Preamble. This is a reprint of the article:

M. Schulze Darup and M. Mönnigmann. Improved automatic computation of Hessian matrix spectral bounds. *SIAM Journal of Scientific Computing*, 38(4): 2068–2090, 2016.

The digital object identifier (DOI) of the original article is:

`10.1137/15M1025773`

Improved automatic computation of Hessian matrix spectral bounds

Moritz Schulze Darup[†] *and* M. Mönnigmann[†]

Abstract

This paper presents a fast and powerful method for the computation of eigenvalue bounds for Hessian matrices $\nabla^2\varphi(x)$ of nonlinear wice continuously differentiable functions $\varphi : \mathcal{U} \subseteq \mathbb{R}^n \to \mathbb{R}$ on hyperrectangles $\mathcal{B} \subset \mathcal{U}$. The method is based on a recently proposed procedure [9] for an efficient computation of spectral bounds using extended codelists. Both that approach and the one presented here substantially differ from established methods in that they deliberately do not use any interval matrices and thus result in a favorable numerical complexity of order $\mathcal{O}(n)\,N(\varphi)$, where $N(\varphi)$ denotes the number of operations needed to evaluate φ at a point in its domain. We improve the method presented Mönnigmann (in [9]) by exploiting sparsity, which naturally arises in the underlying codelists. The new method provides bounds that are as good as or better than those from the most accurate existing method in about 82% of the test cases.

1. Introduction

We present important improvements for a recently proposed method (see [9]) for the efficient calculation of spectral bounds for Hessian matrices on hyperrectangles. The improvements build on a systematic treatment of sparsity of the involved matrices, which will be shown to result in significantly tighter eigenvalue bounds. The problem can concisely be summarized as follows. Let $\varphi : \mathcal{U} \subseteq \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function on an open set $\mathcal{U} \subseteq \mathbb{R}^n$ and let $\mathcal{B} = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n]$ be a closed hyperrectangle in $\mathcal{U}$. We seek bounds $\underline{\lambda}, \overline{\lambda} \in \mathbb{R}$ such that the relations $\underline{\lambda} \leq \lambda \leq \overline{\lambda}$ hold for all eigenvalues λ of all matrices $H \in \{\nabla^2\varphi(x) \,|\, x \in \mathcal{B}\}$. More precisely, we solve the following problem:

$$\text{Find } \underline{\lambda}, \overline{\lambda} \in \mathbb{R} \text{ such that } \underline{\lambda} \leq \min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2\varphi(x)) \text{ and } \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2\varphi(x)) \leq \overline{\lambda}, \tag{1.1}$$

where $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ denote the smallest and largest eigenvalue, respectively, of the symmetric matrix $H \in \mathbb{R}^{n\times n}$. A bound $\underline{\lambda}$ (resp., $\overline{\lambda}$) is called *tight* if there exists at least

[†] M. Schulze Darup and M. Mönnigmann are with Automatic Control and Systems Theory, Department of Mechanical Engineering, Ruhr-Universität Bochum, 44801 Bochum, Germany. E-mails: `moritz.schulzedarup@rub.de` and `martin.moennigmann@rub.de`.

one $x \in \mathcal{B}$ such that $\underline{\lambda} = \lambda_{\min}(\nabla^2\varphi(x))$ (resp., $\overline{\lambda} = \lambda_{\max}(\nabla^2\varphi(x))$). Note that the problem statement (1.1) does not necessarily imply that $\underline{\lambda}$ and $\overline{\lambda}$ are tight.

Eigenvalue bounds $\underline{\lambda}$, $\overline{\lambda}$ are used, for example, in numerical optimization methods to detect convexity or to construct convex underestimators of nonconvex functions [1–3]. If (1.1) yields $\underline{\lambda} \geq 0$, then φ is convex on the interior of the hyperrectangle $\mathcal{B}$. While no conclusion on the convexity can be drawn if (1.1) results in $\underline{\lambda} < 0$, the bound $\underline{\lambda}$ can still be used to construct a convex underestimator for φ on $\mathcal{B}$. Specifically,

$$\breve{\varphi}(x) = \varphi(x) - \frac{1}{2}\underline{\lambda}\sum\nolimits_{i=1}^{n}\left(\underline{x}_i - x_i\right)\left(\overline{x}_i - x_i\right) \tag{1.2}$$

is convex, coincides with φ at the vertices of $\mathcal{B}$, and bounds φ from below everywhere else in $\mathcal{B}$. Since a large fraction of the total computation time is spent on the calculation of convex underestimators in global optimization methods [1], fast methods for solving (1.1) are of interest. We briefly note that (1.1) must also be solved in certain problems in automatic control and systems theory. An illustrative example is given in [8].

Existing approaches to solving (1.1) proceed in two steps. First, a symmetric interval matrix (also called *interval Hessian*) that contains all Hessians $\nabla^2\varphi(x)$ on $\mathcal{B}$ is calculated:

$$\begin{gathered}\text{Find } \underline{H} = \underline{H}^T, \overline{H} = \overline{H}^T \in \mathbb{R}^{n\times n} \text{ such that } \underline{H}_{ij} \leq \left(\nabla^2\varphi(x)\right)_{ij} \leq \overline{H}_{ij} \\ \text{for every } i, j \in \{1, \ldots, n\} \text{ and every } x \in \mathcal{B}.\end{gathered} \tag{1.3}$$

This task can efficiently be carried out by combining interval arithmetics (IA; see [10], for example) and automatic differentiation (AD; see [4, 11], for example). In the second step, spectral bounds can be found by solving the following problem, which is similar to but different from (1.1):

$$\text{Find } \underline{\lambda}, \overline{\lambda} \in \mathbb{R} \text{ such that } \underline{\lambda} \leq \min_{H\in\mathcal{H}} \lambda_{\min}(H) \text{ and } \max_{H\in\mathcal{H}} \lambda_{\max}(H) \leq \overline{\lambda}, \tag{1.4}$$

where $\mathcal{H} = \{H \in \mathbb{R}^{n\times n} \,|\, H_{ij} \in [\underline{H}_{ij}, \overline{H}_{ij}],\, H = H^T\}$ is the set of all symmetric matrices that respect the bounds $\underline{H}$ and $\overline{H}$. Various approaches exist to solving (1.4) (see, e.g., [2, 5, 6, 12]). However, since $\{\nabla^2\varphi(x) \,|\, x \in \mathcal{B}\} \subseteq \mathcal{H}$, problem (1.4) is conservative compared to the original problem (1.1). In fact it is the very point of the method introduced in [9] and refined in the present paper to avoid computing interval matrices of the form (1.3) when solving (1.1) in order to avoid this conservatism.

We briefly summarize the computational complexity of the existing methods. The computation of the matrices $\underline{H}$ and $\overline{H}$ in (1.3) requires $\mathcal{O}(n^2)\,N(\varphi)$ (resp., $\mathcal{O}(n)\,N(\varphi)$) operations if the forward (resp., backward) mode of AD is used, where $N(\varphi)$ denotes the number of operations needed to evaluate φ at a point in its domain [4]. After the interval Hessian has been calculated, solving (1.4) requires between $\mathcal{O}(n^2)$ operations for the interval variant of Gershgorin's circle criterion [2, 5] and $\mathcal{O}(2^n\, n^3)$ operations for Hertz and Rohn's method [6, 12]. The latter method is an important benchmark in that it yields *tight* spectral bounds for the matrix set $\mathcal{H}$. Hertz and Rohn's method therefore provides the most accurate option to solve (1.1) via (1.3) and (1.4). Nevertheless, the resulting eigenvalue bounds are in general not tight due to the conservatism in (1.3).

The total numerical effort of any approach that uses (1.3) and (1.4) corresponds to the sum of the efforts for calculating $\underline{H}, \overline{H}$ and solving (1.4). Thus, the numerical effort for the established methods varies between $\mathcal{O}(n)\,N(\varphi) + \mathcal{O}(n^2)$ (backward mode AD combined with Gershgorin's circle criterion) and $\mathcal{O}(n^2)\,N(\varphi) + \mathcal{O}(2^n\, n^3)$ operations (forward mode AD combined with Hertz and Rohn's method). The major advantage of the direct method presented in [9] is its low computational complexity, which was shown to be of order $\mathcal{O}(n)\,N(\varphi)$.

It is the purpose of this paper to improve the method introduced in [9] such that sparsity can exploited to find tighter eigenvalue bounds. Here sparsity refers not to the Hessian $\nabla^2\varphi(x)$ itself but to the intermediate expressions that are typical for AD and IA based algorithms. Essentially, these methods represent a function $x \to \varphi(x)$, $x = (x_1, \ldots, x_n)^T$ by a sequence of single arithmetic operations called a *codelist* (see section 3). Codelists result in very simple intermediate functions by construction. Since many of these intermediate functions only depend on very few of the x_i, they have very sparse intermediate Hessians as functions of x. This sparsity impedes a precise estimation of eigenvalues, because it introduces superfluous zero eigenvalues in the intermediate Hessians (see Example 1). It is the very point of the present paper to improve the method from [9] by treating the sparsity in the intermediate expressions in a systematic fashion. The improvements do not increase the numerical effort compared to the original method in [9]. In fact, sparsity needs to be investigated once during the automatic generation of the extended codelist. The computations required to evaluate the codelist to obtain eigenvalue bounds on a specific hyperrectangle $\mathcal{B}$ are no more expensive than those for the nonsparse case treated in [9]. While the computational effort remains the same, the improved method results in significantly tighter eigenvalue bounds than the original procedure from [9]. To show this, we investigate 1522 examples (taken from the COCONUT collection [14]; see [13] for details) and compare the eigenvalue bounds resulting from the improved procedure to those obtained with the original one [9] and to bounds obtained with the interval Hessian (1.3) and Gershgorin's circle criterion and Hertz and Rohn's method.

We summarize the major aspects of the direct method for the computation of eigenvalue bounds from [9] in section 3. Our main result, the exploitation of sparsity for the improvement of the eigenvalue bounds from [9], is stated in section 4. We analyze 1522 numerical examples from [13, 14] in section 5.[1] Conclusions are given in section 6.

2. Notation and preliminaries

We frequently use index sets $\mathcal{J} \subseteq \mathcal{N}$, where $\mathcal{N} := \mathbb{N}_{[1,n]}$ and where $\mathbb{N}_{[m,n]} := \{i \in \mathbb{N} \,|\, m \leq i \leq n\}$. The complement of an index set $\mathcal{J}$ is defined as $\mathcal{J}^c := \mathcal{N} \setminus \mathcal{J}$. The cardinality of an index set $\mathcal{J}$ is denoted by $|\mathcal{J}|$.

It is convenient to state eigenvalue bounds as intervals (e.g., $\lambda \in [\underline{\lambda}, \overline{\lambda}]$). Intervals $[\underline{a}, \overline{a}] \subset \mathbb{R}$ with $\underline{a} \leq \overline{a}$ are further abbreviated by $[a] := [\underline{a}, \overline{a}]$ whenever appropriate. Interval equality $[\underline{a}, \overline{a}] = [\underline{b}, \overline{b}]$ is understood to mean $\underline{a} = \underline{b}$ and $\overline{a} = \overline{b}$. We frequently carry out calculations on intervals with standard IA rules. The required rules are summarized in Lemma 2.1 and Table 1.

Lemma 2.1 (basic interval operations [10])**:** *Let $[a] = [\underline{a}, \overline{a}]$ and $[b] = [\underline{b}, \overline{b}]$ be intervals in $\mathbb{R}$. Let $a \in [\underline{a}, \overline{a}]$, $b \in [\underline{b}, \overline{b}]$, and $c \in \mathbb{R}$ be arbitrary real numbers. Then, the relations in the second column of Table 1 hold under the additional restrictions stated in the last column.*

A lowercase letter surrounded by brackets may refer to a real interval $[x] = [\underline{x}, \overline{x}] \subset \mathbb{R}$ (as introduced above) or a hyperrectangle $[x] = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n] \subset \mathbb{R}^n$ (with $n \geq 1$). In the latter case, the interval operations listed in Table 1 are understood to apply to every component. For a hyperrectangle $[x] \subset \mathbb{R}^n$ and a nonempty index set $\mathcal{J} \subseteq \mathcal{N}$ with the m elements $j_1 < \cdots < j_m$, the term $[x_{\mathcal{J}}]$ refers to the hyperrectangle $[x_{\mathcal{J}}] = [x_{j_1}] \times \cdots \times [x_{j_m}]$. Throughout the paper, $\mathcal{B}$ denotes some hyperrectangle $[x] \subset \mathbb{R}^n$.

It is furthermore convenient to use *null matrices* in $\mathbb{R}^{m \times r}$, which we denote by $0_{m,r}$, when dealing with sparsity. For the special cases $m = 0$ or $r = 0$ we obtain an *empty matrix.* Formally,

[1] Results were obtained with `Jcodegen`, a code generator available from the authors on request or to be used online on www.rus.rub.de/software/jcodegen. `Jcodegen` generates ANSI-C code for the algorithm described in Proposition 4.17 for a given function φ. In particular sparsity is treated automatically. The specific hyperrectangle $\mathcal{B}$ is passed to the resulting code as a runtime parameter.

the empty square matrix $0_{0,0}$ has no eigenvalues. It proves useful to assign the eigenvalue bounds $[\lambda] = [0, 0]$ to it. Finally, the Cartesian unit vector along the kth direction is denoted by $e_k \in \mathbb{R}^n$.

Table 1: Basic IA.

Operation/Bounds		Definition		Restriction
$a + b$	$\in [a] + [b]$	$:= [\underline{a} + \underline{b}, \overline{a} + \overline{b}]$		
$a\,b$	$\in [a]\,[b]$	$:= [\min\{\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b}\}, \max\{\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b}\}]$		
$1/a$	$\in 1/[a]$	$:= [1/\overline{a}, 1/\underline{a}]$		$0 \notin [a]$
a^m	$\in [a]^m$	$:= [\underline{a}^m, \overline{a}^m]$	if $\underline{a} > 0$ or m odd	
		$:= [\overline{a}^m, \underline{a}^m]$	if $\overline{a} < 0$ and m even	
		$:= [0, \max\{\underline{a}^m, \overline{a}^m\}]$	otherwise	
$\sqrt{a}$	$\in \sqrt{[a]}$	$:= [\sqrt{\underline{a}}, \sqrt{\overline{a}}]$		$\underline{a} \geq 0$
$\exp a$	$\in \exp([a])$	$:= [\exp(\underline{a}), \exp(\overline{a})]$		
$\ln(a)$	$\in \ln([a])$	$:= [\ln(\underline{a}), \ln(\overline{a})]$		$\underline{a} > 0$
$a + c$	$\in [a] + c$	$:= [\underline{a} + c, \overline{a} + c]$		
$c\,a$	$\in c\,[a]$	$:= [c\,\underline{a}, c\,\overline{a}]$	if $c \geq 0$	
		$:= [c\,\overline{a}, c\,\underline{a}]$	otherwise	

3. Direct computation of eigenvalue bounds for Hessian matrices on hyperrectangles

We summarize the method introduced in [9] for the direct solution of (1.1) as needed in the present paper. We assume the function φ can be evaluated at an arbitrary point $x \in \mathcal{U}$ by carrying out a finite sequence of operations of the form

$$\begin{array}{rcl} y_1 & = & x_1, \\ & \vdots & \\ y_n & = & x_n, \\ y_{n+1} & = & \Phi_{n+1}(y_1, \dots, y_n), \\ y_{n+2} & = & \Phi_{n+2}(y_1, \dots, y_n, y_{n+1}), \\ & \vdots & \\ y_{n+t} & = & \Phi_{n+t}(y_1, \dots, y_n, y_{n+1}, \dots, y_{n+t-1}), \\ \varphi & = & y_{n+t}, \end{array} \tag{3.1}$$

where each Φ_{n+k}, $k = 1, \dots, t$, represents one of the elementary operations listed in the first column of Table 2. We treat the same operations as in [9] for ease of comparison. Note that additional unary operations can be added according to the rules given in [9]. We refer to (3.1) as the *codelist* of the function φ.

