Theor Chem Acc (2009) 123:477–485
DOI 10.1007/s00214-009-0567-3

REGULAR ARTICLE

Locating seam minima for macromolecular systems

Søren Madsen · Frank Jensen

Received: 22 August 2008 / Accepted: 6 April 2009 / Published online: 25 April 2009

Abstract The performance of three different functions (penalty, Lagrange–Newton, and projection) used in combination with three different Newton-based optimization algorithms for solving large-scale constrained optimizations is investigated. The test cases correspond to locating minima on seams between two force field energy functions, which can be used to model transitions structures in chemical reactions. The Lagrange–Newton function used in combination with a standard Newton–Raphson optimization is found to be the most efficient for systems up to ~500 atoms, while an iterative algorithm becomes preferable for larger systems.

Keywords Force field · Constrained optimization · Macromolecules · Seam minimum

1 Introduction

The exploration of potential energy surfaces for describing chemical systems is well-established, and a variety of methods exist for locating minima and saddle points on such surfaces [1]. A description of photochemical reactions requires at least two energy surfaces, and the intersection of these surfaces plays an important role in describing the dynamics of the photo-chemical or photo-physical behavior [2–5]. For two surfaces with different symmetry (space or spin) the relevant point is a minimum on the seam of the two surfaces [7–9]. This corresponds to a minimum in the $N-1$ dimensional space, where the constraint is that the energies of the two surfaces should be equal. For two surfaces with the same symmetry, the relevant point is a conical intersection, which corresponds to a minimum in the $N-2$ dimensional space, where the additional constraint is related to the adiabatic coupling element between the two surfaces [2–5].

The mathematical problem of locating minima on seams or conical intersections is a constrained optimization, and several methods have been proposed for solving this problem [10–24]. The application of these methods has primarily been for modest sized systems with less than 100 variables, where the surfaces have been calculated using electronic structure methods. In a different context, we have proposed to use minima on seams to model chemical reactions using computational inexpensive force field methods [19, 25, 26], and the target systems here may contain several thousand atoms. In order to use this method as a general tool for modeling reactivity, we require a computational inexpensive method for locating minima on seams for systems with a large number of variables, and in the present paper we investigate the optimum strategy for solving such optimization problems. Although only the problem of a single constraint (energy equality) is considered at present, it is expected that the findings will also apply for solving problems with more constraints, such as locating conical intersections [4–6, 10, 11]. It is also possible that our finding will be useful for solving other types of large-scale constrained optimizations.

2 Theory

We consider the problem of locating a minimum on a seam for two different functions E_1 and E_2, which are taken as force field energy functions in the present case. The

S. Madsen · F. Jensen (✉)
Department of Chemistry, University of Aarhus,
Langelandsgade 140, 8000 Århus, Denmark
e-mail: frj@chem.au.dk

mathematical problem can be formulated as minimizing the sum of the two energies, subject to the constraint that the two energies are equal, see Eq. 1.

$$\text{Minimize}\, E_1(\mathbf{x}) + E_2(\mathbf{x}) \quad \text{subject to}\, E_1(\mathbf{x}) = E_2(\mathbf{x}). \tag{1}$$

Three different approaches have been proposed for solving this [10–24] and related [27–29] constrained optimization problems:

1. A penalty function [2].
2. A Lagrange–Newton function [10, 11, 13–19].
3. A projection function [20–24].

The penalty approach defines a target function where the constraint condition is added with a suitable weighting factor. We have chosen the simple harmonic form shown in Eq. 2, but other choices are also possible [2].

$$\text{Pen}(\mathbf{x}) = E_1(\mathbf{x}) + E_2(\mathbf{x}) + w(E_1(\mathbf{x}) - E_2(\mathbf{x}))^2. \tag{2}$$

Minimization of the penalty function will locate a point where the constraint condition is fulfilled with an accuracy related to the magnitude of the weight factor w. The advantage of the penalty approach is that the minimization of the target function can be done using standard techniques for function minimization, including methods which can be used to force the convergence and methods where analytical gradients are not available. The disadvantage is that the constraint condition is only satisfied approximately, and a large value of w may be required for meeting the constraint condition with an acceptable accuracy, and this can cause numerical problems. The numerical aspect may be to some extend circumvented by increasing the value of w as the optimization converges, but it may be difficult to establish stable criteria for adjusting the magnitude of w during the optimization, especially as it is likely that the optimum strategy will depend on the number of variables.

The Lagrange–Newton (LN) approach defines a target function as shown in Eq. 3.

$$\text{LN}(\mathbf{x}, \sigma) = E_1(\mathbf{x}) + E_2(\mathbf{x}) + \sigma(E_1(\mathbf{x}) - E_2(\mathbf{x})). \tag{3}$$

In Eq. 3 the constraint is enforced by means of a Lagrange multiplier σ. In contrast to the penalty method, the optimization of the LN function is to a first-order saddle point in the full parameter space $(\mathbf{x}, \sigma)$. As is well-known, it is more difficult to locate saddle points than minima, but the special structure of the function in Eq. 3 means that the eigenvector which should have a negative eigenvalue is dominated by σ, and this allows formulation of an optimization which in practice is as stable as a regular minimization [19]. A disadvantage is that knowledge of the Hessian eigenvector space is required.

