Math. Prog. Comp. (2010) 2:21–57
DOI 10.1007/s12532-010-0011-7

FULL LENGTH PAPER

On solving trust-region and other regularised subproblems in optimization

Nicholas I. M. Gould · Daniel P. Robinson · H. Sue Thorne

Received: 11 February 2009 / Accepted: 19 January 2010 / Published online: 10 February 2010

Abstract The solution of trust-region and regularisation subproblems that arise in unconstrained optimization is considered. Building on the pioneering work of Gay, Moré and Sorensen, methods that obtain the solution of a sequence of parametrized linear systems by factorization are used. Enhancements using high-order polynomial approximation and inverse iteration ensure that the resulting method is both globally and asymptotically at least superlinearly convergent in all cases, including the notorious hard case. Numerical experiments validate the effectiveness of our approach. The resulting software is available as packages TRS and RQS as part of the GALAHAD optimization library, and is especially designed for large-scale problems.

Keywords Trust-region subproblem · Regularisation · Software

Mathematics Subject Classification (2000) 65F22 · 65H05 · 65K05 · 90C20 · 90C26 · 90C30

N. I. M. Gould (✉) · H. S. Thorne
Computational Science and Engineering Department, Rutherford Appleton Laboratory,
Chilton, Oxfordshire OX11 0QX, UK
e-mail: nick.gould@stfc.ac.uk

H. S. Thorne
e-mail: sue.thorne@stfc.ac.uk

D. P. Robinson
Numerical Analysis Group,
Mathematical Institute, Oxford University,
24–29 St Giles', Oxford OX1 3LB, UK
e-mail: robinson@maths.ox.ac.uk

1 Introduction

Given a symmetric matrix $H \in \mathbb{R}^{n\times n}$, a symmetric positive-definite matrix $M \in \mathbb{R}^{n\times n}$, a vector $c \in \mathbb{R}^n$ and positive scalars Δ, σ and $p > 2$, we are interested in computing solutions of the optimization problems

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ q(x) \overset{\text{def}}{=} c^T x + \frac{1}{2}x^T Hx \ \text{ subject to } \ \|x\|_M \le \Delta \tag{1.1}$$

and

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ r(x) \overset{\text{def}}{=} c^T x + \frac{1}{2}x^T Hx + \frac{\sigma}{p}\|x\|_M^p, \tag{1.2}$$

where the M-norm of x is $\|x\|_M \overset{\text{def}}{=} \sqrt{x^T Mx}$. Both problems arise as subproblems in unconstrained optimization; problem (1.1) occurs when computing the step in trust-region methods [11,37], while (1.2) plays the same role in more recent regularisation approaches [7,29,39,49]; for the latter $p = 3$ is by far the most common choice, although $p < 3$ has been mentioned for applications involving Hölder- but not Lipschitz-continuous derivatives [29]. In addition, (1.1) occurs as an important subproblem in combinatorial optimization, e.g., [6,41] as well as in other application areas, e.g., [5,36].

Although it is now common to try to find approximate solutions to (1.1) and (1.2) using iterative methods [7,18,19,25,31,45,46], there are still many problems for which a factorization of $H + \lambda M$ for given λ is both feasible and efficient. Our intention here is to revisit the possibility of solving our problems using factorization, and in particular to reassess the pioneering ideas of Gay-Moré-Sorensen [21,38] in the light of modern sparse factorization.

In Sect. 2 we discuss optimality conditions for the trust-region subproblem and see how they lead to a robust framework for its solution. Details are given in Sect. 3, and here it is shown that the underlying method may always be made at least superlinearly convergent. Theoretical and practical developments for the regularisation problem (1.2) are similar [15], but for brevity we omit them here. The resulting software is outlined in Sect. 4, and we follow by describing experiments which indicate the effectiveness of our enhancements. We conclude and suggest future extensions in Sect. 5.

Notation: I is the appropriately-dimensioned identity matrix, e_i is its ith column, and $\|\cdot\|$ denotes the Euclidean norm $\|\cdot\|_2$. We suppose that the matrix pencil (H, M) has (necessarily real) eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ with associated eigenvectors u_i, $1 \le i \le n$, and recall that the generalised *Rayleigh quotient*

$$\rho_M(x) \overset{\text{def}}{=} \frac{x^T Hx}{x^T Mx}$$

satisfies the Rayleigh-quotient inequality $\lambda_1 \le \rho_M(x) \le \lambda_n$ for all non-zero x; for brevity we let $\rho(x) \overset{\text{def}}{=} \rho_I(x)$. We denote the *gap* of the eigenvalue λ_i of the pencil to be

$$\mathrm{gap}(\lambda_i) = \min_{\lambda_j \neq \lambda_i} |\lambda_i - \lambda_j|,$$

where by convention $\mathrm{gap}(\lambda_i) = \infty$ if $\lambda_j = \lambda_i$ for all $1 \leq j \leq n$.

2 Theoretical considerations

In this section, where appropriate, we reduce the problem to one for which $M = I$, and thus $\|\cdot\|_M = \|\cdot\|$. From a theoretical viewpoint nothing is lost in general by this since by assumption M may be decomposed as $M = R^T R$ for some non-singular R, and problem (1.1) becomes

$$\underset{\overline{x}\in\mathbb{R}^n}{\text{minimize}}\ \overline{c}^T\overline{x} + \frac{1}{2}\overline{x}^T\overline{H}\overline{x} \ \text{ subject to } \ \|\overline{x}\| \leq \Delta$$

involving data $\overline{c} = R^{-T}c$, $\overline{H} = R^{-T}HR^{-1}$ and the desired solution $x = R^{-1}\overline{x}$. In practice, we may wish to avoid decomposing M and, in particular, forming $\overline{H}$, and we return to this when we describe practical issues. We note in passing that $\|\overline{c}\| = \|c\|_{M^{-1}}$ and that eigenvalues of $\overline{H}$ are generalised eigenvalues of the pencil (H, M); if $\overline{u}$ is an eigenvector of $\overline{H}$, $u = R^{-1}\overline{u}$ is a generalised eigenvector of the pencil (H, M).

For any scalar λ, we let $\overline{x}(\lambda)$ be the (minimum-norm) solution to

$$(\overline{H} + \lambda I)\overline{x}(\lambda) = -\overline{c} \tag{2.1}$$

whenever the system (2.1) is consistent; equivalently $x(\lambda) \overset{\text{def}}{=} R^{-1}\overline{x}(\lambda)$ satisfies

$$(H + \lambda M)x(\lambda) = -c. \tag{2.2}$$

If $\overline{H}$ has eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and a spectral decomposition $\overline{H} = \overline{U}\Lambda\overline{U}^T$, where Λ is the diagonal matrix of eigenvalues and $\overline{U} = (\overline{u}_1 \ldots \overline{u}_n)$ is the corresponding orthogonal matrix of eigenvectors, it follows that

$$\overline{x}(\lambda) = \overline{U}\overline{y}(\lambda), \quad \text{where} \quad \overline{y}_i(\lambda) = -\frac{\gamma_i}{\lambda + \lambda_i} \quad \text{and} \quad \gamma_i = \overline{u}_i^T\overline{c} \quad \text{for} \quad 1 \leq i \leq n.$$

Throughout this paper we will be concerned with the behaviour of powers of the M-norm of $x(\lambda)$ as λ varies. To this end, we define

$$\lambda_{\mathrm{s}} \overset{\text{def}}{=} \max(0, -\lambda_1),$$

and have the following general result.

Lemma 1 *Let* $\lambda_s = \max(0, -\lambda_1)$, *where* λ_1 *is the leftmost eigenvalue of the pencil* (H, M), *and suppose that* $x(\lambda)$ *satisfies* (2.2). *Then the function*

$$\pi(\lambda; \beta) \overset{\text{def}}{=} \|x(\lambda)\|_M^{\beta}$$

is strictly decreasing from $\pi(\lambda_s; \beta)$ *to zero and strictly convex for* $\lambda \in (\lambda_s, \infty)$ *when* $\beta > 0$*, and strictly increasing from* $\pi(\lambda_s; \beta)$ *to infinity and concave for* $\lambda \in (\lambda_s, \infty)$ *when* $\beta \in [-1, 0)$.

Proof The result follows directly from [8, Lem. 2.1] since

$$\|x(\lambda)\|_M = \|\overline{x}(\lambda)\| = \|\overline{y}(\lambda)\| = \sqrt{\sum_{i=1}^{n} \left(\frac{\gamma_i}{\lambda + \lambda_i}\right)^2}. \tag{2.3}$$

□

2.1 The trust-region problem

Quite remarkably, there is a characterisation of global optimality for the trust-region problem (1.1).

Theorem 1 [21, Thm. 2.1], [38, Lem. 2.1] *Any global minimizer* x_* *of* (1.1) *satisfies the equation*

$$(H + \lambda_* M)x_* = -c, \tag{2.4}$$

where $H + \lambda_* M$ *is positive semi-definite,* $\lambda_* \geq 0$*, and* $\lambda_*(\|x_*\|_M - \Delta) = 0$*. If* $H + \lambda_* M$ *is positive definite, then* x_* *is unique.*

This result is constructive. A minimizer lies strictly within the trust region only if H is positive definite and $\|H^{-1}c\|_M < \Delta$, while if H is positive definite and $\|H^{-1}c\|_M = \Delta$ the trust-region constraint is active but effectively irrelevant at the minimizer. Otherwise, with one notable exception—the "hard case" [38] which we will discuss shortly—the multiplier $\lambda_* > \lambda_{\text{s}}$ and $\|x_*\|_M = \Delta$. In this, by contrast "easy" case, we seek the (unique) root of the scalar nonlinear "secular" equation

$$\pi(\lambda; \beta) \equiv \|x(\lambda)\|_M^\beta = \Delta^\beta \tag{2.5}$$

in the interval $(\lambda_{\text{s}}, \infty)$. We are helped in this task by Lemma 1, which shows that $\pi(\lambda; \beta)$ is either strictly convex and decreasing or strictly concave and increasing for $\beta \in [-1, \infty)\backslash\{0\}$. In particular, if we partition the real line into

$$\begin{aligned}
\mathcal{N} &\overset{\text{def}}{=} \{\lambda \mid \lambda \in (-\infty, \lambda s]\} \equiv \{\lambda \mid H + \lambda M \text{ is negative semi-definite}\},\\
\mathcal{L} &\overset{\text{def}}{=} \{\lambda \mid \lambda \in (\lambda_{\text{s}}, \lambda_*]\} \quad \equiv \{\lambda \mid H + \lambda M \text{ is positive definite and } \|x(\lambda)\|_M \geq \Delta\} \quad \text{and}\\
\mathcal{G} &\overset{\text{def}}{=} \{\lambda \mid \lambda \in (\lambda_*, \infty)\} \quad \equiv \{\lambda \mid H + \lambda M \text{ is positive definite and } \|x(\lambda)\|_M < \Delta\}
\end{aligned}$$

and denote $\mathcal{F} \overset{\text{def}}{=} \mathcal{L} \cup \mathcal{G}$ (see Fig. 1), both Newton's and the secant method for (2.5) are guaranteed to converge globally (monotonically, linearly and ultimately at least superlinearly) to the required root if started from any value(s) in $\mathcal{L}$ [8, Lem. A.1]. Moreover, since this is true for all $\beta \in [-1, \infty)\backslash\{0\}$, we are at liberty to choose the

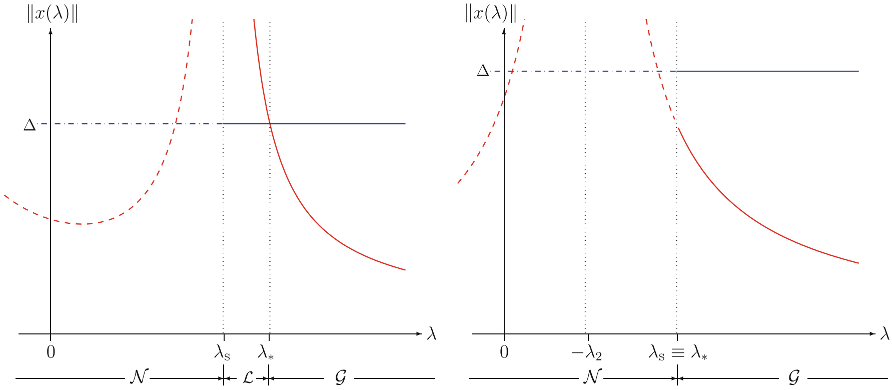

Fig. 1 The sets $\mathcal{N}$, $\mathcal{L}$ and $\mathcal{G}$ and $\|x(\lambda)\|$ for the problem of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within a ℓ_2-norm trust region of radius 4 ("easy" case, *left*) and those for $-\frac{1}{4}x_1^2 - \frac{1}{8}x_2^2 + x_2$ within a trust region of radius 5 ("hard" case, *right*)

β for which the Newton correction gives the best correction, and it can be shown that this occurs when $\beta = -1$ [8, section 2.3.3]—this formalises earlier suggestions that it might be wise to consider the secular equation $1/\|x(\lambda)\|_M = 1/\Delta$ with negative β, since this avoids the poles present at $\lambda = -\lambda_i$ when $\beta > 0$ [32,43]. With fast ultimate convergence assured in the easy case, the art is thus to be able to find an initial $\lambda \in \mathcal{L}$. We return to this in Sect. 3.