The codelist (3.1) can be used to evaluate the function value $\varphi(x)$ at a specific point x in its domain. Using AD [11] the codelist (3.1) can be extended in such a way that the gradient $\nabla\varphi(x)$ or the Hessian $\nabla^2\varphi(x)$ at the point x is calculated. Moreover, using AD and IA, (3.1) can be modified such that interval extensions, interval gradients, or interval Hessians of φ on hyperrectangles $\mathcal{B} \subset \mathcal{U}$ are computed. In fact, extended codelists are commonly used to solve problem (1.3) as part of the established procedures for the computation of eigenvalue bounds (see, e.g., [2]). In contrast, the method introduced in [9] only requires the interval gradient but not the interval Hessian. Essentially, the codelist is extended by arithmetic operations that compute the eigenvalue bounds for the Hessian of the intermediate function in every codelist line. Formally, this leads to the extended codelist which we introduce in the following theorem.

Table 2: Rules for the calculation of y_k, $[y_k]$, $[\nabla y_k]$, and $[\lambda_k]$ in the kth line of the codelist (3.1). $[\nabla y_k]$ refers to the interval gradient of line k with respect to x. The interval operators $[\Lambda_s([a])]$ and $[\Lambda_t([a],[b])]$ are defined in (3.2) and (3.3).

`op` Φ_k	y_k	$[y_k]$	$[\nabla y_k]$	$[\lambda_k]$
`var`	x_k	$[x_k]$	$[e_k, e_k]$	$[0,0]$
`add`	$y_i + y_j$	$[y_i] + [y_j]$	$[\nabla y_i] + [\nabla y_j]$	$[\lambda_i] + [\lambda_j]$
`mul`	$y_i\, y_j$	$[y_i]\,[y_j]$	$[y_j][\nabla y_i]+[y_i][\nabla y_j]$	$[y_j][\lambda_i] + [y_i][\lambda_j] + [\Lambda_t([\nabla y_i],[\nabla y_j])]$
`powNat`	y_i^m	$[y_i]^m$	$m\,[y_i]^{m-1}\,[\nabla y_i]$	$m[y_i]^{m-2}((m-1)[\Lambda_s([\nabla y_i])]+[y_i][\lambda_i])$
`oneOver`	$1/y_i$	$1/[y_i]$	$-[y_k]^2\,[\nabla y_i]$	$[y_k]^2\,(2\,[y_k]\,[\Lambda_s([\nabla y_i])] - [\lambda_i])$
`sqrt`	$\sqrt{y_i}$	$[\sqrt{[y_i]}]$	$1/(2\,[y_k])\,[\nabla y_i]$	$1/(2\,[y_k])([\lambda_i]+1/(-2\,[y_i])[\Lambda_s([\nabla y_i])])$
`exp`	$\exp(y_i)$	$[\exp([y_i])]$	$[y_k]\,[\nabla y_i]$	$[y_k]\,([\Lambda_s([\nabla y_i])] + [\lambda_i])$
`ln`	$\ln(y_i)$	$[\ln([y_i])]$	$1/[y_i]\,[\nabla y_i]$	$1/[y_i]\,([\lambda_i] - 1/[y_i]\,[\Lambda_s([\nabla y_i])])$
`addC`	$y_i + c$	$[y_i]+[c,c]$	$[\nabla y_i]$	$[\lambda_i]$
`mulByC`	$c\,y_i$	$c\,[y_i]$	$c\,[\nabla y_i]$	$c\,[\lambda_i]$

Theorem 3.1 (algorithm for direct eigenvalue bound computation [9, Proposition 4.2])**:** *Assume φ is twice continuously differentiable on $\mathcal{U}$ and can be written as a codelist* (3.1)*. Let $\mathcal{B} = [x_1] \times \cdots \times [x_n] \subset \mathcal{U}$ be arbitrary. Then, for all $x \in \mathcal{B}$, we have $\varphi(x) \in [\varphi]$, $\nabla\varphi(x) \in [\nabla\varphi]$, and $[\lambda_{\min}(\nabla^2\varphi(x)), \lambda_{\max}(\nabla^2\varphi(x))] \subseteq [\lambda_\varphi]$, where $[\varphi]$, $[\nabla\varphi]$, and $[\lambda_\varphi]$ are calculated by the following algorithm:*

1. *For $k = 1,\dots,n$, set $[y_k] = [\underline{x}_k, \overline{x}_k]$, $[\nabla y_k] = [e_k, e_k]$, and $[\lambda_k] = [0,0]$.*

2. *For $k = n+1,\dots,n+t$, calculate $[y_k]$, $[\nabla y_k]$, and $[\lambda_k]$ according to the third, fourth, and fifth columns of Table 2, respectively.*

3. *Set $[\varphi] = [y_{n+t}]$, $[\nabla\varphi] = [\nabla y_{n+t}]$, and $[\lambda_\varphi] = [\lambda_{n+t}]$.*

In Table 2, we use the interval operators $[\Lambda_s([a])]$ and $[\Lambda_t([a],[b])]$, which are defined according to

$$[\Lambda_s([a])] := \begin{cases} [a]^2 & \text{if } m = 1, \\ [0, \sum_{i=1}^m \max\{\underline{a}_i^2, \overline{a}_i^2\}] & \text{otherwise,} \end{cases} \tag{3.2}$$

$$\text{and} \quad [\Lambda_t([a],[b])] := \begin{cases} 2\,[a]\,[b] & \text{if } m = 1, \\ [-\beta,\beta] + \sum_{i=1}^m [\underline{a}_i, \overline{a}_i]\,[\underline{b}_i, \overline{b}_i] & \text{otherwise} \end{cases} \tag{3.3}$$

for hyperrectangles $[a] = [\underline{a}_1, \overline{a}_1] \times \cdots \times [\underline{a}_m, \overline{a}_m] \subset \mathbb{R}^m$ and $[b] = [\underline{b}_1, \overline{b}_1] \times \cdots \times [\underline{b}_m, \overline{b}_m] \subset \mathbb{R}^m$, where $\beta := \sqrt{(\sum_{i=1}^m \max\{\underline{a}_i^2, \overline{a}_i^2\})(\sum_{i=1}^m \max\{\underline{b}_i^2, \overline{b}_i^2\})}$. We refer to [9, Lems. 2.2 and 2.3] for details on $[\Lambda_s([a])]$ and $[\Lambda_t([a],[b])]$.

4. Improved computation of eigenvalue bounds using sparsity

If sparsity is exploited, tighter eigenvalue bounds can be obtained than those that result from the method summarized in Section 3. This is evident from the following motivating example.

Example 1 (method from [9] applied to $\varphi(x) = x_1^2 + x_2^2$)**:** *Consider the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ with $\varphi(x) = x_1^2 + x_2^2$. Theorem 3.1 results in the following extended codelist. Note that the expressions for y_k listed in* (4.1) *do not result from Theorem 3.1 but are given only for illustration of the codelist* (3.1) *of φ.*

k	y_k	$[y_k]$	$[\nabla y_k]$	$[\lambda_k]$
1	x_1	$[x_1]$	$[e_1, e_1]$	$[0, 0]$
2	x_2	$[x_2]$	$[e_2, e_2]$	$[0, 0]$
3	y_1^2	$[y_1]^2$	$2\,[y_1]\,[\nabla y_1]$	$2([\Lambda_s([\nabla y_1])] + [y_1]\,[\lambda_1])$
4	y_2^2	$[y_2]^2$	$2\,[y_2]\,[\nabla y_2]$	$2([\Lambda_s([\nabla y_2])] + [y_2]\,[\lambda_2])$
5	$y_3 + y_4$	$[y_3] + [y_4]$	$[\nabla y_3] + [\nabla y_4]$	$[\lambda_3] + [\lambda_4]$
	$\varphi = y_5$	$[\varphi] = [y_5]$	$[\nabla \varphi] = [\nabla y_5]$	$[\lambda_\varphi] = [\lambda_5]$

(4.1)

Evaluating the extended codelist (4.1) *for the hyperrectangle $\mathcal{B} = [0, 1] \times [0, 1]$ by computing $[y_k]$, $[\nabla y_k]$, and $[\lambda_k]$ and storing the results line by line yields*

$$\begin{array}{lll} [y_1] = [0,1], & [\nabla y_1] = ([1,1],[0,0])^T, & [\lambda_1] = [0,0], \\ [y_2] = [0,1], & [\nabla y_2] = ([0,0],[1,1])^T, & [\lambda_2] = [0,0], \\ [y_3] = [0,1], & [\nabla y_3] = ([0,2],[0,0])^T, & [\lambda_3] = [0,2], \\ [y_4] = [0,1], & [\nabla y_4] = ([0,0],[0,2])^T, & [\lambda_4] = [0,2], \\ [y_5] = [0,2], & [\nabla y_5] = ([0,2],[0,2])^T, & [\lambda_5] = [0,4], \end{array} \tag{4.2}$$

where $[\Lambda_s([\nabla y_1])] = [0, 1]$ and $[\Lambda_s([\nabla y_2])] = [0, 1]$ according to (3.3)*. Thus, we obtain the eigenvalue bounds $[\lambda_\varphi] = [\lambda_5] = [0, 4]$ for $\nabla^2(\varphi(x))$ on $\mathcal{B}$. Now, consider the functions $g, h : \mathbb{R}^2 \to \mathbb{R}$ with $g(x) = x_1^2$ and $h(x) = x_2^2$. From*

$$\nabla^2 g(x) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \quad \textit{and} \quad \nabla^2 h(x) = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix},$$

we infer that both $\nabla^2 g(x)$ and $\nabla^2 h(x)$ have the eigenvalues 0 *and* 2 *for every $x \in \mathcal{B}$. Hence, the eigenvalue bounds $[\lambda_g] = [\lambda_3] = [0, 2]$ and $[\lambda_h] = [\lambda_4] = [0, 2]$ that result in lines 3 and 4 of extended codelist* (4.1) *are tight. The eigenvalue bounds $[\lambda_5] = [\lambda_3] + [\lambda_4] = [0, 4]$ that result in the subsequent line are conservative, however. In fact, the Hessian of φ reads*

$$\nabla^2 \varphi(x) = \nabla^2 g(x) + \nabla^2 h(x) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

for all $x \in \mathcal{B}$ and the tight eigenvalue bounds obviously read $[\lambda_\varphi^*] = [2, 2]$.

The Hessian matrices $\nabla^2 g(x)$ and $\nabla^2 h(x)$ in Exmp. 1 have zero eigenvalues which disappear when adding the two functions to $f(x) = g(x) + h(x)$. The situation illustrated in Example 1 arises naturally in the codelists introduced in Section 3, because codelists build up functions of many variables from functions of very few of these variables. In order to mitigate eigenvalue bound overestimation in these cases, we need to consider functions like $g(x)$ and $h(x)$ in Example 1 as functions of only those variables that they actually depend on nonlinearly. To this end, some simple terminology and intermediate results are introduced in section 4.1. Subsequently, sparse sums, products, and compositions are treated in Sects. 4.2, 4.3, and 4.4, respectively. Section 4.5 summarizes how to compute the improved eigenvalue bounds based on the rules introduced in Sects. 4.2–4.4.

4.1. Sparsity handling using reduced Hessians and reduced gradients

As pointed out in Example 1, sparsity occurs if functions depend at most linearly on some variables x_i, where *at most linear dependence* is defined as follows.

Definition 4.1 (independence and at most linear dependence)**:** *Let $f : \mathcal{U} \to \mathbb{R}$ be a continuously differentiable function on an open set $\mathcal{U} \subseteq \mathbb{R}^n$. Let $i \in \mathcal{N}$. The function f is said to depend at most linearly on x_i if there exists a $c \in \mathbb{R}$ such that*

$$\frac{\partial f}{\partial x_i}(x) = c \text{ for all } x \in \mathcal{U}. \tag{4.3}$$

If (4.3) *holds with $c = 0$, the function is said to be independent of x_i*

Assume a function f is known to depend at most linearly on x_j for all $j \in \mathcal{L}_f$, where $\mathcal{L}_f \subseteq \mathcal{N}$ is a given index set. Then, only the eigenvalues of the *reduced Hessian* (see Def. 4.2) associated with the index set $\mathcal{J} = \mathcal{L}_f^c$ are nontrivial, i.e., not necessarily equal to zero.

Definition 4.2 (reduced Hessian $\nabla^2_{\mathcal{J}} f(x)$)**:** *Let $f : \mathcal{U} \to \mathbb{R}$ be a twice continuously differentiable function on an open set $\mathcal{U} \subseteq \mathbb{R}^n$. Let $\mathcal{J} \subseteq \mathcal{N}$ be an index set and let $m = |\mathcal{J}|$. If $m = 0$ set $\nabla^2_{\mathcal{J}} f(x) = 0_{0,0}$; otherwise denote the m elements of $\mathcal{J}$ by $j_1 < \cdots < j_m$ in ascending order and define the reduced Hessian $\nabla^2_{\mathcal{J}} f(x) \in \mathbb{R}^{m\times m}$ by its elements*

$$\left(\nabla^2_{\mathcal{J}} f(x)\right)_{ik} = \frac{\partial^2 f(x)}{\partial x_{j_i} \partial x_{j_k}},$$

where $i, k \in \mathbb{N}_{[1,m]}$.

We also need to consider reduced gradient vectors.

Definition 4.3 (reduced gradient $\nabla_{\mathcal{J}} f(x)$)**:** *Let $f : U \to \mathbb{R}$ be a continuously differentiable function on an open set $\mathcal{U} \subseteq \mathbb{R}^n$. Let $\mathcal{J} \subseteq \mathcal{N}$ be a nonempty index set and let $m = |\mathcal{J}|$. Denote the m elements of $\mathcal{J}$ by $j_1 < \cdots < j_m$ in ascending order and define the reduced gradient $\nabla_{\mathcal{J}} f(x) \in \mathbb{R}^m$ by its elements*

$$\left(\nabla_{\mathcal{J}} f(x)\right)_i = \frac{\partial f(x)}{\partial x_{j_i}},$$

where $i \in \mathbb{N}_{[1,m]}$.

