A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLPs — I. Theoretical Advances

C.S. Adjiman, S. Dallwig[1], C.A. Floudas[2] and A. Neumaier[1]
Department of Chemical Engineering, Princeton University,
Princeton, NJ 08544, USA

Abstract

In this paper, the deterministic global optimization algorithm, αBB, (α-based Branch and Bound) is presented. This algorithm offers mathematical guarantees for convergence to a point arbitrarily close to the global minimum for the large class of twice-differentiable NLPs. The key idea is the construction of a converging sequence of upper and lower bounds on the global minimum through the convex relaxation of the original problem. This relaxation is obtained by (i) replacing all nonconvex terms of special structure (i.e., bilinear, trilinear, fractional, fractional trilinear, univariate concave) with customized tight convex lower bounding functions and (ii) by utilizing some α parameters as defined by Maranas and Floudas (1994b) to generate valid convex underestimators for nonconvex terms of generic structure. In most cases, the calculation of appropriate values for the α parameters is a challenging task. A number of approaches are proposed, which rigorously generate a set of α parameters for general twice-differentiable functions. A crucial phase in the design of such procedures is the use of interval arithmetic on the Hessian matrix or the characteristic polynomial of the function being investigated. Thanks to this step, the proposed schemes share the common property of computational tractability and preserve the global optimality guarantees of the algorithm. However, their accuracy and computational requirements differ so that no method can be shown to perform consistently better than others for all problems. Their use is illustrated on an unconstrained and a constrained example.

The second part of this paper (Adjiman *et al.*, 1997) is devoted to the discussion of issues related to the implementation of the αBB algorithm and to extensive computational studies illustrating its potential applications.

---

[1]Institut für Mathematik,Universität Wien, Strudlhofgasse 4, A-1090 Wien, Austria
[2]Author to whom all correspondence should be addressed.

1 Introduction

A large proportion of all optimization problems arising in an industrial or scientific context are characterized by the presence of nonconvexities in some of the participating functions. Phase equilibrium, minimum potential energy conformation of molecules, distillation sequencing, reactor network design, batch process design are all but a few of the nonconvex nonlinear programming problems (NLPs) relevant to the chemical industry. The nonconvexities represent major hurdles in attaining the global optimal solution and circumventing them is a central theme of nonlinear optimization theory. Recent progress on global optimization methods and their applications to process synthesis and design, process control and computational chemistry is reviewed in Floudas and Grossmann (1995), Floudas and Pardalos (1996), Grossmann (1996) and Floudas (1997).

Many of the deterministic methods proposed to date rely on the generation of valid convex underestimators for the nonconvex functions involved. Successive improvements of these estimates, together with the identification of the global solution of the resulting convex programs, eventually lead to the determination of the global optimal solution of the original nonconvex problem. The GOP algorithm developed by Floudas and Visweswaran (1990, 1993, see also Visweswaran and Floudas, 1990, 1993, 1996a,b) is an instance of a decomposition method in which the special structure of the problem is exploited in order to construct valid underestimators. The branch-and-bound algorithm of Al-Khayyal and Falk (1983) takes advantage of the properties of bilinear functions for which the convex envelope can be determined explicitly (Al-Khayyal, 1990). A branch-and-bound algorithm applicable to many nonconvex problems relevant to chemical engineering was proposed by Smith and Pantelides (1996). The generation of convex underestimators is based on a symbolic reformulation of the problem that transforms complex nonconvex terms into simpler terms such as bilinear, univariate concave, convex, linear fractional and simple power terms. This is achieved through the addition of new variables and constraints to the original problem. Finally, the αBB algorithm (Maranas and Floudas (1994a,b), Androulakis *et al.* (1995)) is based on a branch-and-bound scheme in which a convex lower bounding function is generated at each node. This algorithm is applicable to the broad class of twice-differentiable functions as shown by Liu and Floudas (1993), and has been successfully used to identify all solutions of nonlinearly constrained systems of equations (Maranas and Floudas, 1995). In order to derive the required valid underestimator for a twice-differentiable function without making any further assumptions concerning its mathematical structure, Maranas and Floudas (1992, 1994a,b) suggested the subtraction of a separable positive quadratic term. Within this term, a nonnegative parameter α is assigned to each variable. The magnitude of these α parameters greatly influences the convergence rate of the algorithm. In the general case, the determination of α values which result in the construction of a tight yet valid convex underestimator is a difficult task. This matter is directly linked to the properties of the Hessian matrix of the function being studied over the domain of interest. Successful α computation methods must therefore curtail the intrinsic complexities of the Hessian matrix analysis. Because of the vast array of problems that can in theory be tackled by the αBB algorithm, the development of new methods to evaluate α values for arbitrarily complex twice-differentiable functions is of primary importance. In Part I of this paper, a general outline of the basic principles of the αBB algorithm is presented. It is followed by the detailed discussion of methodologies that address the α calculation issue. Finally, these procedures are applied to a constrained process design example.

2 The αBB Global Optimization Algorithm

The αBB algorithm operates within a branch-and-bound framework and is designed to solve nonconvex minimization problems of the generic type represented by formulation (1). The theoretical properties of the algorithm guarantee that such a problem can be solved to global optimality with finite ϵ-convergence.

$$\begin{array}{lrcl} \min_{\boldsymbol{x}} & f(\boldsymbol{x}) & & \\ s.t. & \boldsymbol{g}(\boldsymbol{x}) & \leq & 0 \\ & \boldsymbol{h}(\boldsymbol{x}) & = & 0 \\ & \boldsymbol{x} & \in & X \subseteq \Re^n \end{array} \tag{1}$$

where f, $\boldsymbol{g}$ and $\boldsymbol{h}$ belong to $\mathcal{C}^2$, the set of twice-differentiable functions, and $\boldsymbol{x}$ is a vector of size n.

Each iteration of the algorithm consists of a *branching* step and a *bounding* step. In the latter, a lower bound is obtained by constructing valid convex underestimators for the functions in the problem and solving the resulting convex NLP to global optimality. An upper bound is calculated either by solving the original problem locally over each subdomain of the solution space or by performing a problem evaluation at the solution of the lower bounding problem. The identification of the global optimal solution hinges on the validity of the lower bounding problems as well as the construction of increasingly tight lower bounding problems for successive partitions of the solution space. Such properties lead to the generation of a *nondecreasing sequence* of lower bounds which progresses towards the optimal solution. As can be expected, the convergence characteristics of the αBB algorithm are significantly affected by the quality of the underestimators used. The derivation of the lower bounding problem from the original problem plays an important role in the algorithmic procedure. Therefore, the success of the αBB algorithm largely depends on the ability to reconcile two conflicting goals, accuracy and efficiency, during that phase.

A determining step in the convexification strategy is the decomposition of each nonlinear function into a sum of terms belonging to one of several categories: linear, bilinear, trilinear, fractional, fractional trilinear, convex, univariate concave or general nonconvex. Not only can these terms be readily identified, but techniques can be devised in order to generate valid and in some cases very tight convex underestimators. Although it is possible to construct customized underestimators for other mathematical structures such as signomial expressions, they are not considered in this paper. A detailed description of the treatment of such terms can be in found in Maranas and Floudas (1997). In constructing a convex underestimator for the overall function, it is first noted that the linear and convex terms do not require any transformation. The convex envelope of the bilinear, trilinear, fractional, fractional trilinear and univariate concave terms can be constructed by following simple rules.

2.1 Underestimating Bilinear Terms

In the case of a bilinear term xy, Al-Khayyal and Falk (1983) showed that the tightest convex lower bound over the domain $[x^L, x^U] \times [y^L, y^U]$ is obtained by introducing a new variable w_B which replaces every occurrence of xy in the problem and satisfies the following relationship:

$$w_B = \max\{x^L y + y^L x - x^L y^L;\ x^U y + y^U x - x^U y^U\}. \tag{2}$$

This lower bound can be relaxed and included in the minimization problem by adding two linear inequality constraints,

$$\begin{array}{rcl} w_B & \geq & x^L y + y^L x - x^L y^L, \\ w_B & \geq & x^U y + y^U x - x^U y^U. \end{array} \tag{3}$$

Moreover, an upper bound can be imposed on w to construct a better approximation of the original problem (McCormick, 1976). This is achieved through the addition of two linear constraints:

$$\begin{array}{rcl} w_B & \leq & x^U y + y^L x - x^U y^L, \\ w_B & \leq & x^L y + y^U x - x^L y^U. \end{array} \tag{4}$$

2.2 Underestimating Trilinear Terms

A trilinear term of the form xyz can be underestimated in a similar fashion (Maranas and Floudas, 1995). A new variable w_T is introduced and bounded by the following eight inequality constraints:

$$\begin{array}{rcl} w_T & \geq & xy^L z^L + x^L y z^L + x^L y^L z - 2x^L y^L z^L, \\ w_T & \geq & xy^U z^U + x^U y z^L + x^U y^L z - x^U y^L z^L - x^U y^U z^U, \\ w_T & \geq & xy^L z^L + x^L y z^U + x^L y^U z - x^L y^U z^U - x^L y^L z^L, \\ w_T & \geq & xy^U z^L + x^U y z^U + x^L y^U z - x^L y^U z^L - x^U y^U z^U, \\ w_T & \geq & xy^L z^U + x^L y z^L + x^U y^L z - x^U y^L z^U - x^L y^L z^L, \\ w_T & \geq & xy^L z^U + x^L y z^U + x^U y^U z - x^L y^L z^U - x^U y^U z^U, \\ w_T & \geq & xy^U z^L + x^U y z^L + x^L y^L z - x^U y^U z^L - x^L y^L z^L, \\ w_T & \geq & xy^U z^U + x^U y z^U + x^U y^U z - 2x^U y^U z^U. \end{array} \tag{5}$$

2.3 Underestimating Fractional Terms

Fractional terms of the form x/y are underestimated by introducing a new variable w_F and two new constraints (Maranas and Floudas, 1995) which depend on the sign of the bounds on x.

$$\begin{array}{rcl} w_F & \geq & \begin{cases} x^L/y + x/y^U - x^L/y^U & \text{if } x^L \geq 0 \\ x/y^U - x^L y/y^L y^U + x^L/y^L & \text{if } x^L < 0 \end{cases} \\ w_F & \geq & \begin{cases} x^U/y + x/y^L - x^U/y^L & \text{if } x^U \geq 0 \\ x/y^L - x^U y/y^L y^U + x^U/y^U & \text{if } x^U < 0 \end{cases} \end{array} \tag{6}$$

2.4 Underestimating Fractional Trilinear Terms

For fractional trilinear terms, eight new constraints are required (Maranas and Floudas, 1995). The fractional trilinear term xy/z is replaced by the variable w_{FT} and the constraints for $x^L, y^L, z^L \geq 0$ are given by

$$\begin{array}{l} w_{FT} \geq xy^L/z^U + x^L y/z^U + x^L y^L/z - 2x^L y^L/z^U, \\ w_{FT} \geq xy^L/z^U + x^L y/z^L + x^L y^U/z - x^L y^U/z^L - x^L y^L/z^U, \\ w_{FT} \geq xy^U/z^L + x^U y/z^U + x^U y^L/z - x^U y^L/z^U - x^U y^U/z^L, \\ w_{FT} \geq xy^U/z^U + x^U y/z^L + x^L y^U/z - x^L y^U/z^U - x^U y^U/z^L, \\ w_{FT} \geq xy^L/z^U + x^L y/z^L + x^U y^L/z - x^U y^L/z^L - x^L y^L/z^U, \\ w_{FT} \geq xy^U/z^U + x^U y/z^L + x^L y/z - x^L y^U/z^U - x^U y^U/z^L, \\ w_{FT} \geq xy^L/z^U + x^L y/z^L + x^U y^L/z - x^U y^L/z^L - x^L y^L/z^U, \\ w_{FT} \geq xy^U/z^L + x^U y/z^L + x^U y^U/z - 2x^U y^U/z^L. \end{array} \tag{7}$$

2.5 Underestimating Univariate Concave Terms

Univariate concave functions are trivially underestimated by their linearization at the lower bound of the variable range. Thus the convex envelope of the concave function $ut(x)$ over $[x^L, x^U]$ is the linear function of x:

$$ut(x^L) + \frac{ut(x^U) - ut(x^L)}{x^U - x^L}(x - x^L). \tag{8}$$

The generation of the best convex underestimator for a univariate concave function does not require the introduction of additional variables or constraints.

2.6 Underestimating General Nonconvex Terms

For the most general nonconvexities, a slightly modified version of the underestimator proposed by Maranas and Floudas (1994b) is used. A function $f(\boldsymbol{x}) \in \mathcal{C}^2(\Re^n)$ is underestimated over the entire domain $[\boldsymbol{x}^L, \boldsymbol{x}^U]$ by the function $\mathcal{L}(\boldsymbol{x})$ defined as

$$\mathcal{L}(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{n} \alpha_i (x_i^L - x_i)(x_i^U - x_i) \tag{9}$$

where the α_i's are positive scalars.

Since the summation term in Equation (9) is negative over the entire region $[\boldsymbol{x}^L, \boldsymbol{x}^U]$, $\mathcal{L}(\boldsymbol{x})$ is a guaranteed underestimator of $f(\boldsymbol{x})$. Furthermore, since the quadratic term is convex, all nonconvexities in the original function $f(\boldsymbol{x})$ can be overpowered given sufficiently large values of the α_i parameters: $\mathcal{L}(\boldsymbol{x})$ is therefore a *valid convex underestimator*. Since $\mathcal{L}(\boldsymbol{x})$ is convex if and only if its Hessian matrix $H_{\mathcal{L}}(\boldsymbol{x})$ is positive semi-definite, a useful convexity condition is derived by noting that $H_{\mathcal{L}}(\boldsymbol{x})$ is related to the Hessian matrix $H_f(\boldsymbol{x})$ of $f(\boldsymbol{x})$ by

$$H_{\mathcal{L}}(\boldsymbol{x}) = H_f(\boldsymbol{x}) + 2\ \Delta, \tag{10}$$

where Δ is a diagonal matrix whose diagonal elements are the α_i's. Δ is referred to as the *diagonal shift matrix*, since the addition of the quadratic term to the function $f(\boldsymbol{x})$, as shown in Equation (9), corresponds to the introduction of a *shift* in the diagonal elements of its Hessian matrix $H_f(\boldsymbol{x})$. The following theorem can then be used to ensure that $\mathcal{L}(\boldsymbol{x})$ is indeed a convex underestimator:

Theorem 2.6.1 *$\mathcal{L}(\boldsymbol{x})$, as defined in Equation (9), is convex if and only if $H_f(\boldsymbol{x}) + 2\ \Delta = H_f(\boldsymbol{x}) + 2\ \mathrm{diag}\,(\alpha_i)$ is positive semi-definite for all $\boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U]$.*

A number of deterministic methods have been devised in order to automatically identify an appropriate diagonal shift matrix. They are discussed in detail in Section 3.

