CHAPTER 5

Efficient calculation of the generalised Langevin equation

In the previous chapters, we have introduced our S-QN method. Since the calculation of the curvature-dependent mobility and its factorized form is very compute expensive, we also introduced an efficient factorized update scheme. The factorized update scheme (FSU) enables direct updates of the noise term and therefore avoids the compute expensive (Cholesky) factorization. In the previous chapter, where we applied S-QN to a minimal protein model, the small number of degrees of freedom (n) rendered FSU appropriate for the calculation of the mobility matrix. For very large systems, however, storage and modification of full matrices becomes inefficient and L-FSU is more appropriate. Nevertheless, in Chapter 3 we have only considered L-FSU for a quadratic potential, i.e. the case where the spurious drift term vanishes. In this chapter, we clarify how to efficiently combine L-FSU and the scheme introduced by Hütter and Öttinger [39], that applies when the spurious drift term becomes significant and involves the calculation of the inverse of B.

5.1. Introduction

Our starting point is the generalized Langevin equation:

$$\mathrm{d}\mathbf{x} = [-B(\mathbf{x})\nabla\Phi(\mathbf{x}) + k_B T\nabla\cdot B(\mathbf{x})]\mathrm{d}t + \sqrt{2k_BT}J(\mathbf{x})\mathrm{d}W(t), \tag{5.1}$$

where $B(\mathbf{x}) = J(\mathbf{x})J(\mathbf{x})^T$. It is clear that factorization of $B(\mathbf{x})$ and calculation of $\nabla\cdot B(\mathbf{x})$ are of the highest computational complexity.

Avoid factorizing the mobility

Using our proposed factorized secant update (FSU) scheme from Chapter 3, a direct update of J is possible;

$$J_{k+1} = J_k + \frac{\alpha\mathbf{s}_k\mathbf{y}_k^T J_k - \alpha^2 J_k J_k^T \mathbf{y}_k\mathbf{y}_k^T J_k}{\mathbf{y}_k^T\mathbf{s}_k}, \tag{5.2}$$

where $\mathbf{s}_k = \mathbf{x}_{k-1} - \mathbf{x}_k$, $\mathbf{y}_k = \nabla\Phi(\mathbf{x}_{k-1}) - \nabla\Phi(\mathbf{x}_k)$ and

$$\alpha^2 = \frac{\mathbf{y}_k^T\mathbf{s}_k}{\mathbf{y}_k^T J_k J_k^T \mathbf{y}_k}. \tag{5.3}$$

Avoid calculating the divergence of the mobility

Using the scheme of Hütter and Öttinger [39], no explicit calculation of the divergence of the mobility is needed;

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k \tag{5.4}$$

$$\begin{aligned}\Delta\mathbf{x}_k &= -\frac{1}{2}[B(\mathbf{x}_k + \Delta\mathbf{x}_k^p)\nabla\Phi(\mathbf{x}_k + \Delta\mathbf{x}_k^p) + B(\mathbf{x}_k)\nabla\Phi(\mathbf{x}_k)]\Delta t \\ &+ \frac{1}{2}[B(\mathbf{x}_k + \Delta\mathbf{x}_k^p)B^{-1}(\mathbf{x}_k) + I]\sqrt{2k_BT}J(\mathbf{x}_k)\Delta W_t,\end{aligned} \tag{5.5}$$

$$\Delta\mathbf{x}_k^p = -B(\mathbf{x}_k)\nabla\Phi(\mathbf{x}_k)\Delta t + \sqrt{2k_BT}J(\mathbf{x}_k)\Delta W_t. \tag{5.6}$$

where (5.6) is the predictor step and (5.4) is the correction. Although this predictor-correction scheme avoids the calculation of the divergence of B, other costs are involved, notably the significant costs for the calculation of the inverse.
Fortunately the DFP update scheme (equivalent to FSU) for B_k is of such a form

that the Sherman-Morrison theorem can be applied to calculate the exact inverse of $B^{-1}(\mathbf{x}_k) = G(\mathbf{x}_k) = G_k$ in (5.5)

$$G_k = (I - \frac{\mathbf{y}_{k-1}\mathbf{s}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{s}_{k-1}})G_{k-1}(I - \frac{\mathbf{s}_{k-1}\mathbf{y}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{s}_{k-1}}) + \frac{\mathbf{y}_{k-1}\mathbf{y}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{s}_{k-1}}. \quad (5.7)$$