Note that the index set is assumed to be nonempty in Def. 4.3 of the reduced gradient. An empty index set corresponds to a codelist line that does not depend on any variable. Since this case never occurs, we exclude it. The empty index set is, in contrast, included in Def. 4.2 of the reduced Hessian, because this case does occur whenever a codelist line defines a linear or affine function.

We can easily evaluate eigenvalue bounds for the Hessian of a function from eigenvalue bounds for its reduced Hessian. This is stated precisely in Lemma 4.4.

Lemma 4.4 (spectral bounds for Hessian from reduced Hessian)**:** *Let f denote a twice continuously differentiable function $f : \mathcal{U} \to \mathbb{R}$ on an open set $\mathcal{U}$. Let the index set $\mathcal{L}_f \subseteq \mathcal{N}$ be such that f depends at most linearly on x_i for all $i \in \mathcal{L}_f$. Let $\mathcal{B} \subset \mathcal{U}$ and let the interval $[\lambda_f^\dagger] \subset \mathbb{R}$ be such that*

$$\underline{\lambda}_f^\dagger \le \min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2_{\mathcal{L}_f^c} f(x)) \qquad \text{and} \qquad \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2_{\mathcal{L}_f^c} f(x)) \le \overline{\lambda}_f^\dagger. \tag{4.4}$$

Then, the eigenvalues of the Hessian $\nabla^2 f(x)$ on $\mathcal{B}$ lie in the interval

$$[\lambda_f] = \begin{cases} [\lambda_f^\dagger] & \text{if } \mathcal{L}_f = \emptyset, \\ [0, 0] & \text{if } \mathcal{L}_f = \mathcal{N}, \\ [\min\{\underline{\lambda}_f^\dagger, 0\}, \max\{\overline{\lambda}_f^\dagger, 0\}] & \text{otherwise.} \end{cases} \tag{4.5}$$

Proof. We consider the cases in (4.5) separately. $\mathcal{L}_f = \emptyset$ implies $\mathcal{L}_f^c = \mathcal{N}$ and consequently $\nabla^2 f(x) = \nabla^2_{\mathcal{L}_f^c} f(x)$, which proves the first case. In the second case, i.e., $\mathcal{L}_f = \mathcal{N}$, we have $\nabla^2 f(x) = \nabla^2_{\mathcal{L}_f} f(x)$. Since f depends at most linearly on x_i for all $i \in \mathcal{L}_f$, we find $\nabla^2_{\mathcal{L}_f} f(x) = 0_{n,n}$. Thus, the eigenvalue bounds $[\lambda_f] = [0,0]$ hold. Regarding the third case, we note that $\emptyset \subset \mathcal{L}_f \subset \mathcal{N}$ implies $\emptyset \subset \mathcal{L}_f^c \subset \mathcal{N}$. Thus, $m = |\mathcal{L}_f|$ satisfies $0 < m < n$. Without loss of generality we assume $\mathcal{L}_f^c = \mathbb{N}_{[1,m]}$. Then

$$\nabla^2 f(x) = \begin{pmatrix} \nabla^2_{\mathcal{L}_f^c} f(x) & 0_{m,n-m} \\ 0_{n-m,m} & \nabla^2_{\mathcal{L}_f} f(x) \end{pmatrix} = \begin{pmatrix} \nabla^2_{\mathcal{L}_f^c} f(x) & 0_{m,n-m} \\ 0_{n-m,m} & 0_{n-m,n-m} \end{pmatrix}.$$

Now consider an arbitrary but fixed $x \in \mathcal{B}$. We obtain

$$\lambda_{\min}(\nabla^2 f(x)) = \min\{\lambda_{\min}(\nabla^2_{\mathcal{L}_f^c} f(x)), 0\} \quad \text{and} \tag{4.6}$$

$$\lambda_{\max}(\nabla^2 f(x)) = \max\{\lambda_{\max}(\nabla^2_{\mathcal{L}_f^c} f(x)), 0\} \tag{4.7}$$

based on the block-diagonal structure of $\nabla^2_{\mathcal{L}_f^c} f(x)$. Bounding (4.6) below and bounding (4.7) above for all $x \in \mathcal{B}$ yields

$$[\min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2 f(x)), \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2 f(x))] \subseteq [\min\{\underline{\lambda}_f^\dagger, 0\}, \max\{\overline{\lambda}_f^\dagger, 0\}]$$

according to (4.6) and (4.7) and condition (4.4). ■

4.2. Improved eigenvalue bounds for the sum of two functions

We collect some recurring conditions first.

Conditions 4.5: *Let g and h denote twice continuously differentiable functions $g : \mathcal{U} \to \mathbb{R}$ and $h : \mathcal{U} \to \mathbb{R}$ on an open set $\mathcal{U} \subset \mathbb{R}^n$. Let the index sets $\mathcal{L}_g \subseteq \mathcal{N}$ and $\mathcal{L}_h \subseteq \mathcal{N}$ be such that g (resp., h) depends at most linearly on x_i for all $i \in \mathcal{L}_g$ (resp., all $i \in \mathcal{L}_h$). Moreover, let the index sets $\mathcal{I}_g \subseteq \mathcal{L}_g$ and $\mathcal{I}_h \subseteq \mathcal{L}_h$ with $\mathcal{I}_g \subset \mathcal{N}$ and $\mathcal{I}_h \subset \mathcal{N}$ be such that g (resp., h) is independent of x_i for all $i \in \mathcal{I}_g$ (resp., all $i \in \mathcal{I}_h$). Let $\mathcal{B} \subset \mathcal{U}$ and assume there exist intervals $[\lambda_g^\dagger] \subset \mathbb{R}$ and $[\lambda_h^\dagger] \subset \mathbb{R}$ such that*

$$\underline{\lambda}_g^\dagger \le \min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2_{\mathcal{L}_g^c} g(x)) \qquad \textit{and} \qquad \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2_{\mathcal{L}_g^c} g(x)) \le \overline{\lambda}_g^\dagger, \tag{4.8}$$

$$\underline{\lambda}_h^\dagger \le \min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2_{\mathcal{L}_h^c} h(x)) \qquad \textit{and} \qquad \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2_{\mathcal{L}_h^c} h(x)) \le \overline{\lambda}_h^\dagger. \tag{4.9}$$

Note that $\mathcal{L}_g$ and $\mathcal{L}_h$ may be equal to $\mathcal{N}$, while $\mathcal{I}_g$ and $\mathcal{I}_h$ are assumed to be proper subsets of $\mathcal{N}$. This difference arises because a codelist line may depend at most linearly on all variables x_i, but it will never be independent of all x_i. Now assume Conds. 4.5 hold and we intend to calculate eigenvalue bounds for $\nabla^2 f(x)$ on a hyperrectangle $\mathcal{B}$ for $f(x) = g(x) + h(x)$. We could determine eigenvalue bounds for the full Hessians $\nabla^2 g(x)$ and $\nabla^2 h(x)$ with Lemma 4.4 and apply the rule for the eigenvalue bounds of the sum of full Hessians (line `add` in Table 2 reproduced from [9]). However, we show in Lemma 4.8 below that it is advantageous to, roughly speaking, carry out calculations with the sparse Hessians as long as possible and to apply Lemma 4.4 as late as possible. We first state the rules for determining $\mathcal{L}_f$, $\mathcal{I}_f$ and the eigenvalues of the reduced Hessian of f in Lems. 4.6 and 4.7, respectively. The trivial proof of Lemma 4.6 is omitted for brevity.

Lemma 4.6 (index sets for sums)**:** *Assume Conds. 4.5 hold and consider the function $f : \mathcal{U} \to \mathbb{R}$ with $f(x) = g(x) + h(x)$. Let $\mathcal{L}_f = \mathcal{L}_g \cap \mathcal{L}_h$ and $\mathcal{I}_f = \mathcal{I}_g \cap \mathcal{I}_h$. Then, f depends at most linearly on x_i for all $i \in \mathcal{L}_f$ and f is independent of x_i for all $i \in \mathcal{I}_f$.*

Lemma 4.7 (spectral bounds for reduced Hessian of sums)**:** *Assume Conds. 4.5 hold and consider the function* $f : \mathcal{U} \to \mathbb{R}$*,* $f(x) = g(x) + h(x)$*. Let* $\mathcal{L}_f = \mathcal{L}_g \cap \mathcal{L}_h$*. Then,*

$$\underline{\lambda}_f^\dagger \leq \min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2_{\mathcal{L}_f^c} f(x)) \qquad \textit{and} \qquad \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2_{\mathcal{L}_f^c} f(x)) \leq \overline{\lambda}_f^\dagger, \tag{4.10}$$

where $[\lambda_f^\dagger]$ *is computed according to the rules listed in Table 3.*

Table 3: Rules for the computation of eigenvalue bounds $[\lambda_f^\dagger]$ for the reduced Hessian $\nabla^2_{\mathcal{L}_f^c} f(x)$ of a sum $f(x) = g(x) + h(x)$. Let $\mathcal{L}_\cup$ be short for $\mathcal{L}_\cup := \mathcal{L}_g \cup \mathcal{L}_h$. See the end of section 4.2 for a discussion of the eight cases.

Case	$[\lambda_f^\dagger]$	Condition
1	$[0, 0]$	$\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N}$
2	$[\lambda_g^\dagger]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N}$
3	$[\lambda_h^\dagger]$	$\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N}$
4	$[\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger\}, \max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger\}]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge \mathcal{L}_\cup = \mathcal{N}$
5	$[\lambda_g^\dagger] + [\lambda_h^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_g = \mathcal{L}_h$
6	$[\lambda_g^\dagger] + [\min\{\underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_h^\dagger, 0\}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_g \subset \mathcal{L}_h$
7	$[\min\{\underline{\lambda}_g^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\}] + [\lambda_h^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{L}_g$
8	$[\min\{\underline{\lambda}_g^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\}] + [\min\{\underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_h^\dagger, 0\}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_g \not\subseteq \mathcal{L}_h \wedge \mathcal{L}_h \not\subseteq \mathcal{L}_g$

Proof. We prove the fourth case in Table 3 since it will be instrumental for Example 2. All other cases in Table 3 can be proven analogously. The reduced Hessian of f reads $\nabla^2_{\mathcal{L}_f^c} f(x) = \nabla^2_{\mathcal{L}_f^c} g(x) + \nabla^2_{\mathcal{L}_f^c} h(x)$. From $\mathcal{L}_g \subset \mathcal{N}$, $\mathcal{L}_h \subset \mathcal{N}$, and $\mathcal{L}_\cup = \mathcal{L}_g \cup \mathcal{L}_h = \mathcal{N}$, we infer $\emptyset \subset \mathcal{L}_g \subset \mathcal{N}$ and $\emptyset \subset \mathcal{L}_h \subset \mathcal{N}$ and consequently $\emptyset \subset \mathcal{L}_g^c \subset \mathcal{N}$ and $\emptyset \subset \mathcal{L}_h^c \subset \mathcal{N}$. Thus, the cardinalities $r = |\mathcal{L}_g^c|$ and $s = |\mathcal{L}_h^c|$ satisfy $0 < r < n$ and $0 < s < n$. Moreover, $\mathcal{L}_g \cup \mathcal{L}_h = \mathcal{N}$ implies $\mathcal{L}_g^c \cap \mathcal{L}_h^c = \emptyset$. Hence, there does not exist any index $i \in \mathcal{N}$ such that both $i \in \mathcal{L}_g^c$ and $i \in \mathcal{L}_h^c$. We assume $\mathcal{L}_g^c = \mathbb{N}_{[1,r]}$ and $\mathcal{L}_h^c = \mathbb{N}_{[r+1,r+s]}$ without loss of generality. Note that $\mathcal{L}_f^c = \mathcal{L}_g^c \cup \mathcal{L}_h^c$ implies $\mathcal{L}_f^c = \mathbb{N}_{[1,r+s]}$ and $m = |\mathcal{L}_f^c| = r + s$ under this assumption. Thus, $\nabla^2_{\mathcal{L}_f^c} f(x)$ equals

$$\begin{pmatrix} \nabla^2_{\mathcal{L}_g^c} g(x) & 0_{r,s} \\ 0_{s,r} & 0_{s,s} \end{pmatrix} + \begin{pmatrix} 0_{r,r} & 0_{r,s} \\ 0_{s,r} & \nabla^2_{\mathcal{L}_h^c} h(x) \end{pmatrix} = \begin{pmatrix} \nabla^2_{\mathcal{L}_g^c} g(x) & 0_{r,s} \\ 0_{s,r} & \nabla^2_{\mathcal{L}_h^c} h(x) \end{pmatrix}. \tag{4.11}$$

The block-diagonal structure implies

$$\lambda_{\min}(\nabla^2_{\mathcal{L}_f^c} f(x)) = \min\{\lambda_{\min}(\nabla^2_{\mathcal{L}_g^c} g(x)), \lambda_{\min}(\nabla^2_{\mathcal{L}_h^c} h(x))\} \quad \text{and} \tag{4.12}$$

$$\lambda_{\max}(\nabla^2_{\mathcal{L}_f^c} f(x)) = \max\{\lambda_{\max}(\nabla^2_{\mathcal{L}_g^c} g(x)), \lambda_{\max}(\nabla^2_{\mathcal{L}_h^c} h(x))\} \tag{4.13}$$

for an arbitrary but fixed $x \in \mathcal{B}$. Bounding (4.12) below and bounding (4.13) above for all $x \in \mathcal{B}$ yields

$$[\min_{x\in\mathcal{B}} \lambda_{\min}(\nabla^2 f(x)), \max_{x\in\mathcal{B}} \lambda_{\max}(\nabla^2 f(x))] \subseteq [\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger\}, \max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger\}],$$

where we used (4.12) and (4.13) and Conditions 4.5. Thus, the eigenvalues of $\nabla^2_{\mathcal{L}_f^c} f(x)$ on $\mathcal{B}$ lie in the interval $[\lambda_f^\dagger] = [\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger\}, \max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger\}]$ as claimed in Table 3. ∎

We anticipated the bounds from Lemma 4.7 can be shown to be as tight as or tighter than those from the original method proposed in [9] that does not account for sparsity. This can now be shown in Lemma 4.8 below. Recall that the bounds in [9] result in $[\lambda_f] = [\lambda_g] + [\lambda_h]$ for $f(x) = g(x) + h(x)$ according to [9, Proposition 3.2(iii)].

Table 4: Fundamental cases underlying Table 3.