In addition, special expressions for the α parameters have been developed for the construction of a discontinuous underestimator for bilinear terms which matches the convex envelope of Section 2.1. This strategy is described in Appendix A.

2.7 Overall Valid Convex Underestimator

Based on the underestimators discussed for each of the term types identified, a convex underestimator for any given twice-differentiable function can now be obtained through a decomposition approach. A function $f(\boldsymbol{x})$ with continuous second-order derivatives can be written as

$$\begin{aligned} f(\boldsymbol{x}) &= LT(\boldsymbol{x}) + CT(\boldsymbol{x}) + \sum_{i=1}^{bt} b_i x_{B_i,1} x_{B_i,2} + \sum_{i=1}^{tt} t_i x_{T_i,1} x_{T_i,2} x_{T_i,3} \\ &+ \sum_{i=1}^{ft} f_i \frac{x_{F_i,1}}{x_{F_i,2}} + \sum_{i=1}^{ftt} ft_i \frac{x_{FT_i,1} x_{FT_i,2}}{x_{FT_i,3}} + \sum_{i=1}^{ut} UT_i(x^i) + \sum_{i=1}^{nt} NT_i(\boldsymbol{x}) \end{aligned}$$

where $LT(\boldsymbol{x})$ is a linear term; $CT(\boldsymbol{x})$ is a convex term; bt is the number of bilinear terms, $x_{B_i,1}$ and $x_{B_i,2}$ denote the two variables that participate in the ith bilinear term and b_i is its coefficient; tt is the number of trilinear terms, $x_{T_i,1}$, $x_{T_i,2}$ and $x_{T_i,3}$ denote the three variables that participate in the ith trilinear term and t_i is its coefficient; ft is the number of fractional terms, $x_{F_i,1}$ and $x_{F_i,2}$ denote the two variables that participate in the ith fractional term and f_i is its coefficient; ftt is the number of fractional trilinear terms, $x_{FT_i,1}$, $x_{FT_i,2}$ and $x_{FT_i,3}$ denote the three variables that participate in the ith fractional trilinear term and ft_i is its coefficient; ut is the number of univariate concave terms, $UT_i(x^i)$ is the ith univariate concave term, x^i denotes the variable that participates in UT_i; nt is the number of general nonconvex terms, $NT_i(\boldsymbol{x})$ is the ith general nonconvex term.
The corresponding lower bounding function is

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}, \boldsymbol{w}) = \quad & LT(\boldsymbol{x}) + CT(\boldsymbol{x}) + \sum_{i=1}^{bt} b_i w_{B_i} \\
& + \sum_{i=1}^{tt} t_i w_{T_i} + \sum_{i=1}^{ft} f_i w_{F_i} + \sum_{i=1}^{ftt} ft_i w_{FT_i} \\
& + \sum_{i=1}^{ut} \left(UT_i(x^{i,L}) + \frac{UT_i(x^{i,U}) - UT_i(x^{i,L})}{x^{i,U} - x^{i,L}} (x - x^{i,L}) \right) \\
& + \sum_{i=1}^{nt} \left(NT_i(\boldsymbol{x}) + \sum_{j=1}^{n} \alpha_{ij} (x_j^L - x_j)(x_j^U - x_j) \right)
\end{aligned}
\tag{11}
$$

where α_{ij} corresponds to term i and variable j and satisfies Theorem 2.6.1. The w_{B_i} variables are defined by Equations (3) and (4). The w_{T_i}, w_{F_i} and w_{FT_i} variables must satisfy constraints of the forms given by Equations (5), (6) and (7) respectively.

Every customized underestimator discussed is a function of the size of the domain under consideration. Because the αBB algorithm follows a branch-and-bound approach, this domain is systematically reduced at each new node of the tree so that tighter lower bounding functions can be generated through updates of Equations (3–9). Thus the lower bounds on the problem form a non-decreasing sequence, as required for the identification of the global optimal solution. In addition, the quality of the variable bounds provided by the user can be expected to greatly influence the convergence of the algorithm and the investment of computational time in the calculation of tighter bounds is likely to result in improved performance. This fact will be exemplified in the computational studies presented in Part II of this paper.

In theory, any twice-differentiable function can be treated as a single general nonconvex term. However, the decomposition of functions into terms can improve the performance of αBB algorithm in two ways. First, the use of tight underestimators for certain types of terms increases the quality of the lower bounds generated for the optimum solution. Second, it will be shown in Section 3 that the construction of a convex underestimator for a general nonconvex term is dependent on the dimensionality of the term to order $\mathcal{O}(n^2)$ or higher. The separation of large terms into terms involving a smaller number of variables therefore results in a decrease in computational expense. Interestingly, the structure of many physical problems lends itself naturally to such a decomposition.

2.8 Equality Constraints

In order to generate a valid lower bound on the global solution of the nonconvex problem, the underestimating NLP generated in each domain must be convex. This implies that all inequality constraints in the lower bounding problem must be convex, all equality constraints must be linear and that the size of the feasible region must be increased relative to that of the original nonconvex problem. One of two strategies can be used to underestimate a nonlinear equality depending on the type of terms it involves. The first approach is used for equalities in which only linear, bilinear, trilinear, fractional and fractional trilinear terms appear. The nonlinear terms are replaced by new variables which participate linearly in the problem. The equality resulting from the substitution is therefore linear. Moreover, since the set of values these new variables can take on is a superset of the values that can be attained by the nonlinear terms, the linear equality corresponds to an enlarged feasible region. Thus, given the equality

$$\begin{aligned} LT(\boldsymbol{x}) &+ \sum_{i=1}^{bt} b_i x_{B_i,1} x_{B_i,2} + \sum_{i=1}^{tt} t_i x_{T_i,1} x_{T_i,2} x_{T_i,3} \\ &+ \sum_{i=1}^{ft} f_i \frac{x_{F_i,1}}{x_{F_i,2}} + \sum_{i=1}^{ftt} ft_i \frac{x_{FT_i,1} x_{FT_i,2}}{x_{FT_i,3}} = 0, \end{aligned}$$

the following underestimator can be used

$$LT(\boldsymbol{x}) + \sum_{i=1}^{bt} b_i w_{B_i} + \sum_{i=1}^{tt} t_i w_{T_i} + \sum_{i=1}^{ft} f_i w_{F_i} + \sum_{i=1}^{ftt} ft_i w_{FT_i} = 0,$$

where the notation is as previously specified, and the appropriate inequality constraints for the $\boldsymbol{w}$ variables are added to the problem.

If the nonlinear equality constraint contains convex or general nonconvex terms, the equality obtained by simple substitution of the corresponding underestimators is nonlinear. If it contains univariate concave terms, it is linear but it corresponds to a different feasible region. In the presence of convex, general nonconvex or univariate concave terms, the original equality $h(\boldsymbol{x}) = 0$ must therefore be rewritten as two inequalities of opposite signs,

$$\begin{cases} h(\boldsymbol{x}) & \leq 0 \\ -h(\boldsymbol{x}) & \leq 0. \end{cases}$$

These two inequalities must then be underestimated independently. The univariate concave terms appearing in the nonconvex equality become convex in one of the two inequalities while the convex terms become concave and the general nonconvex terms become convex or remain nonconvex.

The only remaining obstacle to the rigorous formulation of a valid convex lower bounding problem resides in the selection of appropriate values for the α parameters in Equation (9).

3 Rigorous calculation of α for general NLPs

The focus of this section is the development of methods that generate rigorously an appropriate diagonal shift matrix, that is, a set of α parameters satisfying Theorem 2.6.1. This allows the construction of a convex underestimator $\mathcal{L}(\boldsymbol{x})$ for a twice-differentiable function $f(\boldsymbol{x})$ over a specified domain. Two classes of approaches to this problem are defined:

- Uniform diagonal shift of the Hessian matrix of $f(\boldsymbol{x})$,
- Non-uniform diagonal shift of the Hessian matrix of $f(\boldsymbol{x})$.

Before the proposed procedures are described, two fundamental issues must be examined: the computational complexity of the α calculation problem and the design of criteria for the assessment of the calculation methods.

3.1 Tractability of α Calculation Techniques

As seen in Equation (10) and Theorem 2.6.1, the diagonal shift matrix Δ is closely linked to the Hessian matrix $H_f(\boldsymbol{x})$ of the function being underestimated. For general twice-differentiable functions, the elements of the Hessian matrix $H_f(\boldsymbol{x})$ are likely to be nonlinear functions of the variables, so that the derivation of a matrix Δ valid over the entire underestimation domain is a very difficult task. Yet, satisfying the convexity condition of Theorem 2.6.1 is essential for the preservation of the guarantee of global optimality. The difficulties arising from the presence of the variables in the convexity condition can be alleviated through the transformation of the exact $\boldsymbol{x}$-dependent Hessian matrix to an interval matrix $[H_f]$ such that $H_f(\boldsymbol{x}) \subseteq [H_f]$, $\forall \boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U]$. The elements of the original Hessian matrix are treated as independent when calculating their natural interval extensions (Ratschek and Rokne, 1988; Neumaier, 1990). The interval Hessian matrix family $[H_f]$ is then used to formulate a theorem in which the α calculation problem is relaxed.

Theorem 3.1.1 *Consider a general function $f(\boldsymbol{x})$ with continuous second-order derivatives and its Hessian matrix $H_f(\boldsymbol{x})$. Let $\mathcal{L}(\boldsymbol{x})$ be defined by Equation (9). Let $[H_f]$ be a real symmetric interval matrix such that $H_f(\boldsymbol{x}) \subseteq [H_f]$, $\forall \boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U]$. If the matrix $[H_\mathcal{L}]$ defined by $[H_\mathcal{L}] = [H_f] + 2\ \Delta = [H_f] + 2\ \mathrm{diag}\,(\alpha_i)$ is positive semi-definite, then $\mathcal{L}(\boldsymbol{x})$ is convex over the domain $[\boldsymbol{x}^L, \boldsymbol{x}^U]$.*

The domain of validity of the underestimator, $[\boldsymbol{x}^L, \boldsymbol{x}^U]$, participates in the interval convexity condition implicitly, through the interval matrix. The use of interval arithmetic serves two purposes: it reduces the computational complexity of the α calculations and it allows the preservation and transfer of global information.

3.2 Assessment of α calculation methods

The quality of the underestimator generated by any given α calculation method can be measured in terms of the separation distance between the nonconvex function and its underestimator: the tighter the lower bounding scheme, the faster the convergence. For this purpose, the maximum separation distance between $f(\boldsymbol{x})$ and $\mathcal{L}(\boldsymbol{x})$, d_{max} can be used. Maranas and Floudas (1994b), showed that it is directly proportional to the α_i's and given by

$$d_{max} = \max_{\boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U} (f(\boldsymbol{x}) - \mathcal{L}(\boldsymbol{x})) = \frac{1}{4} \sum_{i=1}^{n} \alpha_i (\boldsymbol{x}_i^U - \boldsymbol{x}_i^L)^2 \qquad (12)$$

In addition, the α parameters and the bounds on the variables can be shown to affect the maximum number of iterations required in order to achieve ϵ-convergence (Maranas and Floudas, 1994b).

The maximum separation distance is used in subsequent sections to evaluate the accuracy of each calculation method. Accuracy in itself is not sufficient and the trade-off between accuracy and computational expense plays a pivotal role in the assessment of any given method. Finally, regardless of the method being used, the size of the intervals within the Hessian matrix affects the final accuracy of the computation. The interval arithmetic required for this step should therefore be performed in a way that limits overestimates.

The examples presented in Parts I and II serve to illustrate how each proposed method affects the overall performance of the αBB algorithm, both in terms of the number of iterations and computational expense. Throughout the description of the α calculation procedures, the small example presented in the next section is used as an illustration.

3.3 Illustrative Example

The illustrative example is a nonlinear optimization problem in two variables with bound constraints.

$$\begin{aligned} \min_{x,y} f(x,y) &= \cos x \sin y - \tfrac{x}{y^2+1} \\ -1 &\leq x \leq 2 \\ -1 &\leq y \leq 1 \end{aligned} \tag{13}$$

Three minima were identified with the local optimization software MINOS5.5 (Murtagh and Saunders, 1983) in 1000 runs: $f^1 = -2.02181$ at $(x^1, y^1) = (2, 0.10578)$, $f^2 = -0.99495$ at $(x^2, y^2) = (0.63627, -1)$ and $f^3 = 0.95465$ at $(x^3, y^3) = (-1, 1)$.

The objective function is the sum of two nonlinear terms, shown in Figure 1. Two different approaches are available for the derivation of interval Hessian matrices: in *Case 1*, a single Hessian is obtained for the objective function whereas in *Case 2*, a Hessian matrix is derived for each term.

3.3.1 Case 1

Based on the Hessian matrix $H_f(x,y)$ of $f(x,y)$, an interval Hessian family $[H_f]$ which contains $H_f(x,y)$ over the domain of interest is obtained.