The projection method builds on the condition that the direction perpendicular to the seam at a given point is determined by the (normalized) gradient difference vector, $\mathbf{t}$ [18]. An operator, $\mathbf{P}$, for projecting onto the seam subspace and a complementary operator, $\mathbf{Q}$, is defined in Eq. 4 and $\mathbf{t}$ is given by Eq. 5.

$$\mathbf{P} = \mathbf{1} - \mathbf{t}\mathbf{t}^t \quad \text{and} \quad \mathbf{Q} = \mathbf{t}\mathbf{t}^t. \tag{4}$$

$$\mathbf{t}(\mathbf{x}) = \frac{\mathbf{g_1}(\mathbf{x}) - \mathbf{g_2}(\mathbf{x})}{|\mathbf{g_1}(\mathbf{x}) - \mathbf{g_2}(\mathbf{x})|}. \tag{5}$$

Within the seam subspace, the target function should be minimized, allowing the problem to be formulated as locating a zero point for the gradient function shown in Eq. 6, where we have introduced a factor of 2 in the last term to keep the expression similar to the gradient of the penalty function, Eq. 7.

$$\begin{aligned}\nabla\text{Pro}(\mathbf{x}) = {} & \mathbf{P}(\mathbf{g}_1(\mathbf{x}) + \mathbf{g}_2(\mathbf{x})) \\ & + 2w\mathbf{Q}(E_1(\mathbf{x}) - E_2(\mathbf{x}))(\mathbf{g}_1(\mathbf{x}) - \mathbf{g}_2(\mathbf{x})).\end{aligned} \tag{6}$$

The projection method may be considered as a variation of either the penalty or LN approach, where the energy minimization is carried out only in the seam subspace while the constraint condition is only considered in the complimentary space. The relative importance of the two terms may be adjusted by a weight factor w [30]. In the present work we have chosen a value of 1 for w. It should be recognized that there is no obvious function associated with the gradient in Eq. 6 and a rigorous forcing of the convergence by monitoring changes in function values is therefore not possible. Furthermore, the second derivative of the (unknown) projection function, which corresponds to the Jacobian of the gradient function in Eq. 6, is non-symmetric, which leads to complications when using second-order optimization methods.

The projection and LN methods are the most popular methods for locating seam minima and conical intersections, where the optimization typically has been done by a quasi Newton–Raphson (NR) method using analytical gradients and an updated Hessian. Although the use of analytical Hessians is possible with the LN approach, the computational cost for generating this information with electronic structure methods is normally not cost-efficient. During the course of the present work, Keal et al. [30] published a study where they investigated the performance of using either of the above three methods for locating conical intersections for systems with up to 69 degrees of freedom, and found that the LN was the most efficient followed by the projection method.

For either of the above three functions one may employ different optimization strategies, and we will here consider the limited memory BFGS [31–35] and NR methods, where the latter can be used either with an exact or an updated Hessian. The penalty function can be optimized using either of these methods, while the saddle point

optimization of the LN function prevents a straightforward use of the BFGS algorithm. The non-symmetric Jacobian of the projection function, on the other hand, suggests that this function is best optimized using a BFGS approach.

While the problem of constrained optimization is quite general, the applications in computational chemistry have typically been for cases where the two functions are energies depending on a set of atomic coordinates. For such systems there are only $3N-6$ independent variables, as the three translational and three rotational degrees of freedom do not lead to changes in the energy. In practise, however, the energies are calculated from the full 3N set of coordinates, and numerical issues mean that the translational and rotational (TR) invariance is not automatically fulfilled. In an optimization these 6 degrees of freedom often require special attention in order to render the algorithm numerical stable.

We are in the present case interested in locating seam minima for functions with thousands of variables corresponding to atomic coordinates, and the handling of the Hessian matrix in NR-based methods here requires special attention. Our applications employ computationally inexpensive energy functions of the force field type, and calculation of first and second analytical derivatives is therefore relatively undemanding. The use of a regular NR optimization implies storage and diagonalization of the Hessian, which leads to memory demands of order N^2 and computational time of order N^3. Especially the latter rapidly becomes the computational bottleneck for optimizations using force field energy functions. One may instead employ a gradient-only optimization routine, such as the BFGS method, or an iterative NR method.

The NR method can be used with an updating of either the Hessian or its inverse. Not surprisingly we find that the use of an updated Hessian significantly degrades the performance compared to using an analytical Hessian. Since the latter is computationally inexpensive in the present case, we have not pursued the use of an updated Hessian further. The BFGS method corresponds to updating an inverse Hessian using gradient vectors, and the cost per iteration is correspondingly much lower than using an analytical Hessian, which offsets the lower quality of the updated Hessian.

Conjugate gradient methods are very popular for large-scale optimization. We have also tried conjugate gradient methods and found that it is essential to have a good preconditioner to obtain useful convergence. The limited memory BFGS method, however, in all cases performed better than conjugate gradient methods for the problems considered in this paper.

In the present paper we examine the problem of choosing the best combination of target function and optimization algorithms for locating minima on seams of two force field energy functions depending on up to a few thousands of variables. Although this is a rather specialized application, it is likely that our findings carry over to the general problem of performing large-scale constrained optimization in a set of (partly) redundant variables.

3 Optimization methods

In order to establish the notation, we briefly review the employed optimization schemes, more details can be found in the references [10–24, 36]. The two energy functions, depending on a common set of atomic coordinates, $\mathbf{x}$, will be denoted E_1 and E_2 with corresponding gradients $\mathbf{g}_1$ and $\mathbf{g}_2$ and Hessian matrices $\mathbf{H}_1$ and $\mathbf{H}_2$. The gradients and Hessians of the penalty and LN functions are given in Eqs. 7 and 8. The gradient of the projection function is given in Eq. 6.