Case	Reduced Hessian $\nabla^2_{\mathcal{L}_f^c} f(x)$	Contribution	Condition
(i)	$0_{0,0}$	none	$\mathcal{L}_g^c = \emptyset \wedge \mathcal{L}_h^c = \emptyset$
(ii)	$\nabla^2_{\mathcal{L}_f^c} g(x)$	first Hessian	$\mathcal{L}_g^c \supset \emptyset \wedge \mathcal{L}_h^c = \emptyset$
(iii)	$\nabla^2_{\mathcal{L}_f^c} h(x)$	second Hessian	$\mathcal{L}_g^c = \emptyset \wedge \mathcal{L}_h^c \supset \emptyset$
(iv)	$\nabla^2_{\mathcal{L}_f^c} g(x) + \nabla^2_{\mathcal{L}_f^c} h(x)$	both Hessians	$\mathcal{L}_g^c \supset \emptyset \wedge \mathcal{L}_h^c \supset \emptyset$

Lemma 4.8 (improved bounds for sums)**:** *Assume Conditions 4.5 hold and let f, $[\lambda_f^\dagger]$, and $\mathcal{L}_f$ be as in Lemma 4.7. Let $[\lambda_f]$, $[\lambda_g]$, and $[\lambda_h]$ be the eigenvalue bounds for the Hessians $\nabla^2 f(x)$, $\nabla^2 g(x)$, and $\nabla^2 h(x)$ on $\mathcal{B}$, calculated according to* (4.5)*. Then,*

$$[\lambda_f] \subseteq [\lambda_g] + [\lambda_h]. \tag{4.14}$$

Proof. We prove the relation for the fourth case in Table 3. The remaining cases can be proven analogously. As pointed out in the proof of Lemma 4.7, we have $\emptyset \subset \mathcal{L}_g \subset \mathcal{N}$ and $\emptyset \subset \mathcal{L}_h \subset \mathcal{N}$. Thus, the right-hand side in (4.14) yields

$$\begin{aligned}
[\lambda_g] + [\lambda_h] &= [\min\{\underline{\lambda}_g^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\}] + [\min\{\underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_h^\dagger, 0\}] \\
&= [\min\{\underline{\lambda}_g^\dagger, 0\} + \min\{\underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\} + \max\{\overline{\lambda}_h^\dagger, 0\}], \\
&= [\min\{\underline{\lambda}_g^\dagger + \underline{\lambda}_h^\dagger, \underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger + \overline{\lambda}_h^\dagger, \overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger, 0\}],
\end{aligned} \tag{4.15}$$

where the equations hold according to the third case in (4.5), by definition of the sum of two intervals (see Table 1), and by definition of $\min\{\cdot\}$ and $\max\{\cdot\}$, respectively. To evaluate the left-hand side in (4.14), we have to analyze the index set $\mathcal{L}_f$. We obviously have $\mathcal{L}_f = \mathcal{L}_g \cap \mathcal{L}_h \subset \mathcal{N}$. Thus, the second case in (4.5) does not apply. However, from the conditions characterizing the fourth case in Table 3, it is not clear whether $\mathcal{L}_f = \emptyset$ or $\mathcal{L}_f \supset \emptyset$. Thus, according to (4.5), the left-hand side in (4.14) results in

$$[\lambda_f] = \begin{cases} [\lambda_f^\dagger] & \text{if } \mathcal{L}_f = \emptyset, \\ [\min\{\underline{\lambda}_f^\dagger, 0\}, \max\{\overline{\lambda}_f^\dagger, 0\}] & \text{if } \emptyset \subset \mathcal{L}_f \subset \mathcal{N}. \end{cases}$$

However, since $[a] \subseteq [\min\{\underline{a}, 0\}, \max\{\overline{a}, 0\}]$, the relation $[\lambda_f] \subseteq [\min\{\underline{\lambda}_f^\dagger, 0\}, \max\{\overline{\lambda}_f^\dagger, 0\}]$ holds in both cases. Since $[\lambda_f^\dagger] = [\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger\}, \max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger\}]$ according to Lemma 4.7 (resp., Table 3), we obtain

$$\begin{aligned}
[\lambda_f] &\subseteq [\min\{\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger\}, 0\}, \max\{\max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger\}, 0\}], \\
&= [\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger, 0\} \max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger, 0\}].
\end{aligned} \tag{4.16}$$

Comparing (4.15) and (4.16) yields

$$[\min\{\underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger, 0\}] \subseteq [\min\{\underline{\lambda}_g^\dagger + \underline{\lambda}_h^\dagger, \underline{\lambda}_g^\dagger, \underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger + \overline{\lambda}_h^\dagger, \overline{\lambda}_g^\dagger, \overline{\lambda}_h^\dagger, 0\}],$$

which proves (4.14). ■

Lemmas 4.7 and 4.8 are based on the eight cases listed in Table 3. Since they are not obvious at first sight, it is instructive to see how these eight cases arise from the four simpler ones listed in Table 4. The first case in Table 4 applies if both $\nabla^2_{\mathcal{L}_f^c} g(x)$ and $\nabla^2_{\mathcal{L}_f^c} h(x)$ vanish because of $\mathcal{L}_g^c = \mathcal{L}_h^c = \emptyset$. Since these two conditions, i.e., $\mathcal{L}_g^c = \emptyset \wedge \mathcal{L}_h^c = \emptyset$, are equivalent to $\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N}$, case (i) in Table 4 is equivalent to case 1 in Table 3. Analogously, cases (ii) and (iii) in Table 4, where either $\nabla^2_{\mathcal{L}_f^c} g(x)$ or $\nabla^2_{\mathcal{L}_f^c} h(x)$ contribute to $\nabla^2_{\mathcal{L}_f^c} f(x)$, are equivalent to cases 2 and 3 in Table 3, respectively. It remains to relate case (iv) in Table 4 to cases 4–8 in Table 3. In fact, the conditions of the cases 4–8 in Table 3 all imply $\mathcal{L}_g^c \supset \emptyset$ and $\mathcal{L}_h^c \supset \emptyset$, which are the defining conditions for case (iv) in Table 4. Figure 1 illustrates that every instance of case (iv) from Table 4 actually uniquely belongs to one of the cases 4–8 from Table 3.

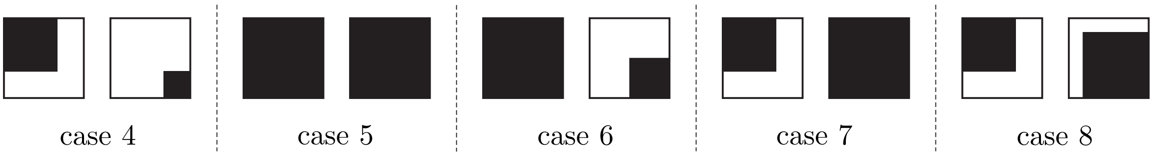

Figure 1: Illustration of the sparsity patterns of the reduced Hessians $\nabla^2_{\mathcal{L}_f^c} g(x)$ and $\nabla^2_{\mathcal{L}_f^c} h(x)$ for cases 4–8 in Table 3 and Lemma 4.7. White areas correspond to zero blocks in the Hessians, black areas to nontrivial blocks. The black areas in case 4 correspond to the nontrivial submatrices $\nabla^2_{\mathcal{L}_g^c} g(x)$ and $\nabla^2_{\mathcal{L}_h^c} h(x)$ on the left-hand side of (4.11).

4.3. Improved eigenvalue bounds for the composition of two functions

We collect some recurring conditions again first.

Conditions 4.9: *Assume Conditions 4.5 hold. Let $r : \mathcal{V} \to \mathbb{R}$ be a twice differentiable function on an open set $\mathcal{V} \subset \mathbb{R}$ and assume $g(x) \in \mathcal{V}$ for every $x \in \mathcal{U}$. Moreover, assume there exist intervals $[r'] \subset \mathbb{R}$ and $[r''] \subset \mathbb{R}$ and a hyperrectangle $[\nabla g] \subset \mathbb{R}^n$ such that*

$$\underline{r}' \le r'(g(x)) \le \overline{r}', \quad \underline{r}'' \le r''(g(x)) \le \overline{r}'', \quad \text{and} \quad \underline{\nabla g}_i \le (\nabla g(x))_i \le \overline{\nabla g}_i \tag{4.17}$$

for every $x \in \mathcal{B}$ and every $i \in \mathcal{N}$, where $r'(z)$ and $r''(z)$ refer to the first and the second derivative of $r(z)$, respectively.

The following lemma, which we state without proof, provides rules for the identification of at most linear dependencies and independencies of compositions.

Lemma 4.10 (index sets for compositions)**:** *Assume Conditions 4.9 hold and consider the function $f : \mathcal{U} \to \mathbb{R}$, $f(x) = r(g(x))$. Let*

$$\mathcal{I}_f = \mathcal{I}_g \qquad \text{and} \qquad \mathcal{L}_f = \begin{cases} \mathcal{L}_g & \text{if } r \text{ is an affine function,} \\ \mathcal{I}_g & \text{otherwise.} \end{cases} \tag{4.18}$$

Then, f depends at most linearly on x_i for all $i \in \mathcal{L}_f$ and f is independent of x_i for all $i \in \mathcal{I}_f$.

Bounds for compositions can now be calculated as follows.

Lemma 4.11 (spectral bounds for reduced Hessian of compositions)**:** *Assume Conditions 4.9 hold and consider the function $f : \mathcal{U} \to \mathbb{R}$, $f(x) = r(g(x))$. Let $\mathcal{L}_f$ be defined as in Lemma 4.10. Then the bounds* (4.10) *hold for $[\lambda_f^\dagger]$ computed according to the rules listed in Table 5.*

Table 5: Rules for the computation of eigenvalue bounds $[\lambda_f^\dagger]$ for the reduced Hessian $\nabla^2_{\mathcal{L}_f^c} f(x)$ of compositions $f(x) = r(g(x))$.

Case	$[\lambda_f^\dagger]$	Condition
1	$[r''] \, [\Lambda_s([\nabla_{\mathcal{L}_f^c} g])]$	$\mathcal{L}_g = \mathcal{N}$
2	$[r''] \, [\Lambda_s([\nabla_{\mathcal{L}_f^c} g])] + [r'] \, [\lambda_g^\dagger]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_g$
3	$[r''] \, [\Lambda_s([\nabla_{\mathcal{L}_f^c} g])] + [r'] \, [\min\{\underline{\lambda}_g^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\}]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_g$

Proof. We prove the last case in Table 5. The remaining cases can be proven analogously. The reduced Hessian of f reads $\nabla^2_{\mathcal{L}_f^c} f(x) = r''(g(x)) \, \nabla_{\mathcal{L}_f^c} g(x) \, \nabla^T_{\mathcal{L}_f^c} g(x) + r'(g(x)) \, \nabla^2_{\mathcal{L}_f^c} g(x)$. Combining the two conditions of case 3 in Table 5 yields $\mathcal{L}_f \subset \mathcal{L}_g \subset \mathcal{N}$, which implies $\mathcal{L}_f^c \supset \mathcal{L}_g^c \supset \emptyset$. Thus, $m = |\mathcal{L}_f^c|$ and $r = |\mathcal{L}_g^c|$ satisfy $m > r > 0$. We assume $\mathcal{L}_f^c = \mathbb{N}_{[1,m]}$ and $\mathcal{L}_g^c = \mathbb{N}_{[1,r]}$ without loss of generality. Under these assumptions, we obtain

$$\nabla^2_{\mathcal{L}_f^c} f(x) = r''(g(x)) \, \nabla_{\mathcal{L}_f^c} g(x) \, \nabla^T_{\mathcal{L}_f^c} g(x) + r'(g(x)) \begin{pmatrix} \nabla^2_{\mathcal{L}_g^c} g(x) & 0_{r,s} \\ 0_{s,r} & 0_{s,s} \end{pmatrix}, \tag{4.19}$$

where $s = m - r > 0$. Since $\nabla g(x) \in [\nabla g]$ for every $x \in \mathcal{B}$, we find

$$\underline{\Lambda}_s([\nabla_{\mathcal{L}_f^c} g]) \le \lambda_{\min}(\nabla_{\mathcal{L}_f^c} g(x) \, \nabla^T_{\mathcal{L}_f^c} g(x)) \quad \text{and} \quad \lambda_{\max}(\nabla_{\mathcal{L}_f^c} g(x) \, \nabla^T_{\mathcal{L}_f^c} g(x)) \le \overline{\Lambda}_s([\nabla_{\mathcal{L}_f^c} g])$$

for every $x \in \mathcal{B}$ according to [9, Lemma 2.2]. Combining this intermediate result with the bounds on $r'(g(x))$ and $r''(g(x))$ from Conditions 4.9 yields the eigenvalue bounds

$$[\lambda_f^\dagger] = [r''] \, [\Lambda_s([\nabla_{\mathcal{L}_f^c} g])] + [r'] \, [\min\{\underline{\lambda}_g^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\}]$$

on $\mathcal{B}$. ■

Finally, Lemma 4.12 below shows that for the composition of two functions, the bounds from Lemma 4.11 are as tight as or tighter than those from the original method proposed in [9]. Recall the bounds in [9] result in $[\lambda_f] = [r''] \, [\Lambda_s([\nabla g])] + [r'] \, [\lambda_g]$ for $f(x) = r(g(x))$ according to [9, Prop 3.4].

Lemma 4.12 (improved bounds for compositions)**:** *Assume Conditions 4.9 hold and let f, $[\lambda_f^\dagger]$, and $\mathcal{L}_f$ be as in Lemma 4.11. Let $[\lambda_f]$ and $[\lambda_g]$ be the eigenvalue bounds for the Hessians $\nabla^2 f(x)$ and $\nabla^2 g(x)$ on $\mathcal{B}$, calculated according to (4.5). Then,*

$$[\lambda_f] \subseteq [r''] \, [\Lambda_s([\nabla g])] + [r'] \, [\lambda_g].$$

Since the proof is very similar to the proof of Lemma 4.8, we omit it.

4.4. Improved eigenvalue bounds for the product of two functions

We begin by collecting recurring conditions again.

Conditions 4.13: *Assume Conditions 4.5 hold and assume there exist intervals $[g] \subset \mathbb{R}$ and $[h] \subset \mathbb{R}$ and hyperrectangles $[\nabla g] \subset \mathbb{R}^n$ and $[\nabla h] \subset \mathbb{R}^n$ such that*

$$\underline{g} \le g(x) \le \overline{g}, \quad \underline{h} \le h(x) \le \overline{h}, \quad \underline{\nabla g}_i \le (\nabla g(x))_i \le \overline{\nabla g}_i, \quad \textit{and} \quad \underline{\nabla h}_i \le (\nabla h(x))_i \le \overline{\nabla h}_i$$

for every $x \in \mathcal{B}$ and every $i \in \mathcal{N}$.

The following lemma provides rules for the identification of at most linear dependencies and independencies of products.