Using FSU and the update scheme of Hütter and Öttinger together with (5.7) decreases the computational cost from $O(n^3)$ to $O(n^2)$.

5.2. Theory

The Sherman-Morrison formula is a special form of the Woodbury formula for the inverse of a rank k correction matrix and is given by

$$\left[A + \mathbf{u}\mathbf{v}^T\right]^{-1} = A^{-1} - \frac{A^{-1}\mathbf{u}\mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{u}}, \quad (5.8)$$

for an invertible square matrix A and vectors $\mathbf{u}$ and $\mathbf{v}$. Since the update scheme (5.2) is of the same form as the left hand side of equation (5.8), the inverse of J_{k+1} can be written as

$$J_{k+1}^{-1} = F_{k+1} = F_k - F_k\frac{\mathbf{s}_k - \alpha_k J_k J_k^T\mathbf{y}_k}{\mathbf{y}_k^T\mathbf{s}_k}\mathbf{y}_k^T, \quad (5.9)$$

where $J_k^{-1} = F_k$. It is easily checked that $J_{k+1}F_{k+1} = I$. In equation (5.5) only B^{-1} occurs and the factorized form of B^{-1} seems redundant. However deriving F_k enables us to rewrite B^{-1} in a suitable way for a limited memory construction of B^{-1}. Rewrite $B^{-1}(\mathbf{x}_{k+1})$ as

$$F_{k+1}^T F_{k+1} = W_k^T W_{k-1}^T ... W_0^T F_0^T F_0 W_0 ... W_{k-1} W_k, \quad (5.10)$$

with

$$W_k = (I - \frac{\mathbf{s}_k - \alpha_k J_k J_k^T\mathbf{y}_k}{\mathbf{y}_k^T\mathbf{s}_k}\mathbf{y}_k^T) = (I - \frac{\mathbf{s}_k - \alpha_k\mathbf{h}_k}{\mathbf{y}_k^T\mathbf{s}_k}\mathbf{y}_k^T) \quad (5.11)$$

$$= (I + \frac{1}{\alpha_k \nu_k}\mathbf{v}_k\mathbf{y}_k^T), \quad (5.12)$$

where $\mathbf{v}_k = \mathbf{h}_k - \mathbf{s}_k/\alpha_k$, $\mathbf{h}_k = J_k J_k^T\mathbf{y}_k$ and $\nu_k = \mathbf{h}_k^T\mathbf{y}_k$. The inverse of the mobility is now casted in a factorized form (5.10) where a recursive expression (obtained by loop unrolling), can serve as a basis for limited-memory implementation. Since the case for $k < m$ is trivial, we only consider the case where k is larger than or equal to the truncation parameter m. Following the derivation of L-FSU in appendix B, we derived the limited-memory case of B_{k+1}^{-1} for $k \geq m$

$$F_{k+1}^T F_{k+1} = \tilde{W}_k^T \tilde{W}_{k-1}^T \quad \ldots \quad \tilde{W}_{k-m+1} F_0 F_0^T \tilde{W}_{k-m+1} \quad \ldots \quad \tilde{W}_{k-1}\tilde{W}_k. \tag{5.13}$$

The ˜ above W_k indicates that W_k is defined as in equation (5.11) with truncated J_k, i.e.

$$J_k = \tilde{V}_{k-1} \ldots \tilde{V}_{k-m+1} J_0, \tag{5.14}$$

where $\tilde{V}_k$ is defined as

$$\tilde{V}_k = (I - \frac{1}{\tilde{\mathbf{h}}_k^T \mathbf{y}_k}(\tilde{\mathbf{h}}_k - \frac{\mathbf{s}_k}{\tilde{\alpha}_k})\mathbf{y}_k^T), \tag{5.15}$$

with truncated J_k in $\tilde{\mathbf{h}}_k$ and $\tilde{\alpha}_k$.