$$\begin{aligned}
&\nabla \mathrm{Pen} = \mathbf{g}_1 + \mathbf{g}_2 + 2w(E_1 - E_2)(\mathbf{g}_1 - \mathbf{g}_2).\\
&\nabla\nabla^t \mathrm{Pen} = \mathbf{H}_1 + \mathbf{H}_2 + 2w(\mathbf{g}_1 - \mathbf{g}_2)(\mathbf{g}_1 - \mathbf{g}_2)^t \\
&\qquad\qquad + 2w(E_1 - E_2)(\mathbf{H}_1 - \mathbf{H}_2).
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\nabla \mathrm{LN} = \mathbf{g}_1 + \mathbf{g}_2 + \sigma(\mathbf{g}_1 - \mathbf{g}_2).\\
&\partial_\sigma \mathrm{LN} = E_1 - E_2.\\
&\nabla\nabla^t \mathrm{LN} = \mathbf{H}_1 + \mathbf{H}_2 + \sigma(\mathbf{H}_1 - \mathbf{H}_2).\\
&\partial_\sigma \nabla \mathrm{LN} = \mathbf{g}_1 - \mathbf{g}_2.\\
&\partial_\sigma^2 \mathrm{LN} = 0.
\end{aligned} \tag{8}$$

A standard NR optimization of a target function leads to a predicted step given by

$$\mathbf{\Delta x} = -(\mathbf{H} - \lambda \mathbf{I})^{-1}\mathbf{g} \quad \text{and} \quad |\Delta \mathbf{x}|^2 = R^2, \tag{9}$$

where the level shift parameter λ is used to control both the step-direction and -size. A popular strategy for choosing λ is by requiring that the steplength must be smaller than or equal to the current trust radius R [37]. The trust radius is continuously updated based on the ratio between the predicted and actual function change. Solution of Eq. 9 involves diagonalization of the Hessian which rapidly becomes impractical as the number of variables increases.

The BFGS method [31–35] is a quasi-Newton method which uses gradient information to construct approximations to the inverse Hessian in iteration $j+1$ using Eq. 10.

$$\begin{aligned}
\mathbf{H}_{j+1}^{-1} \approx \mathbf{H}_j^{-1} &+ \frac{\mathbf{s}_j^t\mathbf{y}_j + \mathbf{y}_j^t\mathbf{H}_j^{-1}\mathbf{y}_j}{\mathbf{s}_j^t\mathbf{y}_j}\mathbf{s}_j\mathbf{s}_j^t \\
&- \frac{\mathbf{H}_j^{-1}\mathbf{y}_j\mathbf{s}_j^t + \mathbf{s}_j\mathbf{y}_j^t\mathbf{H}_j^{-1}}{\mathbf{s}_j^t\mathbf{y}_j}.
\end{aligned} \tag{10}$$

The initial approximation to $\mathbf{H}^{-1}$ is usually a (scaled) identity matrix, i.e. $\mathbf{H}_0^{-1} = \alpha\mathbf{I}$. The vectors $\mathbf{s}$ and $\mathbf{y}$ are given in Eq. 11.

$$\mathbf{s}_j = \frac{\Delta \mathbf{x}_j}{|\Delta \mathbf{x}_j|} \quad \text{and} \quad \mathbf{y}_j = \mathbf{g}(\mathbf{x}_{j+1}) - \mathbf{g}(\mathbf{x}_j). \tag{11}$$

When H–1 is approximated by using only M previous gradients in Eq. 10, the product of the Hessian and gradient required in Eq. 9 can be calculated without storing the full Hessian, and the resulting method is called the limited memory BFGS method [35]. The step length is determined by performing a line search in the Δx -direction, such that the gradient is minimized. We have used the secant method to perform the line search in Eq. 9.

As an alternative to diagonalization of the full Hessian, Eq. 9 may be solved in a reduced vector space derived from the current gradient [36], a method which is closely related to the generalized minimum of the residual algorithm [38]. The idea is to generate a subspace of the full set of variables which spans most of the gradient and predicted step, and solve the NR equations exactly in this subspace. The efficiency of this approach relies on being able to generate a reasonable small subspace capable of representing a good approximation to the full solution, in a computationally efficient fashion. A residual gradient can be defined as in Eq. 12 where the geometry step $\mathbf{\Delta x}$ is initiated as zero. Note that it is not necessary to store the Hessian explicitly, as only its product with the $\mathbf{\Delta x}$ vector is required. This product vector may be formed in a direct fashion during the calculation of the Hessian elements.

$$\mathbf{R}_g = (\mathbf{H} - \lambda \mathbf{I})\mathbf{\Delta x} + \mathbf{g}. \tag{12}$$

A reduced space $\mathbf{b}_i$-vector is calculated from Eq. 13.

$$\mathbf{b}_i = \mathbf{A}\mathbf{R}_g. \tag{13}$$

The $\mathbf{b}_i$-vector is orthogonalized to all previous $\mathbf{b}_i$-vectors and normalized. The $\mathbf{A}$ matrix can be chosen in different ways. Equation 14 shows a relationship between the reduced gradient and the error vector relative to the Newton step.

$$\begin{aligned} &\mathbf{R}_g = (\mathbf{H} - \lambda \mathbf{I})\epsilon, \\ &\epsilon \equiv \mathbf{\Delta x} + (\mathbf{H} - \lambda \mathbf{I})^{-1}\mathbf{g}. \end{aligned} \tag{14}$$

This suggests that the optimal choice for $\mathbf{A}$ is $(\mathbf{H} - \lambda \mathbf{I})^{-1}$ in order to include the direction of the error vector in the reduced space, but this is computationally inefficient. We thus seek low-cost approximations to the level shifted inverse Hessian in order to generate the reduced space vectors. The simplest form of $\mathbf{A}$ is to use the shifted Hessian diagonal $\mathbf{H}_D$ given in Eq. 15. Other choices will be discussed in Sect. 5.

$$\mathbf{A} = (\mathbf{H}_D - \lambda \mathbf{I})^{-1}. \tag{15}$$

The components of the reduced gradient, $\mathbf{g}_i^r$, and Hessian, $\mathbf{H}_{ij}^r$, in the $\mathbf{b}$-space is calculated from Eq. 16.

