World Records in the Size of Simulated Ising Models

Dietrich Stauffer
Institute for Theoretical Physics,
Cologne University, D-50923 Köln, Euroland
e-mail: stauffer@thp.uni-koeln.de

Received on 15 July, 2000

Monte Carlo simulations with up to 176^5 spins confirm the Chen-Dohm theory for the five-dimensional Ising model. Also world record sizes 1000192^2, 9984^3, 880^4, 48^6 and 21^7 spins were simulated in the literature, and we describe the needed multi-spin coding algorithm.

Brazil does good Ising research, but Ising was not born there. Thus in contrast this paper reports on some work done in the town where Ernst Ising was born a century ago: Monte Carlo simulations of large Ising models in d dimensions: nearest-neighbour spin 1/2 model without any further quantum effects, periodic or helical boundaries on hypercubical lattices of L^d spins.

It is a waste of memory to store every spin of an Ising model in a full computer word of, say, 32 bits. Friedberg and Cameron [1] seem to have started putting spins into single bits, Claudio Rebbi [2] made the technique popular as multi-spin coding, but in biology such bitstrings were known from Eigen's quasispecies model [3], also of 1972. Since in three dimensions we have to sum up over six neighbours, giving zero to six antiparallel spins, the energy requires three bits. In more than three dimensions, more bits are needed. Here we use four bits per spin, allowing simulations in up to seven dimensions and fitting nicely into modern 32- or 64-bit words. (Rebbi used 3 bits on 60-bit words.) I do not deal here with the more efficient but less versatile method of using one bit per spin and doing the sum over the nearest neighbours through purely logical operations. Instead, the program allows for varying the dimensionality d by just changing one parameter. Its principles were already explained long ago [4, 5].

Let us assume we already have a working three-dimensional Ising program with multi-spin coding using 4 bits per spin, on a supercomputer like the Cray-T3E with 64 bits per word and thus 16 spins per word. The spin words are called $IS(ii,i)$ with ii going from 1 to $LL = L/16$ and $i = 1 \ldots L^{d-1}$ in d dimensions. To add a fourth or fifth dimension we just have to add two or four neighbours. If we do this with an innermost loop over the 6 (or $2d$) neighbours then vector computing is slowed down. If we deal with traditional if-commands then both vector and scalar computers are slowed down. The trick is to use if-commands depending on a constant dimension $idim$ defined at the beginning through a Fortran PARAMETER statement (or analogous constructs in lesser languages). Then a statement like $if(idim.eq.3)$ *goto* 8 is evaluated already at compile time, the compiler might even give you a warning that the following line is never reached, and after compilation the executation is no longer slowed down by the numerous if-statements.

Of course, it is still a waste of memory to have four bits per spin when one suffices. A way out is compression: only the current hyperplane and the two neighbouring hyperplanes are stored as IS with four bits per spin, while for all other planes four words are compressed into one after shifting them by zero, one, two or three bit positions. They can be re-read from there by logical-AND statements with some $MASK$ of 0001 sequences, called $MK(0), \ldots MK(3)$ here. Normally this would reduce the main spin array $ISM(ii,i)$ to $ii = 1 \ldots L/64$, $i = 1 \ldots L^{d-1}$.

However, in this case the minimum linear dimension would be $L = 128$ since at least two words are needed for each lattice line in multi-spin coding. This would rectrict the applications of the program very much. Therefore, this 4:1 compression was instead performed for the second index i, thus requiring a main spin array $ISM(ii,i)$ with $ii = 1 \ldots L/16$, $i = 1 \ldots L^{d-1}/4$. Unfortunately, this makes programming complicated, involving the variables $I0, IM, IP, INDEX, INDX$ after the loop DO 4 in my program. And I cannot use it for two dimensions anymore, where compression was made in the ii-direction instead. (In case of need, ask stauffer@thp.uni-koeln.de for a simpler program ispardd9.f in two to seven dimensions without compression; the complicated one is ispardd0.f.) More than twenty spins were updated every microsecond on every processor of a fast Cray-T3E in Jülich, for five dimensions.)