Lemma 4.14 (index sets for products)**:** *Assume Conditions 4.13 hold and consider the function $f : \mathcal{U} \to \mathbb{R}$, $f(x) = g(x)\, h(x)$. Let $\mathcal{I}_f = \mathcal{I}_g \cap \mathcal{I}_h$ and $\mathcal{L}_f = \mathcal{I}_g \cap \mathcal{I}_h$. Then, f depends at most linearly on x_i for all $i \in \mathcal{L}_f$ and f is independent of x_i for all $i \in \mathcal{I}_f$.*

Based on Conditions 4.13 and Lemma 4.14, we are able to compute bounds on the spectrum of $\nabla^2_{\mathcal{L}_f^c} f(x)$ according to the rules summarized in Lemma 4.15 and Table 6. As a preparation, we introduce the interval operators

$$[\Lambda_r([a],[b])] := [\min\{\underline{a},\underline{b}\}, \max\{\overline{a},\overline{b}\}] \qquad \text{and} \tag{4.20}$$

$$[\Lambda_\star([a],[b],[c])] := \frac{1}{2}\left[\underline{a}+\underline{b}-\sqrt{(\underline{a}-\underline{b})^2+d}, \overline{a}+\overline{b}+\sqrt{(\overline{a}-\overline{b})^2+d}\right] \tag{4.21}$$

for real intervals $[a],[b],[c] \subset \mathbb{R}$, where $d = 4\max\{\underline{c}^2, \overline{c}^2\}$. Definition (4.20) is introduced only for the sake of a compact notation. Whenever it is more instructive, we use the notation on the right-hand side of (4.20).

Lemma 4.15 (spectral bounds for reduced Hessian of products)**:** *Assume Conditions 4.13 hold and consider the function $f : \mathcal{U} \to \mathbb{R}$ with $f(x) = g(x)\, h(x)$. Let $\mathcal{L}_f$ be defined as in Lemma 4.14. Then the bounds* (4.10) *hold for $[\lambda_f^\dagger]$ computed according to the rules listed in Table 6.*

Proof. We prove case 10 from Table 6. Cases 4 and 7 can be shown analogously. The remaining cases can be proven in the same fashion as those treated in the proofs of Lems. 4.7 and 4.11. The reduced Hessian of f, which reads

$$\nabla^2_{\mathcal{L}_f^c} f(x) = \nabla_{\mathcal{L}_f^c} g(x) \nabla^T_{\mathcal{L}_f^c} h(x) + \nabla_{\mathcal{L}_f^c} h(x) \nabla^T_{\mathcal{L}_f^c} g(x) + h(x)\, \nabla^2_{\mathcal{L}_f^c} g(x) \tag{4.22}$$

in all cases, is a two-by-two matrix with a particularly simple block structure in case 10. To see this, first note that g and h are independent of all but one variable each (the conditions $|\mathcal{I}_g| = n-1$ and $|\mathcal{I}_h| = n-1$ imply $|\mathcal{I}_g^c| = 1$ and $|\mathcal{I}_h^c| = 1$). Moreover, $\mathcal{I}_g \cup \mathcal{I}_h = \mathcal{N}$ implies $\mathcal{I}_g^c \cap \mathcal{I}_h^c = \emptyset$, which implies that g and h depend on two different variables. Without loss of generality we assume g depends on x_1, and h depends on x_2, i.e., $\mathcal{I}_g^c = \{1\}$ and $\mathcal{I}_h^c = \{2\}$. As a further preparation note that $\mathcal{L}_f = \mathcal{I}_g \cap \mathcal{I}_h$, which holds according to Lemma 4.14, implies $\mathcal{L}_f^c = \mathcal{I}_g^c \cup \mathcal{I}_h^c$, which evaluates to $\mathcal{L}_f^c = \{1,2\}$. Since $\mathcal{L}_f^c = \{1,2\}$ and g only depends on x_1 (resp., h only depends on x_2), we have

$$\nabla_{\mathcal{L}_f^c} g(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} g(x) \\ \frac{\partial}{\partial x_2} g(x) \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} g(x) \\ 0 \end{pmatrix} \quad \text{and} \quad \nabla^2_{\mathcal{L}_f^c} g(x) = \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} g(x) & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.23}$$

respectively,

$$\nabla_{\mathcal{L}_f^c} h(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} h(x) \\ \frac{\partial}{\partial x_2} h(x) \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{\partial}{\partial x_2} h(x) \end{pmatrix} \quad \text{and} \quad \nabla^2_{\mathcal{L}_f^c} h(x) = \begin{pmatrix} 0 & 0 \\ 0 & \frac{\partial^2}{\partial x_2^2} h(x) \end{pmatrix}. \tag{4.24}$$

Substituting (4.23) and (4.24) into (4.22) yields

$$\nabla^2_{\mathcal{L}_f^c} f(x) = \begin{pmatrix} h(x)\, \frac{\partial^2}{\partial x_1^2} g(x) & \frac{\partial}{\partial x_1} g(x) \frac{\partial}{\partial x_2} h(x) \\ \frac{\partial}{\partial x_1} g(x) \frac{\partial}{\partial x_2} h(x) & g(x)\, \frac{\partial^2}{\partial x_2^2} h(x) \end{pmatrix},$$

where all entries are scalars. Now, consider the matrix set

$$\mathcal{H} = \{H \in \mathbb{R}^{2\times 2} \,|\, H_{11} \in [h]\,[\lambda_g^\dagger],\ H_{22} \in [g]\,[\lambda_h^\dagger],\ H_{12} \in [\nabla_{\mathcal{I}_g^c} g][\nabla_{\mathcal{I}_h^c} h],\ H = H^T\}$$

and observe $\{\nabla^2_{\mathcal{L}_f^c} f(x) \in \mathbb{R}^{2\times 2} \mid x \in \mathcal{B}\} \subseteq \mathcal{H}$. To see this, note that $\{\nabla^2_{\mathcal{L}_g^c} g(x) \in \mathbb{R} \mid x \in \mathcal{B}\} \subseteq [\lambda_g^\dagger]$ and $\{\nabla^2_{\mathcal{L}_h^c} h(x) \in \mathbb{R} \mid x \in \mathcal{B}\} \subseteq [\lambda_h^\dagger]$, since the eigenvalue of a matrix $M \in \mathbb{R}^{1\times 1}$ is $\lambda = M_{1,1}$. According to Lemma A.1 stated in the appendix, eigenvalue bounds for the matrix set $\mathcal{H}$ and consequently for $\nabla^2_{\mathcal{L}_f^c} f(x)$ on $\mathcal{B}$ read $[\Lambda_\star([h]\,[\lambda_g^\dagger], [g]\,[\lambda_h^\dagger], [\nabla_{\mathcal{I}_g^c} g][\nabla_{\mathcal{I}_h^c} h])]$ as claimed in Table 6. ■

Table 6: Rules for the computation of eigenvalue bounds $[\lambda_f^\dagger]$ for the reduced Hessian $\nabla^2_{\mathcal{L}_f^c} f(x)$ of a product $f(x) = g(x)\,h(x)$. The expressions $[\lambda_t]$, $[\lambda_{g,0}]$, $[\lambda_{h,0}]$, $\mathcal{L}_\cup$, and $\mathcal{L}_\cap$ are short for $[\lambda_t] = [\Lambda_t([\nabla_{\mathcal{L}_f^c} g], [\nabla_{\mathcal{L}_f^c} h])]$, $[\lambda_{g,0}] = [\min\{\underline{\lambda}_g^\dagger, 0\}, \max\{\overline{\lambda}_g^\dagger, 0\}]$, $[\lambda_{h,0}] = [\min\{\underline{\lambda}_h^\dagger, 0\}, \max\{\overline{\lambda}_h^\dagger, 0\}]$, $\mathcal{L}_\cup = \mathcal{L}_g \cup \mathcal{L}_h$ and $\mathcal{L}_\cap = \mathcal{L}_g \cap \mathcal{L}_h$. Condition $C_\star$ reads $(\mathcal{I}_g \cup \mathcal{I}_h = \mathcal{N}) \wedge (|\mathcal{I}_g| = n-1) \wedge (|\mathcal{I}_h| = n-1)$.

Case	$[\lambda_f^\dagger]$	Condition
1	$[\lambda_t]$	$\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N}$
2	$[\lambda_t] + [h]\,[\lambda_g^\dagger]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_g$
3	$[\lambda_t] + [h]\,[\lambda_{g,0}]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_g \wedge \neg C_\star$
4	$[\Lambda_\star([h]\,[\lambda_g^\dagger], [0,0], [\nabla_{\mathcal{I}_g^c} g][\nabla_{\mathcal{I}_h^c} h])]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h = \mathcal{N} \wedge C_\star$
5	$[\lambda_t] + [g]\,[\lambda_h^\dagger]$	$\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_h$
6	$[\lambda_t] + [g]\,[\lambda_{h,0}]$	$\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_h \wedge \neg C_\star$
7	$[\Lambda_\star([0,0], [g]\,[\lambda_h^\dagger], [\nabla_{\mathcal{I}_g^c} g][\nabla_{\mathcal{I}_h^c} h])]$	$\mathcal{L}_g = \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge C_\star$
8	$[\lambda_t] + [\Lambda_r([\Lambda_r([h]\,[\lambda_g^\dagger], [g]\,[\lambda_h^\dagger])], [0,0])]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge \mathcal{L}_\cup = \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_\cap$
9	$[\lambda_t] + [\Lambda_r([h]\,[\lambda_g^\dagger], [g]\,[\lambda_h^\dagger])]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge \mathcal{L}_\cup = \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_\cap \wedge \neg C_\star$
10	$[\Lambda_\star([h]\,[\lambda_g^\dagger], [g]\,[\lambda_h^\dagger], [\nabla_{\mathcal{I}_g^c} g][\nabla_{\mathcal{I}_h^c} h])]$	$\mathcal{L}_g \subset \mathcal{N} \wedge \mathcal{L}_h \subset \mathcal{N} \wedge C_\star$
11	$[\lambda_t] + [h]\,[\lambda_g^\dagger] + [g]\,[\lambda_h^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_g = \mathcal{L}_h$
12	$[\lambda_t] + [h]\,[\lambda_g^\dagger] + [g]\,[\lambda_{h,0}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_g \subset \mathcal{L}_h$
13	$[\lambda_t] + [h]\,[\lambda_{g,0}] + [g]\,[\lambda_h^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_f = \mathcal{L}_h \subset \mathcal{L}_g$
14	$[\lambda_t] + [\Lambda_r([h]\,[\lambda_g^\dagger] + [g]\,[\lambda_h^\dagger], [0,0])]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_g = \mathcal{L}_h$
15	$[\lambda_t] + [\Lambda_r([h]\,[\lambda_g^\dagger] + [g]\,[\lambda_{h,0}], [0,0])]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_g \subset \mathcal{L}_h$
16	$[\lambda_t] + [\Lambda_r([h]\,[\lambda_{g,0}] + [g]\,[\lambda_h^\dagger], [0,0])]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_f \subset \mathcal{L}_h \subset \mathcal{L}_g$
17	$[\lambda_t] + [h]\,[\lambda_{g,0}] + [g]\,[\lambda_{h,0}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_g \nsubseteq \mathcal{L}_h \wedge \mathcal{L}_h \nsubseteq \mathcal{L}_g$

Note that Lems. 4.14 and 4.15 can, in principle, also be applied if one function, say, h, is constant (i.e., $h(x) = c$). In this case, however, Lems. 4.10 and 4.11 for the choice $r(z) = c\,z$ provide better results.

Finally, Lemma 4.16 shows that for the product of two functions, the bounds from Lemma 4.15 are as tight as or tighter than those from the original method proposed in [9]. Recall that the bounds in [9] result in $[\lambda_f] = [\Lambda_t([\nabla g], [\nabla h])] + [h]\,[\lambda_g] + [g]\,[\lambda_h]$ for $f(x) = g(x)\,h(x)$ according to [9, Proposition 3.2(iv)]. We omit the proof of Lemma 4.16, since it is similar to its counterparts in section 4.2.

Lemma 4.16 (improved bounds for products)**:** *Assume Conditions 4.13 hold and let f, $[\lambda_f^\dagger]$, and $\mathcal{L}_f$ be as in Lemma 4.15. Let $[\lambda_f]$, $[\lambda_g]$, and $[\lambda_h]$ be the eigenvalue bounds for the Hessians $\nabla^2 f(x)$, $\nabla^2 g(x)$, and $\nabla^2 h(x)$ on $\mathcal{B}$, calculated according to* (4.5)*. Then,*

$$[\lambda_f] \subseteq [\Lambda_t([\nabla g], [\nabla h])] + [h]\,[\lambda_g] + [g]\,[\lambda_h].$$

4.5. Numerical computation of improved eigenvalue bounds

In this section, we combine the results from Sects. 4.1 through 4.4 in order to compute improved eigenvalue bounds using a codelist. Formally, this leads to the extended codelist in Proposition 4.17.