Alternatively, the limited memory case of B_{k+1}^{-1} can be found by solving the rewritten $B_{k+1}B_{k+1}^{-1} = I$ as

$$[J_0^T(\prod_{i=1}^{m} \tilde{V}_{k-m+i}^T)]^T J_0^T(\prod_{i=1}^{m} \tilde{V}_{k-m+i}^T)[Z_0(\prod_{i=1}^{m} \tilde{Z}_{k-m+i})]^T Z_0(\prod_{i=1}^{m} \tilde{Z}_{k-m+i}) = I, \tag{5.16}$$

where $\tilde{V}_i$ is given in (5.15). A solution for equation (5.16) is found by solving $\tilde{Z}_k$ in

$$\tilde{V}_k^{-1} = \tilde{Z}_k, \tag{5.17}$$

and set $Z_0^T = J_0^{-T}$. Equation (5.17) can easily solved by applying the Sherman-Morrison formula on (5.15) which gives

$$\begin{aligned}
\tilde{Z}_k &= (I - \frac{1}{\tilde{\mathbf{h}}_k^T \mathbf{y}_k}(\tilde{\mathbf{h}}_k - \frac{\mathbf{s}_k}{\tilde{\alpha}_k})\mathbf{y}_k^T)^{-1} \\
&= I - \frac{-1}{\tilde{\mathbf{h}}_k^T \mathbf{y}_k}(\tilde{\mathbf{h}}_k - \frac{\mathbf{s}_k}{\tilde{\alpha}_k})\mathbf{y}_k^T / (1 + \frac{-\mathbf{y}_k^T}{\tilde{\mathbf{h}}_k^T \mathbf{y}_k}(\tilde{\mathbf{h}}_k - \frac{\mathbf{s}_k}{\tilde{\alpha}_k})) = I + \frac{1}{\tilde{\mathbf{h}}_k^T \mathbf{y}_k}(\tilde{\mathbf{h}}_k - \frac{\mathbf{s}_k}{\tilde{\alpha}_k})\mathbf{y}_k^T / \tilde{\alpha}_k \\
&= I + \frac{1}{\alpha_k \nu_k}\mathbf{v}_k \mathbf{y}_k^T.
\end{aligned} \tag{5.18}$$

Evidently (5.18) is the same as the derived $\tilde{W}_k$ from the truncated $F_{k+1}^T F_{k+1}$.

5.3. Discussion and Conclusion

Having derived a limited memory scheme for B^{-1}, we are now able to calculate the general Langevin equation in a complete limited memory way. Using the limited memory method will not only limit the storage needed, but the computational cost will decrease significantly. We finalize this chapter by considering the total computational cost of the general Langevin equation with adaptive mobility using the limited memory method. Only considering the computational load of the displacement (5.5) is sufficient since the predictor term (5.6) also appears in the displacement and thus reusable. Written out the discretized displacement gives

$$\Delta \mathbf{x}_k = $$

$$-\frac{1}{2} B(\mathbf{x}_k^p) \nabla \Phi(\mathbf{x}_k^p) \Delta t \tag{5.19a}$$

$$-\frac{1}{2} B(\mathbf{x}_k) \nabla \Phi(\mathbf{x}_k) \Delta t \tag{5.19b}$$

$$+\frac{1}{2} B(\mathbf{x}_k^p) B^{-1}(\mathbf{x}_k) \sqrt{2k_B T}\, J(\mathbf{x}_k) \Delta W_t \tag{5.19c}$$