The hard case may happen when $u_i^T c \equiv \overline{u}_i^T \overline{c} = 0$ for all i for which $\lambda_i = \lambda_1 \leq 0$. Under these circumstances, (2.2) will be consistent when $\lambda = \lambda_S$—denote the minimum norm solution in this case as x_S—but if $\|x_S\| < \Delta$, there is no solution to the secular equation in $[\lambda_S, \infty)$ or equivalently $\mathcal{L}$ is empty (again see Fig. 1). In fact the required solution in this case is $x_S + \alpha_S u_1$, where the scalar α_S is chosen so that $\|x_S + \alpha_S u_1\|_M = \Delta$. Notice here that to obtain the exact solution in the hard case requires the eigenvalue λ_1, a corresponding eigenvector u_1 of the pencil (H, M) and the "trajectory" vector

$$x_S = \lim_{\lambda \to \lambda_S} x(\lambda)$$

from (2.2).

2.2 The regularisation problem

As in the trust-region case, we may characterize global optimality for the regularisation problem (1.2).

Theorem 2 [15, Thm. 2.3] *Any global minimizer x_* of* (1.2) *when $p > 2$ satisfies the equation*

$$(H + \lambda_* M)x_* = -c, \tag{2.6}$$

where $H + \lambda_* M$ is positive semi-definite, and $\lambda_* = \sigma \|x_*\|_M^{p-2}$. If $H + \lambda_* M$ is positive definite, then x_* is unique.

As in the trust-region case, the result suggests how to find the global minimizer of $r(x)$. Specifically, in all but a "hard" case, we seek the unique root $\lambda_* > \lambda_s$ of the scalar nonlinear "secular" equation

$$\|x(\lambda)\|_M^{\beta} = (\lambda/\sigma)^{\beta/(p-2)} \tag{2.7}$$

for some appropriately chosen β. Further details are given in [15].

3 Algorithmic considerations

3.1 Matrix factorization

We aim to solve (2.2) via a factorization of the symmetric matrix $H + \lambda M$. Since we are only concerned with λ for which $H + \lambda M$ is positive semi-definite, Cholesky or LDL^T factorization (with permutations in the singular case) is appropriate. As we are interested in the sparse case, symmetric permutations should be applied (implicitly) to $H + \lambda M$ prior to the factorization (the "analysis" phase) in order to limit fill in the factors. However, as a priori we do not know whether $H + \lambda M$ is definite, precautions should be in place to report if an indefinite matrix has been encountered (and immediately stop the factorization if this occurs). These features are common to a number of well-known sparse, symmetric linear equation solvers—such methods are generally reliable and effective [24,28]. We use the commercial package[1] MA57 [16] but provide a slightly-less effective alternative SILS (based on the earlier MA27 [17]) for those unable to access MA57.

3.2 The secular function and its properties

Suppose that $x(\lambda)$ satisfies (2.2). We consider properties of the *secular* function

$$\pi(\lambda) \overset{\text{def}}{=} x^T(\lambda) M x(\lambda) \equiv \|x(\lambda)\|_M^2. \tag{3.1}$$

3.2.1 Derivatives

In order to solve the secular equations $\pi(\lambda) - \Delta^2 = 0$ and $\pi(\lambda) - (\lambda/\sigma)^{2/(p-2)} = 0$ by Newton-like or higher-order iteration, we need to evaluate $\pi(\lambda)$ and its derivatives. Denoting the k-th derivative with respect to λ by a superscript (k), we have the following result.

Theorem 3 *Suppose that $H + \lambda M$ is positive definite, that $x(\lambda)$ satisfies (2.2), and that $x^{(0)}(\lambda) \overset{\text{def}}{=} x(\lambda)$ and $\alpha_0 \overset{\text{def}}{=} 1$. Then, for $k = 0, 1, \ldots$, the derivatives of $\pi(\lambda) =$*

[1] MA57 is available without charge to academics.

$x^T(\lambda)Mx(\lambda)$ satisfy

$$\pi^{(2k+1)}(\lambda) = 2\alpha_k x^{(k)T}(\lambda)Mx^{(k+1)}(\lambda) \tag{3.2}$$

$$\text{and } \pi^{(2k+2)}(\lambda) = \alpha_{k+1} x^{(k+1)T}(\lambda)Mx^{(k+1)}(\lambda), \tag{3.3}$$

where

$$(H+\lambda M)x^{(k+1)}(\lambda) = -(k+1)Mx^{(k)}(\lambda) \tag{3.4}$$

and

$$\alpha_{k+1} = 2\frac{(2k+3)}{(k+1)}\alpha_k. \tag{3.5}$$

Proof It follows immediately by differentiating (2.2) that $(H+\lambda M)x^{(1)}(\lambda) = -Mx(\lambda)$ and then by induction and continued differentiation that (3.4) holds. Now suppose that

$$\pi^{(2k)} = \alpha_k x^{(k)T}(\lambda)Mx^{(k)}(\lambda);$$

this is true for $k = 0$ by definition. Differentiating gives (3.2), and a second differentiation reveals

$$\pi^{(2k+2)}(\lambda) = 2\alpha_k[x^{(k+1)T}(\lambda)Mx^{(k+1)}(\lambda) + x^{(k)T}(\lambda)Mx^{(k+2)}(\lambda)]. \tag{3.6}$$

But it follows from (3.4) that

$$\begin{aligned}(k+1)x^{(k)T}(\lambda)Mx^{(k+2)}(\lambda) &= -x^{(k+1)T}(\lambda)(H+\lambda M)x^{(k+2)}(\lambda)\\ &= (k+2)x^{(k+1)T}(\lambda)Mx^{(k+1)}(\lambda)\end{aligned}$$

and hence (3.6) gives

$$\pi^{(2k+2)}(\lambda) = 2\alpha_k\left(1+\frac{k+2}{k+1}\right)x^{(k+1)T}(\lambda)Mx^{(k+1)}(\lambda)$$

which is (3.3) and (3.5). □

Corollary 1 *Suppose that $H+\lambda M$ is positive definite with LDL^T factorization $H+\lambda M = LDL^T$. Let $Ly(\lambda) = -c$, $Dz(\lambda) = y(\lambda)$ and $L^Tx(\lambda) = z(\lambda)$. Starting with $x^{(0)}(\lambda) = x(\lambda)$, define $y^{(k+1)}(\lambda), z^{(k+1)}(\lambda)$ and $x^{(k+1)}(\lambda)$ recursively for $k = 0, 1, \ldots,$ via*

$$\begin{gathered}Ly^{(k+1)}(\lambda) = -(k+1)Mx^{(k)}(\lambda),\ \ Dz^{(k+1)}(\lambda) = y^{(k+1)}(\lambda)\\ \text{and }\ L^Tx^{(k+1)}(\lambda) = z^{(k+1)}(\lambda).\end{gathered}$$

Let $\alpha_0 = 1$*. Then, for* $k = 0, 1, \ldots$*, the derivatives of* $\pi(\lambda) = x^T(\lambda)Mx(\lambda)$ *satisfy*

$$\begin{aligned} \pi^{(2k+1)}(\lambda) &= 2\alpha_k x^{(k)T}(\lambda)Mx^{(k+1)}(\lambda) \equiv -\beta_k y^{(k+1)T}(\lambda)z^{(k+1)}(\lambda) \\ \textit{and} \;\; \pi^{(2k+2)}(\lambda) &= \alpha_{k+1} x^{(k+1)T}(\lambda)Mx^{(k+1)}(\lambda) \end{aligned}$$

where

$$\beta_k = \frac{2}{(k+1)}\alpha_k \;\; \textit{and} \;\; \alpha_{k+1} = (2k+3)\beta_k.$$

Proof Since $H + \lambda M = LDL^T$, the definitions of $y^{(k+1)}(\lambda)$, $z^{(k+1)}(\lambda)$ and $x^{(k+1)}(\lambda)$ correspond to solving (3.4) by parts. The alternative expression for $\pi^{(2k+1)}(\lambda)$ follows from the identity

$$\begin{aligned} (k+1)x^{(k)T}(\lambda)Mx^{(k+1)}(\lambda) &= -(L^T x^{(k+1)}(\lambda))^T (DL^T x^{(k+1)}(\lambda)) \\ &= -y^{(k+1)T}(\lambda)z^{(k+1)}(\lambda). \end{aligned}$$

The remainder of the result follows immediately from Theorem 3. □

Notice how each odd-power derivative of π requires a product with M and solves with L and D, while every even powered derivative needs a solve with L^T. A slight simplification occurs if a Cholesky rather than LDL^T factorization is used. In particular, $y^{(k)}(\lambda)$ and $z^{(k)}(\lambda)$ are identical, and the odd-order derivatives become $\pi^{(2k+1)}(\lambda) = -2\beta_k \|y^{(k+1)}(\lambda)\|^2$. Variants on this theme for regularised linear-least-squares problems have been given by Gander [20, Thm.5.1].

3.2.2 Taylor series approximations to $\pi(\lambda)$

Armed with derivatives of $\pi(\lambda)$, it is now possible to contemplate Taylor series approximations to $\pi(\lambda; \beta)$. Consider first the special case when $\beta = 2$ and thus $\pi(\lambda, 2) = \pi(\lambda)$.

Theorem 4 *Let* $\pi(\lambda) = \|x(\lambda)\|_M^2$*, where* $x(\lambda)$ *satisfies* (2.2)*. Suppose that* λ_c *is a given value such that* $\lambda_c > \lambda_s$*. Let* $\pi_k(\delta)$ *be the k-th order Taylor series approximation to* $\pi(\lambda_c + \delta)$*. Then*

$$\pi(\lambda_c + \delta) \leq \pi_k(\delta) \;\; \textit{for even } k > 0 \textit{ and } \;\; \pi(\lambda_c + \delta) \geq \pi_k(\delta) \;\; \textit{for odd } k > 0 \tag{3.7}$$

when $\delta > 0$*, while*

$$\pi(\lambda_c + \delta) \geq \pi_{k+1}(\delta) \geq \pi_k(\delta) \;\; \textit{for all } k > 0 \tag{3.8}$$

when $\lambda_s - \lambda_c < \delta < 0$*. The inequalities in* (3.7)–(3.8) *are strict whenever* $c \neq 0$*.*

Proof It follows trivially from (2.3) that the jth derivative, $\pi^{(j)}(\lambda)$, of $\pi(\lambda)$ is

$$\pi^{(j)}(\lambda) = (-1)^j (j+1)! \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^{j+2}}. \tag{3.9}$$

Thus if we define the kth order Taylor approximation

$$\pi_k(\delta) \overset{\text{def}}{=} \sum_{j=0}^{k} \frac{\pi^{(j)}(\lambda_{\text{C}})}{j!} \delta^j \tag{3.10}$$

to $\pi(\lambda_{\text{C}} + \delta)$, we see from Taylor's theorem that the error

$$\pi(\lambda_{\text{C}} + \delta) - \pi_k(\delta) = \frac{1}{(k+1)!} \pi^{(k+1)}(\lambda_{\text{C}} + \xi)\delta^{k+1} \tag{3.11}$$

for some ξ strictly between 0 and δ so long as $\delta > \lambda_s - \lambda_{\text{C}}$. But, since (3.9) shows that even derivatives of $\pi(\lambda)$ are non-negative and odd derivatives non-positive for all $\lambda_{\text{C}} > \lambda_{\text{S}}$, (3.11) gives (3.7) when $\delta > 0$ and

$$\pi(\lambda_{\text{C}} + \delta) \geq \pi_k(\delta) \quad \text{for all } k$$

when $\lambda_s - \lambda_{\text{C}} < \delta < 0$. This and the relationship

$$\pi_{k+1}(\delta) - \pi_k(\delta) = (k+2)(-\delta)^{k+1} \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda_{\text{C}} + \lambda_i)^{k+3}},$$

which follows from (3.9) and (3.10), give (3.8) for negative δ.

When $c \neq 0$, at least one of the γ_i in (3.9) is nonzero, and this is sufficient to ensure that the inequalities that result from (3.9) in the above arguments are strict. □

Now suppose that $\tau(\lambda)$ is a given monotonically non-decreasing function on $(\lambda_{\text{S}}, \infty)$ and that $\pi(\lambda_{\text{S}}) > \tau(\lambda_{\text{S}})$. For example $\tau(\lambda)$ might be constant, e.g., Δ^2 as in (2.5), or increasing, e.g., $(\lambda/\sigma)^{2/(p-2)}$ as in (2.7). In this case, Lemma 1 implies that there is a unique root, say $\lambda_* > \lambda_{\text{S}}$ of the equation

$$\pi(\lambda) = \tau(\lambda), \tag{3.12}$$

or equivalently

$$\pi(\lambda_{\text{C}} + \delta) = \tau(\lambda_{\text{C}} + \delta). \tag{3.13}$$

for the correction δ to λ_{C}.

