman and A.D. Sokal, *Phys. Rev. D* **40** (1989) 2035

[3] A. Brandt, *Guide to multigrid development*, Proceedings of the 1st European Multigrid conference (1981)

[4] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics 4 (Springer-Verlag, Berlin, 1985)

[5] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes* (Cambridge UP, Cambridge, 1989)

[6] M. Hasenbusch, G. Mack and S. Meyer, *Nucl. Phys. B (Proc. Suppl.)* **20** (1991) 110

[7] S. Meyer, M. Hasenbusch, *Phys. Rev. Lett. D* **45** (1992) 4376

[8] U. Wolff, preprint CERN-TH 6636/92, proc. Physics Computing '92, Prag, Czecheslovakia, August 24-28, 1992

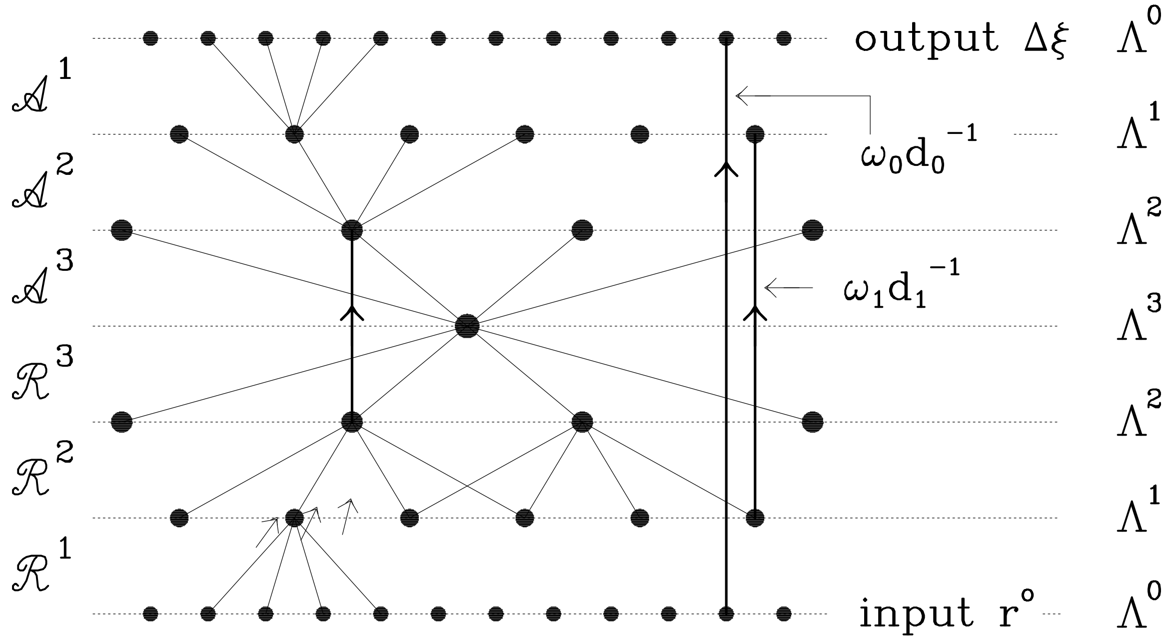
[9] M. Grabenstein and K. Pinn, *Phys. Rev. D* **45** (1992) 4372 and DESY 92-094

[10] R.C. Brower, K. Moriarty, C. Rebbi and E. Vicari, *Nucl. Phys.B* **20** (1991) 89