$$+\frac{1}{2} \sqrt{2k_B T}\, J(\mathbf{x}_k) \Delta W_t, \tag{5.19d}$$

where $\mathbf{x}_k^p = \mathbf{x}_k + \Delta \mathbf{x}_k^p$. Terms (5.19b) and (5.19d) has already been discussed in the previous chapter: the computational costs are $10mn + 2n$ and $2mn + n$ respectively, where m is the truncation parameter or history depth and n the system size (dimension). The computational cost of term (5.19b) is an addition of the cost for the force times the limited factorized $B(\mathbf{x}_k)$ (5mn+n) and the cost for calculating $\mathbf{h}_k$, needed for the next time step. The computational cost for term (5.19a) is $5mn + n$ since it can be calculated in the same way as for (5.19b). Rest us to calculate the cost for term (5.19c). Since the vector $\sqrt{2k_B T}\, J(\mathbf{x}_k) \Delta W_t$ has already been calculated, we concentrate on the calculation of the vector times the limited factorized $B(\mathbf{x}_k)^{-1}$, which can be casted in the following scheme

$$\mathbf{d} = \sqrt{2k_B T}\, J(\mathbf{x}_k) \Delta W_t; \tag{5.20}$$

$$\begin{cases} \textbf{for } i = k-1, ..., \max(0, k-m) \\ \quad \gamma_i = \mathbf{y}_i^T \mathbf{d}; \\ \quad \mathbf{v}_i = \mathbf{h}_i - \mathbf{s}_i/\alpha_i; \\ \quad \mathbf{d} = \mathbf{d} + (\gamma_i/\alpha_i \nu_i)\mathbf{v}_i; \\ \textbf{end} \end{cases} \tag{5.21}$$

$$\mathbf{d} = F_0^T F_0 \mathbf{d}; \tag{5.22}$$

$$\begin{cases} \textbf{for } i = \max(0, k-m), ..., k-1 \\ \quad \omega_i = \mathbf{v}_i^T \mathbf{d}; \\ \quad \mathbf{d} = \mathbf{d} + (\omega_i/\alpha_i \nu_i)\mathbf{y}_i; \\ \textbf{end} \end{cases} \tag{5.23}$$

$$\textbf{stop with result } \mathbf{d} = F(\mathbf{x}_k)^T F(\mathbf{x}_k)\mathbf{d} = B(\mathbf{x}_k)^{-1} \sqrt{2k_B T} J(\mathbf{x}_k)\Delta W_t, \tag{5.24}$$

where the ˜ has been omitted for simpler notation. The first loop recursion (5.21) and second loop recursion (5.23) require $3mn$ and $2mn$ multiplications respectively; if $F_0^T F_0$ is diagonal, then n additional multiplications are needed. The vector (5.24) can be used to multiply with the limited version of $B(\mathbf{x}_k + \Delta\mathbf{x}_k^p)$, which are an additional $5mn + n$ multiplications. This gives us a total of $10mn + 2n$ multiplications for term (5.19c). The computational costs are summarized and compared in the table below. Clearly the computational complexity of the limited memory method is reduced significantly compared to the conventional calculation methods and FSU.

