John von Neumann Institute for Computing

Simulating Materials with Strong Correlations on BlueGene/L

Andreas Dolfen, Yuan Lung Luo, Erik Koch

published in

Parallel Computing: Architectures, Algorithms and Applications ,
C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr, F. Peters (Eds.),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **38**, ISBN 978-3-9810843-4-4, pp. 601-608, 2007.
Reprinted in: Advances in Parallel Computing, Volume **15**,
ISSN 0927-5452, ISBN 978-1-58603-796-3 (IOS Press), 2008.

http://www.fz-juelich.de/nic-series/volume38

Simulating Materials with Strong Correlations on BlueGene/L

Andreas Dolfen, Yuan Lung Luo, and Erik Koch

Institut für Festkörperforschung
Forschungszentrum Jülich, 52425 Jülich, Germany
E-mail: {a.dolfen, y.luo, e.koch}@fz-juelich.de

Understanding the physics of strongly correlated materials is one of the grand-challenges in condensed-matter physics. Simple approximations such as the local density approximation fail, due to the importance of the Coulomb repulsion between localized electrons. Instead we have to resort to non-perturbative many-body techniques. Such calculations are, however, only feasible for quite small model systems. This means that the full Hamiltonian of a real material has to be approximated by a model Hamiltonian comprising only the most important electronic degrees of freedom, while the effect of all other electrons can merely be included in an average way. Realistic calculations of strongly correlated materials need to include as many of the electronic degrees of freedom as possible.

We use the Lanczos method for calculating ground state and dynamical properties of such model Hamiltonians. In the method we have to handle the full many-body state of the correlated system. The method is thus limited by the available main memory. The principal problem for a distributed-memory implementation is that the central routine of the code, the application of the Hamiltonian to the many-body state, leads, due to the kinetic energy term, to very non-local memory access. We describe a solution for this problem and show that the new algorithm scales very efficiently on BlueGene/L. Moreover, our approach is not restricted to correlated electrons but can also be used to simulate quantum spin systems relevant for quantum computing.

1 Motivation

Essentially all of condensed matter physics is described by the non-relativistic Schrödinger equation $i\hbar\frac{\partial}{\partial t}|\Psi\rangle = H|\Psi\rangle$, with the Hamiltonian

$$H = -\sum_{\alpha=1}^{N_n}\frac{\vec{P}_\alpha^2}{2M_\alpha} - \sum_{j=1}^{N_e}\frac{\vec{p}_j^{\,2}}{2m} - \sum_{j=1}^{N_e}\sum_{\alpha=1}^{N_n}\frac{Z_\alpha e^2}{|\vec{r}_j - \vec{R}_\alpha|} + \sum_{j<k}^{N_e}\frac{e^2}{|\vec{r}_j - \vec{r}_k|} + \sum_{\alpha<\beta}^{N_n}\frac{Z_\alpha Z_\beta e^2}{|\vec{R}_\alpha - \vec{R}_\beta|}$$

where Z_α is the atomic number, M_α the mass, $\vec{R}_\alpha$ the position and $\vec{P}_\alpha$ the momentum of nucleus α. $\vec{p}_j$ and $\vec{r}_j$ denote the j^{th} electron's momentum and position and N_e, N_n the number of electrons, nuclei respectively. To accurately describe materials of technological interest and design new ones with superior properties, all we have to do is solve this equation. There is, however, a severe problem which makes a brute-force approach to the many-body Schrödinger equation infeasible. To illustrate this, let us for consider a single iron atom. With its 26 electrons the total electronic wave function depends on 26 times 3 spatial coordinates. Thus, even without spin, specifying the electronic wave function on a hypercubic grid with merely 10 points per coordinate, we would have to store 10^{78} numbers. This is impossible in practice: Even if we could store a number in a single hydrogen atom, the required memory would weight 10^{51} kg – far more than our home-galaxy, the milky way.