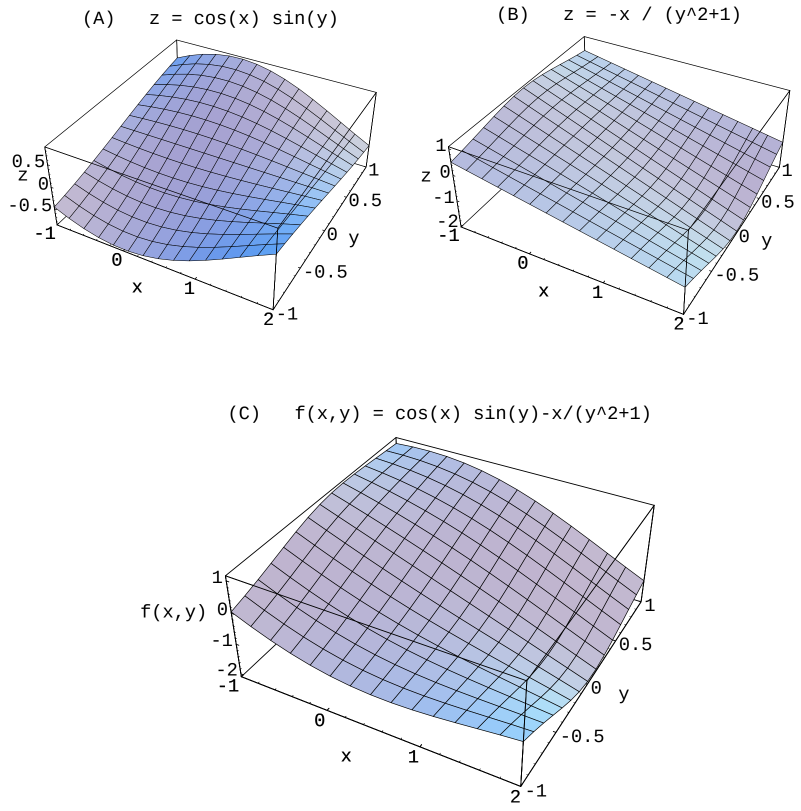

Figure 1: Illustrative Example – The objective function (C) is the sum of two terms (A and B)

$$H_f(x,y) = \begin{pmatrix} -\cos x \sin y & -\sin x \cos y + \frac{2y}{(y^2+1)^2} \\ \\ -\sin x \cos y + \frac{2y}{(y^2+1)^2} & -\cos x \sin y + \frac{2x(y^2+1)^2 - 8xy^2(y^2+1)}{(y^2+1)^4} \end{pmatrix}$$

is such that

$$H_f(x,y) \subseteq [H_f] = \begin{pmatrix} [-0.84148,\ 0.84148] & [-3.00000,\ 2.84148] \\ [-3.00000,\ 2.84148] & [-40.84148,\ 32.84148] \end{pmatrix}$$

for $-1 \leq x \leq 2$ and $-1 \leq y \leq 1$. The second-order derivatives in $H_f(x,y)$ were generated by the automatic differentiation package built in the implementation of the αBB algorithm. Although the expression for the second-order derivative with respect to y could be simplified further, leading to more accurate interval calculations, it has been kept as is to ensure accurate representation of the algorithm's performance. Since $[H_f]$ was obtained through natural interval extensions, it is not the smallest interval Hessian family which contains $H_f(x,y)$. If the exact minimum and maximum values of the Hessian elements are calculated through global optimization, the smallest achievable intervals can be identified. Following this procedure, it was found that the optimal interval Hessian matrix is in fact

$$[H_f]^* = \begin{pmatrix} [-0.84148,\ 0.84148] & [-1.52288,1.38086] \\ [-1.52288,1.38086] & [-2.00608,4.00181] \end{pmatrix}.$$

The largest overestimate occurs for the most nonlinear term, the second-order derivative with respect to y.

3.3.2 Case 2

The trigonometric term (Term A) and the fractional term (Term B) are treated separately.

Term A The Hessian matrix $H_A(x,y)$ and the interval Hessian family $[H_A]$ are given by

$$H_A(x,y) = \begin{pmatrix} -\cos x \sin y & -\sin x \cos y \\ -\sin x \cos y & -\cos x \sin y \end{pmatrix}$$

and

$$[H_A] = \begin{pmatrix} \text{[-0.84148, 0.84148]} & \text{[-1.00000,0.84148]} \\ \text{[-1.00000, 0.84148]} & \text{[-0.84148, 0.84148]} \end{pmatrix}.$$

In this case, $[H_A]$ is the smallest interval family that contains $H_A(x, y)$.

Term B The Hessian matrix $H_B(x, y)$ and the interval Hessian family $[H_B]$ are given by

$$H_B(x, y) = \begin{pmatrix} 0 & \frac{2y}{(y^2+1)^2} \\ \frac{2y}{(y^2+1)^2} & \frac{2x(y^2+1)^2-8xy^2(y^2+1)}{(y^2+1)^4} \end{pmatrix}$$

and

$$[H_B] = \begin{pmatrix} \text{[0,0]} & \text{[-2,2]} \\ \text{[-2,2]} & \text{[-40,32]} \end{pmatrix}.$$

The optimal interval Hessian matrix is

$$[H_B]^* = \begin{pmatrix} \text{[0.00000,0.00000]} & \text{[-0.64952, 0.64952]} \\ \text{[-0.64952, 0.64952]} & \text{[-2.00000, 4.00000]} \end{pmatrix}.$$

3.3.3 Study of the Illustrative Example

In the following sections, every proposed approach for the calculation of α parameters for $f(x, y)$ will be applied to the illustrative example in order to gather the following information:

- Value of α in the initial domain $(x, y) \in [-1, 2] \times [-1, 1]$.
- Maximum separation distance d_{max} between $f(x, y)$ and its underestimator.
- Number of iterations required for convergence with a relative tolerance of 10^{-3}.

These will be obtained for *Case 1* and *Case 2*. Since the CPU time for this problem is less than 1 second, it cannot be used for a meaningful comparison of the different methods.

3.4 Uniform Diagonal Shift Matrix

For this class of methods, the underestimator $\mathcal{L}(\boldsymbol{x})$ is re-formulated using a single α value:

$$\mathcal{L}(\boldsymbol{x}) = f(\boldsymbol{x}) + \alpha \sum_i (x_i^L - x_i)(x_i^U - x_i). \tag{14}$$

All the nonzero elements of the diagonal shift matrix Δ are therefore equal to α. Maranas and Floudas (1994b) showed that $\mathcal{L}(\boldsymbol{x})$ as defined by Equation (14) is convex if and only if

$$\alpha \geq \max\{0, -\frac{1}{2} \min_{i, \boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U} \lambda_i(\boldsymbol{x})\} \tag{15}$$

where the $\lambda_i(\boldsymbol{x})$'s are the eigenvalues of $H_f(\boldsymbol{x})$, the Hessian matrix of the function $f(\boldsymbol{x})$.

If $f(\boldsymbol{x})$ is convex, all eigenvalues of $H_f(\boldsymbol{x})$ are nonnegative for any $\boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U]$ and, by Equation (15), $\alpha = 0$: the original function appears unchanged in the lower bounding problem. For a nonconvex function, a measure of the degree of nonconvexity of the function is introduced through the use of the most negative eigenvalue in the construction of the underestimator: the more nonconvex the function, the smaller its minimum eigenvalue and hence the larger α.

The minimization problem which appears in Equation (15) can be written explicitly as:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}, \lambda} & \lambda \\ s.t. & H_f(\boldsymbol{x}) + \lambda I = 0 \\ & \boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U] \end{array}$$

where I is the identity matrix.

This is, in general, a difficult nonconvex optimization problem. Maranas and Floudas (1994b) suggested that a lower bound on the smallest eigenvalue of $H_f(\boldsymbol{x})$ could be obtained by using the measure of the Hessian matrix. However, this approach requires the solution of a convex programming problem based on the second-order derivatives of the function being underestimated and entails a large amount of computational effort. A valid lower bound on the minimum eigenvalue of $H_f(\boldsymbol{x})$ can be more easily obtained when the interval Hessian matrix $[H_f] \supseteq H_f(\boldsymbol{x})$ is introduced. All the methods presented in this section generate a single α value which satisfies the following sufficient condition for the convexity of $\mathcal{L}(\boldsymbol{x})$:

$$\alpha \geq \left\{0, -\frac{1}{2}\lambda_{min}\left([H_f]\right)\right\} \tag{16}$$

where $\lambda_{min}\left([H_f]\right)$ is the minimum eigenvalue of the interval matrix family $[H_f]$.

One $\mathcal{O}(n^2)$ method and a variety of $\mathcal{O}(n^3)$ methods have been developed to compute a bound on the minimum eigenvalue of a symmetric interval matrix. Before they are exposed, the illustrative example is revisited in the context of the uniform diagonal shift approach.

3.4.1 Eigenvalue Analysis for the Illustrative Example

The minimum eigenvalues for the exact Hessian matrices $H_f(x, y)$, $H_A(x, y)$ and $H_B(x, y)$ are compared to the minimum eigenvalues of the corresponding interval Hessian matrices.

Entire Function The exact minimum eigenvalue of $H_f(x, y)$ is $\lambda^e_{min} = -2.3934$ for $(x, y) \in [-1, 2] \times [-1, 1]$. For $[H_f]$, the minimum eigenvalue is -41.0652 and it is -3.0817 for $[H_f]^*$. Based on Equation (16), a value of α of 20.5326 is the best that can be obtained with the uniform diagonal shift approach. According to the exact eigenvalue calculation, a value of 1.1967 would suffice to guarantee the convexity of the underestimator. This illustrates the importance of careful interval calculations during the construction of the interval Hessian matrix.

Term A The exact minimum eigenvalue of $H_A(x, y)$ is $\lambda^e_{min,A} = -1.0$ and that of $[H_A]$ is -1.84148.

Term B The exact minimum eigenvalue of $H_B(x, y)$ is $\lambda^e_{min,B} = -2.0$. For $[H_B]$, the minimum eigenvalue is -40.0998 and for $[H_B]^*$, the optimal interval Hessian matrix for Term B, it is -2.17636.

As both x and y participate in the two terms, the overall exact α parameter for each of these variables is derived from the sum of the exact minimum eigenvalues for Term A and Term B ($\lambda^e_{min,A} + \lambda^e_{min,B} = -3.0$). Whereas the exact α is 1.1967 in Case 1, the value obtained in Case 2 is $\alpha = 1.5$. Thus, considering the two terms separately yields a looser underestimator. This observation is not maintained for all of the methods to be presented.

3.4.2 $\mathcal{O}(n^2)$ Method

Method I.1: Gerschgorin's Theorem for Interval Matrices

This first method is the straightforward extension of Gerschgorin's theorem (Gerschgorin, 1931) to interval matrices. While its computational complexity is only of order n^2, the bounds it provides on the eigenvalues are often loose.

For a real matrix $A = (a_{ij})$, the well-known theorem states that the eigenvalues are bounded below by λ_{min} such that

$$\lambda_{min} = \min_i \left(a_{ii} - \sum_{j \neq i} |a_{ij}| \right). \tag{17}$$

In the case of interval matrices, the following extended theorem can be used.

Theorem 3.4.1 *For an interval matrix $[A] = ([\underline{a}_{ij}, \overline{a}_{ij}])$, a lower bound on the minimum eigenvalue is given by*

$$\lambda_{min} \geq \min_i \left[\underline{a}_{ii} - \sum_{j \neq i} \max \left(|\underline{a}_{ij}|, |\overline{a}_{ij}| \right) \right].$$

Proof By definition, $\lambda_{min}([A]) \geq \min\limits_{A \in [A]} \lambda_{min}(A)$. Therefore,

$$\begin{aligned}
\lambda_{min}([A]) &\geq \min_{A \in [A]} \min_i \left(a_{ii} - \sum_{j \neq i} |a_{ij}| \right) \\
&\geq \min_i \left[\min_{A \in [A]} (a_{ii}) - \max_{A \in [A]} \left(\sum_{j \neq i} |a_{ij}| \right) \right] \\
&\geq \min_i \left[\underline{a}_{ii} - \sum_{j \neq i} \max \left(|\underline{a}_{ij}|, |\overline{a}_{ij}| \right) \right].
\end{aligned}$$

■

Illustrative Example The application of the extended Gerschgorin theorem to the illustrative example is summarized in Table 1. The calculation of one eigenvalue for the entire function (Case 1) and the use of eigenvalues for each of the two terms in the function (Case 2) yield the same results. Because this scheme relies on the addition and subtraction of the matrix elements, decomposition of the function has no effect on the final outcome when all the bounds on the term eigenvalues are negative, as is the case in this example. If, on the contrary, one of the terms in the decomposition is convex, its positive eigenvalues need not be taken into account, as indicated by Equation (16). By neglecting to add positive contributions, such an approach would result in a decrease of the overall eigenvalues or an increase in α. It therefore seems appropriate not to decompose the nonlinear terms when using this α calculation procedure. Further, it would appear that for a function known to involve convex and nonconvex terms, the inclusion of the convex terms with the general nonlinear terms could lead to an improvement in the quality of the underestimator. Caution must however be exercised as the overestimations resulting from the use of interval arithmetic may cause a failure to identify convexity. If the calculated lower bounds on the positive eigenvalues are negative, the underestimator will be looser than necessary. In addition, decomposition of the nonlinear terms may result in the construction of smaller Hessian matrices and reduce the overall computational effort. No general conclusions can be drawn on the treatment of convex terms when using the extended Gerschgorin theorem.

Case	λ_{min}	α	d_{max}	Iterations
1	-43.85	21.93	71.24	19
2	-43.85	21.93	71.24	20

Table 1: Results for the illustrative example using the Gerschgorin theorem.

3.4.3 $\mathcal{O}(n^3)$ Methods

The following definitions are needed for $\mathcal{O}(n^3)$ methods. Given an interval matrix $[A]$ with elements $[\underline{a}_{ij}, \overline{a}_{ij}]$,

- its radius matrix $\Delta A = (\Delta a_{ij})$ is such that $\Delta a_{ij} = \frac{\overline{a}_{ij} - \underline{a}_{ij}}{2}$,
- its *modified* radius matrix $\widetilde{\Delta A} = (\widetilde{\Delta a}_{ij})$ is such that

$$\widetilde{\Delta a}_{ij} = \begin{cases} 0 & \text{if } i = j \\ \Delta a_{ij} & \text{otherwise} \end{cases}$$

- its midpoint matrix $A_M = (a_{M,ij})$ is such that $a_{M,ij} = \frac{\overline{a}_{ij}+\underline{a}_{ij}}{2}$,
- its *modified* midpoint matrix $\tilde{A}_M = (\tilde{a}_{M,ij})$ is such that

$$\tilde{a}_{M,ij} = \begin{cases} \underline{a}_{ij} & \text{if } i = j \\ a_{M,ij} & \text{otherwise} \end{cases}$$

- a vertex matrix $A_v = (a_{v,ij})$ of $[A]$ is such that $a_{v,ij} = \underline{a}_{ij}$ or $a_{v,ij} = \overline{a}_{ij}$.

Note that $\Delta A = \widetilde{\Delta A} + \mathrm{diag}(\Delta A)$ and $A_M = \tilde{A}_M + \mathrm{diag}(\Delta A)$.

Method I.2: E-Matrix Method

This method is an extension of the theorems developed by Deif (1991) and Rohn (1996) for the calculation of a lower bound on all the eigenvalues of an interval matrix. While they obtained expressions for the real and imaginary parts of the matrices, only the real parts are of concern for the symmetric matrices being considered here.

The following result is required to derive the main result for this method.

Lemma Let $[a] = [\underline{a}, \overline{a}]$ be a single interval with midpoint a_M and radius Δa. Then for all scalars γ and all $a \in [a]$,

$$\gamma\ a \geq \gamma a_M - |\gamma|\Delta a. \tag{18}$$

Proof If $\gamma = 0$, Equation (18) is trivially valid. For $\gamma \neq 0$,

$$\gamma(a - a_M) \geq -|\gamma|\Delta a \Leftrightarrow \begin{cases} (a_M - a) \leq \Delta a & \text{for } \gamma > 0 \\ -(a_M - a) \leq \Delta a & \text{for } \gamma < 0 \end{cases}$$

The right-hand side is always true because the distance between the midpoint and any point in the interval can never be greater than the radius of the interval. ∎

Theorem 3.4.2 *Let E be an arbitrary real symmetric matrix. Given a symmetric interval matrix $[A]$, its modified midpoint matrix $\tilde{A}_M$ and its modified radius matrix $\widetilde{\Delta A}$, the minimum eigenvalue $\lambda_{min}(A)$ of any symmetric matrix $A \in [A]$ is such that*

$$\lambda_{min}(A) \geq \lambda_{min}\left(\tilde{A}_M + E\right) - \rho\left(\widetilde{\Delta A} + |E|\right)$$

where $\rho(M)$ denotes the spectral radius of a matrix M, i.e. its maximal absolute eigenvalue, and $|E|$ is the absolute value taken componentwise.