$$g_i^r = \mathbf{b}_i^t \mathbf{g} \quad \text{and} \quad H_{ij}^r = \mathbf{b}_i^t \mathbf{H} \mathbf{b}_j. \tag{16}$$

The geometry step, $\mathbf{\Delta x}$, is then calculated from Eq. 9 using the reduced gradient and Hessian. If the residual gradient, Eq. 12, is smaller than a suitable threshold, the geometry step is accepted, otherwise an additional $\mathbf{b}$-vector is generated from Eq. 13, followed by prediction of a new geometry step from Eqs. 16 and 9, and a new residual gradient from Eq. 12. For optimization of the LN function the $\mathbf{b}_1$ vector is defined as $(0, 0, 0, \ldots, 1)^t$ to ensure that the seam constraint is satisfied. The reduction in computational effort arises from replacing the diagonalization of the full Hessian with diagonalization of the reduced Hessian and a few matrix–vector multiplications. The size of the reduced space is typically between 50 and 200 vectors. We will denote this method by BNR.

As noted above, 6 of the $3N$ degrees of freedom correspond to overall translation and rotation of the system and the energy is independent on these six coordinates. It might be possible to obtain better convergence if these 6 degrees of freedom are removed from the system. For the NR method, Eq. 9, and the BNR method, Eqs. 12–16, the 6 degrees of freedom can be projected out of the Hessian by the projection operator in Eq. 17 [39].

$$\begin{aligned} \mathbf{P} = \mathbf{I} &- \mathbf{v}_x \mathbf{v}_x^t - \mathbf{v}_y \mathbf{v}_y^t - \mathbf{v}_y \mathbf{v}_y^t \\ &- \mathbf{v}_\alpha \mathbf{v}_\alpha^t - \mathbf{v}_\beta \mathbf{v}_\beta^t - \mathbf{v}_\gamma \mathbf{v}_\gamma^t. \end{aligned} \tag{17}$$

In Eq. 17, $\mathbf{v}_x$, $\mathbf{v}_y$, and $\mathbf{v}_x$ are vectors corresponding to translation in the x, y, and z direction respectively and $\mathbf{v}_\alpha$, $\mathbf{v}_\beta$, and $\mathbf{v}_\gamma$ are vectors corresponding to rotation around the principal axis of rotation.

For the regular NR method, Eq. 9, the 6 redundant degrees of freedom can be removed from the Hessian by forming the $\mathbf{PHP}^t$ matrix product, which ensures that the corresponding 6 eigenvalues becomes zero to within the numerical precision of the computer. This, however, requires two matrix–matrix multiplications that scales as the cube of the matrix size. If the NR equations are solved by diagonalization of the Hessian matrix, the additional effort by projecting out the TR degrees of freedom corresponds to only a few percent of the total computational time of the step and significantly improves the numerical stability of the algorithm.

In the BNR method, however, the step is formed in the space of a number of orthonormal $\mathbf{b}$-vectors generated from Eqs. 12 and 13. If the projected Hessian is used in Eq. 12 the 6 redundant variables are kept outside the $\mathbf{b}$-vector space, but the formation of the projected Hessian then rapidly becomes the computational bottleneck.

In order to avoid the matrix–matrix multiplications associated with the projection, the $\mathbf{b}$-vectors generated from the unprojected Hessian can be orthogonalized

against the 6 TR vectors. To force this orthogonalization, we include the 6 TR vectors explicitly in the **b**-vector space as the first components, such that all subsequent **b** vectors are orthogonal to these degrees of freedom.

We have been unable to find a way to incorporate this projection of TR modes into the BFGS method, Eq. 10, since the projected Hessian, $\mathbf{PHP}^t$, is singular and the inverse Hessian calculated in the BFGS method from the Woodbury matrix identity is only valid for the non-singular case [40].

4 Results

We have tested the efficiency of the seam optimization by modeling the symmetric proton transfer between a neutral (HIE) and protonated (HIP) histidine residue connected by a poly-alanine chain of varying length.

$$\mathrm{Hip(Ala)}_n\mathrm{Hie} \rightarrow \mathrm{Hie(Ala)}_n\mathrm{Hip}. \quad (18)$$

By using values of n in the range 5–400 in reaction (Eq. 18) we have generated systems varying in size between $\sim$100 and $\sim$4,000 atoms, corresponding to $\sim$300 to $\sim$12,000 variables in the target functions.

The starting geometry corresponds to an alpha-helix conformation, while the seam minimum has the two histidines at the ends of the peptide chain in close contact corresponding to distance between the two transferring nitrogen atoms of $\sim$2.7 Å. The optimization characteristics shown below thus reflect application purposes where the initial geometry is far from the seam minimum. Since a harmonic stretch energy term is used in the force field, the energy difference between the two initial structures is huge, typically $\sim$1,000 kJ/mol per atom.

For systems of the present size there are a large number of different seam minima and it is not likely that the different methods, or even the same method with different parameters, will result in the same optimized conformation. Since the starting geometry is far from any of the minima, the total computational cost is approximately independent on the final conformation and the results presented below should reflect the true performance of the methods.

The maximum step size in each optimization step for the NR and BNR methods has been chosen as 1.0 Å. We have used the Tinker package [41] to calculate the energies, gradients, and Hessians of the Amber94 force field in connection with our SEAM program to locate seam minima between the reactant and product energy surfaces.