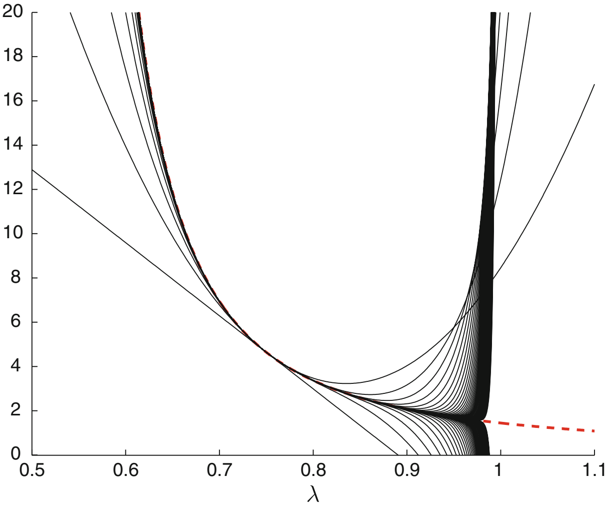

Fig. 2 The secular function $\pi(\lambda)$ for the problem of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within an ℓ_2-norm trust region (*dashed line*) along with the Taylor series approximants $\pi_k(\lambda - \lambda_C)$ of degrees $k = 1$ to 100 about $\lambda_C = 0.75$ (*solid lines*). Observe the threshold $\lambda_B \approx 0.98$ above which no approximant is good

Consider $\lambda_C < \lambda_*$, in which case $\pi_k(0) = \pi(\lambda_C) > \tau(\lambda_C)$, and let $\delta_* \overset{\text{def}}{=} \lambda_* - \lambda_C > 0$. If k is odd, Theorem 4 implies that $\pi_k(\delta_*) \leq \pi(\lambda_C + \delta_*) = \tau(\lambda_C + \delta_*)$. Thus the equation

$$\pi_k(\delta) = \tau(\lambda_C + \delta) \tag{3.14}$$

has at least one root in $(0, \delta_*)$ for odd k, and Newton and other odd-degree Taylor series methods for (3.12) based on finding corrections as positive real roots of (3.14) will underestimate λ_*. Moreover, since $\tau(\lambda_C + \delta) > \pi(\lambda_C + \delta) \geq \pi_k(\delta)$ for $\lambda_C + \delta > \lambda_*$, all positive roots of (3.14) give under-estimators. By contrast, if k is even, $\pi_k(\delta_*) \geq \pi(\lambda_C + \delta_*) = \tau(\lambda_C + \delta_*)$ and any positive root of (3.14) will overestimate λ_*.

Now consider the alternative $\lambda_C > \lambda_*$, in which case $\pi_k(0) = \pi(\lambda_C) < \tau(\lambda_C)$ and $\delta_* < 0$. Theorem 4 gives that $\tau(\lambda_C + \delta) > \pi(\lambda_C + \delta) \geq \pi_k(\delta)$ for all $\delta \in (\delta_*, 0]$ and thus the least-negative root (if any) of (3.14) will not lie to the right of δ_*. Moreover, as k increases (3.8) indicates that the least-negative roots move to the right, and thus the higher the degree of approximation used, the better the lower bound on λ_* provided by the least-negative root of (3.14).

We illustrate these properties in Fig. 2. Notice that, since the secular function is not everywhere analytic, the Taylor series approximations deteriorate as δ increases whatever degree of approximation is used. In particular there is a threshold λ_B—for our example, $\lambda_B \approx 0.98$—above which no approximant π_k is close to π; the actual value depends on the distance of λ_C to the nearest singularity of $\pi(\lambda)$ in the complex plane [4, Thm. 16.20 *et seq.*]. This implies that the suggested root of (3.14) for odd-degree approximations may be a poor estimate of λ_* if $\lambda_C \in \mathcal{L}$ and $\tau(\lambda_B)$ is significantly smaller that $\pi(\lambda_{\text{B}})$. As a consequence many iterations may be required to determine λ_*. Conversely, there appears to be good agreement for negative δ as the degree of approximation increases, and thus scope for optimism that reasonable-order Taylor approximations will perform well if $\lambda \in \mathcal{G}$. Similar results concerning the monotonic (and rapid) convergence of Taylor series methods for roots of more general functions, whose derivatives satisfy appropriate sign conditions, are known [47, Thm 4.2].

There is as always a trade-off between using potentially less accurate lower order approximants against more expensive higher-order ones. For our secular equations, the dominant cost is likely to be in factorizing $H + \lambda M$—although this will be problem/sparsity dependent—and a modest number of derivatives will incur little extra relative cost. Thus better than-first-order (Newton)-like methods seem particularly appealing in our context.

3.2.3 Taylor series approximations to powers of $\pi(\lambda)$

We now turn to the general case in which β may differ from 2, and consider the kth order Taylor series approximation, $\pi_k(\delta; \beta)$, to $\pi(\lambda_{\text{C}} + \delta; \beta)$ for modest values of $k \leq 3$; a higher-order analysis is possible but becomes increasingly messy and of likely decreasing practical value given the increasing cost of evaluating derivatives.

Differentiating $\pi(\lambda; \beta) = \|x(\lambda)\|_M^\beta \equiv [\pi(\lambda)]^{\frac{\beta}{2}}$ with respect to λ and using the chain rule, we obtain

$$\begin{aligned}
\pi^{(1)}(\lambda; \beta) &= \frac{\beta}{2} [\pi(\lambda)]^{\frac{\beta}{2}-1} \pi^{(1)}(\lambda), &(3.15)\\
\pi^{(2)}(\lambda; \beta) &= \frac{\beta}{2} [\pi(\lambda)]^{\frac{\beta}{2}-1} \pi^{(2)}(\lambda)\\
&\quad + \frac{\beta}{2}\left(\frac{\beta}{2} - 1\right) [\pi(\lambda)]^{\frac{\beta}{2}-2} \left[\pi^{(1)}(\lambda)\right]^2, &(3.16)\\
\pi^{(3)}(\lambda; \beta) &= \frac{\beta}{2} [\pi(\lambda)]^{\frac{\beta}{2}-3} \Big([\pi(\lambda)]^2 \pi^{(3)}(\lambda)\\
&\quad + 3\left(\frac{\beta}{2} - 1\right) \pi(\lambda)\pi^{(1)}(\lambda)\pi^{(2)}(\lambda)\\
&\quad + \left(\frac{\beta}{2} - 1\right)\left(\frac{\beta}{2} - 2\right) \left[\pi^{(1)}(\lambda)\right]^3\Big) &(3.17)
\end{aligned}$$

and

$$\begin{aligned}
\pi^{(4)}(\lambda; \beta) = \frac{\beta}{2} [\pi(\lambda)]^{\frac{\beta}{2}-4} \Big(&[\pi(\lambda)]^3 \pi^{(4)}(\lambda)\\
&+ 4\left(\frac{\beta}{2} - 1\right) [\pi(\lambda)]^2 \pi^{(1)}(\lambda)\pi^{(3)}(\lambda)\\
&+ 3\left(\frac{\beta}{2} - 1\right) [\pi(\lambda)]^2 \left[\pi^{(2)}(\lambda)\right]^2
\end{aligned}$$

$$+6\left(\frac{\beta}{2}-1\right)\left(\frac{\beta}{2}-2\right)\pi(\lambda)\left[\pi^{(1)}(\lambda)\right]^2\pi^{(2)}(\lambda)$$
$$+\left(\frac{\beta}{2}-1\right)\left(\frac{\beta}{2}-2\right)\left(\frac{\beta}{2}-3\right)\left[\pi^{(1)}(\lambda)\right]^4\Bigg). \tag{3.18}$$

From this we may deduce the following result.

Lemma 2 *Let $\pi(\lambda;\beta) = \|x(\lambda)\|_M^\beta$, where $x(\lambda)$ satisfies (2.2). Suppose that $\lambda > \lambda_s$. Then*

(i) $\pi^{(1)}(\lambda;\beta) \le 0$ *for all* $\beta > 0$ *while* $\pi^{(1)}(\lambda;\beta) \ge 0$ *for all* $\beta < 0$*;*
(ii) $\pi^{(2)}(\lambda;\beta) \ge 0$ *for all* $\beta > 0$ *while* $\pi^{(2)}(\lambda;\beta) \le 0$ *for all* $\beta \in [-1, 0)$*;*

The above inequalities are strict whenever $c \ne 0$.

Proof Statements (i) follows directly from (3.15) and Theorem 3, while (ii) follows from Lemma 1. □

Theorem 5 *Let $\pi(\lambda;\beta) = \|x(\lambda)\|_M^\beta$, where $x(\lambda)$ satisfies (2.2). Suppose that the value λ_c and perturbation δ satisfy $\lambda_c > \lambda_s$ and $\delta > \lambda_s - \lambda_c$. Let $\pi_1(\delta;\beta)$ be the first-order Taylor series approximation to $\pi(\lambda_c + \delta;\beta)$. Then,*

(i) *for* $\beta > 0$,

$$\pi(\lambda_c + \delta;\beta) \ge \pi_1(\delta;\beta); \quad \textit{and} \tag{3.19}$$

(ii) *otherwise for* $\beta \in [-1, 0)$,

$$\pi(\lambda_c + \delta;\beta) \le \pi_1(\delta;\beta). \tag{3.20}$$

The inequalities in (3.19)–(3.20) are strict whenever $c \ne 0$.

Proof These results follow directly from the relationship $\pi(\lambda_c + \delta;\beta) = \pi_k(\delta;\beta) + \pi^{(k+1)}(\lambda_c + \xi_k;\beta)\delta^{k+1}/(k+1)!$ for some ξ_k between 0 and δ (Taylors' theorem), $\pi_{k+1}(\delta;\beta) = \pi_k(\delta;\beta) + \pi^{(k+1)}(\lambda_c;\beta)\delta^{k+1}/(k+1)!$ (Taylor series) and Lemma 2. □

There is also strong evidence that the following is true.

Conjecture 1 *Let $\pi(\lambda;\beta)$ and λ be as in Lemma 2. Then*

(i) $\pi^{(3)}(\lambda;\beta) \le 0$ *for all* $\beta > 0$ *while* $\pi^{(3)}(\lambda;\beta) \ge 0$ *for all* $\beta \in [-\frac{2}{3}, 0)$*; and*
(ii) $\pi^{(4)}(\lambda;\beta) \ge 0$ *for all* $\beta > 0$ *while* $\pi^{(4)}(\lambda;\beta) \le 0$ *for all* $\beta \in [-\frac{2}{5}, 0)$.

The above inequalities are strict whenever $c \ne 0$.

This conjecture is supported both by an unverified,[2] essentially computer-generated proof [14], and by considerable empirical evidence accrued while testing our software. Of course, for the special case $\beta = 2$, the conjecture immediately follows from (3.9).

[2] At around 250 pages, it is of course unreasonable to expect referees to corroborate such a proof.

Theorem 6 *Let $\pi(\lambda;\beta) = \|x(\lambda)\|_M^\beta$, where $x(\lambda)$ satisfies (2.2). Suppose that $\lambda_c > \lambda_s$ and let $\pi_k(\delta;\beta)$ be the k-th order Taylor series approximation to $\pi(\lambda_c+\delta;\beta)$, and suppose that Conjecture 1 is true. Then,*

(i) *for $\beta > 0$,*

$$\pi(\lambda_c+\delta;\beta) \leq \pi_2(\delta;\beta) \;\; \textit{and} \;\; \pi(\lambda_c+\delta;\beta) \geq \pi_k(\delta;\beta) \;\; \textit{for } k = 1, 3 \tag{3.21}$$

when $\delta > 0$, while

$$\pi(\lambda_c+\delta;\beta) \geq \pi_3(\delta;\beta) \geq \pi_2(\delta;\beta) \geq \pi_1(\delta;\beta) \tag{3.22}$$

when $\lambda_s - \lambda_c < \delta < 0$; and

(ii) *otherwise*

$$\begin{aligned} &\pi(\lambda_c+\delta;\beta) \leq \pi_1(\delta;\beta) \;\; \textit{for} \;\; \beta \in [-1,0) \\ &\pi(\lambda_c+\delta;\beta) \geq \pi_2(\delta;\beta) \;\; \textit{for} \;\; \beta \in [-\tfrac{2}{3},0) \\ &\pi(\lambda_c+\delta;\beta) \leq \pi_3(\delta;\beta) \;\; \textit{for} \;\; \beta \in [-\tfrac{2}{5},0) \end{aligned} \tag{3.23}$$

when $\delta > 0$, while

$$\begin{aligned} &\pi(\lambda_c+\delta;\beta) \leq \pi_3(\delta;\beta) \leq \pi_2(\delta;\beta) \leq \pi_1(\delta;\beta) \;\; \textit{for} \;\; \beta \in [-\tfrac{2}{5},0) \\ &\pi(\lambda_c+\delta;\beta) \leq \pi_2(\delta;\beta) \leq \pi_1(\delta;\beta) \;\; \textit{for} \;\; \beta \in [-\tfrac{2}{3},0) \\ &\pi(\lambda_c+\delta;\beta) \leq \pi_1(\delta;\beta) \;\; \textit{for} \;\; \beta \in [-1,0) \end{aligned} \tag{3.24}$$

when $\lambda_s - \lambda_c < \delta < 0$.

The inequalities in (3.21)–(3.24) are strict whenever $c \neq 0$.