[Technical remarks: This program was written for parallel computing with $NPROC$ processors by dividing the hypercube into $NPROC$ layers of thick-

ness $L/NPROC \geq 2$. The message-passing command $shmem\_get(a,b,c,d)$ gets c words from processor d, starting with word b there, and stores them into the memory of the current processor $NODE = shmem\_my\_pe$ beginning at word a. The line $INFO = BARRIER()$ synchronizes the processors: computation proceeds after all nodes have reached this line. $POPCNT$ counts the number of set bits in a word, $IRTC$ the number of machine cycles, and the final loop $do\ iadd = 1, nproc - 1$ gives a primitive global summation over the magnetization in every node. We use the Heat Bath algorithm.]

```
      PARAMETER(TTC = 0.7, NPROC=88, LL=11, IDIM=5, max=    5,
     1 L=LL*16,L2=L*L,L3=L2*L,L4=L3*L,L5=L4*L,N=L**(IDIM-1)/4,
     2 NPLANE=N/L,  LLP=LL-1, N8=N/NPROC, IMAX=N8+2*NPLANE,
     3 NPLAN4=NPLANE*4, ID2=2*IDIM, LENGTH=N/16)
      DIMENSION IS(LL,3),ISM(LL,IMAX),IEX(0:ID2),MK(0:3)
      COMMON /T3E/ ISM,M,MSUM
      INTEGER SHMEM_N_PES, SHMEM_MY_PE
      ISII(K)=ISHFT(IAND(MASK,ISM(II,1+K/4)),-INDEX)
      DATA ISEED/   2/,MSK/X'1000000000000000'/,
     1       MK/X'1111111111111111',X'2222222222222222',
     2          X'4444444444444444',X'8888888888888888'/
      NODE=SHMEM_MY_PE()
      NUMBER=SHMEM_N_PES()
      IF(IDIM.EQ.3) T=0.2216544
      IF(IDIM.EQ.4) T=0.149695
      IF(IDIM.EQ.5) T=0.113915
      IF(IDIM.EQ.6) T=0.092295
      IF(IDIM.EQ.7) T=0.077706
      T=T/TTC
      IF(NODE.EQ.0) PRINT 11, TTC,IDIM,L,MAX,ISEED,NUMBER,
     1              (LL*IMAX)/131072
      LNPROC=L/NPROC
 11   FORMAT(1X,F9.6,6I5)
      NODEU=NODE-1
      NODED=NODE+1
      IF(NODE.EQ.0) NODEU=NUMBER-1
      IF(NODE.EQ.NUMBER-1) NODED=0
      IBM=2*ISEED-1
      MULT=16807
      DO ITER=0,NODE
        IBM=IBM*65539
      END DO
      DO 2 I=0,ID2
      EX=EXP((4*I-2*ID2)*T)
 2    IEX(I)=2147483648.0*(4.0*EX/(1.0+EX)-2.0)*2147483648.0
      DO 1 I =1,IMAX
           DO 1 II=1,LL
 1      ISM(II,I)=0
      TIME=IRTC()*3.33E-9
      DO 6 ITIME=1,MAX
        INFO = BARRIER()
        CALL SHMEM_GET(ISM(1,1),ISM(1,1+N8),LENGTH,NODEU)
        INFO = BARRIER()
        DO 7 II=1,LL
          IS(II,2)=ISHFT(IAND(MK(3),ISM(II,NPLANE)),-3)
 7        IS(II,3)=      IAND(MK(0),ISM(II,NPLANE+1))
        DO 49 IPLANE=1,LNPROC
        LPLANE=IPLANE*NPLAN4-1
        IF(IPLANE.NE.2) GOTO 9
```

```
          INFO = BARRIER()
          CALL SHMEM_GET(ISM(1,1+NPLANE+N8),ISM(1,NPLANE+1)
     1                ,LENGTH,NODED)
          INFO = BARRIER()
 9      CONTINUE
        DO 4 I=1,NPLAN4
        I0=I+LPLANE
        IM=1 +  I0 /4
        IP=1+(I0+1)/4
        INDEX=MOD(I-1,4)
        INDX =MOD(I  ,4)
        MASK=MK(INDEX)
        NOTMSK=NOT(MASK)
        DO 10 II=1,LL
          IS(II,1)=IS(II,2)
          IS(II,2)=IS(II,3)
 10       IS(II,3)=ISHFT(IAND(MK(INDX),ISM(II,IP)),-INDX)
        DO 4 II=1,LL
          IF(II.EQ. 1) GOTO 40
          IF(II.EQ.LL) GOTO 41
          IEN=IS(II-1,2)+IS(II+1,2)
          GOTO 42
 40       IEN=IS(2,2)+IOR(ISHFT(IS(LL,2),-4),ISHFT(IS(LL,2),60))
          GOTO 42
 41       IEN=IS(LLP,2)+IOR(ISHFT(IS(1,2),4),ISHFT(IS(1,2),-60))
 42       IEN=IEN+IS(II,1)+IS(II,3)+ISII(I0-L)+ISII(I0+L)
          IF(IDIM.EQ.3) GOTO 8
          IEN=IEN+ISII(I0-L2)+ISII(I0+L2)
          IF(IDIM.EQ.4) GOTO 8
          IEN=IEN+ISII(I0-L3)+ISII(I0+L3)
          IF(IDIM.EQ.5) GOTO 8
          IEN=IEN+ISII(I0-L4)+ISII(I0+L4)
          IF(IDIM.EQ.6) GOTO 8
          IEN=IEN+ISII(I0-L5)+ISII(I0+L5)
 8        ICI=0
          DO 3 IB=1,16
            IBM=IBM*MULT
            ICI=ISHFT(ICI,-4)
            IF(IBM.LT.IEX(IAND(15,IEN))) ICI=IOR(ICI,MSK)
 3          IEN=ISHFT(IEN,-4)
          IS(II,2)=ICI
 4        ISM(II,IM)=IOR(IAND(NOTMSK,ISM(II,IM)),ISHFT(ICI,INDEX))
 49   CONTINUE
c     IF(ITIME.NE.(ITIME/10)*10) GOTO 6
      M=0
      DO 5 I =NPLANE+1,NPLANE+N8
      DO 5 II=1,LL
 5    M = M + POPCNT(ISM(II,I))
       INFO = BARRIER()
       IF (NODE .EQ. 0 .AND. NPROC .GT. 1) THEN
         DO IADD = 1,NPROC-1
           CALL SHMEM_GET (MSUM, M, 1, IADD)
           M = M + MSUM
         ENDDO
       ENDIF
      X=1.0-(M*0.5)/(N*L)
      IF(NODE.EQ.0) PRINT 100, ITIME,M,X
```

```
6     CONTINUE
      TIME = IRTC() * 3.33E-9 -TIME
      IF(NODE.EQ.0)PRINT *,4*TIME,NUMBER,
     1   (MAX*L**IDIM/NPROC)/(4000000.0*TIME)
100   FORMAT(1x,I4,I20,F12.9)
      END
```