All the optimization methods we consider can minimize a function but only the NR and BNR methods are able to optimize to a saddle point, as required for the LN function. We can thus combine the target functions and optimization algorithms in the following ways:

1. Penalty function with NR, BNR, or BFGS,
2. Lagrange function with NR or BNR;
3. Projection function with BFGS.

In addition, the NR and BNR methods can be used with and without removal of the TR modes by explicit projection. The BNR method can be used alternatively with orthogonalization of the **b**-vectors against the TR modes. The BFGS method has the number of previous gradient vectors stored as a free parameter, and we will consider the cases of using 5, 7, and 10 gradient vectors to construct the inverse Hessian. As a validation of the implementation, and for comparing with unconstrained optimizations, we also present results with the different methods for minimizing the energy of a single structure, i.e. the left hand side of the reaction (Eq. 18). In all the cases, we terminate the optimization when the gradient norm is below a numerical value of one measured in the units of kilocalories/moles and Angstrom. For the penalty function approach we start out with a weight factor $w = 1$ and then double the value of w when the gradient norm, $|\nabla \mathrm{Pen}(\mathbf{x})|$, is less than 25 until the energy difference, $|E_1 - E_2|$, is smaller than 0.01 kcal/mol.

4.1 Energy function

We have checked the implementation of the optimization algorithms by minimizing the force field energy of a single molecular structure, and Fig. 1 shows the number of iterations and the running time as a function of the number of atoms in the system. The number of iterations is almost independent on the size of the systems for all the optimization algorithms. This independence on system size is atypical and is most likely due to the starting geometry being very close to the minimized geometry. The Newton methods NR and BNR use a small number of iterations while the BFGS method typically uses two orders of magnitude more iterations. The time per iteration, however, is much lower for the BFGS method than for the Newton methods and the larger number of iterations does not necessarily indicate a lower overall efficiency.

The comparison in terms of total running time is shown in the bottom plot of Fig. 1 (note the logarithmic scale). For less than 500 atoms the NR method shows the best performance. In the intermediate range of 500–1,200 atoms the BFGS method is the most efficient, while the BNR method becomes preferable for systems with more than $\sim$1,200 atoms. Since the number of iterations is almost independent on the number of variables, the asymptotic performance of the NR method is $O(N^3)$. For the BFGS method with five gradient vectors we find the asymptotic behavior to be $O(N^{2.1})$. The BNR method determines a variable sized subspace for each iteration and system, and

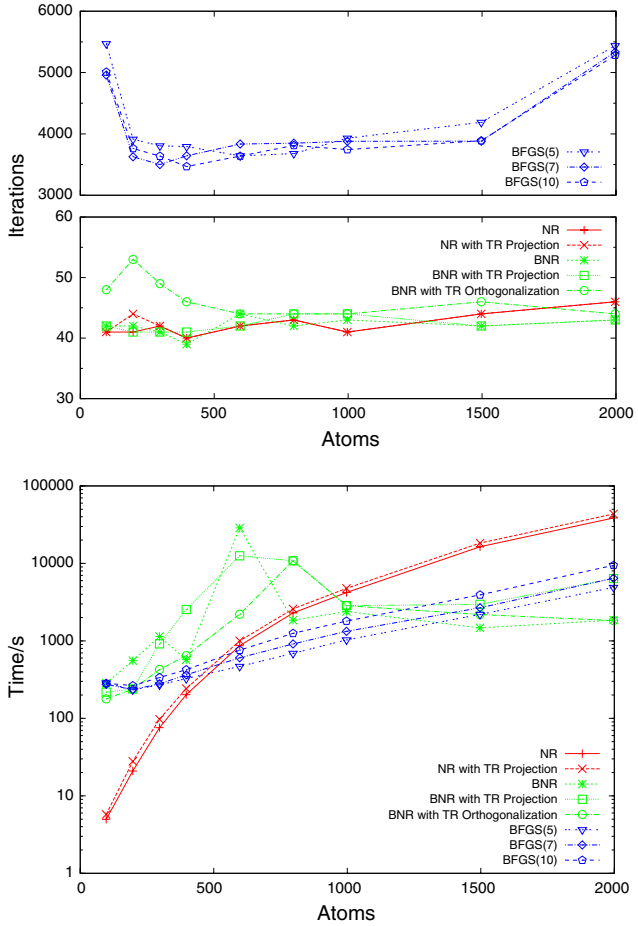

Fig. 1 Number of iterations (*top*) and running time (*bottom*) for minimization of the energy function

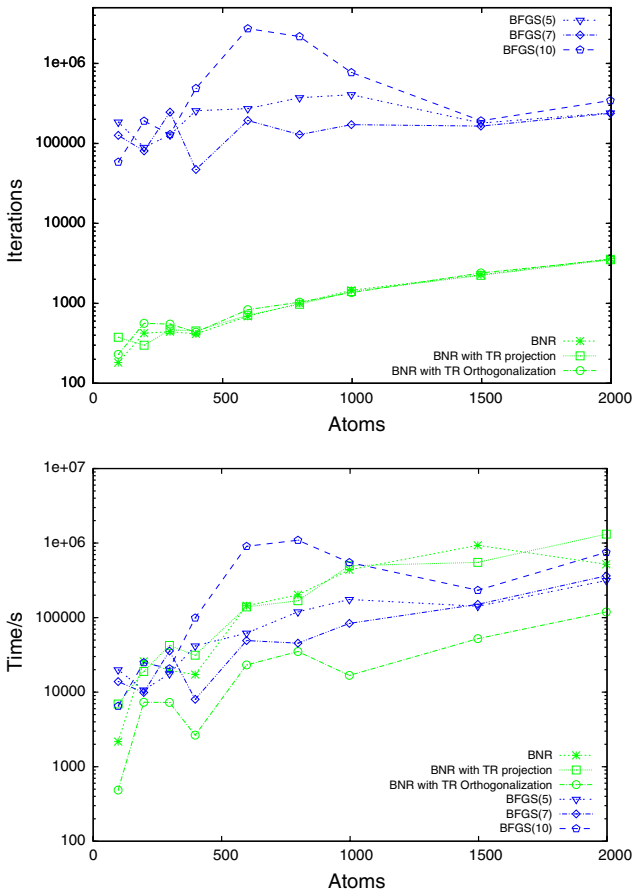

Fig. 2 Number of iterations (*top*) and running time (*bottom*) for minimization of the penalty function

the resulting running time behaves too irregular to determine an asymptotic behavior.