Proposition 4.17 (algorithm for the computation of eigenvalue bounds using sparsity): *Assume φ is twice continuously differentiable on $\mathcal{U}$ and can be written as a codelist* (3.1) *with $t \in \mathbb{N}$ operations. Let $\mathcal{B} = [x_1] \times \cdots \times [x_n] \subset \mathcal{U}$ be arbitrary. Then, for all $x \in \mathcal{B}$, we have $\varphi(x) \in [\varphi]$, $\nabla\varphi(x) \in [\nabla\varphi]$, and $[\lambda_{\min}(\nabla^2\varphi(x)), \lambda_{\max}(\nabla^2\varphi(x))] \subseteq [\lambda_\varphi]$, where $[\varphi]$, $[\nabla\varphi]$, and $[\lambda_\varphi]$ are calculated by the following algorithm:*

1. *For $k = 1, \dots, n$, set $\mathcal{I}_k = \mathcal{N} \setminus \{k\}$, $\mathcal{L}_k = \mathcal{N}$, $[y_k] = [\underline{x}_k, \overline{x}_k]$, $[\nabla y_k] = [e_k, e_k]$, and $[\lambda_k^\dagger] = [0, 0]$.*

2. *For $k = n+1, \dots, n+t$, evaluate $\mathcal{I}_k$ and $\mathcal{L}_k$ according to the third and fourth columns of Table 7, respectively. Calculate $[y_k]$ and $[\nabla y_k]$ according to the third and fourth columns of Table 2, respectively. Compute $[\lambda_k^\dagger]$ depending on $\mathcal{L}_i$, $\mathcal{L}_j$, and $\mathcal{L}_k$ according to the second column of Table 8.*

3. *Compute $[\lambda_{n+t}]$ from $[\lambda_{n+t}^\dagger]$ and $\mathcal{L}_{n+t}$ according to* (4.5) *and set $[\varphi] = [y_{n+t}]$, $[\nabla\varphi] = [\nabla y_{n+t}]$, and $[\lambda_\varphi] = [\lambda_{n+t}]$.*

Table 7: Rules for the computation of the sets $\mathcal{I}_k$ and $\mathcal{L}_k$ in the kth line of the codelist (3.1) for variables and binary operations (left) and compositions (right). Rules for y_k are repeated here for convenience.

op Φ_k	y_k	$\mathcal{I}_k$	$\mathcal{L}_k$
`var`	x_k	$\mathcal{N} \setminus \{k\}$	$\mathcal{N}$
`add`	$y_i + y_j$	$\mathcal{I}_i \cap \mathcal{I}_j$	$\mathcal{L}_i \cap \mathcal{L}_j$
`mul`	$y_i\, y_j$	$\mathcal{I}_i \cap \mathcal{I}_j$	$\mathcal{I}_i \cap \mathcal{I}_j$

op Φ_k	y_k	$\mathcal{I}_k$	$\mathcal{L}_k$
`powNat`	y_i^m	$\mathcal{I}_i$	$\mathcal{I}_i$
`oneOver`	$1/y_i$	$\mathcal{I}_i$	$\mathcal{I}_i$
`sqrt`	$\sqrt{y_i}$	$\mathcal{I}_i$	$\mathcal{I}_i$
`exp`	$\exp(y_i)$	$\mathcal{I}_i$	$\mathcal{I}_i$
`ln`	$\ln(y_i)$	$\mathcal{I}_i$	$\mathcal{I}_i$
`addC`	$y_i + c$	$\mathcal{I}_i$	$\mathcal{L}_i$
`mulByC`	$c\, y_i$	$\mathcal{I}_i$	$\mathcal{L}_i$

Proof. The claims $\varphi(x) \in [\varphi]$ and $\nabla\varphi(x) \in [\nabla\varphi]$ for all $x \in \mathcal{B}$ are covered by Theorem 3.1. It remains to prove that $[\lambda_{\min}(\nabla^2\varphi(x)), \lambda_{\max}(\nabla^2\varphi(x))] \subseteq [\lambda_\varphi]$ for all $x \in \mathcal{B}$. Since $y_k(x) = x_k$ for $k = 1, \dots, n$, the functions $y_k(x)$, $k \in \mathcal{N}$, are independent of x_j for every $j \in \mathcal{I}_k = \mathcal{N} \setminus \{k\}$ and at most linearly dependent on x_j for every $j \in \mathcal{L}_k = \mathcal{N}$. Thus, the reduced Hessian reads $\nabla^2_{\mathcal{L}_k^c} y_k(x) = \nabla^2_\emptyset y_k(x) = 0_{0,0}$ and $[\lambda_k^\dagger] = [0, 0]$ for every $k \in \mathcal{N}$. Now assume eigenvalue bounds $[\lambda_1^\dagger], \dots, [\lambda_l^\dagger]$ for the reduced Hessians $\nabla^2_{\mathcal{L}_1^c} y_1(x), \dots, \nabla^2_{\mathcal{L}_l^c} y_l(x)$ and index sets $\mathcal{I}_1, \dots, \mathcal{I}_l$ and $\mathcal{L}_1, \dots, \mathcal{L}_l$ have been calculated for some $l \in \mathbb{N}_{[n,n+t-1]}$, and let $k = l + 1$. Since $\Phi_k(y_1, \dots, y_{k-1})$ is one of the unary or binary functions listed in Table 2 (and therefore Tabs. 7 and 8), it depends on either one (say, y_i) or two (say, y_i and y_j) of the intermediate variables $y_1, \dots, y_{k-1}$. The remainder of the proof must be carried out for each type of operation Φ_k separately. We state the proof for one of the `mul` cases and claim the remaining cases can be shown accordingly. Let $g(x) = y_i(x)$, $h(x) = y_j(x)$, and $f(x) = y_k(x)$, which implies $f(x) = g(x)\, h(x)$, since the operation in the kth line is of type `mul`. In order to compute eigenvalue bounds $[\lambda_f^\dagger]$ for the reduced Hessian $\nabla^2_{\mathcal{L}_f^c} f(x)$, we first evaluate the index sets $\mathcal{I}_f$ and $\mathcal{L}_f$. According to Lemma 4.14, we obtain

$$\mathcal{I}_f = \mathcal{I}_g \cap \mathcal{I}_h = \mathcal{I}_i \cap \mathcal{I}_j \quad \text{and} \quad \mathcal{L}_f = \mathcal{I}_g \cap \mathcal{I}_h = \mathcal{I}_i \cap \mathcal{I}_j,$$

where we used $\mathcal{I}_g = \mathcal{I}_i$ and $\mathcal{I}_h = \mathcal{I}_j$, which hold by construction. Assuming we have $\mathcal{L}_g \cup \mathcal{L}_h \subset \mathcal{N}$ and $\mathcal{L}_f = \mathcal{L}_g \subset \mathcal{L}_h$, applying Lemma 4.15 (specifically, rule 12 in Table 6) results in

$$\begin{aligned}[\lambda_f^\dagger] &= [\Lambda_t([\nabla_{\mathcal{L}_f^c} g],[\nabla_{\mathcal{L}_f^c} h])] \;\; + [h]\,[\lambda_g^\dagger] \;\; + [g]\,[\min\{\underline{\lambda}_h^\dagger,0\},\max\{\underline{\lambda}_h^\dagger,0\}] \\ &= [\Lambda_t([\nabla_{\mathcal{L}_k^c} y_i],[\nabla_{\mathcal{L}_k^c} y_j])] + [y_j]\,[\lambda_i^\dagger] + [y_i]\,[\min\{\underline{\lambda}_j^\dagger,0\},\max\{\underline{\lambda}_j^\dagger,0\}],\end{aligned} \tag{4.25}$$

where the second equation results from substituting the codelist notation $[g] = [y_i]$, $[h] = [y_j]$, $[\nabla g] = [\nabla y_i]$, $[\nabla h] = [\nabla y_j]$, $[\lambda_g^\dagger] = [\lambda_i^\dagger]$, and $[\lambda_h^\dagger] = [\lambda_j^\dagger]$. Finally, since $\mathcal{L}_k = \mathcal{L}_f$ and consequently $\nabla^2_{\mathcal{L}_k^c} y_k(x) = \nabla^2_{\mathcal{L}_f^c} f(x)$, the eigenvalues of $\nabla^2_{\mathcal{L}_k^c} y_k(x)$ are confined to $[\lambda_k^\dagger] = [\lambda_f^\dagger]$ for all $x \in \mathcal{B}$. Since the second equation in (4.25) is equal to the rule in Table 8 for the case `mul` and $\mathcal{L}_i \cup \mathcal{L}_j \subset \mathcal{N}$ and $\mathcal{L}_k = \mathcal{L}_i \subset \mathcal{L}_j$, this proves the claim for the selected case. ■

Proposition 4.17 is illustrated with two examples. First, we revisit the motivating Example 1. Recall that we evaluated the conservative eigenvalue bounds $[\lambda_\varphi] = [0, 4]$ using the original method from [9].

Example 2 (improved method applied to $\varphi(x) = x_1^2 + x_2^2$ from Example 1)**:** *Consider the function φ from Example 1 again. Proposition 4.17 results in the following extended codelist. Note that we do not list the expressions for $[y_k]$ and $[\nabla y_k]$ in* (4.26) *since they are identical to the corresponding expressions in* (4.1)*. Further note that $\mathcal{I}_k$ and $\mathcal{L}_k$ are independent of $\mathcal{B}$.*

k	$\mathcal{I}_k$		$\mathcal{L}_k$		$[\lambda_k^\dagger]$
1	$\mathcal{N} \setminus \{1\}$	$= \{2\}$	$\mathcal{N}$	$= \{1,2\}$	$[0,0]$
2	$\mathcal{N} \setminus \{2\}$	$= \{1\}$	$\mathcal{N}$	$= \{1,2\}$	$[0,0]$
3	$\mathcal{I}_1$	$= \{2\}$	$\mathcal{I}_1$	$= \{2\}$	$2\,[\Lambda_s([\nabla_{\mathcal{L}_3^c} y_1])] = 2\,[\Lambda_s([\nabla_{\{1\}} y_1])]$
4	$\mathcal{I}_2$	$= \{1\}$	$\mathcal{I}_2$	$= \{1\}$	$2\,[\Lambda_s([\nabla_{\mathcal{L}_4^c} y_2])] = 2\,[\Lambda_s([\nabla_{\{2\}} y_2])]$
5	$\mathcal{I}_3 \cap \mathcal{I}_4$	$= \emptyset$	$\mathcal{L}_3 \cap \mathcal{L}_4$	$= \emptyset$	$[\min\{\underline{\lambda}_3^\dagger, \underline{\lambda}_4^\dagger\}, \max\{\overline{\lambda}_3^\dagger, \overline{\lambda}_4^\dagger\}]$
					$[\lambda_\varphi] = [\lambda_5^\dagger]$

(4.26)

The expressions for $[\lambda_k^\dagger]$ in lines 3 *and* 4 *of the extended codelist in* (4.26) *refer to the first rule associated with the* `powNat`*-operation in Table 8 since $\mathcal{L}_1 = \mathcal{N}$ and $\mathcal{L}_2 = \mathcal{N}$, respectively. Since $\mathcal{L}_3 = \{2\} \subset \mathcal{N}$, $\mathcal{L}_4 = \{1\} \subset \mathcal{N}$, and $\mathcal{L}_3 \cup \mathcal{L}_4 = \mathcal{N}$, we obtain the bounds $[\lambda_5^\dagger] = [\min\{\underline{\lambda}_3^\dagger, \underline{\lambda}_4^\dagger\}, \max\{\overline{\lambda}_3^\dagger, \overline{\lambda}_4^\dagger\}]$ according to the last rule for the* `add`*-operation in Table 8. Finally, since $\mathcal{L}_5 = \emptyset$, we have $[\lambda_\varphi] = [\lambda_5] = [\lambda_5^\dagger]$ according to* (4.5)*.*

Evaluating the extended codelist (4.26) *for the hyperrectangle $\mathcal{B} = [0,1] \times [0,1]$ (as in Example 1) by computing $[y_k]$ and $[\nabla y_k]$ according to* (4.1) *and $[\lambda_k]$ according to* (4.26) *yields $[\lambda_1^\dagger] = [\lambda_2^\dagger] = [0,0]$ and $[\lambda_3^\dagger] = [\lambda_4^\dagger] = [\lambda_5^\dagger] = [2,2]$, where we used $[\Lambda_s([\nabla_{\{1\}} y_1])] = [\Lambda_s([1,1])] = [1,1]$ and $[\Lambda_s([\nabla_{\{2\}} y_2])] = [\Lambda_s([1,1])] = [1,1]$ (see* (4.2) *for numerical results on $[y_k]$ and $[\nabla y_k]$). Thus, using the improved method, we obtain the tight eigenvalue bounds $[\lambda_\varphi] = [\lambda_5^\dagger] = [\lambda_\varphi^*] = [2,2]$.*

Table 8: Rules for the calculation of the eigenvalue bounds $[\lambda_k^\dagger]$ in the kth line of the codelist (3.1) according to Proposition 4.17. The intervals $[\lambda_t]$, $[\lambda_{i,0}]$, and $[\lambda_{j,0}]$ are shorthand notation for $[\lambda_t] = [\Lambda_t([\nabla_{\mathcal{L}_k^c} y_i], [\nabla_{\mathcal{L}_k^c} y_j])]$, $[\lambda_{i,0}] = [\min\{\underline{\lambda}_i^\dagger, 0\}, \max\{\overline{\lambda}_i^\dagger, 0\}]$, and $[\lambda_{j,0}] = [\min\{\underline{\lambda}_j^\dagger, 0\}, \max\{\overline{\lambda}_j^\dagger, 0\}]$. The index sets $\mathcal{L}_\cup$ and $\mathcal{L}_\cap$ are shorthand notation for $\mathcal{L}_\cup = \mathcal{L}_i \cup \mathcal{L}_j$ and $\mathcal{L}_\cap = \mathcal{L}_i \cap \mathcal{L}_j$. Condition $C_\star$ reads $(\mathcal{I}_i \cup \mathcal{I}_j = \mathcal{N}) \wedge (|\mathcal{I}_i| = n-1) \wedge (|\mathcal{I}_j| = n-1)$.