Proof Same as for Theorem 5. □

The limiting ranges on negative β in (3.23) and (3.24) may seem inconvenient, but as we see in Fig. 3 they may be necessary to ensure the Taylor polynomials over/under-estimate $\pi(\lambda;\beta)$.

We can repeat the discussion following Theorem 4 concerning $\pi(\lambda)$ for the more general function $\pi(\lambda;\beta)$. For positive β, all we said about solving (3.12) remains true for

$$\pi(\lambda;\beta) = \tau(\lambda;\beta). \tag{3.25}$$

In particular, if $\lambda_\text{C} < \lambda_*$, the largest positive roots of

$$\pi_k(\delta,\beta) = \tau(\lambda_\text{C}+\delta;\beta) \tag{3.26}$$

for $k = 1$ (and $k = 3$ if Conjecture 1 holds) lead to under-estimators of λ_*, while if $\lambda_* < \lambda_\text{C}$, the least negative roots of (3.26) for $k = 1$ (and $k = 2$ and 3 if the conjecture

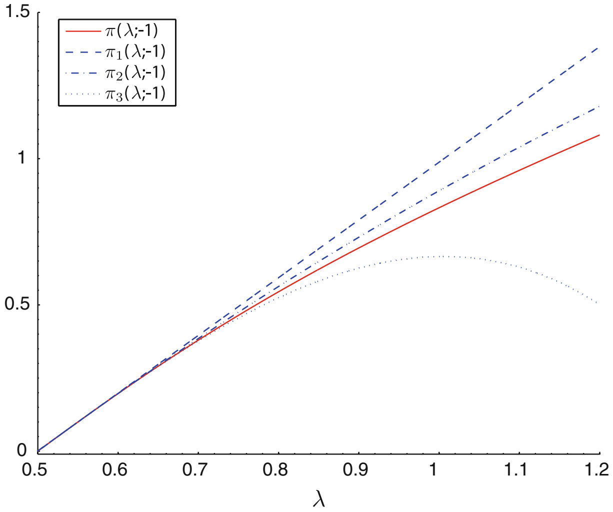

Fig. 3 The function $\pi(\lambda, -1)$ for the problem of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within an ℓ_2-norm trust region along with the Taylor series approximants $\pi_1(\lambda; -1)$ to $\pi_3(\lambda; -1)$ of degrees 1 to 3 about $\lambda_C = 0.55$. Observe that $\pi_2(\lambda; \beta)$ and $\pi_3(\lambda; \beta)$ do not obey (3.23); magnifying the figure shows that they also violate (3.24)

holds) will give estimates to the left of λ_* with the best under-estimator resulting when $k = 3$. If β is negative, and $\tau(\lambda; \beta)$ is a given monotonically non-increasing function on (λ_S, ∞) and that $\pi(\lambda_S; \beta) < \tau(\lambda_S; \beta)$ the same results are true, but now only so long as β is constrained to be larger than -1 for a linear Taylor approximant (and $-\frac{2}{3}$ and $-\frac{2}{5}$, respectively, for the quadratic and cubic Taylor approximants if Conjecture 1 holds). Finding the root of (3.26) for a given β and degree k as described and adding this to λ_C gives what we shall call the *best prediction*, $\lambda_k(\beta)$. We shall also define $\mathcal{B}_k$ to be the interval of allowable values of β for which Taylor approximants of degree k provide guaranteed under-estimates of λ_*; thus according to Theorems 4, 5 and 6, $\mathcal{B}_1 = [-1, \infty)$ and $\mathcal{B}_3 = \{2\}$ (or $\mathcal{B}_3 = [-\frac{2}{5}, \infty)$ if Conjecture 1 holds), while $\mathcal{B}_2 = \{2\}$ (or $\mathcal{B}_2 = [-\frac{2}{3}, \infty)$ if the conjecture holds) when $\lambda < \lambda_*$ and is empty otherwise.

Since the best prediction for each degree k and β in its allowed range gives a viable estimate of λ_*, a natural question is which k and β gives the overall best estimate of λ_*. In Fig. 4 we illustrate how the best predictions behave as a function of β for Taylor approximations of degrees up to three in the trust-region case for which $\tau(\lambda; \beta) = \Delta^\beta$.

It is known for linear Taylor approximants that $\beta = -1$ is best in the trust-region case [8, section 2.3.3]. The figure also suggests that the optimal choice for higher-degree polynomials might occur at the lower end of their allowed β range. How this translates for more general τ, such as for that for regularisation, is less clear, but certainly picking the best $\lambda_k(\beta)$ from a sample of allowable β and k is an algorithmic possibility.

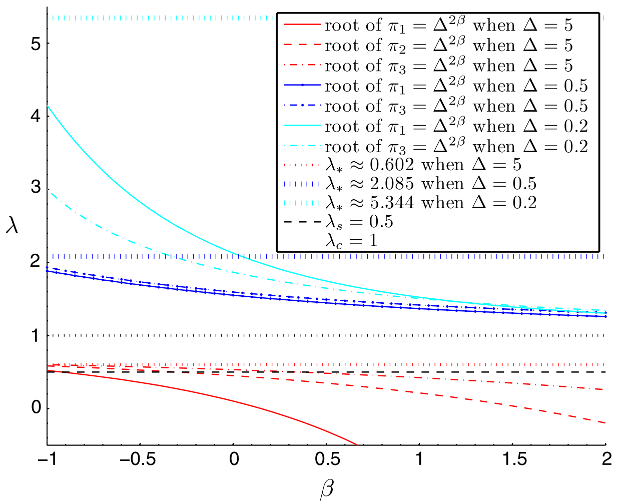

Fig. 4 A plot of the best prediction $\lambda_k(\beta)$ of the scalar polynomial equation $\pi_k(\lambda;\beta) = \Delta^{2\beta}$ as a function of β, for the Taylor series approximants $\pi_k(\lambda;\beta)$ of degrees $k = 1, 2, 3$ to $\pi(\lambda;\beta) \equiv \|x(\lambda)\|^{\beta}$ about $\lambda_C = 1$ for the problems of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within ℓ_2-norm trust regions of radii $\Delta = 5, 0.5$ and 0.2

3.3 The trust-region problem

3.3.1 Finding a point in $\mathcal{L}$

The standard method [21,38] for finding an initial point in $\mathcal{L}$ (if one exists) is to determine bounds $\lambda_L \geq 0$ and λ_U for which $\lambda_* \in [\lambda_L, \lambda_U]$. If $\lambda_U > \lambda_*$, then necessarily $\lambda_U \in \mathcal{G}$, while $\lambda_L \in \mathcal{N} \cup \mathcal{L}$. The bounds λ_L and λ_U are adjusted by iteration so that the length of $[\lambda_L, \lambda_U]$ shrinks by a non-trivial amount at each step and thus ultimately will collapse to $\{\lambda_*\}$. If $\mathcal{L}$ is non-empty (the "easy case"), the interval will be adjusted a finite number of times; as soon as a point in $\mathcal{L}$ has been determined, no further adjustments of λ_L and λ_U are required as subsequent iterates remain in $\mathcal{L}$. The cases when $\mathcal{L}$ is empty (either the "hard case" or when the solution lies interior to the trust-region) will be discussed later.

Suppose that we have found an interval $[\lambda_L, \lambda_U]$ surrounding λ_*, but that the current estimate $\lambda_C \in [\lambda_L, \lambda_U]$ of λ_* is not in $\mathcal{L}$. If $\lambda_C \in \mathcal{G}$, an improvement λ_+ may be sought by applying one or more iterations of a suitable root finder—Newton's method applied to (2.5) for $\beta \geq -1$ is guaranteed to overshoot the root (and thus lie in $\mathcal{N} \cup \mathcal{L}$), but other iterations might not; we will return to this later. There are three outcomes. With luck, $\lambda_+ \in \mathcal{L}$ and we are done. Otherwise, we have the opportunity to improve one of the interval bounds; the lower one if $\lambda_+ \in \mathcal{N}$ and the upper if $\lambda_+ \in \mathcal{G}$. If $\lambda_C \in \mathcal{N}$, root-finding is unlikely to be fruitful as we lie on the wrong side of the pole of $\|x(\lambda)\|$. In addition—either as a by-product of the root-finding when $\lambda_C \in \mathcal{G}$ or from some auxiliary calculation, for instance from a suitably chosen Rayleigh-quotient, when $\lambda_C \in \mathcal{N}$—we might obtain a new upper bound on λ_1, and this may lead to a further improvement in λ_L; again we will examine this in detail later. Having refined known lower and upper bounds, it remains to choose a new estimate of λ_* with the goal of ensuring that the bounding interval continues to shrink at an at-worst linear rate. Many possibilities have been suggested in the past [11, section 7.3.6], usually involving a convex and/or geometric combination of the current bounds.

We formalize this discussion as Algorithm 3.1. The proposed formula for computing the next λ_{C} ensures that the ratio of widths of successive bounding intervals is at most

$$\max\left[1-\theta, \theta, \frac{\gamma\sqrt{\lambda_{\mathrm{U}}}}{\sqrt{\lambda_{\mathrm{L}}}+\sqrt{\lambda_{\mathrm{U}}}}\right]$$

for some $\theta \in (\theta_{\mathrm{L}}, \theta_{\mathrm{U}}) \subset (0, 1)$ and $\gamma \in \{0, 1\}$ [11, section 7.3.6], and thus that the algorithm has the desired effect of ensuring finite convergence if $\mathcal{L}$ is non-empty.

```
**Algorithm 3.1: Find** $\lambda \in \mathcal{L}$
Given initial $\lambda_{\mathrm{L}} \leq \lambda_* \leq \lambda_{\mathrm{U}}$ and $\lambda_{\mathrm{C}}$. Set constants $0 < \theta_{\mathrm{L}} \leq \theta_{\mathrm{U}} < 1$.
**Loop**:
  If $\lambda_{\mathrm{C}} \in \mathcal{L}$:
    Exit loop with $\lambda \leftarrow \lambda_{\mathrm{C}}$.
  Else if $\lambda_{\mathrm{C}} \in \mathcal{N}$:
    Set $\lambda_{\mathrm{L}} \leftarrow \max(\lambda_{\mathrm{L}}, \lambda_{\mathrm{C}})$.
    Possibly compute an estimate $\lambda_{\mathrm{E}} \geq \lambda_1$ and if so set $\lambda_{\mathrm{L}} \leftarrow \max(\lambda_{\mathrm{L}}, -\lambda_{\mathrm{E}})$.
    Select $\theta \in [\theta_{\mathrm{L}}, \theta_{\mathrm{U}}]$ and $\gamma \in \{0, 1\}$, and set
      $\lambda_{\mathrm{C}} \leftarrow \max(\gamma\sqrt{\lambda_{\mathrm{L}}\lambda_{\mathrm{U}}}, \lambda_{\mathrm{L}} + \theta(\lambda_{\mathrm{U}} - \lambda_{\mathrm{L}}))$.
  Else (i.e., $\lambda_{\mathrm{C}} \in \mathcal{G}$):
    Set $\lambda_{\mathrm{U}} \leftarrow \min(\lambda_{\mathrm{U}}, \lambda_{\mathrm{C}})$.
    Possibly compute an estimate $\lambda_{\mathrm{E}} \geq \lambda_1$ and if so set $\lambda_{\mathrm{L}} \leftarrow \max(\lambda_{\mathrm{L}}, -\lambda_{\mathrm{E}})$.
    Select $\theta_1 \leq \theta_2 \in [\theta_{\mathrm{L}}, \theta_{\mathrm{U}}]$ and $\gamma_1 \leq \gamma_2 \in \{0, 1\}$, and set

    $\lambda_{\mathrm{C}} \in \left[\max\left(\gamma_1\sqrt{\lambda_{\mathrm{L}}\lambda_{\mathrm{U}}}, \lambda_{\mathrm{L}} + \theta_1(\lambda_{\mathrm{U}} - \lambda_{\mathrm{L}})\right), \max\left(\gamma_2\sqrt{\lambda_{\mathrm{L}}\lambda_{\mathrm{U}}}, \lambda_{\mathrm{L}} + \theta_2(\lambda_{\mathrm{U}} - \lambda_{\mathrm{L}})\right)\right].$   (3.27)
```

3.3.2 *Initial values for* λ_{L} *and* λ_{U}

To start Algorithm 3.1, we require suitable initial values λ_{L} and λ_{U}. Since the Rayleigh-quotient inequality and (2.2) give

$$(\lambda_* + \lambda_1)^2 \leq \frac{\overline{x}_*^T(\overline{H} + \lambda_* I)^2\overline{x}_*}{\overline{x}_*^T\overline{x}_*} = \frac{\|\overline{c}\|^2}{\Delta^2} = \frac{\|c\|_{M^{-1}}^2}{\Delta^2} \leq (\lambda_* + \lambda_n)^2 \tag{3.28}$$

for solutions on the trust-region boundary, it follows immediately that

$$\frac{\|c\|_{M^{-1}}}{\Delta} - \lambda_n \leq \lambda_* \leq \frac{\|c\|_{M^{-1}}}{\Delta} - \lambda_1. \tag{3.29}$$

For (3.29) to be useful, it is necessary to find outer bounds (i.e., a lower bound on λ_1 and an upper bound on λ_n) on the extreme eigenvalues of the pencil (H, M). We also know that $\lambda_* \geq \lambda_{\mathrm{S}} = \max(0, -\lambda_1)$, so any known upper bound on λ_1 may be used.