What can we do with this program? As the title promises it allows to simulate world record sizes, with $L = 10^6$ in two dimensions, matching Linke et al. [6], $L = 10^4$ in three dimensions equalizing percolation [7], and 880^4, 176^5 and 48^6 spins in higher dimensions. These world records were established at the same time as the prediction of Ceperley [8] was published, that 10^{12} sites could be simulated. (Since we need at least two hyperplanes per processor, the 512 nodes of the Cray-T3E cannot all be used for higher dimensions. Fortunately, in physics there is little need for more than five dimensions.) Some of the results in lower dimensions were given before [9] and are compatible with those from smaller systems [10]. Now come some new five-dimensional simulations.

According to Chen and Dohm [11], the behavior in five dimensions, above the upper critical dimension of four, is far from trivial: Some finite-size field theories are invalid, and finite-size scaling is not of the usual one-parameter type. This work was criticized [12] as being in contradiction to Monte Carlo simulations, and that criticism in turn was recently refuted [13]. A comparison of field theory, ϕ^4 lattice theory, and the true Ising model requires to fit several parameters since e.g. the critical temperature is not the same. Ref.[12] set one of these parameters to unity, while Ref. [13] allowed it to be different from one. This latter choice removed [13] some discrepancy between the theoretical prediction [11] and Monte Carlo simulation [12]. As a result, several bulk amplitudes differ depending on whether we use the fit of Mainz[12] or Aachen[13], and we now perform new simulations to find out which of these two predictions, [12] or [13], fits our new data better. (The parameters appearing in the theory of [11, 13] have been identified as *bulk* parameters related to the amplitudes of the bulk correlation length, the bulk susceptibility and the bulk magnetization. Thus the values of their parameters can be tested by bulk simulations.)