Proof For all $A = (a_{ij}) \in [A]$,

$$\begin{aligned} \boldsymbol{x}^T A \boldsymbol{x} &= \sum_i a_{ii} x_i^2 + \sum_{j \neq i} x_i a_{ij} x_j & (19) \\ &\geq \sum_i \underline{a}_{ii} x_i^2 + \sum_{j \neq i} x_i a_{ij} x_j . & (20) \end{aligned}$$

Setting $x_i x_j = \gamma$ and using the Lemma in the second term of the right hand side of (20), we find

$$\boldsymbol{x}^T A \boldsymbol{x} \geq \sum_i \underline{a}_{ii} x_i^2 + \sum_{j \neq i} x_i a_{M,ij} x_j - \sum_{j \neq i} |x_i| \Delta a_{ij} |x_j| . \tag{21}$$

The matrix $E = (e_{ij})$ is now introduced. Since $\sum_i e_{ii} x_i^2 - \sum_i e_{ii} x_i^2 + \sum_{j \neq i} x_i e_{ij} x_j - \sum_{j \neq i} x_i e_{ij} x_j = 0$, this term can be added to the right hand side of (21) to yield

$$\begin{aligned} \boldsymbol{x}^T A \boldsymbol{x} \geq & \sum_i (\underline{a}_{ii} + e_{ii}) x_i^2 + \sum_{j \neq i} x_i (a_{M,ij} + e_{ij}) x_j \\ & - \sum_i e_{ii} x_i^2 - \sum_{j \neq i} \left(|x_i| \Delta a_{ij} |x_j| + x_i e_{ij} x_j \right) . \end{aligned} \tag{22}$$

The right hand side of Equation (22) can be further relaxed:

$$\begin{aligned} \boldsymbol{x}^T A \boldsymbol{x} \geq & \sum_i (\underline{a}_{ii} + e_{ii}) x_i^2 + \sum_{j \neq i} x_i (a_{M,ij} + e_{ij}) x_j \\ & - \sum_i \left(0 + |e_{ii}| \right) x_i^2 - \sum_{j \neq i} |x_i| \left(\Delta a_{ij} + |e_{ij}| \right) |x_j| . \end{aligned} \tag{23}$$

This is equivalent to

$$\boldsymbol{x}^T A \boldsymbol{x} \geq \boldsymbol{x}^T (\tilde{A}_M + E) \boldsymbol{x} - |\boldsymbol{x}^T| (\widetilde{\Delta A} + |E|) |\boldsymbol{x}| . \tag{24}$$

Using the Rayleigh quotient, we have $\lambda_{min}(A) = \min_{\boldsymbol{x}^T \boldsymbol{x} = 1} \boldsymbol{x}^T A \boldsymbol{x}$. Hence,

$$\begin{aligned} \lambda_{min}(A) &\geq \min_{\boldsymbol{x}^T \boldsymbol{x} = 1} \boldsymbol{x}^T (\tilde{A}_M + E) \boldsymbol{x} - \max_{\boldsymbol{x}^T \boldsymbol{x} = 1} |\boldsymbol{x}^T| (\widetilde{\Delta A} + |E|) |\boldsymbol{x}| \\ &\geq \lambda_{min}(\tilde{A}_M + E) - \rho(\widetilde{\Delta A} + |E|) . \end{aligned}$$

∎

The optimal choice for the matrix E is a matrix that minimizes the lower bound λ_{min} obtained. Unfortunately, this matrix cannot be determined *a priori*. Two choices of the matrix E have been used to illustrate this α calculation method:

- $E = 0$,
- $E = \text{diag}\,(\Delta A)$.

The second choice of E matrix yields a result equivalent to the theorems of Deif (1991) and Rohn (1996), namely:

$$\lambda_{min}(A) \geq \lambda_{min}\,(A_M) - \rho\,(\Delta A)\,. \tag{25}$$

Illustrative Example The modified midpoint matrix is

$$\tilde{H}_{f,M} = \begin{pmatrix} -0.84148 & -0.07926 \\ -0.07926 & -40.84148 \end{pmatrix},$$

while the modified radius matrix is

$$\widetilde{\Delta H}_f = \begin{pmatrix} 0 & 2.92074 \\ 2.92074 & 0 \end{pmatrix}.$$

In the example, both choices of E generate similar results as shown in Table 2.

Case	$E = 0$				$E = \text{diag}(\Delta H_f)$			
	λ_{min}	α	d_{max}	Iter	λ_{min}	α	d_{max}	Iter
1	-43.77	21.89	71.11	18	-45.04	22.52	73.18	18
2	-43.85	21.93	71.24	18	-45.94	22.97	74.64	18

Table 2: Results for the illustrative example using the E-matrix method.

Method I.3: Mori and Kokame's Method

Mori and Kokame (1994) suggested the use of the lower and upper vertex matrices, $\underline{A} = (\underline{a}_{ij})$ and $\overline{A} = (\overline{a}_{ij})$ respectively, in order to obtain a lower bound on the minimum eigenvalue of an interval matrix.

Theorem 3.4.3 *For any symmetric matrix A in the symmetric interval matrix family $[A]$, the minimum eigenvalue $\lambda_{min}(A)$ of A is such that*

$$\lambda_{min}(A) \geq \lambda_{min}(\underline{A}) - \rho(\overline{A} - \underline{A}).$$

The Mori and Kokame method can be compared to the E-matrix method with $E = \text{diag}(\Delta A)$. Since $\overline{A} - \underline{A} = 2\Delta A$, the value provided by the Mori and Kokame method is greater than $\lambda_{min}(\underline{A}) - 2\rho(\Delta A)$. Comparing this with Equation (25), the E-matrix method with $E = \text{diag}(\Delta A)$ yields better results if and only if $\lambda_{min}(\underline{A}) - \lambda_{min}(A_M) \leq \rho(\Delta A)$. For any vector $\boldsymbol{x}$,

$$\lambda_{min}(\underline{A}) \leq \frac{\boldsymbol{x}^T \underline{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{\boldsymbol{x}^T A_M \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} - \frac{\boldsymbol{x}^T \Delta A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \leq \frac{\boldsymbol{x}^T A_M \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} + \rho(\Delta A).$$

In particular

$$\lambda_{min}(\underline{A}) \leq \lambda_{min}(A_M) + \rho(\Delta A).$$

The E-matrix method with $E = \text{diag}(\Delta A)$ is therefore at least as good as the Mori and Kokame method.

Illustrative Example The lower vertex matrix for the current solution domain is

$$\underline{H_f} = \begin{pmatrix} -0.84148 & -3.0 \\ -3.0 & -40.84148 \end{pmatrix}.$$

The minimum eigenvalue of $\underline{H_f}$ is -41.0652 but that of the midpoint matrix, $H_{f,M}$, is -0.0016. $\lambda_{min}(\underline{H_f}) - \lambda_{min}(H_{f,M})$ is negative and much smaller than the spectral radius of ΔH_f. The Mori and Kokame technique therefore leads to the construction of much looser underestimators than the E matrix approach with $E = \text{diag}(\Delta H_f)$. This is corroborated by the results reported in Table 3.

Case	λ_{min}	α	d_{max}	Iterations
1	-131.14	65.57	213.09	31
2	-133.65	66.83	217.18	32

Table 3: Results for the illustrative example using the Mori and Kokame method.

Method I.4: The Lower Bounding Hessian Method

Unlike other $\mathcal{O}(n^3)$ methods, this technique requires the construction of a single real matrix, referred to as a *lower bounding Hessian matrix*, in order to determine a lower bound on the minimum eigenvalue of the interval Hessian matrix. Using a necessary and sufficient condition proved by Stephens (1997), a bounding Hessian can be defined in terms of a property of its quadratic form.

Definition 1 Given a symmetric interval Hessian matrix $[A]$, the real symmetric matrix L is a lower bounding Hessian of $[A]$ if and only if $\boldsymbol{x}^T L\boldsymbol{x} \leq \boldsymbol{x}^T A\boldsymbol{x}$, $\forall \boldsymbol{x} \in \Re^n$, $\forall A \in [A]$. Similarly, the real symmetric matrix U is an upper bounding Hessian of $[A]$ if and only if $\boldsymbol{x}^T U\boldsymbol{x} \geq \boldsymbol{x}^T A\boldsymbol{x}$, $\forall \boldsymbol{x} \in \Re^n$, $\forall A \in [A]$.

It follows immediately from this definition that the minimum eigenvalue of a lower bounding Hessian is a guaranteed lower bound on the minimum eigenvalue of the interval matrix family.

The procedure proposed by Stephens (1997) for the construction of upper bounding Hessians can be appropriately transformed to build a lower bounding Hessian.

Theorem 3.4.4 *Given an interval Hessian matrix $[A] = ([\underline{a}_{ij}, \overline{a}_{ij}])$, the matrix $L = (l_{ij})$ where*

$$l_{ij} = \begin{cases} \underline{a}_{ii} + \sum\limits_{k \neq i} \frac{\underline{a}_{ik} - \overline{a}_{ik}}{2} & i = j \\ \frac{\underline{a}_{ij} + \overline{a}_{ij}}{2} & i \neq j \end{cases}$$

is a lower bounding Hessian of $[A]$.

Proof Let $[A]$ be an $n \times n$ symmetric interval matrix. Recall that a vertex matrix A_v of $[A]$ is defined by $(A_v)_{ij} = \underline{a}_{ij}$ or $(A_v)_{ij} = \overline{a}_{ij}$.
Let X be the set of n-dimensional vectors with unit norm. For any $\boldsymbol{x} \in X$, the vertex matrix $A_v^* = (a_{v,ij}^*)$ is defined as

$$a_{v,ij}^* = \begin{cases} \underline{a}_{ij} & \text{if } x_i x_j \geq 0 \\ \overline{a}_{ij} & \text{if } x_i x_j < 0 \end{cases}$$

Then, $\boldsymbol{x}^T A_v^* \boldsymbol{x} \leq \boldsymbol{x}^T A \boldsymbol{x} = \sum\limits_{i=1}^{n} a_{ii} x_i^2 + \sum\limits_{i=1}^{n} \sum\limits_{j \neq i} a_{ij} x_i x_j$, $\forall \boldsymbol{x} \in X$, where $A \in [A]$ is a real symmetric matrix. Consequently, L is lower bounding Hessian if and only if $\boldsymbol{x}^T L \boldsymbol{x} \leq \boldsymbol{x}^T A_v^* \boldsymbol{x}$, $\forall \boldsymbol{x} \in X$.

Given a vector $\boldsymbol{x} \in X$, the quadratic form for the matrix L defined in Theorem 3.4.4 is expressed as

$$\begin{aligned}
\boldsymbol{x}^T L \boldsymbol{x} &= \sum_{i=1}^{n} l_{ii} x_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} l_{ij} x_i x_j \\
&= \sum_{i=1}^{n} \underline{a}_{ii} x_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} \frac{\underline{a}_{ij} - \overline{a}_{ij}}{2} x_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} \frac{\underline{a}_{ij} + \overline{a}_{ij}}{2} x_i x_j .
\end{aligned}$$

Using the fact that $[A]$ is symmetric, this is equivalent to

$$\boldsymbol{x}^T L \boldsymbol{x} = \sum_{i=1}^{n} \underline{a}_{ii} x_i^2 + 2 \sum_{i=1}^{n} \sum_{j>i} \left(\frac{\underline{a}_{ij} - \overline{a}_{ij}}{4} (x_i^2 + x_j^2) + \frac{\underline{a}_{ij} + \overline{a}_{ij}}{2} x_i x_j \right) . \tag{26}$$

For the vertex matrix A_v^*, we have

$$\begin{aligned}
\boldsymbol{x}^T A_v^* \boldsymbol{x} &= \sum_{i=1}^{n} a_{v,ii}^* x_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} a_{v,ij}^* x_i x_j \\
&= \sum_{i=1}^{n} \underline{a}_{ii} x_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} a_{v,ij}^* x_i x_j \\
&= \sum_{i=1}^{n} \underline{a}_{ii} x_i^2 + 2 \sum_{i=1}^{n} \sum_{j>i} a_{v,ij}^* x_i x_j . \qquad (27)
\end{aligned}$$

Comparing Equations (26) and (27), we find that $\boldsymbol{x}^T L \boldsymbol{x} \leq \boldsymbol{x}^T A_v^* \boldsymbol{x}$ if and only if

$$\sum_{i=1}^{n} \sum_{j>i} \left(\frac{\underline{a}_{ij} - \overline{a}_{ij}}{4} (x_i^2 + x_j^2) + \frac{\underline{a}_{ij} + \overline{a}_{ij}}{2} x_i x_j \right) \leq \sum_{i=1}^{n} \sum_{j>i} a_{v,ij}^* x_i x_j . \tag{28}$$

In order to prove that this relation holds for all $\boldsymbol{x} \in X$, we first note that, since $(x_i \pm x_j)^2 \geq 0$, $x_i^2 + x_j^2 \geq \pm 2 x_i x_j$. Hence,

$$\frac{\underline{a}_{ij} - \overline{a}_{ij}}{4} (x_i^2 + x_j^2) \leq \pm \frac{\underline{a}_{ij} - \overline{a}_{ij}}{2} x_i x_j$$

and

$$\begin{aligned}
\frac{\underline{a}_{ij} - \overline{a}_{ij}}{4} (x_i^2 + x_j^2) + \frac{\underline{a}_{ij} + \overline{a}_{ij}}{2} x_i x_j &\leq \pm \frac{\underline{a}_{ij} - \overline{a}_{ij}}{2} x_i x_j + \frac{\underline{a}_{ij} + \overline{a}_{ij}}{2} x_i x_j \\
&\leq \begin{cases} \underline{a}_{ij} x_i x_j \\ \overline{a}_{ij} x_i x_j \end{cases} \\
&\leq a_{v,ij}^* x_i x_j .
\end{aligned}$$

Summing over all i and $j > i$, the desired relation is obtained, proving that L is a lower bounding Hessian of $[A]$. ∎

Rather than calculating the minimum eigenvalue of L in order to construct an underestimator of the form given in Equation (9), Stephens (1997) recommends the incorporation of the lower bounding Hessian in the lower bounding function. He proves that given any twice-differentiable function $f(\boldsymbol{x})$ and its lower bounding Hessian L_f over the solution space, the function $f(\boldsymbol{x}) - 1/2\, \boldsymbol{x}^T L_f \boldsymbol{x}$ is convex. However, this expression is not everywhere less than $f(\boldsymbol{x})$ and the addition of a constant c is necessary to ensure that a valid convex underestimator is indeed obtained. The underestimating function is then expressed as

$$f(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{x}^T L_f \boldsymbol{x} + c \tag{29}$$

where the condition $c \leq 1/2\ \min \boldsymbol{x}^T L_f \boldsymbol{x}$ must be satisfied in order to guarantee valid underestimation throughout the solution domain.