term \ scheme	Conventional operation	Conventional cost	FSU operation	FSU cost	L-FSU operation	L-FSU cost
$J(\mathbf{x}_k)J(\mathbf{x}_k)^T\mathbf{y}_k$	NA		NA		store as $\mathbf{h}_k$ $\left[JJ^T\right]_{\text{lim}}\mathbf{v}$	$5mn+n$
$B(\mathbf{x}_k^p)\nabla\Phi(\mathbf{x}_k^p)$	$B_{\text{DFP}}\mathbf{v}$	$2n^2+O(n)$	$B_{\text{DFP}}\mathbf{v}$	$2n^2+O(n)$	$\left[JJ^T\right]_{\text{lim}}\mathbf{v}$	$5mn+n$
$B(\mathbf{x}_k)\nabla\Phi(\mathbf{x}_k)$	update $B(\mathbf{x}_k)$ $B_{\text{DFP}}\mathbf{v}$	$2n^2+O(n)$ $n^2+O(n)$	update $B(\mathbf{x}_k)$ $B_{\text{DFP}}\mathbf{v}$	$2n^2+O(n)$ $n^2+O(n)$	store $\mathbf{y}_{k-1}$,$\mathbf{s}_{k-1}$ $\left[JJ^T\right]_{\text{lim}}\mathbf{v}$	$5mn+n$
$J(\mathbf{x}_k)\Delta W_t(=\mathbf{w})$	Chol$(B(\mathbf{x}_k))$ $J_{\text{Chol}}\mathbf{v}$	$O(n^3)$ $n^2+O(n)$	update $J(\mathbf{x}_k)$ $J_{\text{FSU}}\mathbf{v}$	$4n^2+O(n)$ $n^2+O(n)$	$\left[JJ^T\right]_{\text{lim}}\mathbf{v}$	$2mn+n$
$B(\mathbf{x}_k^p)B^{-1}\mathbf{w}$	inv$(B(\mathbf{x}_k))$ $B_{\text{DFP}}B^{-1}\mathbf{v}$	$O(n^3)$ $3n^2+O(n)$	SM$(B(\mathbf{x}_k))$ $B_{\text{DFP}}B^{-1}\mathbf{v}$	$2\frac{1}{2}n^2+O(n)$ $3n^2+O(n)$	$\left[JJ^TF^TF\right]_{\text{lim}}\mathbf{v}$	$10mn+n$

$B_{xxx}\mathbf{v}$ is the matrix vector multiplication where B is obtained by applying scheme xxx
Chol(B) is the Cholesky decomposition applied on B
inv(B) is the inverse calculation(Gaussian elimination) for the matrix B,
SM(B) is the inverse calculation using the Sherman−Morrison formula for the matrix B
$[..]_{\text{lim}}\,\mathbf{v}$ is the vector obtained using the limited memory methods

Table 5.1: *Computational cost comparison using different schemes.*

Reducing computational complexity and saving on storage is very important for simulating large systems. In this chapter we achieved to construct a limited-memory scheme for the generalized S-QN method. Combining the limited memory scheme with the automated multi-scaling property of the alternative mobility gives us a very powerful method for configurational space sampling with good thermodynamical consistency.

Bibliography

[1] E. Nelson, Dynamical Theories of Brownian Motion, Princeton University Press, Princeton, NJ (1967).

[2] R. Brown, A brief Account of Microscopical Observations made in the Months of June, July, and August, 1827, on the Particles contained in the Pollen of Plants; and on the general Existence of active Molecules in Organic and Inorganic Bodies, Philosophical Magazine N. S. **4**, (1828).

[3] R. Brown, Additional Remarks on Active Molecules, Philosophical Magazine N. S. **6**, (1829).

[4] A. Einstein, "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen.", Annalen der Physik **17** (1905).

[5] A. Einstein, "Eine neue Bestimmung der Moleküldimensionen.", Inaugural-Dissertation zur Erlangung der philosophischen Doctorwürde der hohen Philosophischen Fakultät (Mathematisch-Naturwissenschaftliche Sektion) der Universität Zürich,K.J. Wyss, (Bern, 1905).

[6] M. Smoluchowski, "Zur kinetischen Theorie der Brownschen Molekularbewegung und der Suspensionen", Annalen der Physik **21**, (1906).

[7] P. Langevin, "Sur la theorie du mouvement brownien", C.R. Acad. Sci. **146**, (1908).

[8] J. Perrin, a number of short contributions between May 1908 and September 1909, C.R. Acad. Sci., (1908-1909).