First, we determined the correlation length ξ, defined through the exponential decay of the correlation function, by observing the magnetization profile between two hyperplanes of up spins, fitting an exponential on an intermediate space region somewhat removed from the surface but where the magnetization still differs significantly from its bulk value, Fig.1. We used one-word-per spin coding for typically 35^5 spins above T_c in zero magnetic field and multi-spin coding (4 bits per spin) for 48^5, 80^5, 112^5 at $T = T_c$ in a magnetic field $h = \mu B/k_B T$, where μ is the magnetic dipole moment. Typically, 2000 Monte Carlo steps per spin were simulated. Fig. 2 shows the data for $T > T_c$ in zero field, Fig.3 the case $T = T_c$ in a positive field. In spite of the larger lattices, the results in a magnetic field are less accurate since then, as opposed to paramagnets, the bulk magnetization is nonzero and has to be subtracted from the magnetization profile. We thus estimate $\xi h^{1/3} = 0.27 \pm 0.03$ and $\xi[(T - T_c)/T_c]^{1/2} = 0.36 \pm 0.02$, in better agreement with the Aachen predictions (0.273 and 0.396) than the Mainz estimates (0.390 and 0.549).

The magnetization M can be determined more accurately since no fit to an exponential profile is involved. Fig. 4 shows at $T = T_c$ the amplitude $M/h^{1/3}$ and Fig.5 shows $M/(1 - T/T_c)^{1/2}$ below T_c in zero field. Again the lower horizontal line is the Aachen prediction, which agrees better with the data than the upper horizontal line from Mainz. Moroever, Fig.5 using up to 176^5 spins shows, as predicted by analytic theory [13], a scaling correction $2.25 - 1.1 \cdot (1 - T/T_c)^{1/2}$ which fits better than the correction linear in $T - T_c$ suggested by Cheon et al [11].

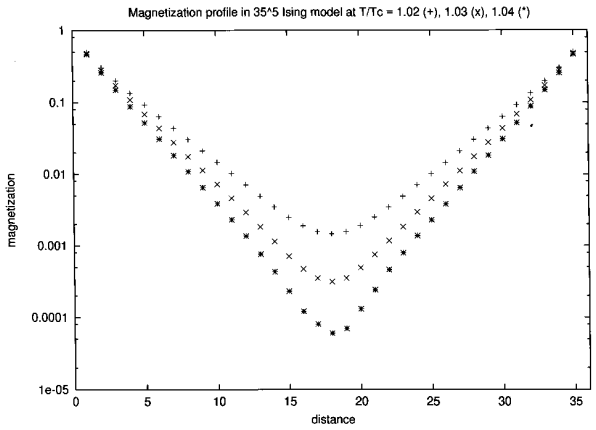

Figure 1. Examples of the magnetization profile. Straight lines in this semilogarithmic plot determine $1/\xi$. Two boundary hyperplanes had all spins up to produce these profiles.

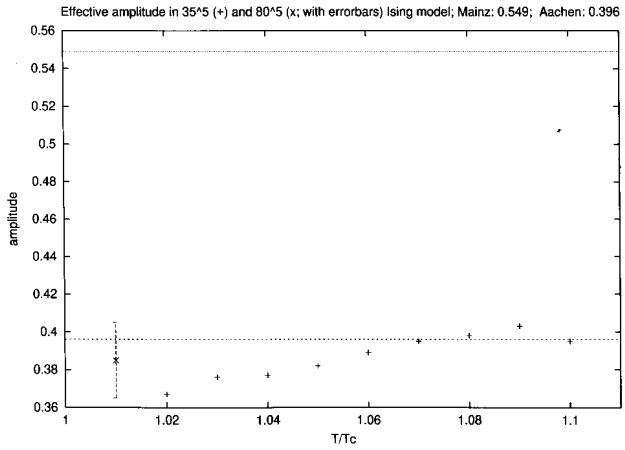

Figure 2. Correlation length amplitudes above T_c in zero field.