This type of underestimator has not been tested on any examples. The rigorous calculation of c requires the solution of an indefinite quadratic program. The underestimator thus obtained differs from the general αBB underestimators in several ways: Equation (29) does not match the original function at the end points and no bound can be provided for the maximum separation distance between $f(\boldsymbol{x})$ and its underestimator. In addition, the function $f(\boldsymbol{x})$ is underestimated even when its lower bounding Hessian is positive semi-definite.

Illustrative Example The lower bounding Hessian matrix for the illustrative example is

$$L_f = \begin{pmatrix} -3.48355 & -0.07926 \\ -0.07926 & -43.7622 \end{pmatrix}.$$

Its minimum eigenvalue is -43.7624. The results are shown in Table 4. In this example, the interval Hessian matrix is such that the widths of the off-diagonal elements are small compared to the lower bounds on the diagonal elements. Furthermore, the midpoint of the off-diagonal intervals, which is used to determine the off-diagonal elements of L, is close to zero. These two factors lead to the construction of a lower bounding matrix which is almost diagonal and whose diagonal entries are almost equal to the lower bounds on the interval diagonal elements. In such a situation, the bound on the minimum eigenvalue is very accurate. This is not the case if the width of the off-diagonal intervals is large, or if their midpoint is not close to zero.

Case	λ_{min}	α	d_{max}	Iterations
1	-43.77	21.89	71.11	18
2	-43.85	21.93	71.24	18

Table 4: Results for the illustrative example using the lower bounding Hessian method.

Method I.5: A Method Based on the Kharitonov Theorem

The Kharitonov theorem (Kharitonov, 1979) is used to determine whether an interval polynomial family $\mathcal{P}(\lambda)$ is stable or not by testing the stability of only four real polynomials in the whole family. Considering the set of all the roots of the interval polynomial family, let $\lambda_{min,Re}$ denote the root with the smallest real part. Adjiman *et al.* (1996) showed that the Kharitonov theorem can be used not only to determine the stability of the family, but also to compute the value of $\lambda_{min,Re}$.

Theorem 3.4.5 *Let an interval polynomial family $\mathcal{P}(\lambda)$ be defined by*

$$\mathcal{P}(\lambda) = [a_0^L, a_0^U] + [a_1^L, a_1^U]\lambda + \cdots + [a_{n-1}^L, a_{n-1}^U]\lambda^{n-1} + \lambda^n$$

where $a_i^L \leq a_i^U, \ \forall i$.

Let $\mathcal{P}_4(\lambda)$ denote the subset of this family containing the following four real polynomials (Kharitonov polynomials):

$$\begin{aligned}
K_1(f,X,\lambda) &= a_0^L + a_1^L\lambda + a_2^U\lambda^2 + a_3^U\lambda^3 + a_4^L\lambda^4 + a_5^L\lambda^5 + a_6^U\lambda^6 + \cdots \\
K_2(f,X,\lambda) &= a_0^U + a_1^U\lambda + a_2^L\lambda^2 + a_3^L\lambda^3 + a_4^U\lambda^4 + a_5^U\lambda^5 + a_6^L\lambda^6 + \cdots \\
K_3(f,X,\lambda) &= a_0^U + a_1^L\lambda + a_2^L\lambda^2 + a_3^U\lambda^3 + a_4^U\lambda^4 + a_5^L\lambda^5 + a_6^L\lambda^6 + \cdots \\
K_4(f,X,\lambda) &= a_0^L + a_1^U\lambda + a_2^U\lambda^2 + a_3^L\lambda^3 + a_4^L\lambda^4 + a_5^U\lambda^5 + a_6^U\lambda^6 + \cdots
\end{aligned}$$

Then $\mathcal{P}(\lambda)$ and $\mathcal{P}_4(\lambda)$ have the same $\lambda_{min,Re}$.

This result greatly decreases the complexity of the $\lambda_{min,Re}$ calculation as the number of polynomials to be considered is reduced to four. The following procedure can then be used to calculate a lower bound on the minimum eigenvalue of the Hessian matrix $H(\boldsymbol{x})$:

1. Construct $H(\boldsymbol{x}) - \lambda I$, where I is the identity matrix.

2. Derive the determinant of $H(\boldsymbol{x}) - \lambda I$ and set it to zero. The resulting polynomial is of the form

$$\mathcal{P}(\boldsymbol{x}, \lambda) = a_0(\boldsymbol{x}) + a_1(\boldsymbol{x})\lambda + a_2(\boldsymbol{x})\lambda^2 + a_3(\boldsymbol{x})\lambda^3 + \cdots$$

3. Using interval arithmetic, obtain an interval polynomial family, $\mathcal{P}(\lambda)$ which contains $\mathcal{P}(\boldsymbol{x}, \lambda)$.

4. Using Theorem 3.4.5, calculate $\lambda_{min,Re}$ whose real part gives a lower bound on the minimum eigenvalues of $H(\boldsymbol{x})$.

The recourse to interval arithmetic in Step 3 is necessary as it transforms a tremendously difficult problem into a tractable one. However, the family of polynomials is enlarged by the process and while $\lambda_{min,Re}$ is the root of a polynomial in $\mathcal{P}(\lambda)$, it may not be the root of any of the polynomials in $\mathcal{P}(\boldsymbol{x}, \lambda)$. Thus, the value obtained is a *valid lower bound* on the minimum eigenvalue, whose quality depends on the type of interval calculations that are involved in the construction of the interval polynomial family. Since the coefficients of $\mathcal{P}(\boldsymbol{x}, \lambda)$ are the result of numerous multiplications, their dependence on $\boldsymbol{x}$ as well as their interdependence may be quite intricate, leading to large overestimates of the intervals they cover. An alternative to the procedure presented above is to perform the interval extensions on the Hessian matrix, prior to the determinant computation. This is likely to aggravate the accuracy problem. Whether one starts from the exact Hessian matrix or from a Hessian matrix that contains it, the final interval polynomial may generate a larger spectrum of eigenvalues than the interval Hessian matrix, as its derivation entails a larger number of operations.

Illustrative Example If the interval calculations are performed after the derivation of the characteristic polynomial of $H_f(x, y)$, the four Kharitonov polynomials are

$$\begin{aligned}
K_1(f, \lambda) &= -42.65921 - 33.68296\lambda + \lambda^2, \\
K_2(f, \lambda) &= 38.36728 + 41.68296\lambda + \lambda^2, \\
K_3(f, \lambda) &= 38.36728 - 33.68296\lambda + \lambda^2, \\
K_4(f, \lambda) &= -42.65921 + 41.68296\lambda + \lambda^2.
\end{aligned}$$

The minimum roots of the four polynomials are -12.2215, -40.7412, 1.1804, -42.6824, and therefore $\alpha = 21.3412$. For this small example, a more accurate value can be obtained by using global optimization to compute the tightest possible intervals for the coefficients of the polynomial. The optimal interval polynomial is $[-2.3382, 0.6744] + [-4.0073, 2.0245]\lambda + \lambda^2$ and the Kharitonov polynomials give a minimum eigenvalue of -2.8460, a value which is very close to the exact minimum eigenvalue of -2.3934.

If, on the other hand, the interval Hessian matrix is used, the following four polynomials are obtained:

$$\begin{array}{rrcl}
K_1(f,\lambda) = & -43.36729 & - & 33.68296\lambda + \lambda^2, \\
K_2(f,\lambda) = & 34.36729 & + & 41.68296\lambda + \lambda^2, \\
K_3(f,\lambda) = & 34.36729 & - & 33.68296\lambda + \lambda^2, \\
K_4(f,\lambda) = & -43.36729 & + & 41.68296\lambda + \lambda^2.
\end{array}$$

The corresponding set of minimum roots is

$$\{-1.2418, -40.8415, 1.0532, -42.6986\},$$

giving $\alpha = 21.3493$. As shown in Table 5, this method performs well when applied to this small example.

Case	λ_{min}	α	d_{max}	Iterations
1	-42.69	21.35	69.36	19
2	-42.50	21.25	69.06	18

Table 5: Results for the illustrative example using the Kharitonov theorem method.

Method I.6: Hertz's Method

The methods described so far either provide a lower bound on the minimum eigenvalue of the interval Hessian matrix or the smallest real part of the roots of the interval characteristic polynomial. The Hertz method allows the computation of the *exact* smallest eigenvalue of the interval Hessian matrix. This gain in accuracy comes at a cost since the number of matrices that need to be constructed in order to arrive at this value is no longer independent of the dimensionality of the problem. As this technique calls for 2^{n-1} vertex matrices, its computational cost may become prohibitive for larger problems.

The Hertz method has been described in detail in Hertz (1992) and Adjiman and Floudas (1996). In essence, it is similar to the Kharitonov theorem: a finite subset of real matrices is constructed from the interval matrix, with the property that its minimum eigenvalue is equal to the minimum eigenvalue of the interval matrix. The required real matrices can be obtained through a systematic procedure.

Theorem 3.4.6 *Let $[A] = ([\underline{a}_{ij}, \overline{a}_{ij}])$ be a symmetric interval Hessian matrix. Consider a vector $\boldsymbol{x} \in R^n$. There are 2^{n-1} possible combinations for the signs of the $x_i x_j$ products $(i \neq j)$. For the kth such combination, let the vertex matrix $A_k \in [A]$ be defined by $A_k = [a_{ij}^k]$ where*

$$a_{ij}^k = \begin{cases} \underline{a}_{ii} & \text{if } i = j, \\ \underline{a}_{ij} & \text{if } x_i x_j \geq 0, i \neq j, \\ \overline{a}_{ij} & \text{if } x_i x_j < 0, i \neq j. \end{cases}$$

Then the smallest eigenvalue of the set of matrices $\{A_k\}$ is the minimum eigenvalue of $[A]$.

Illustrative Example Two vertex matrices are required for problems in two variables. For the illustrative example, they are

$$H_{f,1} = \begin{pmatrix} -0.84148 & -3 \\ -3 & -40.84148 \end{pmatrix} \text{ and } H_{f,2} = \begin{pmatrix} -0.84148 & 2.84148 \\ 2.84148 & -40.84148 \end{pmatrix}.$$

Their minimum eigenvalues are -41.0652 and -41.0423 respectively. The results are shown in Table 6. The α parameter calculated with the Hertz method is based on the exact minimum eigenvalue of the interval Hessian matrix. Consequently, any other uniform diagonal shift method relying on the interval Hessian matrix produces α values greater than or equal to the value generated with the Hertz method.

Case	λ_{min}	α	d_{max}	Iterations
1	-41.07	20.54	66.73	18
2	-41.95	20.98	68.15	18

Table 6: Results for the illustrative example using the Hertz method.

3.5 Non-Uniform Diagonal Shift Matrix

This class of methods allows the calculation of a different α value for each variable in order to construct an underestimator of the form shown in Equation (9). The non-zero elements of the diagonal shift matrix Δ can no longer be related to the minimum eigenvalue of the interval Hessian matrix $[H_f]$. The condition presented in Theorem 3.1.1 for the convexity of the underestimator cannot be simplified.

Before presenting rigorous procedures for the derivation of a diagonal shift matrix Δ such that $[H_f] + 2\Delta$ is positive semi-definite, a few definitions are introduced.

Definition 1 A square matrix A is an M**-matrix** if all its off-diagonal elements are non-positive and there exists a real positive vector $\boldsymbol{u}$ such that $A\ \boldsymbol{u} > 0$. Inequalities are understood component-wise.

Definition 2 The **comparison matrix** $\langle A \rangle$ of the interval matrix $[A] = ([\underline{a}_{ij}, \overline{a}_{ij}])$ is

$$\langle A \rangle_{ij} = \begin{cases} 0 & \text{if } i = j \text{ and } 0 \in [\underline{a}_{ij}, \overline{a}_{ij}] \\ \min\{|\underline{a}_{ij}|, |\overline{a}_{ij}|\} & \text{if } i = j \text{ and } 0 \notin [\underline{a}_{ij}, \overline{a}_{ij}] \\ -|a|_{ij} & \text{if } i \neq j \end{cases}$$

where $|a|_{ij} = \max\{|\underline{a}_{ij}|, |\overline{a}_{ij}|\}$.

Definition 3 A square interval matrix $[A]$ is an H**-matrix** if its comparison matrix $\langle A \rangle$ is an M-matrix. It can be shown that if $[A]$ is an H-matrix, it is regular and 0 is not an eigenvalue of $[A]$ (Neumaier, 1992).

The first rigorous method belonging to this class is a $\mathcal{O}(n^2)$ method, while the two other methods presented are iterative, with varying degrees of complexity.

3.5.1 $\mathcal{O}(n^2)$ Method

Method II.1: Scaled Gerschgorin Theorem

Theorem 3.5.1 *For any vector $\boldsymbol{d} > 0$ and a symmetric interval matrix $[A]$, define the vector α as*

$$\alpha_i = \max\left\{0, -\frac{1}{2}\left(\underline{a}_{ii} - \sum_{j\neq i} |a|_{ij}\frac{d_j}{d_i}\right)\right\}$$

where $|a|_{ij} = \max\{|\underline{a}_{ij}|, |\overline{a}_{ij}|\}$.

Then, for all $A \in [A]$, *the matrix* $A_{\mathcal{L}} = A + 2\Delta$ *with* $\Delta = \mathrm{diag}(\alpha_i)$ *is positive semi-definite.*

Proof First, we show that the diagonal elements of $A_{\mathcal{L}}$ are non-negative. By definition,

$$a_{\mathcal{L},ii}d_i = a_{ii}d_i + \max\left\{0, -\underline{a}_{ii}d_i + \sum_{j\neq i}|a|_{ij}d_j\right\}.$$

<u>If $a_{ii} \geq 0$:</u> Since $d_i > 0$, $a_{\mathcal{L},ii}d_i$ is the sum of two non-negative terms and therefore $a_{\mathcal{L},ii} > 0$.