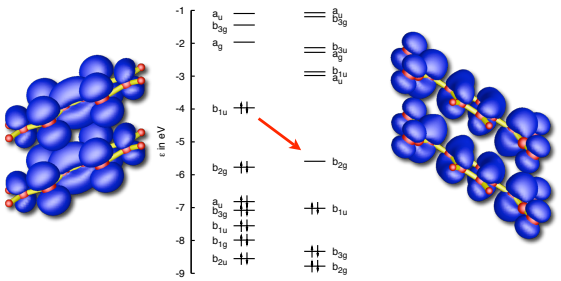

Figure 1. The molecular metal TTF-TCNQ. Centre: molecular levels of the isolated molecules; left: two TTF molecules with the electron density of their highest occupied molecular orbital (HOMO); right: TCNQ with the electron density of the lowest unoccupied molecular orbital (LUMO). The red arrow denotes the charge transfer of 0.6 electrons from the TTF-HOMO to the TCNQ-LUMO.

Still, the quantitative description of solids is not an entirely hopeless enterprise. Even though an exact treatment is a practical impossibility, there are successful approximations that work for wide classes of materials. The most prominent examples are approximations to density functional theory.[1] They effectively map the hard many-body problem to an effective single-particle problem that can be efficiently solved numerically. Essential to these approximations is that the Coulomb repulsion is described on a mean-field level. Such an approximation fails, however, to capture the physics in systems with strong correlations. In these systems the Coulomb repulsion between the electrons is so strong that the motion of a single electron depends on the position of all the others. The electrons thus lose their individuality and the single-electron picture breaks down. To accurately model this, we have to solve the many-electron problem exactly. Clearly we cannot do this for the full Hamiltonian. Instead, we consider a simplified Hamiltonian, which describes only those electrons that are essential to the correlation effects.[2] We illustrate this for the example of TTF-TCNQ, a quasi one-dimensional organic metal.

As shown in Fig. 1, TTF and TCNQ are stable molecules with completely filled molecular orbitals. The highest molecular orbital (HOMO) of TTF is, however, significantly higher in energy than the lowest unoccupied molecular orbtial (LUMO) of TCNQ. Thus in a crystal of TTF and TCNQ, charge is transferred from the TTF-HOMO to the TCNQ-LUMO. This leads to partially filled bands and thus metallic behaviour. In the TTF-TCNQ crystal, like molecules are stacked on top of each other. Electrons can move along these stacks, while hopping between different stacks is extremely weak. Thus the material is quasi one-dimensional.

As pointed out above, we cannot treat all the electrons in the molecular solid. Instead, we focus our efforts on the most important electronic states: the partially filled TTF-HOMO and TCNQ-LUMO. The effects of the other electrons are included by considering their screening effects.[3] The simplest model Hamiltonian which captures both effects, the itinerancy of the electrons as well as the strong Coulomb interaction is the Hubbard model

$$H = -\sum_{\sigma, i \neq j} t_{ij} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow} \; . \tag{1.1}$$

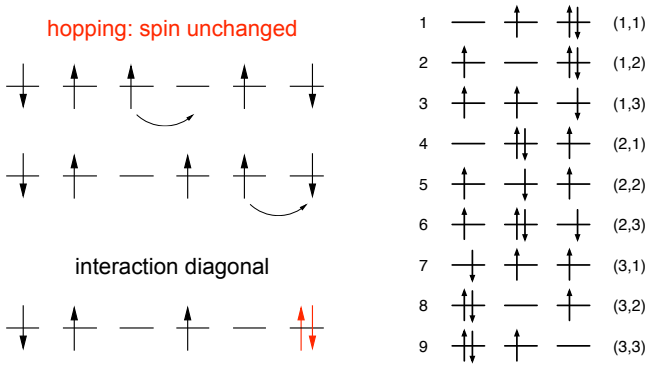

Figure 2. Illustration of the Hubbard model and the configuration basis of the Hilbert space. Upper left: next neighbour hopping (kinetic term); lower left: Coulomb repulsion U; right: basis for a three site system with two up and one down spin electron. The left label denotes the index of the configuration. Equivalently, a state is also unambiguously pointed to by a tuple of up- and down-configuration index.

The first term gives the kinetic energy, where t_{ij} is the amplitude for an electron to hop from the molecule at site i to lattice site j. Note that hopping does not change the spin σ. The second term represents the Coulomb repulsion between two electrons in the same molecular orbital. Both terms are illustrated in Fig. 2.