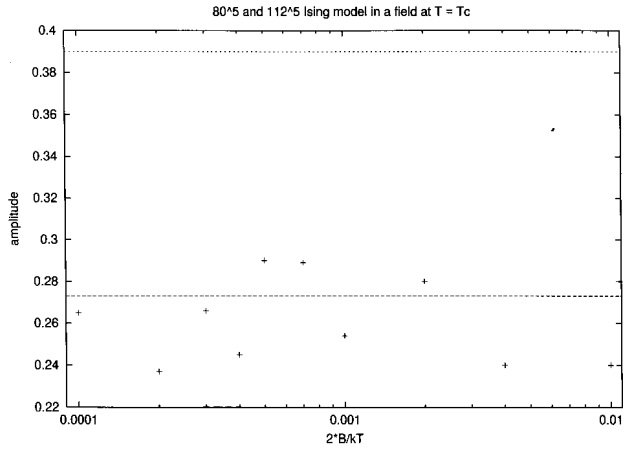

Figure 3. Correlation length amplitudes at T_c in positive field.

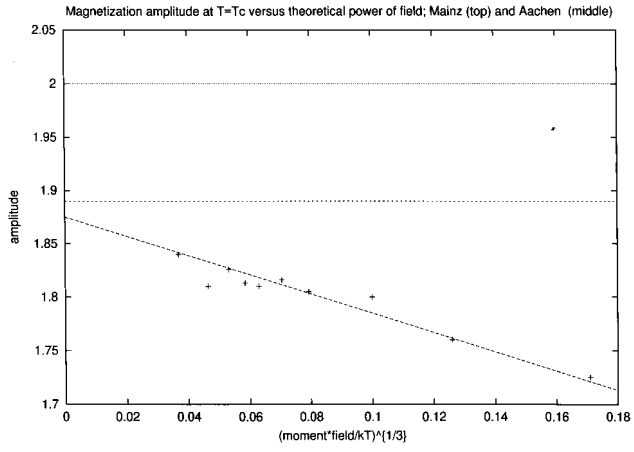

Figure 4. Magnetization amplitudes at T_c in positive field.

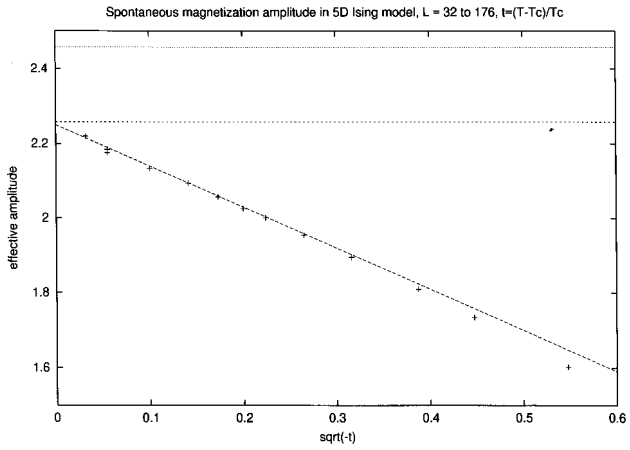

Figure 5. Magnetization amplitudes below T_c in zero field with $t = T/Tc - 1$; the straight line with negative slope suggests scaling corrections $\propto \sqrt{-t}$.

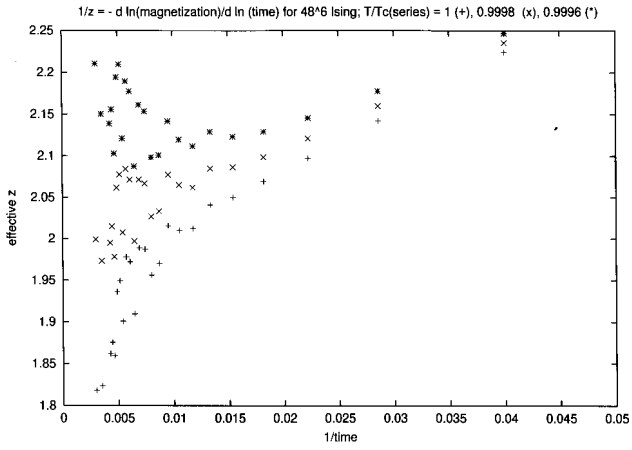

Figure 6. Six-dimensional dynamics. The correct critical temperature requires $z = 2$ asymptotically. Our slight deviation $T/T_c = 0.9998 \pm 0.0001$ may come from the finite $L = 48$.

Thus in all cases the Aachen prediction fits better than the Mainz alternative, and only Fig.2 suggests that Aachen is only approximate.