<u>If $a_{ii} < 0$:</u> Since $\underline{a}_{ii} \leq a_{ii}$, $\underline{a}_{ii} < 0$. Therefore,

$$\underline{a}_{ii}d_i - \sum_{j\neq i}|a|_{ij}d_j < 0.$$

Hence,

$$a_{\mathcal{L},ii}d_i = a_{ii}d_i - \underline{a}_{ii}d_i + \sum_{j\neq i}|a|_{ij}d_j \geq 0.$$

The diagonal elements of $A_{\mathcal{L}}$ are therefore non-negative. Using this property, the comparison matrix $\langle A_{\mathcal{L}}\rangle$ of $A_{\mathcal{L}}$ is given by

$$\begin{aligned}
\langle A_{\mathcal{L}}\rangle_{ij} &= \begin{cases} \underline{a}_{\mathcal{L},ii} & \text{if } i = j \\ -|a_{\mathcal{L}}|_{ij} & \text{if } i \neq j \end{cases} \\
&= \begin{cases} \underline{a}_{ii} + \max\left\{0, -\underline{a}_{ii} + \sum_{k\neq i}|a|_{ik}\frac{d_k}{d_i}\right\} & \text{if } i = j \\ -|a|_{ij} & \text{if } i \neq j \end{cases}
\end{aligned}$$

We now prove that $\langle A_{\mathcal{L}}\rangle \boldsymbol{d} \geq 0$. The ith component of the vector $\langle A_{\mathcal{L}}\rangle \boldsymbol{d}$ is given by

$$(\langle A_{\mathcal{L}}\rangle \boldsymbol{d})_i = \underline{a}_{ii}d_i + \max\left\{0, -\underline{a}_{ii}d_i + \sum_{k\neq i}|a|_{ik}d_k\right\} - \sum_{k\neq i}|a|_{ik}d_k. \tag{30}$$

Two cases must now be discussed.

CASE 1: $-\underline{a}_{ii}d_i + \sum_{k \neq i} |a|_{ik} d_k > 0$

Equation (30) becomes $(\langle A_{\mathcal{L}} \rangle \boldsymbol{d})_i = 0$.

CASE 2: $-\underline{a}_{ii}d_i + \sum_{k \neq i} |a|_{ik} d_k \leq 0$.

Equation (30) becomes $(\langle A_{\mathcal{L}} \rangle \boldsymbol{d})_i = \underline{a}_{ii} d_i - \sum_{k \neq i} |a|_{ik} d_k \geq 0$.

This proves that $\langle A_{\mathcal{L}} \rangle \boldsymbol{d} \geq 0$. Let $(\lambda, \boldsymbol{x})$ be any eigenpair of $A_{\mathcal{L}}$ so that $A_{\mathcal{L}} \boldsymbol{x} = \lambda \boldsymbol{x}$. Since $A_{\mathcal{L}}$ is symmetric, all its eigenpairs are real and the eigenvector $\boldsymbol{x}$ can be scaled in such a way that $\max_i \left| \frac{x_i}{d_i} \right| = \frac{x_j}{d_j} = 1$, for some j. Then,

$$\begin{aligned} \lambda d_j &= \lambda x_j = \sum_k a_{\mathcal{L},jk} x_k \\ &\geq |a_{\mathcal{L},jj}||x_j| - \sum_{k \neq j} |a_{\mathcal{L},jk}||x_k| \\ &\geq |a_{\mathcal{L},jj}| d_j - \sum_{k \neq j} |a_{\mathcal{L},jk}| d_k \geq 0. \end{aligned}$$

Therefore, $\lambda \geq 0$. ∎

The choice of the non-negative vector $\boldsymbol{d}$ in Theorem 3.5.1 is arbitrary. If all its elements are set to 1, the α vector becomes:

$$\alpha_i = \max \left\{ 0, -\frac{1}{2} \left(\underline{a}_{ii} - \sum_{j \neq i} |a|_{ij} \right) \right\},$$

where the second term is the expression used in Method I.1 for the extended Gerschgorin theorem. As a result, the elements of the non-uniform diagonal shift matrix obtained using Method II.1 with $d_i = 1 \; \forall i$ are less than or equal to the elements of the uniform diagonal shift matrix of Method I.1.

A second choice of $\boldsymbol{d}$ is based on the variable ranges. If $\boldsymbol{d} = \boldsymbol{x}^U - \boldsymbol{x}^L$, the off-diagonal contributions to the value of α_i are divided by $x_i^U - x_i^L$. This reflects the fact that variables with a wide range have a larger effect on the quality of the underestimator than variables with a smaller range.

Similar scaling can be used for all Type I methods. The interval Hessian matrix $[A]$ is replaced by the matrix $[B]$ with $B_{ik} = d_i A_{ik} d_k$, where $d_i = x_i^U - x_i^L$. With the resulting α value, α_B, one α_i parameter is then computed for each variable using the expression $\alpha_i = \alpha_B / d_i^2$.

Illustrative Example Both choices for vector $\boldsymbol{d}$, $d_i = 1$ and $d_i = x_i^U - x_i^L$, are considered. As the results in Table 7 show, the use of a non-uniform shift is a very effective way to reduce the number of iterations and the maximum separation distance between f and its underestimator. For $d_i = 1$, the α corresponding to variable y, α_y, is the same as that obtained using the Gerschgorin bound on the minimum eigenvalue (Method I.1). However, the value α_x is decreased by 91%. This results in a 63% decrease in the maximum separation distance for the initial domain. Although the use of $\boldsymbol{d} = \boldsymbol{x}_i^U - \boldsymbol{x}_i^L = (3,2)^T$ does not change the required number of iterations, it shifts a larger part of the underestimation to the y variable as its range is smaller than that of x. This results in a small decrease in the maximum separation distance. Finally, as was the case for Method I.1, the results are the same for Case 1 and Case 2.

	$d_i = 1$				$d_i = x_i^U - x_i^L$			
Case	α_x	α_y	d_{max}	Iter	α_x	α_y	d_{max}	Iter
1	1.93	21.93	26.24	13	1.43	22.68	25.87	13
2	1.93	21.93	26.24	13	1.43	22.68	25.87	13

Table 7: Results for the illustrative example using the scaled Gerschgorin method.

3.5.2 Iterative Methods

Method II.2: H-Matrix Method

In this method, the properties of H-matrices are used to obtain a valid non-uniform diagonal shift matrix.

Theorem 3.5.2 *Consider the symmetric interval matrix $[G]$ and its modified midpoint matrix $\tilde{G}_M$. Let $C = \tilde{G}_M^{-1}$. If*

- *$\tilde{G}_M$ is positive definite, and*
- *the pre-conditioned matrix $C[G]$ is an H-matrix,*

then $[G]$ is positive definite.

Proof If $\tilde{G}_M$ is positive definite, all its eigenvalues are positive and the matrix C is regular. If $C[G]$ is an H-matrix, it is regular and hence 0 is not an eigenvalue of $[G]$. The eigenvalues of an interval matrix are a continuous function of the matrix elements. Thus if there exists $G \in [G]$ such that at least one of the eigenvalues of G, or equivalently CG, is negative, and $\tilde{G}_M \in [G]$ has positive eigenvalues, there must exist a matrix G' on the path between G and $\tilde{G}_M$ such that at least one eigenvalue of CG' is equal to 0. This contradicts the H-matrix property of $C[G]$. If the two conditions of Theorem 3.5.2 are met, $[G]$ must therefore be positive definite. ∎

This theorem can be used for the identification of an appropriate diagonal shift matrix Δ for the interval matrix $[A]$ by setting $[G] = [A] + 2\Delta$. The modified midpoint matrix of $[G]$ is then $\tilde{A}_M + \Delta$, where $\tilde{A}_M$ is the modified midpoint matrix of $[A]$. The conditions for the positive definiteness of the interval matrix $[A] + 2\Delta$ are then

1. $\tilde{A}_M + \Delta$ is positive definite, and
2. the pre-conditioned matrix $C[A] + 2C\Delta$ is an H-matrix, where $C = \left(\tilde{A}_M + \Delta\right)^{-1}$.

An iterative approach can be used to compute a matrix Δ which satisfies the above conditions. While the details of this approach are described in Appendix B, the main steps are given below.

1. Use Method I.2 to construct a uniform shift matrix Δ^E and calculate the corresponding maximum separation distance, d^E_{max}. Set $k = 0$.
2. Compute a modified Cholesky decomposition (Neumaier, 1997) of $\tilde{A}_M$ to determine if it is positive definite. If not, the results of the decomposition provide an initial guess Δ_0 for the diagonal shift matrix.
3. Check whether $C[A] + 2C\Delta_k$ is an H-matrix. If so, Δ_k is returned as the diagonal shift matrix. Otherwise, proceed to Step 4.
4. Construct a new guess Δ_{k+1} such that the corresponding maximum separation distance is d^{k+1}_{max} with $d^k_{max} \leq d^{k+1}_{max} \leq d^E_{max}$. Set $k = k + 1$. Go to Step 3.

If no matrix Δ such that $C[A] + 2C\Delta$ is an H-matrix has been identified after a fixed number of iterations, the matrix Δ^E is returned. As a result, the H-matrix method is at least as accurate as the E-matrix method. It is also more computationally expensive.

Illustrative Example As shown in Table 8, the overall results for Case 1 are the same as for Method I.2, the E-matrix method with $E = 0$. The H-matrix method consistently fails to identify a diagonal shift matrix better than that of Method I.2. For Case 2, however, the number of required iterations is reduced and a significantly improved non-uniform diagonal shift matrix is generated by the H-matrix method.

Case	α_x	α_y	d_{max}	Iterations
1	21.89	21.89	71.11	18
2	0.93	21.93	23.99	16

Table 8: Results for the illustrative example with the H-matrix method.

Method II.3: Minimization of Maximum Separation Distance

Since the maximum separation distance between the original function and its underestimator reflects the quality of the underestimator, this method aims to derive a non-uniform diagonal shift matrix Δ which is optimal with respect to d_{max}. This goal can be expressed as an optimization problem of the form

$$\begin{array}{ll} \min & (\boldsymbol{x}^U - \boldsymbol{x}^L)^T \Delta (\boldsymbol{x}^U - \boldsymbol{x}^L) \\ s.t. & H_f(\boldsymbol{x}) + 2\Delta \geq 0 \\ & \boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U] \end{array}$$

where Δ is a diagonal matrix, and $M \geq 0$ means that the matrix M is positive semi-definite.

Due to the nonconvexity of the above problem, the formulation is relaxed to

$$\begin{array}{ll} \min & (\boldsymbol{x}^U - \boldsymbol{x}^L)^T \Delta (\boldsymbol{x}^U - \boldsymbol{x}^L) \\ s.t. & [H_f] + 2\Delta \geq 0 \end{array}$$

The presence of the interval Hessian matrix in the constraint makes the identification of the solution of this problem difficult. To further simplify it, $[H_f]$ can be replaced by a real matrix whose minimum eigenvalue is smaller than the minimum eigenvalue of $[H_f]$. The lower bounding Hessian L defined in Theorem 3.4.4 is a natural choice and the maximum distance minimization problem becomes

$$\begin{array}{ll} \min & (\boldsymbol{x}^U - \boldsymbol{x}^L)^T \Delta (\boldsymbol{x}^U - \boldsymbol{x}^L) \\ s.t. & L + 2\Delta \geq 0 \end{array} \tag{31}$$

Problem (31), a semi-definite programming problem, is convex and can therefore be solved to global optimality using interior-point methods which have a polynomial worst-case complexity (Vandenberghe and Boyd, 1996). Because of the use of the lower bounding Hessian matrix, the Δ matrix is rigorously valid but not strictly optimal. As far as the quality of the underestimator is concerned, the minimization of maximum distance method is at least as good as the lower bounding Hessian method but no *a priori* comparison with other methods is possible.

Illustrative Example For the illustrative example,

$$\Delta = \begin{pmatrix} \alpha_x & 0 \\ 0 & \alpha_y \end{pmatrix} \text{ and } L = \begin{pmatrix} -3.48355 & -0.07926 \\ -0.07926 & -43.7622 \end{pmatrix}.$$

The results are shown in Table 9. Although the maximum separation distance is explicitly minimized in this method, the results are slightly worse than those given by Method II.1 which uses the scaled Gerschgorin approach. This can be attributed to the use of the lower bounding Hessian matrix rather than the interval Hessian matrix in Method II.3. This method is the only one for which the treatment of the two nonconvex terms as separate entities (Case 2) yields worse results than the analysis of the entire function (Case 1).

Case	α_x	α_y	d_{max}	Iterations
1	1.91	21.95	26.23	14
2	1.91	21.95	26.23	16

Table 9: Results for the illustrative example with minimization of maximum distance.

3.6 On the Role of Interval Calculations

The tractability of all the α calculation methods presented in this paper is ensured through the use of interval arithmetic to obtain bounds on the second-order derivatives. Since the mutual dependence of the Hessian matrix elements is neglected in such an analysis, a certain degree of inaccuracy is introduced. Regardless of the chosen procedure, the quality of the constructed underestimator depends to a large extent on the accuracy of the intervals generated for each Hessian matrix element. As was stated in the presentation of the illustrative example (Section 3.3), an *optimal* interval Hessian matrix can be computed by using global optimization to determine the minimum and maximum values each matrix element can take given a bounded solution space. The resulting matrix is then optimal in the sense that the tightest possible bounds have been obtained for each element. However, this optimal matrix cannot be derived for the general case. Wider but guaranteed intervals can be generated using classical interval arithmetic techniques. One of the main characteristics of such an approach is that the final result differs depending on the initial analytical representation of the matrix elements. For instance, if the same term appears in the numerator and the denominator of a division, natural interval extensions will evaluate these quantities independently, most probably leading to an overestimate. This is the case for the small illustrative example used throughout Section 3. If the analytical Hessian matrix for Term B is expressed as

$$H_B(x,y) = \begin{pmatrix} 0 & \frac{2y}{(y^2+1)^2} \\ \\ \frac{2y}{(y^2+1)^2} & \frac{2x}{(y^2+1)^2}\left(1-\frac{4y^2}{y^2+1}\right) \end{pmatrix},$$

the interval Hessian matrix obtained through standard interval arithmetic is now

$$[H_B] = \begin{pmatrix} [0,0] & [-2,2] \\ [-2,2] & [-12,6] \end{pmatrix}.$$

The second-order derivative with respect to y is found to belong to a much smaller interval than that obtained through automatic differentiation. The minimum eigenvalue of the above interval matrix is -12.3246 and the best α value which can be derived by any of the uniform diagonal shift matrix methods is 7.0831. This represents a significant improvement over the value of 20.9707 obtained in Section 3.4.1 with different analytical expressions. Great care should therefore be taken when carrying out automatic differentiation.