For minimizing the energy of a single structure there is little effect of removing the TR degrees of freedom when using the NR method. This may again be an effect due to starting from a near-optimum geometry. For the BNR method we observe a better performance for small systems when the TR directions are removed by projection of the Hessian but for large systems the projection degrades the performance due to the $O(N^3)$ nature of the matrix–matrix multiplications. Orthogonalization of the **b** vectors against the 6 TR vectors provide no N^3 computational penalty per iteration and is found to be the best choice.

4.2 Penalty function

Figure 2 shows the number of iterations and the total running time (note the logarithmic scale) for the seam optimization with the penalty function. In contrast to the unconstrained optimization of the energy function, the number of iterations increases with the system size and the overall computational cost is thus considerably higher.

The NR method performed poorly with the penalty function and failed to converge except for the smallest systems with $\sim$200 atoms. The BNR and BFGS methods in contrast converge in all cases. As with the energy function, the BFGS method uses two or three orders of magnitude more iterations than the BNR methods.

The BFGS method displays different behavior when changing the number of gradient vectors used to generate the inverse Hessian. The cases with 5 and 7 gradient vectors perform the best both with respect to the number of iterations and the total computational time.

In contrast to the unconstrained minimization of the energy of a single structure, location of a seam minimum proved to be very sensitive to removal of the TR degrees of freedom. This can be done in the BNR method by projection of the Hessian matrix, but since this is an N^3 process, it rapidly becomes the computational bottleneck for large systems. With the BNR method it is much more efficient to orthogonalize the **b**-vectors against the TR

vectors by including them as the first 6 **b**-vectors. For the ~2,000 atom case, this reduces the total running time by about a factor of 10 compared to projecting the Hessian and by about a factor of 5 compared to not orthogonalizing the **b**-vectors against the TR degrees of freedom. The most efficient optimization algorithm is for all system sizes found to be the BNR method with removal of the TR modes by orthogonalization.

4.3 Lagrange function

Figure 3 shows the number of iterations and the running time for optimizing the LN function with the NR and BNR methods. The number of iterations used is about the same for all methods and increases with the number of atoms. It is also noticeable that the number of iterations for optimizing the LN functions with the BNR method is about the same as for optimizing the penalty function, despite that the LN optimization is to a saddle point, while the penalty optimization is to a minimum.

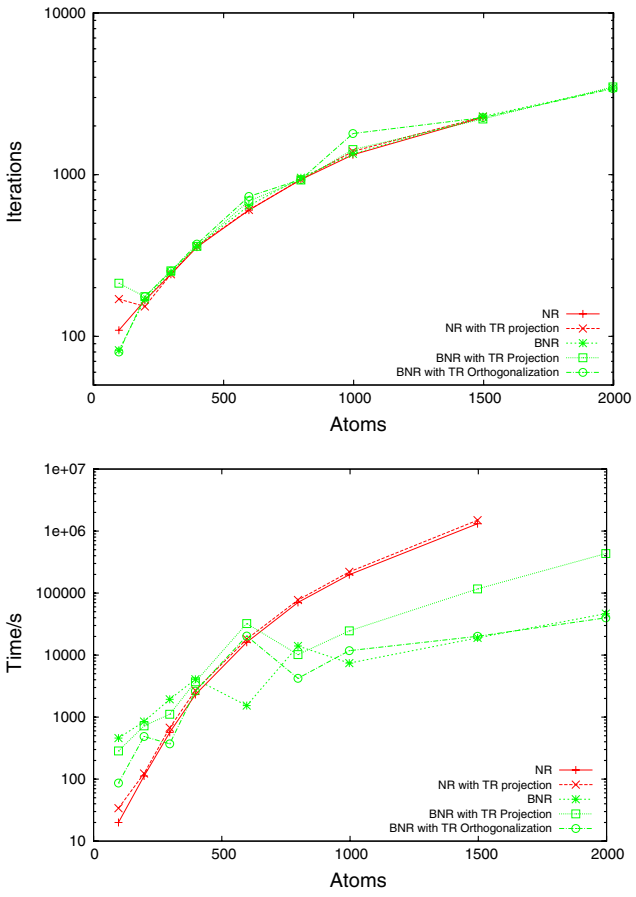

Fig. 3 Number of iterations (*top*) and running time (*bottom*) for optimization of the LN function

For small systems the NR method is the most efficient, but the $O(N^3)$ computational cost for diagonalizing the Hessian means that the total running time increases rapidly with the number of atoms. For system sizes of about 500 atoms the BNR method become faster and already for 1,000 atoms is the BNR method a factor of 10–50 more efficient than the NR method.

For small systems both the orthogonalization and projection methods improve the performance. For all system sizes orthogonalizing the **b**-vectors against the TR directions is computationally more efficient than the projection method.

As for the penalty function, the BNR method with **b**-vector orthogonalization is the overall best method for optimizing the LN function. When comparing the efficiency of locating seam minima with the penalty function, the LN function approach is about three times faster for systems with ~2,000 atoms.

4.4 Projection function

The projection function only works with the BFGS methods since only the gradient of the optimization function is available. The number of iterations and running time is shown in Fig. 4 and is in general slightly better with the projection function than with the penalty function and we again find that 5 or 7 gradient vectors give the best performance with the BFGS algorithm. The total running time, however, is approximately a factor of 5–20 longer than using the LN function with the BNR method. One reason for this is the use of the less efficient BFGS optimization compared to BNR, as seen in Fig. 2. Another possible reason may be related to fact that the projection operator only incorporates the constraint in a linear approximation. As shown in Eqs. 4 and 5, the operator for projection onto the seam subspace is given by the gradient difference direction, which means that the projection operator changes with each geometry steps. It is likely that this change in the minimization subspace contribute to the degraded performance relative to the LN approach, where the coupling between the minimization and maximization subspaces is included by the lagrange multiplier.