Usable outer bounds when $M = I$ are normally found from Gershgorin's theorems or computable overestimates of $\|H\|$ such as

$$-\min(\|H\|_\infty, \|H\|_F) \leq -\|H\| \leq \lambda_1 \leq \lambda_n \leq \|H\| \leq \min(\|H\|_\infty, \|H\|_F)$$

involving the infinity norm $\|H\|_\infty$ or the Frobenius norm $\|H\|_F$ [11, section 7.3.8], [21,38]. For non-unit M, Gershgorin-like methods are also possible so long as M is strictly diagonally dominant. To see this, suppose that $(H - \lambda M)u = 0$ and that k is such that $|u_k| \geq |u_i|$ for $i = 1, \ldots, n$. In this case

$$(h_{k,k} - \lambda m_{k,k})u_k = -\sum_{i=1, i\neq k}^{n} (h_{k,i} - \lambda m_{k,i})u_i$$

and thus

$$|h_{k,k} - \lambda m_{k,k}| \leq \left| \sum_{i=1, i\neq k}^{n} (h_{k,i} - \lambda m_{k,i}) \right| \leq o_k^{\mathrm{H}} + |\lambda| o_k^{\mathrm{M}}, \tag{3.30}$$

where

$$o_k^{\mathrm{H}} \stackrel{\mathrm{def}}{=} \sum_{i=1, i\neq k}^{n} |h_{k,i}| \text{ and } o_k^{\mathrm{M}} \stackrel{\mathrm{def}}{=} \sum_{i=1, i\neq k}^{n} |m_{k,i}|.$$

Hence the eigenvalues of (H, M) lie in the union of the regions defined by

$$|h_{k,k} - \lambda m_{k,k}| \leq o_k^{\mathrm{H}} + |\lambda| o_k^{\mathrm{M}}, \quad k = 1, \ldots, n; \tag{3.31}$$

it is easy to see that each region (3.31) is a trivially computable closed interval because M is both positive definite and strictly diagonally-dominant and thus $m_{k,k} > o_k^{\mathrm{M}}$. Thus outer bounds may easily be found by computing the extrema of each of these interval bounds, and gives

$$\lambda_{\mathrm{L}} = \min_{1\leq k\leq n} \left(\frac{h_{k,k} - o_k^{\mathrm{H}}}{m_{k,k} - o_k^{\mathrm{M}}}, \frac{h_{k,k} - o_k^{\mathrm{H}}}{m_{k,k} + o_k^{\mathrm{M}}} \right) \text{ and } \lambda_{\mathrm{U}} = \max_{1\leq k\leq n} \left(\frac{h_{k,k} + o_k^{\mathrm{H}}}{m_{k,k} - o_k^{\mathrm{M}}}, \frac{h_{k,k} + o_k^{\mathrm{H}}}{m_{k,k} + o_k^{\mathrm{M}}} \right).$$

This technique fails if M is not strictly diagonally dominant as then at least one of the sets defined by (3.31) may be unbounded; it is not clear to us how to get suitable outer bounds in this case. Note also that the first inequality in (3.30) may provide a tighter bound albeit at a slightly higher computational cost. It is also possible to apply optimized diagonal scalings, as suggested by Gay [21], to improve the interval bounds, but we have not done so.

In exceptional cases finding λ_1 may be practicable—for example, if H is tri-diagonal and $M = I$, the Lanczos method is a possibility. But usually the cost of computing λ_1 is high, and an upper bound is preferable. Suitable bounds may be deduced from the Rayleigh-quotient inequality $\lambda_1 \leq \rho_M(x)$ for especially chosen x. In particular, if H_{s} and M_{s} are symmetric sub-matrices of H and M, if $\lambda_{\text{s}1}$ is the leftmost eigenvalue of the pencil $(H_{\text{s}}, M_{\text{s}})$ with associated eigenvector u_{s}, then appropriately padding u_{s} with zeros to obtain a vector $u \in \mathbb{R}^n$, we have that $\lambda_1 \leq \rho_M(u) = u_{\text{s}}^T H_{\text{s}} u_{\text{s}} / u_{\text{s}}^T M_{\text{s}} u_{\text{s}} = \lambda_{\text{s}1}$. Thus, for example, considering one-by-one symmetric sub-matrices gives the bound

$$\lambda_1 \leq \min_{1 \leq i \leq n} h_{i,i}/m_{i,i}.$$

3.3.3 New estimates from $\mathcal{G}$

Next we suppose that we have found a $\lambda_{\text{C}} \in \mathcal{G}$ and now wish to find an improvement λ_+. Since $x(\lambda)$ exists and thus the value and derivatives of $\pi(\lambda; \beta)$ may be computed, the obvious idea outlined in Sect. 3.2.3 is to estimate the root of (2.5) by replacing $\pi(\lambda; \beta)$ by its k-th order Taylor approximant $\pi_k(\delta; \beta)$ for some suitable β. As we mentioned in Sect. 3.2.3, every estimate found in this way must lie in $\mathcal{N} \cup \mathcal{L}$, and to encourage the estimate to lie in $\mathcal{L}$ ideally we should pick the largest best prediction,

$$\max_{k \in \mathbb{N}, \beta \in \mathcal{B}_k} \lambda_k(\beta),$$

where the best prediction $\lambda_k(\beta)$ is as defined on p. 34. In practice, the computationally viable under-estimate $\lambda_+ = \lambda_{\text{T}}$ for

$$\lambda_{\text{T}} = \max\left(\lambda_1(-1), \lambda_2\left(-\frac{2}{3}\right), \lambda_3\left(-\frac{2}{5}\right)\right) \tag{3.32}$$

seems to suffices—of course, unless one accepts Conjecture 1, only a guaranteed estimate such as

$$\lambda_{\text{T}} = \max\left(\lambda_1(-1), \lambda_2(2), \lambda_3(2)\right) \tag{3.33}$$

should really be used; although we have never observed an instance for which (3.32) is inappropriate, a precaution to use (3.33) should (3.32) fail is imposed. Note additionally that such a λ_{T} may be used to improve λ_{L}.

Since we cannot be sure that $\lambda_+ \in \mathcal{L}$, we might at the same time try to improve the current lower bound λ_{L}. More specifically, we aim to construct a close upper bound λ_{E} on λ_1. As we already have a factorization of $H + \lambda_{\text{C}} M$, we reuse this to apply one or more inverse iterations

$$u \leftarrow (H + \lambda_{\text{C}} M)^{-1} M u, \quad u \leftarrow u/\|u\|_M \tag{3.34}$$

to estimate the extreme eigenvector of (H, M), and then use the Rayleigh quotient $\lambda_{\text{E}} = \rho_M(u)$ as our estimate of λ_1. Although the starting vector u used is not critical, it is prudent to use the result from the last λ calculation (if any) as this is already an approximation to the desired eigenvector. We will return to this in Sect. 3.3.6.

An alternative is to use the so-called LINPACK technique to provide a suitable estimate of u_1; this is a vital component of Moré and Sorensen's [38] method and meshes well with their goal to provide a usable, low accuracy solution to (1.1). The LINPACK technique depends crucially on the ability to intercept and alter components of the solution to linear systems involving the Cholesky or LDL^T factors of $H + \lambda_{\text{C}} M$ and specially-crafted right-hand sides as the solution-process proceeds [10]. Unfortunately this limits its utility for modern sparse-factorization packages, where the factors are generally hidden from the user, and thus the LINPACK technique forms no part of our algorithm.

Another possibility is to use the method at the heart of the LAPACK condition number estimator [30,33] to provide a lower bound λ_{H}, say, on $\|(\overline{H} + \lambda_{\text{C}} I)^{-1}\|_1$. We may then use the inequalities

$$\frac{1}{\lambda + \lambda_{\text{C}}} \geq \|(\overline{H} + \lambda_{\text{C}} I)^{-1}\|_2 \geq \frac{\|(\overline{H} + \lambda_{\text{C}} I)^{-1}\|_1}{\sqrt{n}} \geq \frac{\lambda_{\text{H}}}{\sqrt{n}}$$

to derive the upper bound

$$\lambda_1 \leq \sqrt{n}/\lambda_{\text{H}} - \lambda_{\text{C}}$$

on λ_1. The LAPACK method, like the LINPACK technique, uses multiple applications of the Cholesky or LDL^T factors of $H + \lambda_{\text{C}} M$ to provide the estimate λ_{H}. This method has been used successfully in the trust-region context [22, section 5], but as we see little advantage over (3.34) we have not tried it.

3.3.4 New estimates from $\mathcal{N}$

If $\lambda_{\text{C}} \in \mathcal{N}$, it lies to the left of the rightmost pole of $\pi(\lambda)$ and there is little point in using Taylor series approximations to try to estimate λ_*. Moreover, since we use the failure of the factorization to identify that $\lambda_{\text{C}} \in \mathcal{N}$, we are unable to use the factors to apply inverse iteration to estimate λ_1—even if we had used an indefinite factorization, there would of course be no guarantee that inverse iteration from an arbitrary $-\lambda_{\text{C}} > \lambda_1$ would converge to the desired eigenvalue. However, as Gay [21] points out, an LDL^T or Cholesky factorization will continue so long as the leading sub-matrix of $H + \lambda_{\text{C}} M$ is positive definite, and the factors generated up until this point may be used to improve the upper bound on λ_1. Specifically, if failure first occurs when factorizing the k by k sub-matrix of $H + \lambda_{\text{C}} M$, this sub-matrix may be factorized as $L_k D_k L_k^T$, where L_k is k by k unit lower triangular and $D_k = \text{diag}(d_{i,i})$, $1 \leq i \leq k$, with $d_{k,k} \leq 0$. If y_k satisfies $L_k^T y_k = e_k$ and $w_k = (y_k^T \; 0)^T$, it follows that

$$w_k^T (H + \lambda_{\text{C}} M) w_k = y_k^T L_k D_k L_k^T y_k = d_{k,k} \leq 0,$$

and hence

$$\rho_M(w_k) = \frac{d_{k,k}}{w_k^T M w_k} - \lambda_c$$

provides an upper bound on λ_1 which may be used to improve λ_{L}. In practice, a partial factorization of $H + \lambda_{\mathrm{C}} M$—or more especially the means to find y_k—is not always easy to recover from sophisticated sparse factorization packages; we are currently discussing this need with the authors of `MA57`/`MA27`.

3.3.5 Improving an estimate in $\mathcal{L}$

Once we have found $\lambda_{\mathrm{C}} \in \mathcal{L}$, fast (quadratic) convergence is assured using Newton's method. However, as we discussed in Sect. 3.2.3, there is an opportunity to get even faster convergence using a higher-order Taylor approximation to generate an improvement λ_+. In particular, we know that the approximation $\pi_{2k+1}(\delta) \equiv \pi_{2k+1}(\delta, 2)$ underestimates $\pi(\lambda_{\mathrm{C}} + \delta) \equiv \pi(\lambda_{\mathrm{C}} + \delta; 2)$, and the best prediction $\lambda_{2k+1}(2)$ (see p. 34) computed from the largest root of $\pi_{2k+1}(\delta) = \Delta^2$ will lead to a globally convergent method with asymptotic Q-order $2k+2$ (see Theorem A.8); largest roots from $\pi_{2k+1}(\delta; \beta) = \Delta^\beta$ for other $\beta \in \mathcal{B}_{2k+1}$ may be better. Ideally, the impractical

$$\max_{k \in \mathbb{N}, \beta \in \mathcal{B}_{2k+1}} \lambda_{2k+1}(\beta)$$

would be chosen, but $\lambda_+ = \lambda_{\mathrm{T}}$ for

$$\lambda_{\mathrm{T}} = \max\left(\lambda_1(-1), \lambda_3(2), \lambda_3\left(-\frac{2}{5}\right)\right) \tag{3.35}$$

will give quartic global convergence, which suffices for all practical purposes. Again, in theory the guaranteed value

$$\lambda_{\mathrm{T}} = \max\left(\lambda_1(-1), \lambda_3(2)\right)$$

should be used, but in practice we only use this if (3.35) lies in $\mathcal{G}$. For completeness, we summarize the resultant algorithm as follows:

Algorithm 3.2: Given $\lambda \in \mathcal{L}$, find an estimate of λ_*.

```
Given initial $\lambda \in \mathcal{L}$ and some small $\epsilon > 0$.
Loop:
    Compute
        $\lambda_+ = \max(\lambda_1(-1), \lambda_3(2))$.
    If $|\lambda_+ - \lambda| \le \epsilon$
        exit loop with $\lambda_* \leftarrow \lambda_+$.
    Else
        set $\lambda \leftarrow \lambda_+$.
```

If the cost of multiplications by M and solves with L and L^T are significantly cheaper than factorization of $H + \lambda M$, higher-order roots $\lambda_{2k+1}(\beta)$ for $k > 1$ might be added.