[9] C.W. Gardiner. Handbook of Stochastic Methods. Springer-Verlag, Berlin, New York (1983).

[10] L. Verlet, Phys. Rev. **159**, 98 (1967).

[11] L. Verlet, Phys. Rev. **165**, 201 (1967).

[12] R. P. Feynman, R. B. Leighton and M. Sands, The Feynman Lectures on Physics, Vol. 1, Addison-Wesley, (1963).

[13] D. Frenkel and B. Smit, Understanding molecular simulation: from algorithms to applications. Academic Press, (1996).

[14] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.N. Teller and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[15] C. Pangali, M. Rao and B.J. Berne, Chem. Phys. Lett. **55**, 413 (1978).

[16] P.J. Rossky, J.D. Doll and H.L. Friedman, J. Chem. Phys. **69**, 4628 (1978).

[17] D. L. Ermak, J. Chem. Phys. **62**, 4189 (1975).

[18] S. Kirkpatrick, C. Gelatt and M. Vecchi, Science **220**,671 (1983).

[19] V. Cerny, J. Optim. Theory Applic. **45**, 41 (1985)

[20] P. J. M. V. Laarhoven and E. H. L. Aarts, Simulated Annealing: Theory and Applications, D. Rediel, Dordrecht (1987).

[21] J. Nocedal and S.J. Wright. Numerical Optimisation, Springer Series in Operation Research, New York (1999).

[22] R. M. Levy, A. R. Srinivasan, W. K. Olson, and J. A. McCammon, Biopoly- mers **23**, 1099 (1984).

[23] A. E. García, Phys. Rev. Lett., **68**, 2696 (1992).

[24] A. Amadei, A. B. M. Linssen and H. J. C. Berendsen, Struct. Funct. Gen. **17**, 412 (1993).

[25] F.H. Stillinger, Science **267**, 1935 (1996).

[26] F.H. Stillinger and T.A. Weber, Science **225**, 983 (1984).

[27] W.F. van Gunsteren and H.J.C. Berendsen, Molecular Physics **45**, 637 (1982).

[28] I.M de Schepper and E.G.D. Cohen, J. Stat. Phys. **27**, 223 (1982).

[29] E. Leontidis and U.W. Suter, Molecular Physics **83**, 489 (1994).

[30] S. Goldman, J. Chem. Phys. **62**, 441 (1986).

[31] M. Doi, S.F. EDWARDS, The theory of polymer dynamics, Oxford science publications, Oxford (1986).

[32] L. Kaufman, SIAM J. Optim. **10**, 56 (1999).

[33] J.E. Dennis Jr. and R.B. Schnabel, Numerical Methods for Unconstrained Optimisation and Nonlinear Equations, Siam Classics, (1996)

[34] M.J.D. Powell. In: Nonlinear programming, SIAM-AMS proceedings vol. IX, eds. R.W. Cottle and C.E. Lemke, SIAM Publications, Philadelphia (1976).

[35] J.M. Anglada and J.M. Bofill, J. Comput. Chem. **19**, 349 (1998).

[36] H.P. Hratchian and H.B. Schlegel. J. Chem. Theory & Comput. **1**, 61 (2005).

[37] P.E. Gill, W. Murray. J. Inst. Maths. Applns. **9**, 91 (1972).

[38] Grandinetti. In: Methods for operations research, Hain Verlay, (1979).

[39] M. Hütter and H.C. Öttinger, J. Chem. Soc., Faraday Trans., **94**(10), 1403 (1998).

[40] H.D. Vollmer, H. Risken. Z. Physik B. **34**, 313 (1979).

[41] C.W. Gardiner, Journal of Statistical Physics, **30**, No 1, 157 (1983).

[42] H.A. Kramers, Physica, **7**, 284 (1940).

[43] C.J. Cerjan and W.H. Miller, J. Chem. Phys. **75**, 2800 (1981).