[Two side remarks: (i) In six and seven dimensions, Figs. 6 and 7 confirm the series estimate $J/k_BT_c = 0.092295(3)$ and $0.077706(2)$ less accurately [16]: For the correct temperature the intercept of the extrapolation gives the asymptotic kinetic exponent $z = 2$. The 21^7 spins were simulated without multi-spin coding. For Swendsen-Wang cluster flips, Fig.8 shows nicely the exponential decay of the magnetization with time in two dimensions: $z = 0$. Three-dimensional Q2R cellular automata still have an unexplained dynamics, $z = 3.37$, Fig.9. Fig.10 shows results for Metropolis kinetics with Ito's one-bit-per-spin and parallelization in stripes and 10^{10} updates per second. These system sizes can be compared with the present site percolation records [7]

of $L = 4$ million, 10000, 611, 127, 49, and 26 for two to seven dimensions.

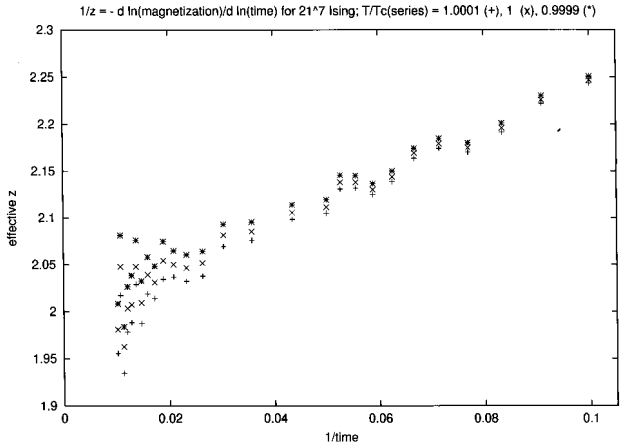

Figure 7. Seven-dimensional dynamics analogous to Fig.6, with 21^7 spins.

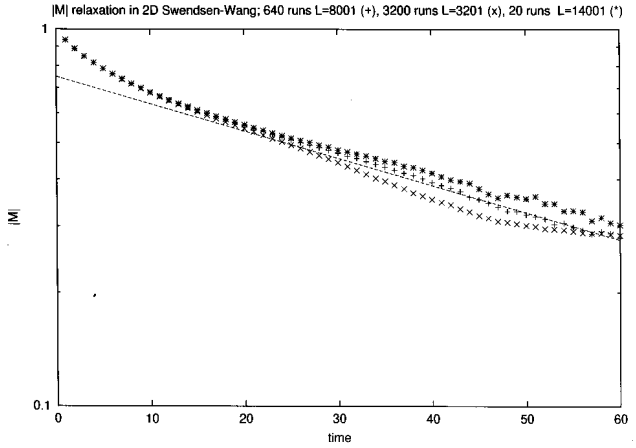

Figure 8. Two-dimensional Swendsen-Wang dynamics[18, 19] for L up to 14001.

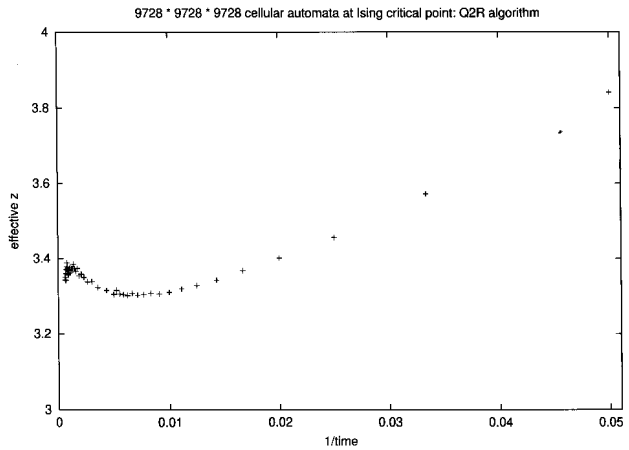

Figure 9. Three-dimensional Q2R cellular automata with 10^{12} spins at criticality.