A second-order optimization scheme for the projection function is hampered by the non-symmetric nature of the Jacobian of the gradient associated with the projection function, Eq. 6. Silicia et al. [24] have recently proposed a second-order optimization scheme where a symmetric Hessian within the seam subspace is defined as the projected derivative of the seam energy gradient. The resulting algorithm employs a threshold parameter below which the direction perpendicular to the seam is handled separately, and this algorithm appears to display a better convergence near the seam minimum.

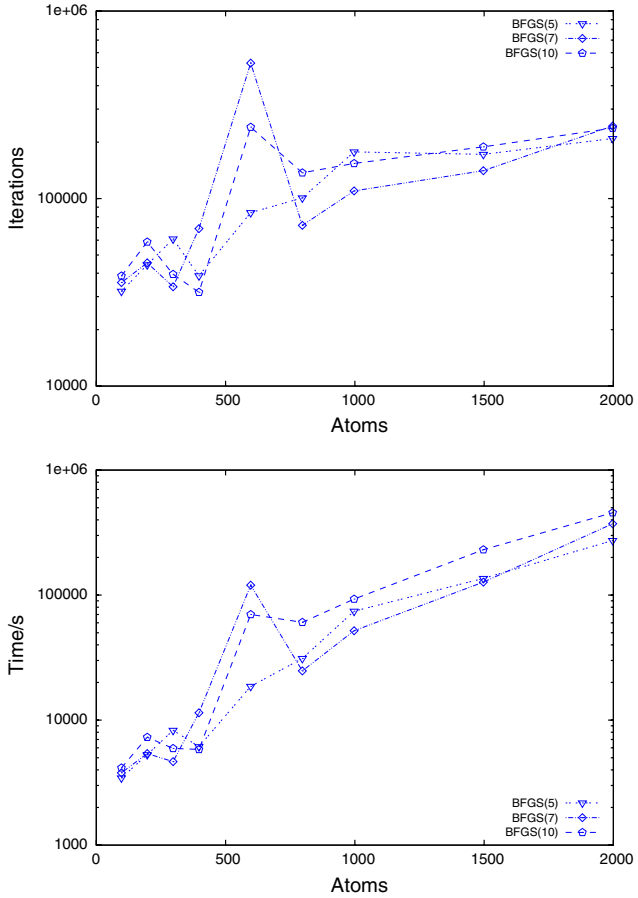

Fig. 4 Number of iterations (*top*) and running time (*bottom*) for minimization of the projection function

5 Improving the BNR method

In the previous section it was found that the LN function combined with the BNR method with orthogonalization of the **b**-vectors against the TR directions provides the best overall performance for locating minima on seams for systems with many variables. A variable factor in the BNR method is how the **b**-vectors are generated. A rapid generation of a small set of **b**-vectors covering a large fraction of the full Newton step is the key to a good performance. While Eq. 15 is a very simple approach, a better approximation to the inverse Hessian may be obtained by the BFGS method in Eq. 10, but the level shifting parameter prevents the use of this directly. If, however, λ is small compared to the Hessian diagonal elements, the level shifting is not important and can be neglected. When shifting of the diagonal elements is important an improvement over the diagonal Hessian can be obtained by using 3×3 blocks along the diagonal, corresponding to coupling between the x, y, and z coordinates of atom i. This scheme is shown in Eq. 19, where $\mathbf{H}_3$ is the Hessian consisting of the 3×3 atomic coordinate blocks, and the

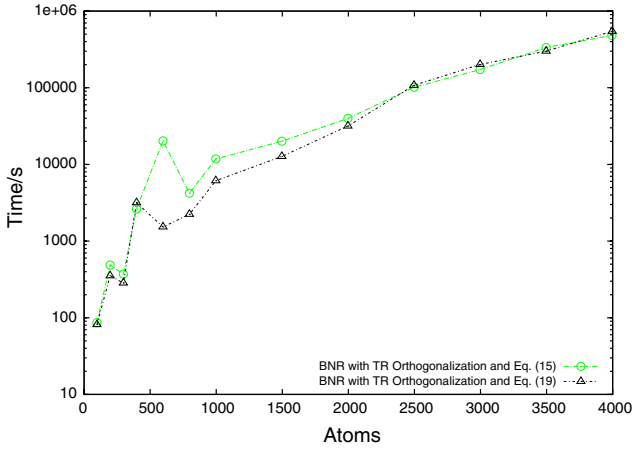

Fig. 5 Running time for optimization of the Lagrange function with the BNR method with orthogonalization of the **b**-vectors against the TR modes and using either Eq. 15 or Eq. 19

inversion of these block matrices can be performed analytically.

$$\mathbf{A} = \begin{cases} \text{Eq. } 10 & \text{if} \quad \lambda \ll max\mathbf{H}_D \\ (\mathbf{H}_3 - \lambda\mathbf{I})^{-1} & \text{otherwise} \end{cases} \tag{19}$$

We use Eq. 10 if $max\mathbf{H}_D/\lambda > 500$ and find that the method works best using 2 gradient vectors in the limited memory BFGS method.

Figure 5 shows the results of the LN function and the BNR method with orthogonalization of the **b**-vectors against the TR modes using either Eq. 15 or Eq. 19 to generate the **b**-vectors for systems with up to 4,000 atoms. The performance is slightly improved using Eq. 19 compared to Eq. 15 but the difference seems to diminish as the number of atoms increases.