Since the Hamiltonian does neither change the number of electrons nor their spin, we need to consider (1.1) only on Hilbert spaces with a fixed number of electrons of spin up $N_\uparrow$ and spin down $N_\downarrow$. For a finite system of L orbitals there are $\binom{L}{N_\sigma}$ different ways to arrange N_σ electrons of spin σ. Thus the dimension of the Hilbert space is given by $\binom{L}{N_\uparrow} \cdot \binom{L}{N_\downarrow}$. For three orbitals, two up- and one down-spin electron, there are 9 different configurations shown in Fig. 2. Even though we significantly simplified the problem, we still have to face the many-body problem: increasing system size, the dimension of the Hilbert space increases steeply. A system with 20 orbitals and 10 electrons of either spin already contains more than 34 billion ($34\,134\,779\,536$) different configurations. Storing a single many-body state for this system takes about 254 GB.

2 Lanczos Method

We have seen that even for small systems the full Hamiltonian matrix is so large that it cannot be stored on a computer. Fortunately the matrix is sparse in real space, since each configuration is only connected to few others by hopping and the Coulomb repulsion part of the Hamiltonian is diagonal. To obtain the ground-state eigenvalue and -vector for our sparse Hamiltonian H we use the Lanczos method: We start from a random vector $|\phi_0\rangle$, leading to a first energy value $E_0 = E[\phi_0] = \langle\phi_0|H|\phi_0\rangle$. In analogy to the gradient descent method we look for the direction of steepest descent in the energy functional which is given by $H|\phi_0\rangle$. The resulting vector is orthogonalized with respect to the starting vector, i.e.

$$\langle\phi_1|H|\phi_0\rangle\,|\phi_1\rangle = H\,|\phi_0\rangle - \langle\phi_0|H|\phi_0\rangle\,|\phi_0\rangle \ .$$

Now the same step is done with $|\phi_1\rangle$ leading to

$$\langle\phi_2|H|\phi_1\rangle\,|\phi_2\rangle = H\,|\phi_1\rangle - \langle\phi_1|H|\phi_1\rangle\,|\phi_1\rangle - \langle\phi_1|H|\phi_0\rangle\,|\phi_0\rangle \ ,$$

or in general,

$$\beta_{n+1}\,|\phi_{n+1}\rangle = H\,|\phi_n\rangle - \alpha_n\,|\phi_n\rangle - \beta_n\,|\phi_{n-1}\rangle \ ,$$

where $n \in 2, \ldots, m$ and

$$\alpha_n = \langle\phi_n|H|\phi_n\rangle\,, \quad \beta_{n+1} = \langle\phi_{n+1}|H|\phi_n\rangle\,.$$

From this equation we see that the Hamiltonian H is tridiagonal in the basis of the Lanczos vectors. In practice we need only of the order of 100 iterations to converge to the ground-state. If we are only interested in the ground-state energy, we only have to store two vectors of the size of the Hilbert space. In order to also get the ground state vector, a third vector is needed. As discussed above the dimension grows swiftly and thus the applicability of the method is limited by the maximum available main memory.

The Lanzcos method also provides a way to calculate Green functions, i.e.

$$G_k(\omega) = \sum_n \frac{\left|\left\langle \psi_n^{N\pm1}\middle|c_k^{(\dagger)}\middle|\psi_0\right\rangle\right|^2}{\omega \mp (E_n^{N\pm1} - E_0^N)} \ .$$

Having calculated the ground-state vector as described above we apply the $c_k^{(\dagger)}$ operator to it, normalize, and use the result as initial vector for a Lanczos iteration. The α_i and β_i then give the Green function

$$G_k(\omega) = \cfrac{\beta_0^2}{\omega - \alpha_0 - \cfrac{\beta_1^2}{\omega-\alpha_1-\cfrac{\beta_2^2}{\omega-\alpha_2-\cfrac{\beta_3^2}{\omega-\alpha_3-\cdots}}}} \ .$$

3 Computational Aspects

The key ingredient of the Lanczos algorithm described above is the sparse matrix vector multiplication. Already for quite small systems this operation takes most of the execution time, and increasing the size of the many-body vector it dominates ever more. Thus it will be the focus our parallelization efforts. On shared memory systems this matrix-vector multiplication is embarrassingly simple: The resulting vector elements can be calculated independently. Thus different threads can work on different chunks of this vector. The factor vector as well as the matrix elements are only read, so that there is no need for locking. An OpenMP parallelization thus needs only a single pragma. Similarly parallelizing also the scalar products, we obtain almost ideal speedup on an IBM p690 frame of JUMP in Jülich. The implementation is however limited to a single node. To use significantly more memory we need to find an efficient distributed memory implementation.