`op` Φ_k	$[\lambda_k^\dagger]$	Condition
`add`	$[0,0]$	$\mathcal{L}_i = \mathcal{N} \wedge \mathcal{L}_j = \mathcal{N}$
	$[\lambda_i^\dagger]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j = \mathcal{N}$
	$[\lambda_j^\dagger]$	$\mathcal{L}_i = \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N}$
	$[\Lambda_r([\lambda_i^\dagger], [\lambda_j^\dagger])]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge \mathcal{L}_\cup = \mathcal{N}$
	$[\lambda_i^\dagger] + [\lambda_j^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_i = \mathcal{L}_j$
	$[\lambda_i^\dagger] + [\lambda_{j,0}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_i \subset \mathcal{L}_j$
	$[\lambda_{i,0}] + [\lambda_j^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{L}_i$
	$[\lambda_{i,0}] + [\lambda_{j,0}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_i \not\subseteq \mathcal{L}_j \wedge \mathcal{L}_j \not\subseteq \mathcal{L}_i$
`mul`	$[\lambda_t]$	$\mathcal{L}_i = \mathcal{N} \wedge \mathcal{L}_j = \mathcal{N}$
	$[\lambda_t] + [y_j]\,[\lambda_i^\dagger]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j = \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i$
	$[\lambda_t] + [y_j]\,[\lambda_{i,0}]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j = \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i \wedge \neg C_\star$
	$[\Lambda_\star([y_j]\,[\lambda_i^\dagger], [0,0], [\nabla_{\mathcal{I}_i^c} y_i][\nabla_{\mathcal{I}_j^c} y_j])]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j = \mathcal{N} \wedge C_\star$
	$[\lambda_t] + [y_i]\,[\lambda_j^\dagger]$	$\mathcal{L}_i = \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_j$
	$[\lambda_t] + [y_i]\,[\lambda_{j,0}]$	$\mathcal{L}_i = \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_j \wedge \neg C_\star$
	$[\Lambda_\star([0,0], [y_i]\,[\lambda_j^\dagger], [\nabla_{\mathcal{I}_i^c} y_i][\nabla_{\mathcal{I}_j^c} y_j])]$	$\mathcal{L}_i = \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge C_\star$
	$[\lambda_t] + [\Lambda_r([\Lambda_r([y_j]\,[\lambda_i^\dagger], [y_i]\,[\lambda_j^\dagger])], [0,0])]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge \mathcal{L}_\cup = \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_\cap$
	$[\lambda_t] + [\Lambda_r([y_j]\,[\lambda_i^\dagger], [y_i]\,[\lambda_j^\dagger])]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge \mathcal{L}_\cup = \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_\cap \wedge \neg C_\star$
	$[\Lambda_\star([y_j]\,[\lambda_i^\dagger], [y_i]\,[\lambda_j^\dagger], [\nabla_{\mathcal{I}_i^c} y_i][\nabla_{\mathcal{I}_j^c} y_j])]$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_j \subset \mathcal{N} \wedge C_\star$
	$[\lambda_t] + [y_j]\,[\lambda_i^\dagger] + [y_i]\,[\lambda_j^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i = \mathcal{L}_j$
	$[\lambda_t] + [y_j]\,[\lambda_i^\dagger] + [y_i]\,[\lambda_{j,0}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i \subset \mathcal{L}_j$
	$[\lambda_t] + [y_j]\,[\lambda_{i,0}] + [y_i]\,[\lambda_j^\dagger]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_j \subset \mathcal{L}_i$
	$[\lambda_t] + [\Lambda_r([y_j]\,[\lambda_i^\dagger] + [y_i]\,[\lambda_j^\dagger], [0,0])]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i = \mathcal{L}_j$
	$[\lambda_t] + [\Lambda_r([y_j]\,[\lambda_i^\dagger] + [y_i]\,[\lambda_{j,0}], [0,0])]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i \subset \mathcal{L}_j$
	$[\lambda_t] + [\Lambda_r([y_j]\,[\lambda_{i,0}] + [y_i]\,[\lambda_j^\dagger], [0,0])]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_j \subset \mathcal{L}_i$
	$[\lambda_t] + [y_j]\,[\lambda_{i,0}] + [y_i]\,[\lambda_{j,0}]$	$\mathcal{L}_\cup \subset \mathcal{N} \wedge \mathcal{L}_i \not\subseteq \mathcal{L}_j \wedge \mathcal{L}_j \not\subseteq \mathcal{L}_i$
`powNat`	$m\,(m-1)\,[y_i]^{m-2}\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])]$	$\mathcal{L}_i = \mathcal{N}$
	$m\,[y_i]^{m-2}((m-1)[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] + [y_i]\,[\lambda_i^\dagger])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i$
	$m\,[y_i]^{m-2}((m-1)[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] + [y_i]\,[\lambda_{i,0}])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i$
`oneOver`	$2\,[y_k]^3\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])]$	$\mathcal{L}_i = \mathcal{N}$
	$[y_k]^2\,(2\,[y_k]\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] - [\lambda_i^\dagger])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i$
	$[y_k]^2\,(2\,[y_k]\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] - [\lambda_{i,0}])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i$
`sqrt`	$1/(-4\,[y_k]^3)\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])]$	$\mathcal{L}_i = \mathcal{N}$
	$1/(2\,[y_k])(1/(-2\,[y_i])[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] + [\lambda_i^\dagger])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i$
	$1/(2\,[y_k])(1/(-2\,[y_i])[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] + [\lambda_{i,0}])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i$
`exp`	$[y_k]\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])]$	$\mathcal{L}_i = \mathcal{N}$
	$[y_k]\,([\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] + [\lambda_i^\dagger])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i$
	$[y_k]\,([\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])] + [\lambda_{i,0}])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i$
`ln`	$-1/[y_i]^2\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])]$	$\mathcal{L}_i = \mathcal{N}$
	$1/[y_i]\,([\lambda_i^\dagger] - 1/[y_i]\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k = \mathcal{L}_i$
	$1/[y_i]\,([\lambda_{i,0}] - 1/[y_i]\,[\Lambda_s([\nabla_{\mathcal{L}_k^c} y_i])])$	$\mathcal{L}_i \subset \mathcal{N} \wedge \mathcal{L}_k \subset \mathcal{L}_i$
`addC`	$[0,0]$	$\mathcal{L}_i = \mathcal{N}$
	$[\lambda_i^\dagger]$	$\mathcal{L}_i \subset \mathcal{N}$
`mulByC`	$[0,0]$	$\mathcal{L}_i = \mathcal{N}$
	$c\,[\lambda_i^\dagger]$	$\mathcal{L}_i \subset \mathcal{N}$

We analyze another example to demonstrate that the new method results in considerable improvements for all functions that involve multiplications. In fact, we know from [9, Rem. 4.3] that $0 \in [\lambda_\varphi]$ for the original method if the `mul`-operation is required in the codelist of any φ with $n \geq 2$. This is a severe drawback of the original method, since it implies that any convex (resp., concave) function $\varphi : \mathbb{R}^n \to \mathbb{R}$ involving `mul`-operations will never be identified to be convex (resp., concave) using the method from [9]. The following example shows that this restriction does not apply for the improved method.

Example 3 (comparison of [9] and improved method for $\varphi(x) = x_1^2 + x_2 \exp(x_2)$)**:** *Consider the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ with $\varphi(x) = x_1^2 + x_2 \exp(x_2)$ on a $\mathcal{B} \subset \mathbb{R}^2$. Theorem 3.1 (i.e., the original method from [9]) results in the following extended codelist, where the expressions for y_k are listed only for illustration of the codelist* (3.1) *of φ. We skip the first three lines, since they are identical to those in* (4.1)*.*

k	y_k	$[y_k]$	$[\nabla y_k]$	$[\lambda_k]$
4	$\exp(y_2)$	$[\exp([y_2])]$	$[y_4]\,[\nabla y_2]$	$[y_4]\,([\Lambda_s([\nabla y_2])] + [\lambda_2])$
5	$y_2\,y_4$	$[y_2]\,[y_4]$	$[y_4][\nabla y_2]+[y_2][\nabla y_4]$	$[y_4][\lambda_2]+[y_2][\lambda_4]+[\Lambda_t([\nabla y_2], [\nabla y_4])]$
6	$y_3 + y_5$	$[y_3] + [y_5]$	$[\nabla y_3] + [\nabla y_5]$	$[\lambda_3] + [\lambda_5]$
	$\varphi = y_6$	$[\varphi] = [y_6]$	$[\nabla\varphi] = [\nabla y_6]$	$[\lambda_\varphi] = [\lambda_6]$

(4.27)

Evaluating this codelist for $\mathcal{B} = [0,1] \times [0,1]$ yields

$$[\lambda_\varphi] = [\lambda_6] = [-\exp(1) + 1, 3\exp(1) + 2] \approx [-1.7183, 10.1548].$$

Proposition 4.17 (i.e., the improved method) results in the following extended codelist. The first three lines are identical to those in (4.26) *in this case. Further note that the expressions for $[y_k]$ and $[\nabla y_k]$ can be found in* (4.1) *(lines 1-3) and* (4.27) *(lines 4-6).*

k	$\mathcal{I}_k$		$\mathcal{L}_k$		$[\lambda_k^\dagger]$
4	$\mathcal{I}_2$	$= \{1\}$	$\mathcal{I}_2$	$= \{1\}$	$[y_4]\,[\Lambda_s([\nabla_{\mathcal{L}_4^c} y_2])] = [y_4]\,[\Lambda_s([\nabla_{\{2\}} y_2])]$
5	$\mathcal{I}_2 \cap \mathcal{I}_4$	$= \{1\}$	$\mathcal{I}_2 \cap \mathcal{I}_4$	$= \{1\}$	$[\Lambda_t([\nabla_{\mathcal{L}_5^c} y_2], [\nabla_{\mathcal{L}_5^c} y_4])]+[y_4]\,[\lambda_2^\dagger]+[y_2]\,[\lambda_4^\dagger]$ $= [\Lambda_t([\nabla_{\{2\}} y_2], [\nabla_{\{2\}} y_4])]+[y_4]\,[\lambda_2^\dagger]+[y_2]\,[\lambda_4^\dagger]$
6	$\mathcal{I}_3 \cap \mathcal{I}_5$	$= \emptyset$	$\mathcal{L}_3 \cap \mathcal{L}_5$	$= \emptyset$	$[\min\{\underline{\lambda}_3^\dagger, \underline{\lambda}_5^\dagger\}, \max\{\overline{\lambda}_3^\dagger, \overline{\lambda}_5^\dagger\}]$
					$[\lambda_\varphi] = [\lambda_6^\dagger]$

(4.28)

Evaluating (4.28) *for $\mathcal{B} = [0,1] \times [0,1]$ yields*

$$[\lambda_\varphi] = [\lambda_6^\dagger] = [2, 3\exp(1)] \approx [2, 8.1548],$$

Just as in Example 3, the improved method results in tight spectral bounds while the original method from [9] provides loose outer approximations. In particular, $0 \in [-1.7183, 10.1548]$ for the original method as predicted by [9, Rem. 4.3] but $0 \notin [2, 8.1548]$ for the improved method presented here. Convexity of φ on $\mathcal{B}$ can therefore be established with the improved but not with the original method.

More generally, the improved method results in eigenvalue bounds that are always as tight as, or tighter than, the original method from [9], as stated in the following proposition.

Proposition 4.18 (accuracy of the improved method)**:** *Assume φ is twice continuously differentiable on $\mathcal{U}$ and can be written as a codelist* (3.1)*. Let $\mathcal{B} = [x_1] \times \cdots \times [x_n] \subset \mathcal{U}$ be arbitrary* *and denote the eigenvalue bounds for $\nabla^2\varphi(x)$ on $\mathcal{B}$ computed according to Theorem 3.1 and Proposition 4.17 by $[\lambda_\varphi^{(\text{Theorem 3.1})}]$ and $[\lambda_\varphi^{(\text{Proposition 4.17})}]$, respectively. Then,*

$$[\lambda_\varphi^{(\text{Proposition 4.17})}] \subseteq [\lambda_\varphi^{(\text{Theorem 3.1})}].$$

Proof. The proof immediately follows from Lems. 4.8, 4.11, and 4.15. ■

In [9, Proposition 4.4] it was shown that the numerical complexity for evaluating the extended codelist resulting from Theorem 3.1 is of order $\mathcal{O}(n)\,N(\varphi)$, where $N(\varphi)$ denotes the number of operations needed to evaluate φ at a point in its domain. It is remarkable that this order of complexity can be maintained for the improved method. This is summarized in the following proposition.

Proposition 4.19 (numerical complexity of the improved method)**:** *Assume φ is twice continuously differentiable on $\mathcal{U}$ and can be written as a codelist* (3.1) *with $t = N(\varphi)$ operations. Let $N([\varphi], [\nabla\varphi], [\lambda_\varphi])$ denote the number of operations that are necessary to calculate the bounds $[\varphi] \subset \mathbb{R}$, $[\nabla\varphi] \subset \mathbb{R}^n$, and $[\lambda_\varphi] \subset \mathbb{R}$ for a given hyperrectangle $\mathcal{B} \subset \mathcal{U}$ using the extended codelist from Proposition 4.17. Then,*

$$N([\varphi], [\nabla\varphi], [\lambda_\varphi]) = \mathcal{O}(n)\,N(\varphi).$$

Since the proof of Proposition 4.19 is very similar to that of [9, Proposition 4.4], we only sketch it. The extended codelist that results from Proposition 4.17 involves the index sets $\mathcal{I}_k$ and $\mathcal{L}_k$, which were not required in the original method. These index sets do not depend on the particular hyperrectangle $\mathcal{B}$ as illustrated in Example 2, but they are uniquely determined by the function φ itself. Consequently, all index sets need to be determined only once. This step can be carried out at the time of construction of the extended codelist. In particular, it need not be repeated at the time of evaluating the codelist for a particular $\mathcal{B}$. Once $\mathcal{I}_k$ and $\mathcal{L}_k$ have been determined, each line of the extended codelist is specified by the rules in Table 8 (and Table 2). It is easy to show that the evaluation of every expression in the second column of Table 8 requires at most $\mathcal{O}(n)$ basic operations (like additions, multiplications, or comparisons of two real numbers; see [9, section 4.1] for further details). Thus, under the assumption that $[\varphi]$ and $[\nabla\varphi]$ are known, we need $\mathcal{O}(n)\,N(\varphi)$ basic operations for the computation of $[\lambda_\varphi]$. Since the calculation of $[\varphi]$ and $[\nabla\varphi]$ requires $\mathcal{O}(1)$ and $\mathcal{O}(n)$ basic operations according to standard results in AD and IA (see, e.g., [4,7]), we obtain

$$N([\varphi], [\nabla\varphi], [\lambda_\varphi]) = \mathcal{O}(1)\,N(\varphi) + \mathcal{O}(n)\,N(\varphi) + \mathcal{O}(n)\,N(\varphi) = \mathcal{O}(n)\,N(\varphi).$$

5. Numerical experiments for a large number of examples

In this section, we analyze 1522 numerical examples taken from the COCONUT collection of optimization problems [14]. We consider all COCONUT problems with $1 < n \leq 10$ variables and extract those cost and constraint functions that can be decomposed into the operations listed in Tabs. 2 and 8. For each function $\varphi : \mathbb{R}^n \to \mathbb{R}$, we consider 100 random hyperrectangles $\mathcal{B} \subset \mathcal{D}$ in the domain $\mathcal{D}$ of φ specified in the respective COCONUT probLemma For ease of comparison, the set of examples as well as the associated hyperrectangles are identical to the examples considered in [13].

For each of the resulting $1522 \cdot 100$ sample problems, we solve problem (1.1) using the improved algorithm ($\text{A}^\dagger$ for short) in Proposition 4.17. We compare the resulting eigenvalue bounds with those obtained from two established methods using interval Hessians (see problem (1.3)) and either Gershgorin's circle criterion (G for short) or Hertz and Rohn's method (H for short) for the computation of spectral bounds of interval matrices (see problem (1.4)). We choose G and H as reference procedures due to the favorable computational complexity of G and since H provides tight eigenvalue bounds for problem (1.4) (cf. section 1). We refer to the original papers [2,5,6,12] or the summaries in [9,13] for a detailed description of methods G and H.

Table 9: Classes used to aggregate results in Table 10. Symbols $[\lambda_{\mathrm{A}^\dagger}] = [\underline{\lambda}_{\mathrm{A}^\dagger}, \overline{\lambda}_{\mathrm{A}^\dagger}]$, $[\lambda_{\mathrm{G}}] = [\underline{\lambda}_{\mathrm{G}}, \overline{\lambda}_{\mathrm{G}}]$, and $[\lambda_{\mathrm{H}}] = [\underline{\lambda}_{\mathrm{H}}, \overline{\lambda}_{\mathrm{H}}]$ denote eigenvalue bounds calculated by the improved algorithm $\mathrm{A}^\dagger$ (see Proposition 4.17), Gershgorin's circle criterion, and Hertz and Rohn's method, respectively.