3.3.6 Fast convergence in the hard case

We now consider an inverse-iteration/Rayleigh-quotient-based algorithm for computing approximations to $-\lambda_1$ that are asymptotically greater than $-\lambda_1$. In the easy case, the algorithm generates iterates that will ultimately lie in $\mathcal{L}$, but the main purpose is to cope with the hard-case or *near* hard-case when $|\mathcal{L}|$ may be quite small.

```
Algorithm 3.3: Potential hard case.
Given $\lambda_0^A > -\lambda_1$ and $z_0$ such that $\|z_0\|_M = 1$. Set real constants $0 < \omega_{\mathrm{L}} \leq \omega_{\mathrm{U}}$, $0 < \gamma_{\mathrm{U}} \leq 1$ and
$1 < \gamma_{\mathrm{L}} \leq 2 - \gamma_{\mathrm{U}}$, and integer constant $1 \leq n_{\mathrm{u}} \leq \infty$.
For $k = 0$ until converged
    Choose $1 \leq n_k \leq n_{\mathrm{u}}$.                                  [number of inverse iterations]
    Initialize $w_0 = z_k$.
    For $i = 1 : n_k$                                                          [inverse iteration]
        Set $w_i = (H + \lambda_k^A M)^{-1} M w_{i-1}$ and normalize $w_i \leftarrow w_i / \|w_i\|_M$.
    Set $z_{k+1} = w_{n_k}$ and compute $\rho_M(z_{k+1}) = z_{k+1}^T H z_{k+1}$.      [Rayleigh quotient]
    Choose $\omega_k \in [\omega_{\mathrm{L}}, \omega_{\mathrm{U}}]$ and $\gamma_k \in [\gamma_{\mathrm{L}}, 2n_k - \gamma_{\mathrm{U}}]$.
    Set $\lambda_{k+1}^A = -\rho_M(z_{k+1}) + \omega_k \left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\gamma_k}$.
```

Algorithm 3.3 assumes that an initial estimate λ_0^A is available that satisfies $\lambda_0^A > -\lambda_1$, i.e., $\lambda_0^A \in \mathcal{G}$ region. The algorithm then repeats the following steps until convergence. First, a positive integer n_k is chosen which represents the number of inverse iterations that will be performed. Next, the requested number of inverse iterations are computed using $-\lambda_k^A$ as the fixed estimate of λ_1 and results in a new best approximation z_{k+1} to $_{(-/+)}u_1$—for simplicity, in this discussion we presume that λ_1 has algebraic multiplicity one, although our analysis below does not require this. The second-order Rayleigh-quotient $\rho_M(z_{k+1})$ is then computed to (over-)estimate λ_1. Values $\omega_k \in [\omega_{\mathrm{L}}, \omega_{\mathrm{U}}]$ and $\gamma_k \in [\gamma_{\mathrm{L}}, 2n_k - \gamma_{\mathrm{U}}]$ are now assigned; the restrictions imposed by ω_{L} and $2n_k - \gamma_{\mathrm{U}}$ are required to guarantee (ultimately) that $\lambda_k^A > -\lambda_1$, while the restrictions $\omega_k \leq \omega_{\mathrm{U}}$ and $\gamma_k \geq \gamma_{\mathrm{L}} > 1$ are needed to ensure that the sequence $\{\lambda_k^A\}$ is monotonically decreasing (see Lemma 5). Using these, an improved estimate λ_{k+1}^A of $-\lambda_1$ is computed with the aim of being greater than $-\lambda_1$ and thus in $\mathcal{F}$; this is in contrast to the negative of the Rayleigh-quotient estimate $-\rho_M(z_{k+1})$, which is *always* less than $-\lambda_1$ and thus lies in the $\mathcal{N}$ region. The exact form of the correction is chosen so that it does not interfere with the superlinear convergence of the Rayleigh-quotient iteration, but at the same time is ultimately large enough to ensure that iteration converges from $\mathcal{G}$ in the hard case. This process is then repeated.

Numerous standard results relating to both inverse iteration and the Rayleigh quotient may be found in [12,40,48]; for simplicity, these results typically assume that the eigenvalue for which convergence occurs is simple and that $M = I$. Lemmas 3 and 4 extend two of these results to the generalized eigenvalue problem and they account for the possibility that eigenvalues may not be simple. The analysis that follows may be simplified if we consider the iteration in the scaled variables $\overline{z}_k = Rz_k$ and $\overline{w}_i = Rw_i$, in which case the iteration becomes

```
Initialize $\overline{w}_0 = \overline{z}_k$.
**For** $i = 1 : n_k$
    Set $\overline{w}_i = (\overline{H} + \lambda_k^A I)^{-1}\overline{w}_{i-1}$ and normalize $\overline{w}_i \leftarrow \overline{w}_i / \|\overline{w}_i\|$.
Set $\overline{z}_{k+1} = \overline{w}_{n_k}$ and compute $\rho(\overline{z}_{k+1}) = \overline{z}_{k+1}^T \overline{H}\overline{z}_{k+1}$.
```

Lemma 3 *Let λ_1 be the left-most eigenvalue of the pencil (H, M) with corresponding eigenspace* $\mathrm{eig}(\lambda_1)$. *Then*

$$|\rho_M(x) - \lambda_1| = O(\|x - u\|^2)$$

as $x \to u$ for any $u \in \mathrm{eig}(\lambda_1)$.

Proof The proof follows by applying [48, see p. 204] to the transformed problem in the "bar" variables and then transforming back. □

Lemma 4 *Let λ_1 be the left-most eigenvalue of the pencil (H, M) with corresponding eigenspace* $\mathrm{eig}(\lambda_1)$. *Define $n_1 = \max\{i : \lambda_i = \lambda_1\}$ and assume that gap$(\lambda_1) < \infty$. Suppose that inverse iteration is applied to an initial vector $z_0 = \sum_{i=1}^{n} \alpha_i u_i$ such that $z_0 \not\perp \mathrm{eig}(\lambda_1)$ and with eigenvalue approximation μ that satisfies $|\lambda_1 - \mu| <$ gap$(\lambda_1)/2$. If $\{z_k\}$ denotes the sequence of inverse iterates, then there is a constant Γ for which*

$$|z_k - {}_{(-/+)}u| \le \Gamma \left|\frac{\mu - \lambda_1}{\mu - \lambda_J}\right|^k \quad \textit{and} \quad |\rho_M(z_k) - \lambda_1| \le \Gamma \left|\frac{\mu - \lambda_1}{\mu - \lambda_J}\right|^{2k}$$

for all $k \ge 1$, where λ_J is defined by $|\lambda_J - \lambda_1| =$ gap(λ_1) and

$$u = \frac{\sum_{i=1}^{n_1} \alpha_i u_i}{\|\sum_{i=1}^{n_1} \alpha_i u_i\|_M},$$

and where the ${}_{(-/+)}$ means that for each value of k either the plus or the minus applies.

Here the statement $z_0 \not\perp \mathrm{eig}(\lambda_1)$ should be interpreted to say that $z_0^T M u \neq 0$ for all $u \in \mathrm{eig}(\lambda_1)$.

Proof Essentially, the proof follows from [48, see pp. 204–207]. Specifically, we may apply [48, Thm 27.2]—taking multiple eigenvalues into account—to the transformed problem in "bar" variables and then transform back. □

We note that in the previous lemma, the assumption that $\mathrm{gap}(\lambda_1) < \infty$ was used to make the lemma easier to state. If $\mathrm{gap}(\lambda_1) = \infty$, then λ_1 has multiplicity n and every vector in $\mathbb{R}^n$ is an eigenvector associated with λ_1. Therefore, z_0 is an eigenvector associated with λ_1 and $\rho_M(z_0) = \lambda_1$.

The next lemma gives two important properties of the sequence $\{\lambda_k^A\}$ generated by Algorithm 3.3.

Lemma 5 *Let λ_1 be the left-most eigenvalue of the pencil (H, M). Then the sequence $\{\lambda_k^A\}$ generated by Algorithm 3.3 satisfies*

(i) $\lambda_{k+1}^A < \lambda_k^A$ *and*
(ii) $\lambda_{k+1}^A > -\lambda_1$

for all $k \geq 0$ provided λ_0^A is sufficiently close to $-\lambda_1$.

Proof Suppose that λ_k^A satisfies

$$0 < \Gamma(\lambda_k^A + \lambda_1)^{\gamma_{\mathrm{U}}}/\kappa \leq \omega_{\mathrm{L}}/2, \quad \text{where } \kappa \overset{\text{def}}{=} \min\left(1, \text{gap}(\lambda_1)^{2n_{\mathrm{u}}}\right), \tag{3.36a}$$

$$|\lambda_k^A + \lambda_1| < 1, \quad \text{and} \tag{3.36b}$$

$$\begin{aligned} 0 < \lambda_k^A + \rho_M(z_{k+1}) &= (\lambda_k^A + \lambda_1) + (-\lambda_1 + \rho_M(z_{k+1})) \\ &< \min\left(1, \omega_{\mathrm{U}}^{-1/(\gamma_{\mathrm{L}}-1)}\right). \end{aligned} \tag{3.36c}$$

Using condition (3.36c), the inequalities $\omega_k \leq \omega_{\mathrm{U}}$ and $\gamma_k \geq \gamma_{\mathrm{L}}$, and the definition of λ_{k+1}^A, we may write

$$\begin{aligned} \lambda_{k+1}^A &= -\rho_M(z_{k+1}) + \omega_k \left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\gamma_k} \\ &\leq -\rho_M(z_{k+1}) + \omega_{\mathrm{U}} \left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\gamma_{\mathrm{L}}} \\ &< -\rho_M(z_{k+1}) + \lambda_k^A + \rho_M(z_{k+1}) = \lambda_k^A, \end{aligned}$$

so that part (i) holds for all λ_k^A satisfying (3.36). Moreover, we have

$$\begin{aligned} \lambda_{k+1}^A + \lambda_1 &= -\rho_M(z_{k+1}) + \lambda_1 + \omega_k \left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\gamma_k} && (3.37) \\ &\geq -\Gamma\left(\frac{\lambda_k^A + \lambda_1}{\text{gap}(\lambda_1)}\right)^{2n_k} + \omega_k(\lambda_k^A + \lambda_1)^{\gamma_k} && (3.38) \\ &= (\lambda_k^A + \lambda_1)^{\gamma_k}\left(\omega_k - \frac{\Gamma(\lambda_k^A + \lambda_1)^{2n_k-\gamma_k}}{\text{gap}(\lambda_1)^{2n_k}}\right) && (3.39) \\ &\geq (\lambda_k^A + \lambda_1)^{\gamma_k}\left(\omega_{\mathrm{L}} - \frac{\Gamma(\lambda_k^A + \lambda_1)^{\gamma_{\mathrm{U}}}}{\kappa}\right) && (3.40) \\ &\geq \frac{1}{2}\omega_{\mathrm{L}}(\lambda_k^A + \lambda_1)^{\gamma_k} > 0. && (3.41) \end{aligned}$$

Equation (3.37) follows from the definition of λ_{k+1}^A, while inequality (3.38) follows from Lemma 4 and the inequality $\lambda_k^A + \rho_M(z_{k+1}) \geq \lambda_k^A + \lambda_1$. The relationships (3.39)–(3.41) follow from factorization, the restrictions that ω_{L} and γ_{U} place on ω_0 and γ_0, and (3.36a)–(3.36b). Therefore, part (ii) holds for all λ_k^A satisfying (3.36).