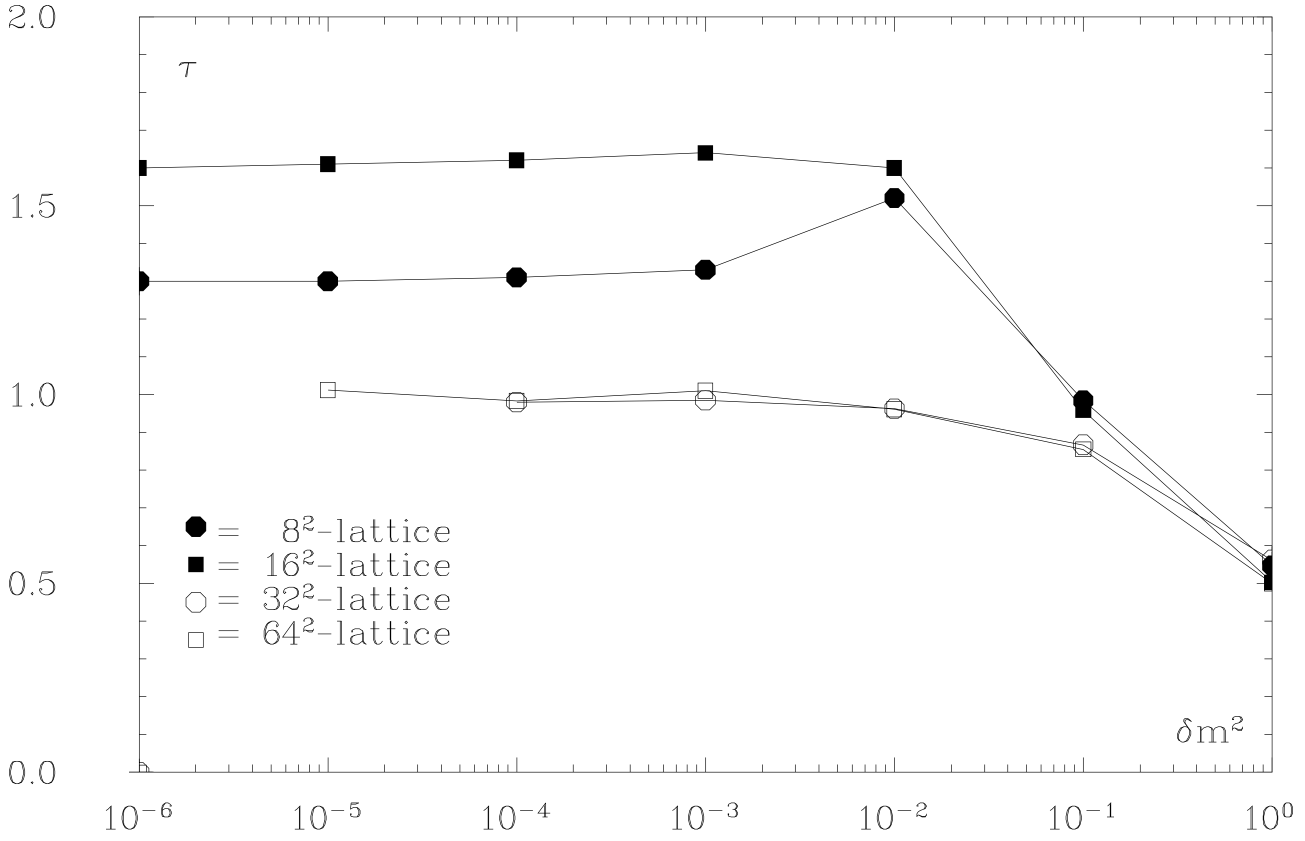
[11] A. Hulsebos, J. Smit and J.C. Vink, *Nucl. Phys. B* **368** (1992) 379

[12] T. Kalkreuter, *Phys. Lett. B* **276** (1992) 485

[13] J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City CA, 1991)



output $\Delta\xi$
Λ^0
$\mathscr{A}^1$
$\omega_0 d_0^{-1}$
Λ^1
$\mathscr{A}^2$
Λ^2
$\mathscr{A}^3$
$\omega_1 d_1^{-1}$
Λ^3
$\mathscr{R}^3$
Λ^2
$\mathscr{R}^2$
Λ^1
$\mathscr{R}^1$
input r°
Λ^0



2.0
1.5
1.0
0.5
0.0
τ
δm²
10⁻⁶ 10⁻⁵ 10⁻⁴ 10⁻³ 10⁻² 10⁻¹ 10⁰
= 8²-lattice
= 16²-lattice
= 32²-lattice
= 64²-lattice