[44] B.A. Berg and T. Neuhaus. Phys. Rev. Lett. **68**, 9 (1992).

[45] E. Aarts and J. Korst. Simulated Annealing and Boltzmann Machines. Wiley, Great

Britain (1989).

[46] C.D. Chau, G.J.A. Sevink and J.G.E.M. Fraaije, J. Chem. Phys. **128**, 244110 (2008).

[47] B. Dünweg, *Accelerated algorithms 2, Computer simulations of surfaces and interfaces, proceedings of the NATO Advanced Study Institute*, edited by B. Dünweg, D. P. Landau, A. Milchev (Kluwer Academic Publishers, Dordrecht / Boston / London 2003).

[48] U.H.E. Hansmann, Chem. Phys. Lett. **281**, 140 (1997).

[49] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).

[50] E. Marinari and G. Parisi, Europhys. Lett. **19**, 451 (1992).

[51] P. Liu, B. Kim, R. A. Friesner and B. J. Berne, Proc. Nat. Acad. Sci. USA. **102**, 13749 (2005).

[52] N. Nakajima, J. Higo, A. Kidera and H. Nakamura, Chem. Phys. Lett. **278**, 297 (1997).

[53] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

[54] A. Laio, M. Parrinello, Proc. Natl. Acad. Sci. USA. **99**, 12562 (2002).

[55] A. F. Voter, J. Chem. Phys. **106**, 4665 (1997).

[56] B. Space, H. Rabitz, and A. Askar, J. Chem. Phys. **99**, 9070 (1993).

[57] M.E. Tuckerman, B.J. Berne, and A. Rossi, J. Chem. Phys. **94**, 1465 (1991).

[58] G. Zhang and T. Schlick, J. Chem. Phys. **101**, 4995 (1994).

[59] G. G. Batrouni, G. R. Katz, A. S. Kronfeld, G. P. Lepage, B. Svetitsky and K. G. Wilson, Phys. Rev. D **32**, 2736 (1985).

[60] F. J. Alexander, B. M. Boghosian, R. C. Brower, and S. R. Kimura, Phys. Rev. E **64**, 066704 (2001).

[61] B. Dünweg, private discussion, 2009.

[62] D. Goldfarb, Math. Comp. **30**, 796 (1976).

[63] K.W. Brodlie, A. R. Gourlay and J. Greenstadt, J. Inst. Math. Appl. **11**, 73 (1973).

[64] D. C. Liu and J. Nocedal, Math. Program. **45**, 503 (1989).

[65] D.J. Wales and J.P.K. Doye, J. Phys. Chem. A **101**, 5111 (1997).

[66] R. Fletcher and M.J.D. Powell, Computer J. **6**, 163 (1963).

[67] M. T. Chu, R. E. Funderlic, and G. H. Golub, SIAM. J. Matrix Anal. & Appl. **20**, 428 (1998).

[68] R.B. Schnabel, Math. Program. **15**, 247 (1978).

[69] W.C. Davidon, Math. Program. **9**, 1 (1975).

[70] C.R. Rao and S.K. Mitra, *Generalized Inverse of Matrices and its Applications*, Wiley, (New York 1971).

[71] M. B. Reed, Int. J. Comput. Math. **86**, 606 (2009).

[72] J. L. McCauley , Phys. A **382**, 445 (2007).

[73] J. L. McCauley , MPRA Paper No. 2128; http://mpra.ub.uni-muenchen.de/2128/, (2007).

[74] P. Eastman and S. Doniach, Proteins **30**, 215 (1998).

[75] B. Mishra and T. Schlick, J. Chem. Phys. **105**, 299 (1996).

[76] L. N. Mazzoni and L. Casetti, Phys. Rev. Lett. **97**, 218104 (2006).

[77] A. Buckley and A. Lenir, Math. Program. **27**, 155 (1983).