The kinetic energy term of the Hamiltonian (1.1) has non-diagonal terms and therefore leads to non-local memory access patterns. A naive distributed memory parallelization is to emulate a shared memory by direct remote memory access via MPI one-sided communication. This approach leads, however, to a severe speed-down, i.e. the more processors we use, the longer we have to wait for the result. This is shown in the inset of Fig. 4.

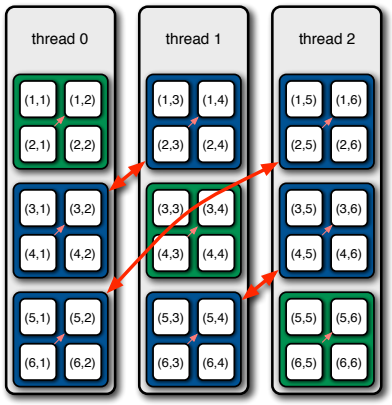

Figure 3. Transpose operation that makes memory access thread-local when calculating the operation of the Hamiltonian on the state-vector. The communication (red arrows) is realized by a call to `MPI_Alltoall`, which is very efficiently implemented on BlueGene/L. The small grey arrows indicate the local operations needed to complete the matrix-transpose.

To obtain an efficient distributed memory implementation we use a simple yet important observation: As pointed out above, the kinetic energy term conserves spin. Thus, performing the up-electron hopping takes only different up-hopping configurations into account while the down-electron configuration remains unchanged. If we group all up configurations for a fixed down configuration together in a single thread this hopping can thus be carried out locally. Fig. 2 illustrates this: for a fixed index $i_\downarrow$, all $i_\uparrow$ configurations follow and can be stored in a thread. We see, that this basis can be naturally indexed by a tuple $(i_\downarrow, i_\uparrow)$ (right labels in Fig. 2) instead of the global index (left labels). We can therefore equivalently regard the vectors as matrices $v(i_\downarrow, i_\uparrow)$ with indices $i_\downarrow$ and $i_\uparrow$. Now it is easy to see that a matrix transpose reshuffles the data elements such that the down configurations are sequentially in memory and local to the thread.

Therefore, the efficiency of the sparse matrix-vector multiplication rests on the performance of the matrix transpose operation. We implement it with `MPI_Alltoall`. This routine expects, however, the data packages which will be sent to a given process to be stored contiguously in memory. This does not apply to our case, since we would like to store the spin-down electron configurations sequentially in memory. Thus, the matrix is stored column wise. For `MPI_Alltoall` to work properly, we would have to bring the data elements in row-major order. This could be done by performing a local matrix transpose. The involved matrices are, however, in general rectangular, leading to expensive local-copy and reordering operations. We can avoid this by calling `MPI_Alltoall` for each column separately. After calling `MPI_Alltoall` for each column (red arrows in Fig. 3) only a local strided transposition has to be performed (small pink arrows) to obtain the fully transposed matrix or Lanczos vector.[4,5] The speed-up (left plot of Fig. 4) shows that collective communication is indeed very efficient, particularly on the BlueGene/L.

The implementation described so far uses `MPI_Alltoall` which assumes that the matrix to be transposed is a square matrix and that the dimension $dim_\uparrow = dim_\downarrow$ is divisible by the number of MPI processes. To overcome these restrictions we have generalized