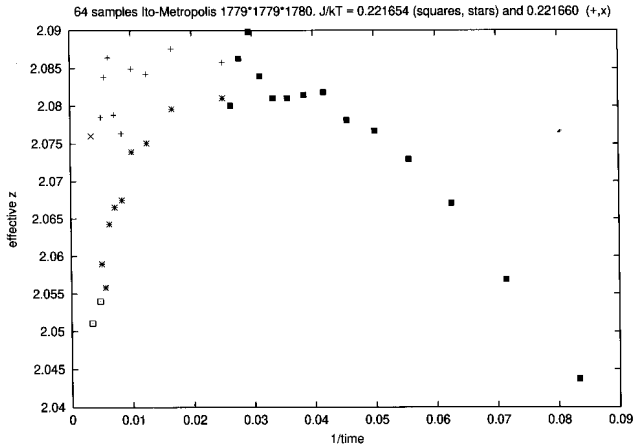

Figure 10. Three-dimensional Ito-Metropolis algorithm, suggesting $z \simeq 2.06$ depending on the choice [10] of T_c. The speed was 11 Giga updates per second, and we assume $\beta/\nu = 0.515$.

(ii) At the last conference in Belo Horizonte in which I took part, the Brazilian computational physics meeting, I learned [14] that simulations of dipole forces are easy. As a result, the stability of arrays of parallel strips of Ising spins interacting also with long-range dipole forces, as for iron monolayers on stepped surfaces [15], was investigated by Michelsen [17]. He found the Curie point to be stable for large systems, and not just a gradual freezing-in as in a glass.]

I thank X.S. Chen and V. Dohm for stimulating discussions and for sending the various 5D amplitude predictions, N. Ito for checking a parallel algorithm, P.M.C. de Oliveira for a critical reading of the manuscript, SFB 341 for partial support, and the German Supercomputer Center Jülich for time on their Cray-T3E.

References

[1] R. Friedberg and J.E. Cameron, J. Chem. Phys. **52**, 6049 (1972)

[2] M. Creutz, L. Jacobs and C. Rebbi, Phys. Rev. Lett. **42**, 1390 (1979)

[3] M. Eigen, J. McCaskill and P. Schuster, Adv. Chem. Phys. **75**, 149 (1990). See also H.J. Muller, Mutat. Res. **1**, 2 (1964).

[4] D. Stauffer, F.W. Hehl, N. Ito, V. Winkelmann and J.G. Zabolitzky, *Computer Simulation and Computer Algebra*, Springer, Heidelberg-Berlin 1993

[5] D. Stauffer, page 5 in: Computer simulation studies in condensed matter physics VI, D. P. Landau, K. K. Mon, H. B. Schüttler, eds., Springer, Heidelberg 1993

[6] A. Linke, D.W. Heermann, P. Altevogt, M. Siegert, Physica A **222**, 205 (1995)

[7] N.Jan and D. Stauffer, Int. J. Mod. Phys. C **9**, 341 (1998); D. Tiggemann and A.B. MacIsaac, priv.comm.

[8] L. Ceperley, Rev. Mod. Phys. **71**, S 438 (1999)

[9] D. Stauffer, Physica A **244**, 344 (1997), Int. J. Mod. Phys.C **10**, 807, 809, 931 (1999)

[10] N. Ito, K. Hukushima, K. Ogawa and Y. Ozeki, J. Phys. Soc. Jpn. **69**, 1931 (2000)

[11] X. S. Chen and V. Dohm, Int. J. Mod. Phys. C **9**, 1007 and 1073 (1998); see also M. Cheon, I. Chang and D. Stauffer, Int. J. Mod. Phys. C **10**, 131 (1999) and Phys. Rev. E, in press (Jan. 2001).

[12] E. Luijten, K. Binder and H. W. J. Blöte, Eur. Phys. J. B **9**, 289 (1999); K. Binder et al., Physica A **281**, 112 (2000).

[13] X. S. Chen and V. Dohm, e-print cond-mat 0005022 and priv.comm.

[14] T.G. Rappoport, T.S. de Menezes, L.C. Sampaio, M.P. Albuquerque and F. Mello, Int. J. Mod. Phys. C **9**, 821 (1998)

[15] P. Sen, D. Stauffer and U. Gradmann, Physica A **245**, 361 (1997)

[16] M. Gofman, J. Adler, A. Aharony, A.B. Harris and D. Stauffer, J. Stat. Phys. **71**, 1221 (1993)

[17] A. Michelsen, Masters Thesis, Hamburg and Cologne, Feb. 2000

[18] J.S. Wang and R.H. Swendsen, Physica A **167**, 565 (1990)

[19] R. Hackl, H.G. Matuttis, J.M. Singer, T. Husslein, and I. Morgenstern, Int. J. Mod. Phys. C **4**, 1117 (1993)