Finally, we show by induction that λ_k^A satisfies (3.36) for all $k \geq 0$. First, note that Lemma 4 guarantees that (3.36) will hold for λ_0^A by choosing λ_0^A sufficiently close to $-\lambda_1$. Now suppose that λ_k^A satisfies (3.36). In then follows from (i) and (ii) that λ_{k+1}^A will also satisfy (3.36). This completes the induction step. □

Theorem 7 *Let λ_1 be the left-most eigenvalue of the pencil (H, M). Suppose that there exists positive constants $\bar{\gamma}$ and $\bar{n}$ such that $\gamma_k = \bar{\gamma}$ and $n_k = \bar{n}$ for all k sufficiently large. Then the sequence $\{\lambda_k^A\}$ generated by Algorithm 3.3 converges to $-\lambda_1$ with Q-rate equal to $\bar{\gamma}$, provided λ_0^A is chosen sufficiently close to $-\lambda_1$.*

Proof For k sufficiently large and λ_0^A sufficiently close to $-\lambda_1$, we have

$$
\begin{aligned}
|\lambda_{k+1}^A + \lambda_1| &= |-\rho_M(z_{k+1}) + \lambda_1 + \omega_k \left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\bar{\gamma}}| && (3.42)\\
&\le |-\rho_M(z_{k+1}) + \lambda_1| + \omega_k |\lambda_k^A + \rho_M(z_{k+1})|^{\bar{\gamma}} && (3.43)\\
&\le |-\rho_M(z_{k+1}) + \lambda_1| + 2^{\bar{\gamma}} \omega_k |\lambda_k^A + \lambda_1|^{\bar{\gamma}} && (3.44)\\
&\le \Gamma \left|\frac{\lambda_k^A + \lambda_1}{\text{gap}(\lambda_1)}\right|^{2\bar{n}} + 2^{\bar{\gamma}} \omega_{\text{U}} |\lambda_k^A + \lambda_1|^{\bar{\gamma}} && (3.45)\\
&= \left[\frac{\Gamma(\lambda_k^A + \lambda_1)^{2\bar{n}-\bar{\gamma}}}{\text{gap}(\lambda_1)^{2\bar{n}}} + 2^{\bar{\gamma}} \omega_{\text{U}}\right] |\lambda_k^A + \lambda_1|^{\bar{\gamma}} && (3.46)\\
&\le \left[\frac{\Gamma(\lambda_0^A + \lambda_1)^{\gamma_{\text{U}}}}{\text{gap}(\lambda_1)^{2\bar{n}}} + 2^{\bar{\gamma}} \omega_{\text{U}}\right] |\lambda_k^A + \lambda_1|^{\bar{\gamma}} = c|\lambda_k^A + \lambda_1|^{\bar{\gamma}}, && (3.47)
\end{aligned}
$$

where

$$
c = \frac{\Gamma(\lambda_0^A + \lambda_1)^{\gamma_{\text{U}}}}{\text{gap}(\lambda_1)^{2\bar{n}}} + 2^{\bar{\gamma}} \omega_{\text{U}} \quad \text{and } \Gamma \text{ was defined in the statement of Lemma 4.}
$$

Equations (3.42) and (3.43) follow from the definition of λ_{k+1}^A and the triangle inequality. Equation (3.44) follows from the inequality $\lambda_k^A + \rho_M(z_{k+1}) \le 2(\lambda_k^A + \lambda_1)$, which follows from Lemma 4 for λ_0^A sufficiently close to $-\lambda_1$. Equations (3.45) and (3.46) follow from Lemma 4, the definition of gap(λ_1), the inequality $\omega_k \le \omega_{\text{U}}$, and factorization. Finally, Eq. (3.47) follows from the properties of $\{\lambda_k^A\}$ described in Lemma 5 and definition of γ_k. □

This theorem essentially says that we can obtain any Q-order convergence we wish at the expense of performing an ever increasing number of inverse iterations. More precisely, we can obtain the Q-convergence "goal" $\bar{\gamma}$ by setting $\gamma_k = \bar{\gamma}$ and consequently choosing n_k, the number of inverse iterations performed, to satisfy $2n_k - \gamma_{\text{U}} > \bar{\gamma}$ (this should be done for all k sufficiently large). For example, we could obtain superlinear convergence by ultimately setting $\gamma_k = 1.5$ and by performing a single inverse iteration ($n_k = 1$), or we could obtain super-cubic-convergence by ultimately setting $\gamma_k = 3.5$ and by performing two steps of inverse iteration ($n_k = 2$).

A reasonable implementation would be to use Algorithm 3.3 once $\lambda_{\text{U}} - \lambda_{\text{L}}$ is relatively small. In the easy case, Algorithm 3.3 will quickly produce an iterate that lands in the $\mathcal{L}$-region, while in the hard-case the algorithm converges rapidly to $-\lambda_1 = \lambda_s$. We also note that the algorithm produces iterates z_k that approximate the eigenspace associated with λ_1, which is required for computing a solution to problem (1.1) in the hard-case. Initially, using a single step of inverse iteration with $\gamma_k \approx 1.5$ is reasonable. In general, this will force λ_+ into $\mathcal{L}$ rapidly, with subsequent fast convergence as described in Sect. 3.3.5. If the hard-case is suspected, it may be wise to increase γ_k from 1.5 to 3 and to perform two steps of inverse iteration, but only after the value $\gamma_k = 1.5$ has been "successful"; the resultant cubic-convergence seems sufficient for all practical purposes.

Formally, if $\lambda_{\text{C}} \in \mathcal{G}$, we compute a new estimate λ_+ as follows. Let λ_{T} be the Taylor series under-estimate (3.32), let λ_{E} be the closest upper bound of λ_1 found so far, let $\bar{\gamma}$ be the desired order of convergence in the hard case, and let θ_{G} and θ_{H} be given constants in (0, 1) and $\omega_{\text{L}} > 0$—for example, in practice we use $\theta_{\text{G}} = 0.5 = \theta_{\text{H}}$ and $\omega_{\text{L}} = 1$. Then whenever $|\lambda_{\text{C}} + \lambda_{\text{E}}| \leq \theta_{\text{H}}\lambda_{\text{C}}$, we suspect we might be in the hard case, so use the Rayleigh quotient $\rho_M(z_{k+1})$ to try to improve λ_{E} and λ_{L} by assigning $\lambda_{\text{E}} \leftarrow \min(\lambda_{\text{E}}, \rho_M(z_{k+1}))$ and $\lambda_{\text{L}} \leftarrow \max(\lambda_{\text{L}}, -\lambda_{\text{E}})$, and subsequently set

$$\lambda_+ = \min\left(\lambda_{\text{L}} + \theta_{\text{G}}(\lambda_{\text{C}} - \lambda_{\text{L}}), \max\left(\lambda_{\text{T}}, \lambda_{\text{L}}, -\lambda_{\text{E}} + \omega_{\text{L}}\left(\frac{\lambda_{\text{C}} + \lambda_{\text{E}}}{\lambda_{\text{C}}}\right)^{\bar{\gamma}}\right)\right) \tag{3.48}$$

so as to mimic Algorithm 3.3. Otherwise, if $\lambda_{\text{T}} \geq \lambda_{\text{L}}$, the Taylor estimate gives the best estimate in $\mathcal{F}$ found so far, and we assign

$$\lambda_+ = \lambda_{\text{T}}. \tag{3.49}$$

If neither happens, we simply revert to improving the interval of uncertainty by setting

$$\lambda_+ = \lambda_{\text{L}} + \theta_{\text{G}}(\lambda_{\text{C}} - \lambda_{\text{L}}). \tag{3.50}$$

3.3.7 Interior solution, sequences of related problems and initial values

The one remaining issue is when, if at all, to test for the possibility that the solution to (1.1) lies interior to the trust region, and thus that the required $\lambda_* = 0$. Clearly this is impossible if $\lambda_{\text{L}} > 0$, so any investigation should be delayed until the initial λ_{L} has been computed [38].

In a trust-region context, a sequence of problems of the form (1.1) will be solved. There will generally be two possibilities. In the first, the data H and c will be unchanged, but Δ will have been reduced to Δ_+. Define the usual sets $\mathcal{G}$, $\mathcal{L}$ and $\mathcal{N}$ with respect to λ, and let $\mathcal{G}_+$, $\mathcal{L}_+$ and $\mathcal{N}_+$ be their analogs with respect to λ_+. In this case, if the previous $\lambda_- \in \mathcal{L} \cup \{\lambda_{\text{S}}\}$, then $\lambda_- \in \mathcal{L}_+$ and is a good starting point for the new problem. Potentially better, λ may have been sampled at points $\lambda_+ > \lambda_-$ when solving the previous problem, and corresponding values of $\pi(\lambda_+)$ will be known. Thus finding the largest previous λ_+ for which $\pi(\lambda_+) \geq \Delta^+$ will also give a value in $\mathcal{L}_+$. The other possibility is that H and c might have changed but $\Delta_+ \geq \Delta$. Little useful information is then available, but as a heuristic starting from λ_- is a possibility; if small changes to H and c have occurred, it is likely that $\lambda_- \in \mathcal{G}_+$.

In the absence of better initial information, we simply choose the initial λ as 0 if $\lambda_{\text{L}} = 0$, and as

$$\lambda_{\text{C}} = \max\left(\gamma\sqrt{\lambda_{\text{L}}\lambda_{\text{U}}}, \lambda_{\text{L}} + \theta(\lambda_{\text{U}} - \lambda_{\text{L}})\right) \tag{3.51}$$

for some $\theta \in (0, 1)$ and $\gamma \in \{0, 1\}$ (c.f., Algorithm 3.1) otherwise.

3.3.8 Summary

In summary, our complete algorithm is as follows. First choose λ_{L} and λ_{U} as Sect. 3.3.2. If $\lambda_{\text{L}} = 0$ choose the initial estimate $\lambda_{\text{C}} = 0$, but otherwise use (3.51). Now use Algorithm 3.1 to try to find a $\lambda \in \mathcal{L}$, choosing the new estimate when $\lambda_{\text{C}} \in \mathcal{G}$ by projecting λ_+ from (3.48)–(3.50) as appropriate into the interval (3.27), and finding approximations to λ_{E} as mentioned in Sect. 3.3.3 and 3.3.4. If Algorithm 3.1 succeeds, use the resulting λ as input to Algorithm 3.2 to find an approximation to λ_*. If by contrast, $\lambda_{\text{U}} - \lambda_{\text{L}} < \epsilon$ for some specified small $\epsilon > 0$ in Algorithm 3.1, terminate with the approximate hard-case estimate $\lambda_* = \frac{1}{2}(\lambda_l + \lambda_u)$.

3.3.9 Early termination

While our current implementation iterates to find a highly-accurate solution, there is no reason why a lower-accuracy estimate along the lines of those proposed by Moré and Sorensen [38] or Gertz and Gill [22, Section 5] should not be produced. Gertz and Gill [22, Lem. 5.2], based on [38, Lem. 3.4], show that any $x = x(\lambda) + u$ for which $\|x(\lambda) + u\|_M \leq (1 + \eta)\Delta$ and

$$u^T(H + \lambda M)u \leq \eta(2 - \eta)\left[x^T(\lambda)(H + \lambda M)x(\lambda) + \lambda\Delta^2\right] \tag{3.52}$$

for some (possibly zero) u and (small) $\eta \in (0, 1)$ gives a suitable approximate solution to (1.1). These rules may be applied to terminate our algorithm: we reduce the interval $[\lambda_{\text{L}}, \lambda_{\text{U}}]$ until either $\|x(\lambda)\|_M \leq (1+\eta)\Delta$ (in which case $u = 0$ suffices), or ultimately, if the hard-case occurs, the eigenvector estimates $u = z_k$ generated by Algorithm 3.3 will satisfy (3.52).

4 Software and numerical experiments

The ideas developed in this paper have been implemented as a pair of thread-safe Fortran 95 packages—respectively, TRS and RQS for problems (1.1) and (1.2)—as part of version 2.3 of the GALAHAD optimization library[3] [27]. The packages provide a number of options. The matrix H (and optionally M) may be given in a variety of sparse and dense matrix formats. The highest degree of the Taylor polynomials used may be specified (up to three), as may the number of inverse iterations performed. Iterative refinement may be used when solving (2.2), and this is particularly important in the "hard" or "nearly hard" cases since then (2.2) may be very ill conditioned. Any a priori knowledge of initial λ_{L}, λ_{U} and λ may optionally be provided, and this

[3] Available from http://galahad.rl.ac.uk/galahad-www/.

proves useful when a sequence of problems is solved. Finally, there is an option to replace the trust region constraint in TRS by the equation $\|x\|_M = \Delta$ since there is currently much interest in solving optimization problems on Riemannian manifolds including the hyper-ellipsoid [1,2]; in this case there is no longer the requirement that λ be positive, merely that $\lambda \geq -\lambda_1$, and the algorithm is adapted in the obvious way. Currently the possible improvement when $\lambda_{\mathrm{C}} \in \mathcal{N}$ mentioned in Sect. 3.3.4 has not been implemented, as we await the necessary features from the sparse factorization packages we are using.

As a comparison, we use the MINPACK-2 package[4] dgqt which is an implementation of the Moré-Sorensen [38] approach; we slightly modified this software to record and print required details, and to allow consistent stopping rules (namely, (4.1) and (4.2) below).

By way of a simple example, consider the data

$$H = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 2 & 0 \\ 4 & 0 & 3 \end{pmatrix}, \quad c_1 = \begin{pmatrix} 5 \\ 0 \\ 4 \end{pmatrix}, \quad c_2 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, \quad \text{and} \quad c_3 = \begin{pmatrix} 0 \\ 2 \\ 0.0001 \end{pmatrix}$$

with $M = I$ and $\Delta = 1$. If we pick $c = c_1$, the resulting problem (1.1) is an example of the "easy case". By contrast, $c = c_2$ gives rise to the "hard case", and $c = c_3$ is the "nearly hard case". By default, TRS picks its initial value of λ automatically as described in Sect. 3.3.7. Running TRS when $c = c_1$ and stopping as soon as

$$|\|x(\lambda)\| - \Delta| < 10^{-12} \max(1, \Delta) \tag{4.1}$$

gives

```
   it       lambda_l               lambda               lambda_u
G   1 2.123105625617661E+00 4.468089744720383E+00 4.468089744720383E+00
   it     ||x||-radius             lambda               d_lambda
L   2 3.479156233026082E-04 3.999056146822190E+00 0.000000000000000E+00
L   3 9.769962616701378E-15 3.999999999999973E+00 9.438531777834491E-04
Normal stopping criteria satisfied
3 factorizations. Solution and Lagrange multiplier = -4.5000E+00  4.0000E+00
```

Here the characters G and L indicate that the current value of λ lies in the $\mathcal{G}$ and $\mathcal{L}$ regions, respectively, while lambda_l, lambda_u, ||x||-radius and d_lambda are, respectively, the current λ_{L} and λ_{U}, the residual $|\|x(\lambda)\| - \Delta|$ and the change in λ. By contrast dgqt (started with the same initial λ) yields

```
   it       lambda_l               lambda               lambda_u
G   1 0.000000000000000E+00 4.468089744720383E+00 4.468089744720383E+00
   it     ||x||-radius             lambda               d_lambda
L   2 1.398428589050060E-02 3.962817739881419E+00 3.693192836831821E-02
L   3 9.224016079900643E-05 3.999749668249738E+00 2.503204418455550E-04
L   4 4.166259115478965E-09 3.999999988691583E+00 1.130841750579190E-08
L   5 0.000000000000000E+00 4.000000000000001E+00 0.000000000000000E+00
5 factorizations. Solution and Lagrange multiplier = -4.5000E+00  4.0000E+00
```

[4] Available from ftp://info.mcs.anl.gov/pub/MINPACK-2/gqt/.