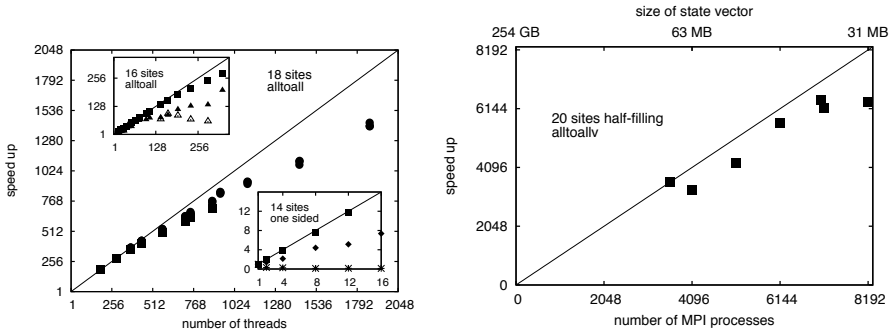

Figure 4. Speed-up of the Lanczos code. Left: IBM BlueGene/L JUBL (squares (CO mode) and circles (VN mode)) and IBM Regatta JUMP (triangles and diamonds: collective comm., stars: 1-sided comm.) for different problem sizes; right: speedup of the Lanczos code on IBM BlueGene/L JUBL in CO mode for 20 sites half-filled Hubbard model.

the algorithm to `MPI_Alltoallv`.[4] The right plot of Fig. 4 shows the speed-up of the generalized algorithm for a 20 sites system with half-filling (10 electrons of either spin). The corresponding size of the many-body vector is about 254 GB. The plot shows that we can efficiently use about 8192 threads and suggests that for larger system sizes we should be able to exploit even more processors.

4 Cluster Perturbation Theory and Spin-Charge Separation

Our parallel implementation of the Lanczos method enables us to efficiently calculate angular-resolved spectral functions for quite large systems. However, we still can have at most as many different momenta as we have sites. To resolve exciting physics like spin-charge separation we need, however, a much higher resolution. A way to achieve this is cluster perturbation theory (CPT).[6] The general idea is to solve a finite cluster with open boundary conditions exactly and then treat hopping between clusters in strong coupling perturbation theory, leading to an effective infinite chain.

We use the CPT technique in combination with the Lanczos method to study the quasi one-dimensional molecular metal TTF-TCNQ. Its low dimensionality in tandem with strong Coulomb repulsion compared to the kinetic energy leads to strong correlations and interesting many-body effects. Fig. 5 shows the angular-resolved spectral function for TCNQ in a CPT calculation for a 20 sites t-U Hubbard model. At the Γ-point we observe signatures of spin-charge separation: The electron dispersion splits into a holon and a spinon branch. To generate this plot about three BlueGene/L rack-days are needed: The calculation of the ground state is negligible and takes considerably less than half an hour on 2048 processors in VN mode on a BlueGene/L system. To calculate the Green's function for photoemission and inverse photoemission about 400 Green's functions each have to be calculated, where the former calculation takes a total of about 15 hours whereas the latter one takes about two days.

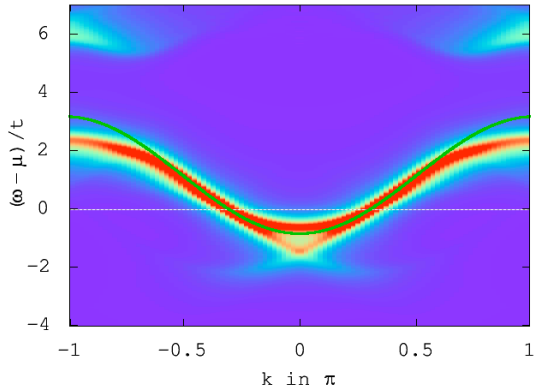

Figure 5. Angular-resolved spectral function obtained by CPT for a 20 sites TCNQ-like t-U Hubbard model with 6 electrons of either spin ($U = 1.96$eV, $t = 0.4$eV). The white line shows the chemical potential, the green cosine the independent-particle band. Signatures of spin-charge separation are clearly observed in the vicinity of the Γ-point.

5 Quantum Spins and Decoherence

A simplified version of the transposition scheme using `MPI_Alltoall` can be used to simulate quantum spin systems. We study the decoherence of two 1/2-spins (qubits) coupled to independent baths (a and b) of 1/2-spins. The Hamiltonian of this system is

$$H = \sum_{i \in \mathrm{BATH}_a} J_i \vec{s}_0 \vec{s}_i + \sum_{j \in \mathrm{BATH}_b} J_j \vec{s}_1 \vec{s}_j \ .$$