To reduce the occurrence of such overestimation, it may be possible to exploit the structure of the problem and thus supply second-order derivatives which are more appropriate for interval evaluation than those generated through automatic differentiation. This is the case, for example, of the minimization of the potential energy of the pseudoethane molecule presented by Maranas and Floudas (1994b). The highly nonlinear objective function can be expressed in the dihedral angle space or in the inter-atomic distance space. Although the first formulation is more desirable as it results in a one-variable optimization problem, the corresponding Hessian expression is grossly overestimated by natural interval extensions. The α value at the first level of the branch-and-bound tree is 37378.9 and the global optimal solution is identified in 21 iterations and 1.2 CPU sec. On the other hand, if the Hessian is first obtained in the distance space and then transformed to the dihedral angle coordinate system, a substantial improvement in the accuracy of the eigenvalue bound is achieved. The first α is then 620.8 and the algorithm converges in 15 iterations and 0.6 CPU sec. Finally, using the exact minimum eigenvalue at each iteration, the solution is found in 10 iterations, with an initial α value of 6.7. The importance of computing an accurate interval Hessian matrix cannot be overemphasized.

4 A Constrained Process Design Problem

This example, which involves 7 variables, 12 nonlinear constraints and 4 linear constraints, provides an illustration of the performance of the αBB algorithm and the different α calculation methods for a larger problem than the illustrative example used in the previous section.

The simplified alkylation process considered for this example was discussed in detail by Bracken and McCormick (1968). As shown in Figure 2, an olefin feed (100% butene), a pure isobutane recycle and a 100% isobutane make up stream are introduced in a reactor together with an acid catalyst. The reactor product stream is then passed through a fractionator where the isobutane and the alkylate product are separated. The spent acid is also removed from the reactor.

Bracken and McCormick (1968) proposed a model for this process which allowed the formulation of a profit-maximization problem. The 10 variable NLP they derived was transformed to a 7 variable problem by Dembo (1976). A slightly modified version of his formulation is used here.

The variables are defined as follows: x_1 is the olefin feed rate in barrels per day; x_2 is the acid addition rate in thousands of pounds per day; x_3 is the alkylate yield in barrels per day; x_4 is the acid strength (weight percent); x_5 is the motor octane number; x_6 is the external isobutane-to-olefin ratio; x_7 is the F-4 performance number. The profit maximization problem is then expressed as:

$$\text{Profit} = -\min \quad 1.715x_1 + 0.035x_1x_6 + 4.0565x_3 + 10.0x_2 - 0.063x_3x_5$$

subject to:

$$\begin{aligned}
0.0059553571x_6^2x_1 + 0.88392857x_3 - 0.1175625x_6x_1 - x_1 &\leq 0 \\
1.1088x_1 + 0.1303533x_1x_6 - 0.0066033x_1x_6^2 - x_3 &\leq 0 \\
6.66173269x_6^2 + 172.39878x_5 - 56.596669x_4 - 191.20592x_6 - 10000 &\leq 0 \\
1.08702x_6 + 0.32175x_4 - 0.03762x_6^2 - x_5 + 56.85075 &\leq 0 \\
0.006198x_7x_4x_3 + 2462.3121x_2 - 25.125634x_2x_4 - x_3x_4 &\leq 0 \\
161.18996x_3x_4 + 5000.0x_2x_4 - 489510.0x_2 - x_3x_4x_7 &\leq 0 \\
0.33x_7 - x_5 + 44.333333 &\leq 0 \\
0.022556x_5 - 0.007595x_7 &\leq 1 \\
0.00061x_3 - 0.0005x_1 &\leq 1 \\
0.819672x_1 - x_3 + 0.819672 &\leq 0 \\
24500.0x_2 - 250.0x_2x_4 - x_3x_4 &\leq 0 \\
1020.4082x_4x_2 + 1.2244898x_3x_4 - 100000x_2 &\leq 0 \\
6.25x_1x_6 + 6.25x_1 - 7.625x_3 - 100000 &\leq 0 \\
1.22x_3 - x_6x_1 - x_1 + 1 &\leq 0
\end{aligned}$$

$$\begin{array}{rcccr}
1500 & \leq & x_1 & \leq & 2000 \\
1 & \leq & x_2 & \leq & 120 \\
3000 & \leq & x_3 & \leq & 3500 \\
85 & \leq & x_4 & \leq & 93 \\
90 & \leq & x_5 & \leq & 95 \\
3 & \leq & x_6 & \leq & 12 \\
145 & \leq & x_7 & \leq & 162
\end{array}$$

The maximum profit is \$1772.77 per day, and the optimal variable values are $x_1^* = 1698.18$, $x_2^* = 53.66$, $x_3^* = 3031.30$, $x_4^* = 90.11$, $x_5^* = 95.00$, $x_6^* = 10.50$, $x_7^* = 153.53$.

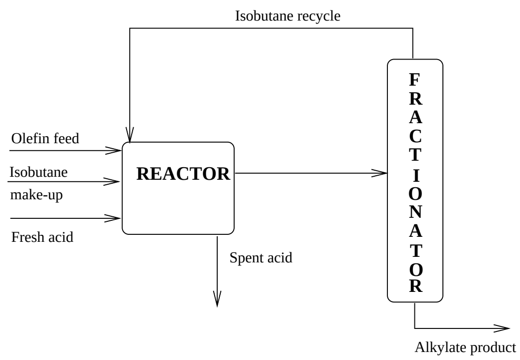

Figure 2: Simplified alkylation process flowsheet.

The presence of constraints allows the introduction of some enhancements of the algorithm such as variable bounds updates. These are performed via a reformulation of the original problem into a bound problem where the objective is to maximize or minimize one of the variables and where the original constraints have been underestimated and convexified. In this example, an update of all the variable bounds therefore necessitates the solution of 14 convex problems. Such an operation entails considerable computational expense. Although bound tightening is likely to result in the construction of more accurate underestimators, its cost may outweigh the benefits. Two bounding strategies were used for this example: an update of all variable bounds at the onset of the algorithm, or an update of all bounds at each iteration of the branch-and-bound tree. Between these two extremes, several approaches could be selected: one could choose to update the bounds of a fraction of the variables — those that are thought to affect the underestimators most significantly — or updates could be performed only at a few levels of the branch-and-bound tree. These alternatives are not explored in this paper. The results are summarized in Table 10.

All calculations were performed on an HP9000/730. The results presented in the 'Single Up.' column were generated by updating all variable bounds before the first iteration. The 'One Up./Iter' results were obtained by updating all bounds at every iterations. Although the second approach results in tighter underestimators, and hence a smaller number of iterations, the time requirements for each iteration are significantly larger than when no bounds updates are performed. Thus, the overall CPU requirements may be larger when all variable bounds are updated at each iteration. A comparison of the progress of the lower bound for each strategy as a function of iteration number and as a function of CPU time for the scaled Gerschgorin theorem method with $d_i = (x_i^U - x_i^L)$ (Method II.1), shown in Figure 3, illustrates this point. The percentage of overall computational effort dedicated to the construction of the convex lower bounding problem, t_U, is small for almost all methods. It is significantly larger for Methods I.5 and II.3 as they require the solution of a polynomial and a semi-definite programming problem respectively. t_U decreases when bound updates are performed at each iteration as a large amount of time is spent solving the bound updates problems.

In this example, the scaled Gerschgorin approach (Method II.1) with $d_i = (x_i^U - x_i^L)$ gives the best results both in terms of number of iterations and CPU time. Its performance and that of other methods are further assessed by solving a variety of problems presented in Part II of this paper (Adjiman *et al.*, 1997).

Method		Single Up.			One Up./Iter		
		Iter.	CPU sec.	t_U (%)	Iter.	CPU sec.	t_U (%)
Gerschgorin (I.1)		74	37.5	0.5	31	41.6	0.0
E-Matrix (I.2)	$E = 0$	61	30.6	1.6	25	37.2	0.2
E-Matrix (I.2)	$E = \text{diag}(\Delta H)$	61	29.2	1.0	25	35.4	0.1
Mori-Kokame (I.3)		69	32.8	1.9	25	31.5	0.2
Lower bounding Hessian (I.4)		61	31.6	1.4	25	33.1	0.2
Kharitonov (I.5)		61	32.8	12.3	25	36.7	1.7
Hertz (I.6)		59	32.9	1.4	25	32.8	0.5
Scaled G. (II.1)	$d_i = 1$	56	24.9	0.3	30	36.5	0.3
Scaled G. (II.1)	$d_i = (x_i^U - x_i^L)$	38	13.6	1.7	17	19.9	0.5
H-Matrix (II.2)		62	32.7	0.6	25	34.5	0.3
Min. Max. distance (II.3)		54	21.8	16.7	23	30.4	5.0

Table 10: Alkylation Process Design Results for different α computation methods. t_U denotes the percentage of total CPU time spent generating convex underestimators.

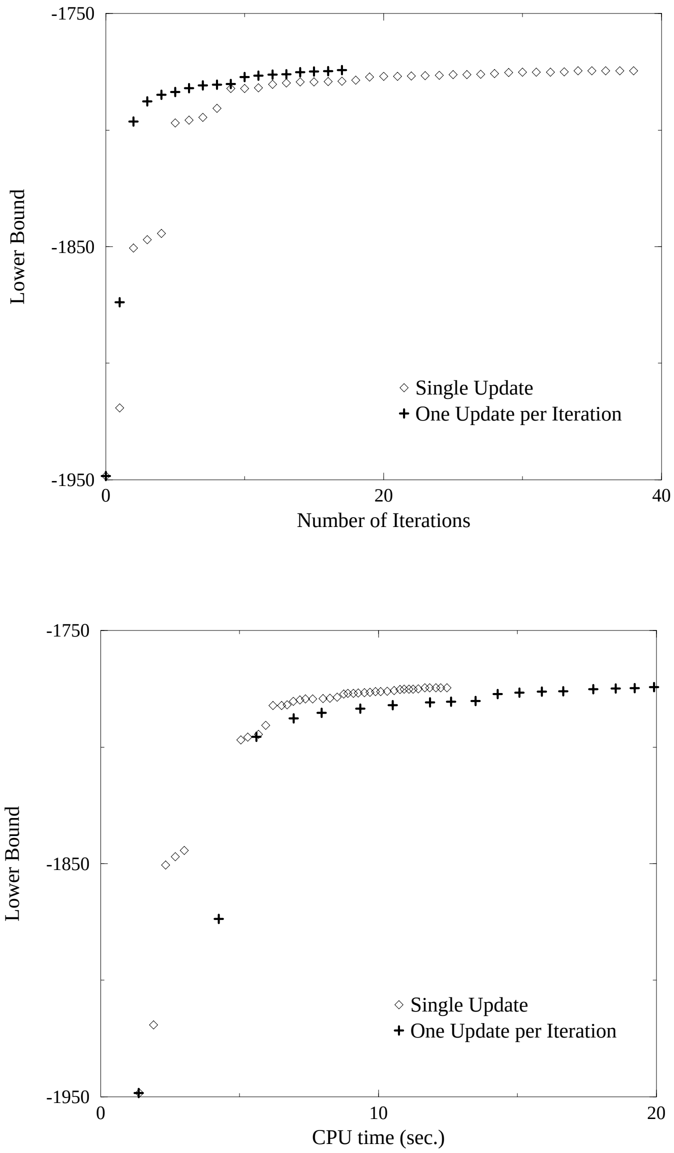

Figure 3: Lower bound vs. Iteration number or CPU time for the scaled Gerschgorin theorem approach

5 Conclusions

As demonstrated in this paper, the αBB algorithm can rigorously identify the global solution of twice-differentiable nonconvex programming problems based on a general convex underestimating scheme. Several methods have been developed for the calculation of the α parameters necessary for the construction of a tight valid convex underestimator. They fall within two classes: a uniform diagonal shift approach which requires the computation of a lower bound on the minimum eigenvalue of an interval Hessian matrix; a non-uniform diagonal shift approach in which a set of α parameters that ensures the positive semi-definiteness of an interval matrix is obtained. The decomposition of the nonlinear functions into a sum of terms constitutes a central concept for the generation of tight underestimators and the successful operation of the algorithm: the general nonconvex terms should involve few variables, reducing the cost of the α computations. In addition, the construction of customized lower bounding functions for different classes of terms, such as bilinear, trilinear, fractional, fractional trilinear or univariate concave, enhances the convergence characteristics of the αBB. The algorithm and all the underestimating techniques were successfully tested on an unconstrained example and a constrained design problem. Algorithmic and implementation related issues, as well as extensive computational studies, are reported in the second part of this paper (Adjiman *et al.*, 1997).

Acknowledgments

The authors gratefully acknowledge financial support from the National Science Foundation, the Air Force Office of Scientific Research, the National Institutes of Health, Exxon Foundation and Mobil Technology Company.

References

C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A Global Optimization Method, αBB, for General Twice–Differentiable NLPs – II. Implementation and Computational Results. accepted for publication, 1997.

C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas. A Global Optimisation Method, αBB, for Process Design. *Computers chem. Engng, Suppl.*, 20:S419–S424, 1996.

C. S. Adjiman and C. A. Floudas. Rigorous Convex Underestimators for General Twice–Differentiable Problems. *J. of Glob. Opt.*, 9:23–40, 1996.

F. A. Al-Khayyal. Jointly Constrained Bilinear Programs and Related Problems : An Overview. *Computers Math. Applic.*, 19(11):53–62, 1990.

F. A. Al-Khayyal and J. E. Falk. Jointly Constrained Biconvex Programming. *Maths Ops Res.*, 8:273–286, 1983.

I.P. Androulakis, C. D. Maranas, and C. A. Floudas. αBB : A Global Optimization Method for General Constrained Nonconvex Problems. *J. of Glob. Opt.*, 7:337–363, 1995.

J. Bracken and G. P. McCormick. *Selected Applications of Nonlinear Programming.* Wiley, 1968.

A. S. Deif. The Interval Eigenvalue Problem. *Z. angew. Math. Mech*, 71(1):61–64, 1991.

R. S. Dembo. A Set of Geometric Programming Test Problems and their Solutions. *Math. Programming*, 10:192–213, 1976.

C. A. Floudas. Deterministic Global Optimization in Design, Control, and Computational Chemistry. In L.T. Biegler, T.F. Coleman, A.R. Conn, and F.N. Santosa, editors, *IMA Volumes in Mathematics and its Applications : Large Scale Optimization with Applications, Part II*, volume 93, pages 129–184, New York, 1997. Springer-Verlag.

C. A. Floudas and I. E. Grossmann. Algorithmic Approaches to Process Synthesis : Logic and Global Optimization. In *FOCAPD'94*, volume 91 of *AICHE Symposium Series*, pages 198–221, 1995.

C. A. Floudas and P. M. Pardalos, editors. *State of the Art in Global Optimization.* Kluwer Academic Publishers, 1996.

C. A. Floudas and V. Visweswaran. A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs : I. Theory. *Computers chem. Engng.*, 14:1397–1417, 1990.

C. A. Floudas and V. Visweswaran. A Primal-Relaxed Dual Global Optimization Approach. *J. of Opt. Th. and Appl.*, 78(2):187–225, 1993.

S. Gerschgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk SSSR, Ser. fiz.-mat.*, 6:749–754, 1931.

I. E. Grossmann, editor. *Global Optimization in Engineering Design*. Kluwer Academic Publishers, 1996.

D. Hertz. The Extreme Eigenvalues and Stability of Real Symmetric Interval Matrices. *IEEE Transactions on Automatic Control*, 37(4):532–535, 1992.

V.L. Kharitonov. Asymptotic Stability of an Equilibrium Position of a Family of Systems of Linear Differential Equations. *Differential Equations*, 78:1483–1485, 1979.

W.B. Liu and C. A. Floudas. A Remark on the GOP Algorithm for Global Optimization. *J. of Glob. Opt.*, 3(3):519–521, 1993.

C. D. Maranas and C.A. Floudas. A Global Optimization Approach for Lennard-Jones Microclusters. *J. Chem. Phys.*, 97(10):7667–7677, 1992.

C. D. Maranas and C.A. Floudas. A deteministic Global Optimization Approach for Molecular Structure Determination. *J. Chem. Phys.*, 100(2):1247–1261, 1994a.

C. D. Maranas and C.A. Floudas. Global Minimum Potential Energy Conformations of Small Molecules. *J. of Glob. Opt.*, 4:135–170, 1994b.

C. D. Maranas and C.A. Floudas. Finding All Solutions of Nonlinearly Constrained Systems of Equations. *J. of Glob. Opt.*, 7(2):143–182, 1995.

C. D. Maranas and C.A. Floudas. Global Optimization in Generalized Geometric Programming. *Computers chem. Engng.*, 21(4):351–370, 1997.

G. P. McCormick. Computability of Global Solutions to Factorable Nonconvex Programs : Part I – Convex Underestimating Problems. *Math. Programming*, 10:147–175, 1976.

T. Mori and H. Kokame. Eigenvalue Bounds for a Certain Class of Interval Matrices. *IEICE Trans. Fundamentals*, E77-A(10):1707–1709, 1994.

B. A. Murtagh and M. A. Saunders. *MINOS 5.4 User's Guide*. Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, CA., 1983.

A. Neumaier. *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

A. Neumaier. An Optimality Criterion for Global Quadratic Optimization. *J. of Glob. Opt.*, 2:201–208, 1992.

A. Neumaier. On Satisfying Second–Order Optimality Conditions Using Modified Cholesky Factorizations. *SIAM J. Optimization*, 1997. submitted.

H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Ellis Horwood Series in Mathematics and its Applications. Halsted Press, 1988.

J. Rohn. Bounds on Eigenvalues of Interval Matrices. Technical Report no. 688, Institute of Computer Science, Academy of Sciences, Prague, 1996.

E.M.B. Smith and C.C. Pantelides. Global Optimisation of General Process Models. In I.E. Grossmann, editor, *Global Optimization in Engineering Design*, pages 355–386. Kluwer Academic Publishers, 1996.

C. Stephens. Interval and Bounding Hessians. In I. M. et al. Bomze, editor, *Developments in Global Optimization*, pages 109–199. Kluwer Academic Publishers, 1997.

L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review*, 38:49–95, 1996.

V. Visweswaran and C. A. Floudas. A Global Optimization Algorithm GOP for Certain Classes of Nonconvex NLPs : II. Application of Theory and Test Problems. *Computers chem. Engng.*, 14:1419–1434, 1990.

V. Visweswaran and C. A. Floudas. New Properties and Computational Improvement of the GOP Algorithm for Problems with Quadratic Objective Function and Constraints. *J. of Glob. Opt.*, 3(3):439–462, 1993.

V. Visweswaran and C. A. Floudas. New Formulations and Branching Strategies for the GOP Algorithm. In I. E. Grossmann, editor, *Global Optimization in Engineering Design*, Kluwer Book Series in Nonconvex Optimization and its Applications, 1996a. Chapter 3.

V. Visweswaran and C. A. Floudas. Computational Results for an Efficient Implementation of the GOP Algorithm and its Variants. In I. E. Grossmann, editor, *Global Optimization in Engineering Design*, Kluwer Book Series in Nonconvex Optimization and its Applications, 1996b. Chapter 4.

A Underestimating Bilinear Terms

The relationship between the general underestimators of the αBB algorithm and the convex envelope for bilinear terms used by McCormick (1976) and Al-Khayyal and Falk (1983) is explored in this Appendix.

A.1 The Convex Envelope of Bilinear Terms

The convex envelope for a bilinear term xy with $x \in [x^L, x^U]$ and $y \in [y^L, y^U]$ is given by Equation (2). This discontinuous function is piecewise linear and the following theorem defines the regions of applicability of each linear expression.

Theorem A.1.1 *The convex envelope of a bilinear term xy with $x \in [x^L, x^U]$ and $y \in [y^L, y^U]$ is*

$$w = \begin{cases} x^L y + y^L x - x^L y^L & \text{if } y \le -\frac{y^U - y^L}{x^U - x^L} x + \frac{x^U y^U - x^L y^L}{x^U - x^L} \\ x^U y + y^U x - x^U y^U & \text{otherwise} \end{cases}$$

Proof Using the definition of w given by Equation (2), let us determine when $w = x^L y + y^L x - x^L y^L$, i.e. when $x^L y + y^L x - x^L y^L \ge x^U y + y^U x - x^U y^U$.

$$\begin{aligned} & x^L y + y^L x - x^L y^L \ge x^U y + y^U x - x^U y^U \\ & \quad \Leftrightarrow \quad -(x^U - x^L) y - (y^U - y^L) x \ge x^L y^L + x^U y^U \\ & \quad \Leftrightarrow \quad y \le -\frac{y^U - y^L}{x^U - x^L} x + \frac{x^U y^U - x^L y^L}{x^U - x^L} \end{aligned}$$

$\blacksquare$

Geometrical Interpretation From Theorem A.1.1, the line separating the two regions of applicability of the linear underestimators is given by

$$y = -\frac{y^U - y^L}{x^U - x^L} x + \frac{x^U y^U - x^L y^L}{x^U - x^L}.$$

This can be equivalently expressed as

$$\begin{aligned} y &= -\frac{y^U - y^L}{x^U - x^L}x + \frac{x^U y^U - x^L y^L}{x^U - x^L} + \frac{x^L y^U - x^L y^U}{x^U - x^L} \\ &= -\frac{y^U - y^L}{x^U - x^L}x + \frac{y^U - y^L}{x^U - x^L}x^L + \frac{x^U y^U - x^L y^U}{x^U - x^L} \\ &= \frac{y^U - y^L}{x^L - x^U}(x - x^L) + y^U \end{aligned}$$

This is the equation of a line passing through (x^L, y^U) and (x^U, y^L). As shown in Figure 4, it crosses the rectangle $[x^L, x^U] \times [y^L, y^U]$ diagonally, from the top left corner to the bottom right corner. Region (1) corresponds to the domain of applicability of $x^L y + y^L x - x^L y^L$ and region (2) to that of $x^U y + y^U x - x^U y^U$.

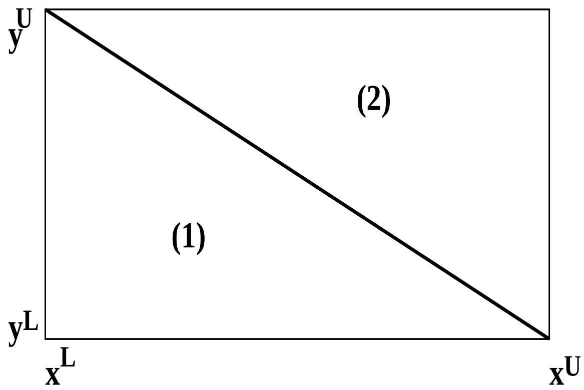

Figure 4: Geometrical Interpretation of Theorem A.1.1.

A.2 Discontinuous α Parameters

Because the convex envelope for bilinear terms is a discontinuous function, no continuous function can define an underestimator of comparable quality. To generate a tight lower bound, the α values used to construct an underestimator of the form given in Equation (9) must be discontinuous. For region (1), the following expressions are suggested:

$$\alpha_x = \frac{1}{2}\,\frac{y - y^L}{x^U - x}; \quad \alpha_y = \frac{1}{2}\,\frac{x - x^L}{y^U - y},$$

while for region (2), the following can be used:

$$\alpha_x = \frac{1}{2}\frac{y^U - y}{x - x^L}; \quad \alpha_y = \frac{1}{2}\frac{x - x^L}{y^U - y}.$$

The values of α_x and α_y are non-negative for all $x \in [x^L, x^U]$ and all $y \in [y^L, y^U]$. Using the proposed α values therefore generates a *valid* underestimator throughout the domain of interest. Although the denominators of the α expressions in region (1) vanish along two borders of the rectangle ($x = x^U$ and $y = y^U$) it can be seen from Figure 4 that this does not pose any problem as the region (2) expressions can be used along these lines in order to evaluate α_x and α_y. Similar observations hold for the region (2) expressions.

The underestimator

$$\mathcal{L}(x,y) = xy + \alpha_x(x - x^U)(x - x^L) + \alpha_y(y - y^U)(y - y^L)$$

then becomes $x^L y + y^L x - x^L y^L$ in region (1) and $x^U y + y^U x - x^U y^U$ in region (2). It is therefore equivalent to the convex envelope of a bilinear term as defined in Theorem A.1.1.

B The H-Matrix Method – Implementation Issues

The central result of the H-matrix method, presented in Section 3.5.2, is used to define sufficient conditions on the diagonal shift matrix Δ so that $[A] + 2\Delta$ is positive definite. This Appendix describes how these conditions can be applied to the computation of a valid Δ. The notation used here is the same as in Section 3.5.2. The following remarks on the sufficient conditions are made in order to facilitate the design of an iterative procedure:

Condition 1: Checking the positive definiteness of a real symmetric matrix The positive definiteness of some real symmetric matrix A can be tested by constructing a Cholesky decomposition of the matrix. If this decomposition fails, the matrix A is not positive definite. Such an approach does not provide any insight on how far the matrix A is from being positive definite and therefore does not guide the choice of an appropriate diagonal shift matrix Δ. The *modified Cholesky decomposition* of Neumaier (1997) can be used to obtain quantitative information. The results of this modified Cholesky decomposition are a lower triangular matrix L and a vector $\boldsymbol{p}$ such that

$$A = LL^T - \mathrm{diag}(\boldsymbol{p}).$$

The matrix $A + \mathrm{diag}(\boldsymbol{p})$ is therefore positive definite. This result can be used for an initial guess of Δ, namely $\Delta_0 = 1/2\ \mathrm{diag}(\boldsymbol{p})$. Interestingly, while the modified Cholesky decomposition is more conclusive than the traditional Cholesky decomposition, the number of operations it requires is almost the same as that required by the traditional Cholesky decomposition. Moreover, when A is positive definite, the vector $\boldsymbol{p}$ vanishes and the result of the traditional Cholesky decomposition is recovered.

Condition 2: Checking the H-matrix property Whether the matrix $[A] + 2\Delta$ is an H-matrix can be tested as follows:

> Construct $[B] := C\left([A] + 2\Delta\right)$, where C is the pre-conditioning matrix $(\tilde{A}_M + \Delta)^{-1}$. If $0 \in B_{ii}$ for some i, $[A] + 2\Delta$ is not an H-matrix. Otherwise, compute $[B]\boldsymbol{e}$ where $\boldsymbol{e} = (1, \cdots, 1)^T$ and $[B]\boldsymbol{u}$ where $\boldsymbol{u} = (\langle B_{11}\rangle^{-1}, \cdots, \langle B_{nn}\rangle^{-1})^T$ and $\langle B\rangle$ is the comparison matrix of $[B]$. If

all the elements of at least one of the resulting vectors are positive, $[A] + 2\Delta$ is an H-matrix.

It should be noted that this test constitutes a *sufficient* condition of the H-matrix property. It may therefore fail to recognize that $[A] + 2\Delta$ is indeed an H-matrix. The use of the pre-conditioning matrix C is an important factor in reducing the failure rate of this test as it helps to diagonalize the matrix $[A] + 2\Delta$. Choosing the identity matrix for C would not change the theoretical analysis but would have a negative effect on the numerical aspects of the method. An exact check for the H-matrix property can be performed using Proposition 3.7.3 of Neumaier (1990). However, this check is much more computationally expensive than the simple matrix and vector multiplications used in the above procedure.

Bounds on Δ: Since the fact that a valid diagonal shift matrix Δ has been identified may not be recognized by the proposed tests, an upper limit should be imposed on the elements of Δ to ensure that the quality of the constructed underestimator is no worse than that obtained with other techniques. As previously mentioned, the maximum separation distance is an effective measure of the tightness of the underestimator. In practice, Method I.2, the E-matrix method with $E = 0$, has been used to determine the largest maximum separation distance d^E_{max} allowed. If no suitable Δ matrix with a maximum separation distance smaller than d^E_{max} is found by the H-matrix method, Δ^E, the uniform diagonal shift matrix of Method I.2, is returned.

Based on the above observations, the following procedure can be set up for the computation of a non-uniform diagonal shift matrix:

Step 1 Given the interval Hessian matrix $[A]$, compute a valid uniform shift matrix Δ^E using Method I.2. All diagonal elements of Δ^E are equal to some non-negative scalar α^E. If $\alpha^E = 0$, $[A]$ is positive semi-definite: return $\Delta = 0$. Otherwise, compute $d^E_{max} = \alpha^E||\boldsymbol{x}^U - \boldsymbol{x}^L||^2$. Set iteration counter $k = 0$.

Step 2 Compute the modified midpoint matrix $\tilde{A}_M$ of $[A]$ and its modified Cholesky decomposition $LL^T - \text{diag}(\boldsymbol{p})$. Set $\Delta_0 = 1/2\text{diag}(\boldsymbol{p})$. Check that

$$d_{max,0} = \sum_i \Delta_{0,ii}(x_i^U - x_i^L)^2 < d^E_{max}.$$

If this is not satisfied, return $\Delta = \Delta^E$.

Step 3 Compute the pre-conditioning matrix $C_k = \left(\tilde{A}_M + \Delta_k\right)^{-1}$ and the interval matrix $[B_k] = C_k([A] + \Delta_k)$. If $[B_k]$ is an H-matrix, return $\Delta = \Delta_k$.

Step 4 If the maximum number of iterations has been reached, return $\Delta = \Delta^E$.

Step 5 Since $[B_k]$ is not an H-matrix, a new guess for Δ is needed. Specifically, the elements of Δ must be augmented. Compute a step-size δ such that

$$\delta = \frac{d^E_{max} - d_{max,k}}{||\boldsymbol{x}^U - \boldsymbol{x}^L||^2}.$$

Then, $\Delta_{k+1,ii} = \Delta_{k,ii} + \delta$. Increase the iteration counter k by 1. Go to Step 3.