[78] D. F. Shanno and K-H Phua, J. Optimiz. Theory App. **25**, 507 (1978).

[79] V. Tozzini, Accounts of Chem. Res. **43**, 220 (2010).

[80] C.D. Chau, G.J.A. Sevink and J.G.E.M. Fraaije, J. Chem. Phys. **128**, 244110 (2008).

[81] C.D. Chau, G.J.A. Sevink and J.G.E.M. Fraaije, Phys. Rev. E. (2009).

[82] J. Nocedal and S.J. Wright, *Numerical Optimisation*, Springer Series in Operation Research, New York (1999).

[83] C.J. Camacho and D. Thirumalai, Proc. Natl. Acad. Sci. USA **92**, (1995).

[84] M. Dadlez and P.S. Kim, Nat. Struct. Biol. **2**, 674 (1995).

[85] L.A. Mirny, V. Abkevich and E.I. Shakhnovich, Fold. Des. **1**, 221 (1996).

[86] K.A. Dill et al. and H.S. Chan, Protein Sci. **4**, 561 (1995).

[87] J.D. Bryngelson, J.N. Onuchic, N.D. Socci and P.G. Wolynes, Proteins **21**, 167 (1995).

[88] P.G. Wolynes, J.N. Onuchic and D. Thirumalai, Science **267**, 1619 (1995).

[89] H.S. Chan and K.A. Dill, J. Chem. Phys. **100**, 9238 (1994).

[90] D. Thirumalai, Theoretical perspectives on in vitro and in vivo protein folding. In: S. Doniach, Editor, Statistical Mechanics, Protein Structure, and Protein Substrate Interactions, Plenum Press, New York (1994).

[91] T. Veitshans, D. Klimov, and D. Thirumalai, Folding Des. **2**, 1 (1997).

[92] L. Angelani and G. Ruocco, EPL **87**, 18002 (2009).

[93] A. Samiotakis, D. Homouz and M. S. Cheung, J. Chem. Phys. **132**, 175101 (2010).

[94] C.G. Broyden, J. Inst. Math. Appl. **6**, 222 (1970).

[95] S. S. Oren and E. Spedicato, Mathematical Programming **10**, 70 (1976).

[96] R. Fletcher, Computer J. **13** 317 (1970).

[97] D.F. Shanno and K.H. Phua , Math. Programming **14**, 149 (1978).

[98] C.G. Broyden, Math. Comput. **21**, 368 (1967).

[99] H. Yabe, H. J. Martinez, and R. A. Tapia, SIAM J. Optim. **15**, 139 (2004) .

[100] S.K. Kearsley, Acta Cryst. A **45**, 208 (1989).

[101] D.F. Shanno, Computers and Chemical Engineering **7**, 569 (1983).

[102] M.J.D. Powell, *The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations*, Nonlinear Programming 3, (O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds.), Academic Press (1978).

[103] M.J.D. Powell, *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, Numerical Analysis, G.A.Watson ed., Lecture Notes in Mathematics, Springer Verlag, Vol. 630, (1978).

[104] D.L. Pincus, S.S. Cho, C. Hyeon and D. Thirumalai, Progress in Molecular Biology and Translational Science **84**, 203 (2008).

[105] P.J. Flory, *Statistical mechanics of chain molecules*, Interscience publishers, New York (1969).

[106] A. Laio, M. Parrinello, Proc. Natl. Acad. Sci. USA. **99**, 12562 (2002).

[107] J.P. Ryckaert, G. Ciccotti G and H.J.C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[108] H.C. Andersen, J. Comput. Phys. **52**, 24 (1983).
[109] B. Hess, H. Berendsen and J.G.E.M. Fraaije, J. Comput. Chem. **18**, 1463 (1997).
[110] O.M.J. Crippen, J. Camp. Phys. **24**, 96 (1977).
[111] A.L. Mackay, Acta Crystallogr. Sect. A **30**, 440 (1974).