Since the Hilbert space of this system grows like 2^L where L is the total number of spins, we face similar problems as in the Lanczos method. In general, for a time-independent Hamiltonian, the time evolution operator is given by $U(t) = \exp(-iHt)$. Using a Suzuki-Trotter decomposition we can rewrite the operator as a product of time evolution operators for small time steps $U(t) \approx \prod^N U(\Delta t)$, where $t = N\Delta t$. We further decompose the time evolution operator such that we have a product of time evolution operators for two-spin systems, i.e. treating each term in the Hamiltonian separately, yielding

$$U(\Delta t) = \prod_{k=1}^{N} \exp(-iH_k \Delta t/2) \cdot \prod_{k=N}^{1} \exp(-iH_k \Delta t/2) \ + \mathcal{O}(\Delta t^3) \ ,$$

where N denotes the terms in the Hamiltonian and $\exp(-iH_k\Delta t/2)$ is given by

$$e^{iJ_k\Delta t/8} \begin{pmatrix} \exp(-iJ_k\Delta t/4) & 0 & 0 & 0 \\ 0 & \cos(J_k\Delta t/4) & -i\sin(J_k\Delta t/4) & 0 \\ 0 & -i\sin(J_k\Delta t/4) & \cos(J_k\Delta t/4) & 0 \\ 0 & 0 & 0 & \exp(-iJ_k\Delta t/4) \end{pmatrix} .$$

Thus we reduce calculating the exponential of the spin-Hamiltonian to simple matrix products. As before the matrix elements are not necessarily local to the threads, but can be made so by appropriate calls to `MPI_Alltoall`. To test this approach we have so far considered systems of up to 26 spins for which we obtain almost ideal speed-up for up to 1024

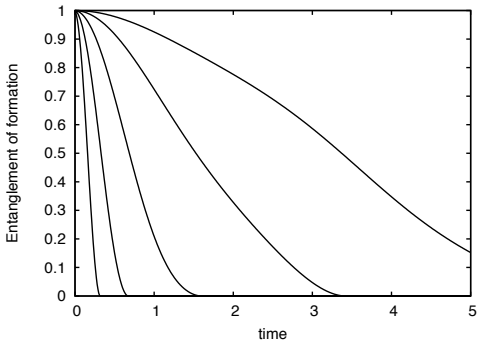

Figure 6. Entanglement-of-formation of a two qubit system, where each qubit couples to a bath of 10 spins. The coupling constants are chosen such that $\sum_i |J_i|^2 = C$ (see text). The curves from left to right show C = 1,1/2,1/4,1/8,1/16. The initial state for the qubits is a Bell state.

processors in VN mode. As a first application we study the sudden death of entanglement. We start with the two qubits in the Bell state $(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)/\sqrt{2}$ and a random, but unpolarized bath. The coupling constants are chosen randomly, but normalized to $\sum_i |J_i|^2 = C$. We find that the entanglement-of-formation[7] goes to zero after a finite time. As can be seen from Fig. 6, the time to this sudden-death of entanglement[8] is roughly proportional to the strength of the coupling to the bath C.

References

1. W. Kohn, *Nobel Lecture: Electronic structure of matter: wave functions and density functionals*, Rev. Mod. Phys., **71**, 1253, (1999).
2. E. Koch and E. Pavarini, *Multiple Scales in Solid State Physics*, Proceedings of the Summer School on Multiscale Modeling and Simulations in Science, (Springer,2007).
3. L. Cano-Cortés, A. Dolfen, J. Merino, J. Behler, B. Delley, K. Reuter and E. Koch, *Coulomb parameters and photoemission for the molecular metal TTF-TCNQ*, Eur. Phys. J. B, **56**, 173, (2007).
4. A. Dolfen, *Massively parallel exact diagonalization of strongly correlated systems*, Diploma Thesis, RWTH Aachen, (2006).
5. A. Dolfen, E. Pavarini and E. Koch, *New horizons for the realistic description of materials with strong correlations*, Innovatives Supercomputing in Deutschland, **4**, 16, (2006).
6. D. Sénéchal, D. Perez and M. Pioro-Ladrière, *Spectral weight of the Hubbard model through cluster perturbation theory*, Phys. Rev. Lett., **84**, 522, (2000).
7. W. K. Wooters, *Entanglement of formation of an arbitrary state of two qubits*, Phys. Rev. Lett., **80**, 2245, (1998).
8. M. P. Almeida, *et al. Environment-induced sudden death of entanglement*, Science, **316**, 579, (2007).