6 Conclusion

We have investigated the computationally most efficient strategy for solving large-scale constrained optimizations, exemplified by locating minima on seams corresponding to two different force field energy functions depending on the same set of atomic coordinates. In agreement with other work [30], we find that the Lagrange–Newton approach performs better than using either a penalty or projection function. For locating minima on seams, it is important that redundant degrees of freedom, corresponding for example to overall translation and rotation of the system, are explicitly removed when generating the geometry step. For systems with less than about 500 atoms, the most efficient approach is to use a straight forward Newton–Raphson optimization scheme with explicit diagonalization of the Hessian. For larger systems it is preferably to solve the problem in a reduced vector space which can be generated in an iterative fashion.

We have in the present work used exact Hessians for the regular Newton–Raphson optimization and for calculating the residual gradient in the iterative Newton–Raphson method. This is motivated by the fact that Hessians are computational inexpensive to calculate when using force field energy functions. The use of an updating procedure for estimating the Hessian possesses no formal changes in the algorithms, but will affect the convergence rate and possibly change the conclusions regarding the most efficient combination of target function and optimization algorithm. We note, however, that Keal et al. [30] also found the Lagrange–Newton approach to be most efficient when used in connection with an updated Hessian. Our choice of starting geometries far from the seam minimum reflects a typical use for modeling chemical reactions within a force field environment [25, 26], and testifies that all of the present methods display a stable global convergence.

For illustrating the convergence properties of the different algorithms, we have employed the Amber94 force field, but a modeling of chemical reactions will require a more realistic function for the stretch energy, as for example a Morse potential, and the use of either implicit or explicit solvent. Neither of these improvements is expected to change the conclusions regarding the most efficient algorithm. The use of a Morse potential for the stretch energy has been implemented by switching from the native harmonic potential in Amber94 to a Morse potential once the seam minimization is close to converged. This strategy only adds a few extra iterations in the final refinement of the seam minimum. The use of an explicit solvent will increase the number of degrees of freedom, but as these will be almost exclusively in the minimization subspace, it should not change the convergence characteristics.

Acknowledgments This work was supported by grants from the Danish Natural Science Research Council, the Danish Center for Scientific Computing and the Lundbeck foundation.

References

1. Schlegel HB (2003) J Comput Chem 24:1514
2. Bernardi F, Ollivucci M, Robb MA, Chem (1996) Soc Rev 25:321
3. Robb MA, Garavelli M, Olivucci M, Bernardi F (2000) Rev Comp Chem 15:87
4. Yarkony DR (1996) Rev Mod Phys 68:985
5. Matsika S, Yarkony DR (2002) Adv Chem Phys 124:557
6. Matsika S (2007) Rev Comp Chem 23:83
7. Lundberg M, Siegbahn PEM (2005) J Phys Chem B 109:10513
8. Harvey JN, Aschi M, Schwarz H, Koch W (1998) Theor Chem Acc 99:95
9. Harvey JN (2007) Phys Chem Chem Phys 9:331
10. Manaa RM, Yarkony DR (1993) J Chem Phys 99:5251
11. Matsika S, Yarkony DR (2001) J Chem Phys 115:2038
12. Ciminelli C, Granucci G, Persico M (2004) Chem Eur J 10:2327
13. Koga N, Morokuma K (1985) Chem Phys Lett 119:371
14. Ragazos IN, Robb MA, Bernardi F, Ollivucci M (1992) Chem Phys Lett 197:217
15. Anglada JM, Bofill JM (1997) J Comput Chem 18:992
16. Yarkony DR (2004) J Phys Chem A 108:3200
17. Dallos M, Lischka H, Shepard R, Yarkony DR, Szalay PG (2004) J Chem Phys 120:7330
18. Farazdel A, Dupuis M (1991) J Comput Chem 12:276
19. Jensen F (1994) J Comput Chem 15:1199
20. Bearpark MJ, Robb MA, Schlegel HB (1994) Chem Phys Lett 223:269
21. Toniolo A, Ben-Nun M, Martinez TJ (2002) J Phys Chem A 106:4679
22. Izzo R, Klessinger M (2000) J Comput Chem 21:52
23. Chachiyo T, Rodriquez JH (2005) J Chem Phys 123:094711
24. Silicia F, Blancafort L, Bearpark MJ, Robb MA (2008) J Chem Theor Comput 4:257
25. Jensen F, Norrby PO (2003) Theor Chem Acc 109:1
26. Olsen PT, Jensen F (2003) J Chem Phys 118:3523
27. Lu D, Zhao M, Truhlar DG (1991) J Comput Chem 12:376
28. Baker J (1992) J Comput Chem 13:240
29. Baker J (1997) J Comput Chem 18:1079
30. Keal TW, Koslowski A, Thiel W (2007) Theor Chem Acc 118:837
31. Broyden CG (1970) J Inst Math Appl 6:76
32. Fletcher R (1970) Comput J 13:317
33. Goldfarb D (1970) Math Comput 24:23
34. Schanno DF (1970) Math Comput 24:647
35. Nocedal J (1980) Math Comput 35:773
36. Hansen MB, Jensen HJA, Jensen F (2009) Int J Quant Chem 109:373
37. Culot P, Dive G, Nguyen VH, Ghuysen JM (1992) Theor Chim Acta 82:189
38. Saad Y, Schultz HM, (1986) SIAM J Sci Stat Comput 7:856
39. Miller WH, Handy NC, Adams JE (1980) J Chem Phys 72:99
40. Hager WH (1989) SIAM Rev 31:221
41. Ponder JW, TINKER (2004) Software tools for molecular design 4.2. Washington University School of Medicine, Saint Louis, MO. http://dasher.wustl.edu/tinke. Accessed 1 April 2007