Class	Verbal definition	Formal definition	
		Lower bound $\underline{\lambda}$	Upper bound $\overline{\lambda}$
1	$\mathrm{A}^\dagger$ worse than G (and H)	$\underline{\lambda}_{\mathrm{A}^\dagger} < \underline{\lambda}_{\mathrm{G}} \leq \underline{\lambda}_{\mathrm{H}}$	$\overline{\lambda}_{\mathrm{H}} \leq \overline{\lambda}_{\mathrm{G}} < \overline{\lambda}_{\mathrm{A}^\dagger}$
2	$\mathrm{A}^\dagger$ equal to G but worse than H	$\underline{\lambda}_{\mathrm{G}} \approx \underline{\lambda}_{\mathrm{A}^\dagger} < \underline{\lambda}_{\mathrm{H}}$	$\overline{\lambda}_{\mathrm{H}} < \overline{\lambda}_{\mathrm{A}^\dagger} \approx \overline{\lambda}_{\mathrm{G}}$
3	$\mathrm{A}^\dagger$ better than G but worse than H	$\underline{\lambda}_{\mathrm{G}} < \underline{\lambda}_{\mathrm{A}^\dagger} < \underline{\lambda}_{\mathrm{H}}$	$\overline{\lambda}_{\mathrm{H}} < \overline{\lambda}_{\mathrm{A}^\dagger} < \overline{\lambda}_{\mathrm{G}}$
4	$\mathrm{A}^\dagger$ equal to H (and equal to or better than G)	$\underline{\lambda}_{\mathrm{G}} \leq \underline{\lambda}_{\mathrm{H}} \approx \underline{\lambda}_{\mathrm{A}^\dagger}$	$\overline{\lambda}_{\mathrm{A}^\dagger} \approx \overline{\lambda}_{\mathrm{H}} \leq \overline{\lambda}_{\mathrm{G}}$
5	$\mathrm{A}^\dagger$ better than H (and G)	$\underline{\lambda}_{\mathrm{G}} \leq \underline{\lambda}_{\mathrm{H}} < \underline{\lambda}_{\mathrm{A}^\dagger}$	$\overline{\lambda}_{\mathrm{A}^\dagger} < \overline{\lambda}_{\mathrm{H}} \leq \overline{\lambda}_{\mathrm{G}}$

For each sample problem, we analyze whether $\mathrm{A}^\dagger$ performs better than, equally good as, or worse than G and H. We independently compare the lower and upper eigenvalue bounds of the particular methods and categorize the results according to the five classes in Table 9. Note that G never performs better than H (since H provides tight bounds for (1.4)). Consequently, the relations $\underline{\lambda}_{\mathrm{G}} \leq \underline{\lambda}_{\mathrm{H}}$ and $\overline{\lambda}_{\mathrm{H}} \leq \overline{\lambda}_{\mathrm{G}}$ always hold. Hence, the list of classes in Table 9 is complete in the sense that every example can be uniquely classified into one of the five classes. It remains to comment on the precise meaning of $a > b$ and $a \approx b$ as used for the classification in Table 9. To this end, we introduce the function

$$\mathrm{dev}(a, b) = \frac{a - b}{1 + 0.5\,|a + b|},$$

which evaluates a weighted difference of $a, b \in \mathbb{R}$. Based on $\mathrm{dev}(a, b)$, we specify

$$a > b \qquad \text{is understood as} \qquad \mathrm{dev}(a, b) > \epsilon \qquad \text{and} \tag{5.1}$$

$$a \approx b \qquad \text{is understood as} \qquad |\mathrm{dev}(a, b)| \leq \epsilon, \tag{5.2}$$

where $\epsilon \in \mathbb{R}_+$ represents an error bound. Note that $|\mathrm{dev}(a, b)|$ is approximately equal to the relative difference (i.e., $2\,|a - b|\,|a + b|^{-1}$) for two large but almost equal numbers $a, b \in \mathbb{R}$ and almost equal to the absolute difference (i.e., $|a - b|$) for two small but almost equal numbers $a, b \in \mathbb{R}$. This behavior is useful since the absolute values of the computed eigenvalue bounds range across several magnitudes.

We summarize numerical results for the analyzed examples in Table 10 (with $\epsilon = 10^{-6}$). We list the percentage of samples that fall into classes 1 to 5 from Table 9 separated by dimension n of the underlying example. In order to compare the improved algorithm in Proposition 4.17 to the original method from [9] (see Theorem 3.1), we also list the classification results using the original algorithm (A for short). The numerical results confirm that the consideration of sparsity significantly improves the tightness of the computed eigenvalue bounds. To see this, note that for each dimension n, the percentages in class 1 (where the established approaches outperform the direct computation of eigenvalue bounds) decrease, while the percentages in classes 4 and 5 (where the direct computation of eigenvalue bounds performs as good as or better than Hertz and Rohn's method) increase using the improved algorithm $\mathrm{A}^\dagger$ instead of the original A. In particular, it is remarkable that the improved algorithm $\mathrm{A}^\dagger$ results in worse eigenvalue bounds than G in only 9.09% of all cases in contrast to 17.77% for the original method A. Moreover, A† provides equally good or better eigenvalue bounds than H in 82.20% = 77.90% + 4.30% of all cases, while the corresponding percentage only reads 20.83% = 16.86% + 3.95% for A.

Table 10: Numerical results for $1522 \cdot 100$ sample problems. For each problem, we computed eigenvalue bounds using the original algorithm A from [9] and the improved variant A† taking sparsity into account. The obtained spectral bounds were classified according to Table 9 and (5.1)–(5.2) with $\epsilon = 10^{-6}$. The table shows, separately for methods A and A†, percentages of problems that fall into the various classes.

Examples		1		2		3		4		5	
n	#	A	A†	A	A†	A	A†	A	A†	A	A†
2	62	56.37	29.06	1.07	1.08	12.24	16.06	18.89	41.27	11.43	12.54
3	1078	2.98	0.95	77.53	0.31	0.85	1.01	17.70	96.72	0.95	1.01
4	67	60.75	35.28	4.84	4.63	8.45	16.76	15.04	32.14	10.92	11.19
5	88	56.80	35.34	3.32	0.21	10.16	14.90	14.91	34.47	14.81	15.09
6	95	35.05	25.65	5.31	4.13	31.91	34.25	13.87	21.38	13.86	14.58
7	27	65.80	27.07	11.93	8.50	0.02	19.44	22.26	44.98	0.00	0.00
8	15	94.23	63.83	3.20	4.23	1.27	29.80	1.30	1.53	0.00	0.60
9	24	57.27	25.02	8.29	0.02	18.71	34.94	4.25	22.58	11.48	17.44
10	66	51.17	12.24	1.00	0.22	22.12	34.11	15.86	41.54	9.86	11.89
All	1522	17.77	9.09	56.11	0.94	5.32	7.78	16.86	77.90	3.95	4.30

Another observation is that the ratios in the particular classes seem to be independent of the dimension n (i.e., there is no trend). This is important since the numerical complexities of the established approaches G and H vary between $\mathcal{O}(n)\,N(\varphi) + \mathcal{O}(n^2)$ and $\mathcal{O}(n^2)\,N(\varphi) + \mathcal{O}(2^n\,n^3)$ operations (see section 1 and the benchmark in [13]), while the direct eigenvalue bound computation requires $\mathcal{O}(n)\,N(\varphi)$. Thus, methods A and A† become numerically very attractive for high dimensions n.

Table 11: Last line of Table 10 for different choices of the error bound ϵ in (5.1) and (5.2).

	1		2		3		4		5	
ϵ	A	A†	A	A†	A	A†	A	A†	A	A†
10^{-5}	17.16	8.78	54.11	0.91	4.29	6.57	20.78	79.74	3.66	4.01
10^{-6}	17.77	9.09	56.11	0.94	5.32	7.78	16.86	77.90	3.95	4.30
10^{-7}	18.04	9.28	57.62	0.66	5.61	8.24	14.67	77.41	4.06	4.41

According to (5.1) and (5.2), the classification in Table 10 depends on the choice of the error bound ϵ. We repeated all calculations for various choices of ϵ and present the results reported in the last line of Table 10 for $\epsilon = 10^{-5}$ and $\epsilon = 10^{-7}$ in Table 11. As expected, the ratios in classes 1 and 5 increase for decreasing ϵ, since we detect $\underline{\lambda}_{\mathrm{A}^\dagger} < \underline{\lambda}_{\mathrm{G}}$ (as well as $\overline{\lambda}_{\mathrm{G}} < \overline{\lambda}_{\mathrm{A}^\dagger}$, $\underline{\lambda}_{\mathrm{H}} < \underline{\lambda}_{\mathrm{A}^\dagger}$, and $\overline{\lambda}_{\mathrm{A}^\dagger} < \overline{\lambda}_{\mathrm{H}}$) for a larger number of examples (cf. (5.1) and (5.2)). However, beside this observation, the results are robust w.r.t. the value of ϵ.

6. Conclusion

We significantly improved a method recently introduced in [9] for the efficient computation of spectral bounds for Hessian matrices of twice continuously differentiable functions on hyperrectangles. The improvements build on the identification and utilization of sparsity that naturally arises in the first lines of every codelist for a function $\varphi : \mathbb{R}^n \to \mathbb{R}$.

The improved method was applied to a set of 1522 examples previously analyzed in [13]. The numerical results show that the consideration of sparsity results in significantly tighter eigenvalue bounds. In fact, the improved method provided equally good or better eigenvalue bounds than Hertz and Rohn's method in 82.20% of the examples, while the corresponding percentage reads only 20.83% for the original procedure.

In addition to illustrating the practical usefulness of the proposed improvements, we provided an important theoretic result. In fact, it is well-known that the original method from [9] results in spectral bounds with $0 \in [\lambda_\varphi]$ for any function that involves the multiplications of two or more variables (see [9, Rem. 4.3]). Consequently, convex functions that involve such a multiplication cannot be detected to be convex with the original method. We showed that this restriction does not apply for the improved method.

Acknowledgments

Funding by Deutsche Forschungsgemeinschaft grant MO-1086/9 is gratefully acknowledged.

References

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, A global optimization method, αBB, for general twice-differentiabe constrained NLPs – II. Implementation and computational results. *Computers Chemical Engineering*, 22(9):1159–1179, 1998.

[2] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Computers Chemical Engineering*, 22(9):1137–1158, 1998.

[3] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, αBB: A global optimization method for general constrained nonconvex problems. *J. Global Optim.*, 7(4):337–363, 1995.

[4] H. Fischer, Automatisches Differenzieren, in *Wissenschaftliches Rechnen: Eine Einführung in das Scientific Computing*, J. Herzberger, ed., Akademie-Verlag, Berlin, pp. 53–103, 1995.

[5] S. Gerschgorin, Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk SSSR, Ser. Fizmat.*, 6:749–754, 1931.

[6] D. Hertz, The extreme eigenvalues and stability of real symmetric interval matrices. *IEEE Trans. Automat. Control*, 37(4):532–535, 1992.

[7] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht, The Netherlands, 1996.

[8] M. Mönnigmann, Positive invariance tests with efficient Hessian matrix eigenvalue bounds, in *Proceedings of the 17th IFAC World Congress*, 2008.

[9] M. Mönnigmann, Fast calculation of spectral bounds for Hessian matrices on hyperrectangles. *SIAM J. Matrix Anal. Appl.*, 32(4):1351–1366, 2011.

[10] A. Neumaier, *Interval Methods for Systems of Equations.* Encyclopedia Math. Appl., Cambridge University Press, Cambridge, UK, 2008.

[11] L. B. Rall, *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Comput. Sci. 120*, Springer, Berlin, 1981.

[12] J. Rohn, Positive definiteness and stability of interval matrices. *SIAM J. Matrix Anal. Appl.*, 15(1):175–184, 1994.

[13] M. Schulze Darup, M. Kastsian, S. Mross, and M. Mönnigmann, Efficient computation of spectral bounds for Hessian matrices on hyperrectangles for global optimization. *J. Glob. Optim.*, 58:631–652, 2014.

[14] O. Shcherbina, A. Neumaier, D. Sam-Haroud, X.-H. Vu, and T.-V. Nguyen, Benchmarking global optimization and constraint satisfaction codes. in *Global Optimization and Constraint Satisfaction*, C. Bliek, C. Jermann, and A. Neumaier, eds. , Springer-Verlag, Berlin, pp. 211–222, 2003.

A. Supplementary results

Lemma A.1: *Let* $[a],[b],[c] \subset \mathbb{R}$ *be real intervals, let* $[\Lambda_\star([a],[b],[c])]$ *be defined as in* (4.21)*, and consider the matrix set*

$$\mathcal{H} = \{H \in \mathbb{R}^{2\times 2} \,|\, H_{11} \in [a],\, H_{22} \in [b],\, H_{12} \in [c],\, H = H^T\}.$$

Then,

$$\underline{\Lambda}_\star([a],[b],[c]) = \min_{H\in\mathcal{H}} \lambda_{\min}(H) \quad \text{and} \quad \max_{H\in\mathcal{H}} \lambda_{\max}(H) = \overline{\Lambda}_\star([a],[b],[c]). \tag{A.1}$$

Proof. The eigenvalue bounds of a symmetric matrix $H = \left(\begin{smallmatrix} a & c \\ c & b \end{smallmatrix}\right) \in \mathcal{H}$ read

$$\lambda_{\min}(H) = \frac{1}{2}\left(a+b-\sqrt{(a-b)^2+4\,c^2}\right), \quad \lambda_{\max}(H) = \frac{1}{2}\left(a+b+\sqrt{(a-b)^2+4\,c^2}\right).$$

We therefore have to show that

$$\underline{a}+\underline{b}-\sqrt{(\underline{a}-\underline{b})^2+d} = \min_{a\in[a],b\in[b],c\in[c]} a+b-\sqrt{(a-b)^2+4\,c^2} \quad \text{and} \tag{A.2}$$

$$\overline{a}+\overline{b}+\sqrt{(\overline{a}-\overline{b})^2+d} = \max_{a\in[a],b\in[b],c\in[c]} a+b+\sqrt{(a-b)^2+4\,c^2}, \tag{A.3}$$

where the left-hand side results from (4.21) and where $d = 4\,\max\{\underline{c}^2, \overline{c}^2\}$. We show that (A.2) holds and claim (A.3) can be proven analogously. First note that the right-hand side in (A.2) can be simplified to

$$\min_{a\in[a],b\in[b],c\in[c]} a+b-\sqrt{(a-b)^2+4\,c^2} = \min_{a\in[a],b\in[b]} a+b-\sqrt{(a-b)^2+d} \tag{A.4}$$

since c occurs only in the radicand. Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$, $f(a,b) = a+b-\sqrt{(a-b)^2+d}$, which occurs on the right-hand side of (A.4), and note that $f(a,b)$ is concave (since $g(a,b) = \sqrt{(a-b)^2+d}$ is convex). Since the hyperrectangle $\mathcal{B} = [a]\times[b]$ is convex, the minimum on the right-hand side of (A.4) is attained at one of the vertices of $\mathcal{B}$. Among the candidate tuples $(\underline{a},\underline{b})$, $(\underline{a},\overline{b})$, $(\overline{a},\underline{b})$, and $(\overline{a},\overline{b})$, it is easy to show that $(\underline{a},\underline{b})$ results in the smallest function value, i.e., $f(\underline{a},\underline{b}) \le \min\{f(\underline{a},\overline{b}), f(\overline{a},\underline{b}), f(\overline{a},\overline{b})\}$. Thus, (A.2) holds. ■