Notice how using the higher (third)-order Taylor model improves the ultimate rate of convergence; this is both typical in practice in the "easy case", and to be expected; as confirmation, when TRS is run with just the traditional (first-order) Taylor model, it too requires 5 factorizations.

Running TRS when $c = c_2$ and stopping as soon as

$$\lambda_{\text{U}} - \lambda_{\text{L}} < 10^{-12} \max(1, \lambda_{\text{U}}) \tag{4.2}$$

gives

```
   it       lambda_l                lambda               lambda_u
G   1 2.123105625617661E+00 3.258147959821393E+00 3.258147959821393E+00
G   2 2.123105625617661E+00 2.328723983198157E+00 2.328723983198157E+00
G   3 2.123105625617661E+00 2.123310177530242E+00 2.123310177530242E+00
G   4 2.123105625617661E+00 2.123105625617669E+00 2.123105625617669E+00
G   5 2.123105625617661E+00 2.123105625617661E+00 2.123105625617669E+00
Hard-case stopping criteria satisfied. Interval width = 8.8818E-15
4 factorizations. Solution and Lagrange multiplier = -1.5466E+00  2.1231E+00
```

By contrast dgqt (again started with the same initial λ) yields

```
   it       lambda_l                lambda               lambda_u
G   1 0.000000000000000E+00 3.258147959821393E+00 3.258147959821393E+00
G   2 2.026074553757914E+00 2.569289916255538E+00 2.569289916255538E+00
G   3 2.102919568297307E+00 2.324437575312084E+00 2.324437575312084E+00
.  .. ..................... ..................... .....................
G  36 2.123105625617661E+00 2.123105625639012E+00 2.123105625639012E+00
G  37 2.123105625617661E+00 2.123105625628336E+00 2.123105625628336E+00
G  38 2.123105625617661E+00 2.123105625622999E+00 2.123105625622999E+00
38 factorizations. Solution and Lagrange multiplier = -1.5466E+00  2.1231E+00
```

Notice now how the inverse Rayleigh-quotient iteration and improved lower bound on λ obtained using the higher (second- and third)-order Taylor models dramatically improves convergence; in the absence of better lower bounds dqdt essentially reverts to bisection to ensure convergence. Once again TRS is superlinearly convergent, and the performance indicated is typical in practice in the "hard case".

Finally, running TRS when $c = c_3$ and stopping as for the previous case, we obtain

```
   it       lambda_l                lambda               lambda_u
G   1 2.123105625617661E+00 3.258147960635930E+00 3.258147960635930E+00
G   2 2.123105625617661E+00 2.328723983342385E+00 2.328723983342385E+00
G   3 2.123105625617661E+00 2.123310177530700E+00 2.123310177530700E+00
   it      ||x||-radius             lambda               d_lambda
L   4 2.275299065674614E-01 2.123160201503088E+00 0.000000000000000E+00
L   5 5.309901864716249E-04 2.123175951499089E+00 1.574999600073568E-05
G   6 2.611244553918368E-13 2.123176000326642E+00 4.882755266777394E-08
Normal stopping criteria satisfied
6 factorizations. Solution and Lagrange multiplier = -1.5467E+00  2.1232E+00
```

By comparison, for dgqt, we find

```
   it       lambda_l                lambda               lambda_u
G   1 0.000000000000000E+00 3.258147960635930E+00 3.258147960635930E+00
G   2 2.026074553651017E+00 2.569289916508920E+00 2.569289916508920E+00
G   3 2.102919568276703E+00 2.324437575415315E+00 2.324437575415315E+00
.  .. ..................... ..................... .....................
G  12 2.123105561462512E+00 2.123463910456750E+00 2.123463910456750E+00
G  13 2.123105609581684E+00 2.123284752461381E+00 2.123284752461381E+00
```

```
G  14 2.123105621609018E+00 2.123195185146074E+00 2.123195185146074E+00
   it     ||x||-radius              lambda                d_lambda
L  15 2.402191047985047E-02 2.123173864436229E+00 2.113422099896687E-06
L  16 2.442362429893041E-04 2.123175977858329E+00 2.246578161241124E-08
L  17 2.750231753445576E-08 2.123176000324110E+00 2.530956130439299E-12
L  18 3.371303236576750E-12 2.123176000326641E+00 3.102986617653514E-16
   it       lambda_l                lambda                lambda_u
G  19 2.123176000326641E+00 2.123176000326642E+00 2.123176000326642E+00
19 factorizations. Solution and Lagrange multiplier = -1.5467E+00  2.1232E+00
```

Here TRS immediately refines the lower bound on the interval of uncertainty to obtain a λ in $\mathcal{L}$, and thereafter converges rapidly to the required root. By contrast, the linear Taylor model used by dgqt is less able to find a good λ_L, and this results in a number of essentially bisection steps until $\mathcal{L}$ is reached. This again is indicative of the behaviour of the methods in practice in the "nearly hard case". As a further comparison, when TRS is restricted to first- and second-order Taylor models, it requires 9 and 8 factorizations, respectively.

We should be cautious not to infer too much from these examples, particularly as dgqt was originally designed to terminate fast with a low-accuracy but usable solution. However, they do illustrate well the new design features we have added.

To see more generally the effect of improved convergence in both easy and hard cases, we consider all the unconstrained problems contained in the CUTEr [26] test set; we restrict our attention to those problems involving 2000 or fewer variables, since the dense Cholesky factorization used by dgqt struggles with larger cases, and this leads to 97 examples. We construct instances of (1.1) by setting $c = \nabla_x f(x_0)$ and $H = \nabla_{xx} f(x_0)$ for the given objective function $f(x)$ and starting point x_0; a spherical trust-region of radius 1 is used. We provided the same initial "guess" $\lambda = 0$ for both packages.

In Table 1 we report the number of factorizations required by TRS and dgqt on each problem; the algorithms terminate as soon as either (4.1) or (4.2) occurs. We also provide a graphical interpretation of this data using performance profiles of the factorization counts in Fig. 5; briefly, given a set of test problems and a set of competing algorithms, the i-th performance profiles $p_i(\alpha)$ indicates the fraction of problems for which the i-th algorithm is within a factor α of the best for a given metric—see [13] for a formal definition of performance profiles and a discussion of their properties.

Both the detailed and summary results indicate the improvements offered by the enhancements discussed in this paper. In most cases, the number of factorizations falls, and for those cases where dgqt requires fewer factorizations, TRS is usually not significantly worse. The average number of factorizations required over all of these examples by TRS is 3.7 compared to 4.7 for dgqt. The worst performance is for problem GROWTHLS, and in detail we see the following for TRS:

```
   it       lambda_l                lambda                lambda_u
N   1 0.000000000000000E+00 0.000000000000000E+00 6.516158914938158E+06
G   2 0.000000000000000E+00 3.258079457469079E+06 3.258079457469079E+06
G   3 0.000000000000000E+00 3.258079457469079E+04 3.258079457469079E+04
G   4 0.000000000000000E+00 3.258079457469079E+02 3.258079457469079E+02
N   5 0.000000000000000E+00 3.258079457469079E+00 3.258079457469079E+02
G   6 3.258079457469079E+00 3.551306608641297E+01 3.551306608641297E+01
N   7 3.258079457469079E+00 1.075659756093366E+01 3.551306608641297E+01
```

Table 1 The numbers of factorizations required to solve the sample set of CUTEr problems using dgqt compared to those using TRS

Problem	dgqt	TRS	Problem	dgqt	TRS	Problem	dgqt	TRS
3PK	8	7	EXTROSNB	2	3	PALMER5C	3	3
AKIVA	1	1	FLETCHCR	4	3	PALMER6C	4	3
ALLINITU	6	5	FMINSURF	6	5	PALMER7C	4	3
ARGLINA	1	1	GENROSE	6	4	PALMER8C	10	3
ARGLINB	2	2	GENROSEB	6	4	PARKCH	9	8
ARGLINC	2	2	GROWTHLS	14	14	PENALTY1	2	2
BARD	6	5	GULF	6	3	PENALTY2	4	3
BEALE	7	3	HAIRY	6	4	PENALTY3	2	2
BIGGS6	7	8	HATFLDD	5	4	PFIT1LS	8	3
BOX3	6	4	HATFLDE	5	4	PFIT2LS	7	3
BRKMCC	1	1	HEART6LS	11	4	PFIT3LS	7	3
BROWNAL	2	2	HEART8LS	5	4	PFIT4LS	7	3
BROWNBS	1	1	HELIX	7	4	POWELLSG	7	5
BROWNDEN	4	3	HIELOW	7	9	POWER	3	3
CHNROSNB	5	4	HIMMELBB	8	6	ROSENBR	1	1
CLIFF	2	3	HIMMELBF	8	4	S308	1	1
CUBE	5	4	HIMMELBG	6	3	SENSORS	4	3
DECONVU	6	4	HIMMELBH	4	3	SINEVAL	2	2
DENSCHNA	1	1	HUMPS	4	3	SISSER	1	1
DENSCHNB	7	5	HYDC20LS	6	4	SNAIL	4	3
DENSCHNC	1	1	JENSMP	1	1	SPARSINE	3	3
DENSCHND	4	3	KOWOSB	7	6	SPARSQUR	4	3
DENSCHNE	4	4	LOGHAIRY	5	8	STRATEC	9	8
DENSCHNF	1	1	MANCINO	3	2	STREG	2	2
DJTL	4	3	MEXHAT	1	1	TOINTGOR	4	3
EDENSCH	2	2	MEYER3	4	4	TOINTPSP	3	2
EG2	1	1	MSQRTALS	5	4	VARDIM	2	4
EIGENALS	4	4	MSQRTBLS	5	4	VAREIGVL	5	4
EIGENBLS	5	3	NONCVXU2	2	2	VIBRBEAM	15	13
EIGENCLS	5	4	NONCVXUN	2	2	WATSON	7	6
ENGVAL2	6	4	OSBORNEA	6	7	WOODS	4	3
ERRINROS	8	7	OSBORNEB	8	10	YFITU	8	5
EXPFIT	6	3						

```
G   8 1.075659756093366E+01 2.313483182367331E+01 2.313483182367331E+01
N   9 1.869105479497000E+01 1.869105479497000E+01 2.313483182367331E+01
N  10 1.869105479497000E+01 1.913543249784033E+01 2.313483182367331E+01
G  11 1.913543249784033E+01 2.113513216075682E+01 2.113513216075682E+01
   it     ||x||-radius             lambda                d_lambda
L  12 8.884499521913303E-02 2.054711382755286E+01 0.000000000000000E+00
L  13 1.199973391163844E-05 2.058132199997437E+01 3.420817242151486E-02
L  14 1.088906742552354E-12 2.058132716354694E+01 5.163572570410224E-06
Normal stopping criteria satisfied
```

Here, the character N records that the current value of λ lies in the $\mathcal{N}$ region. Observe that the initial interval $[\lambda_L, \lambda_U]$ is large, the first few iterations refine estimates from $\mathcal{G}$ which eventually underestimate λ_*. This leads to a cycle to and from $\mathcal{N}$ and eventually to $\mathcal{L}$ from whence fast convergence occurs. It is difficult to imagine how this might be improved in general, and so we feel reassured that TRS behaves as well as might be expected.

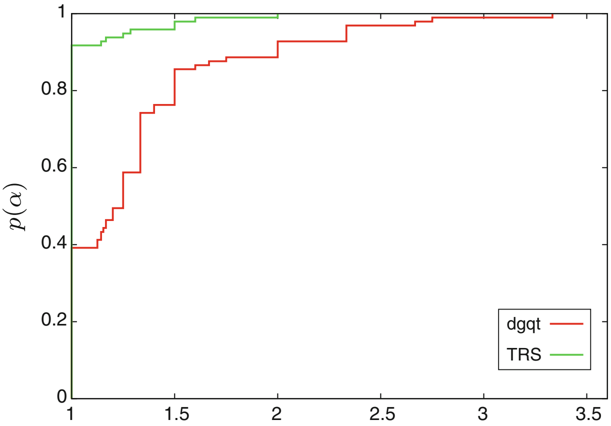

Fig. 5 Performance profile for the numbers of factorizations required to solve the sample set of CUTEr problems using `dgqt` compared to those using `TRS`

Table 2 The numbers of factorizations and the CPU time (in seconds) required to solve the CUTEr problem `BOX` in the trust-region (`TRS`) and cubic regularisation (`RQS`) cases, as the dimension n increases

n	`TRS`		`RQS`	
	Factorizations	CPU	Factorizations	CPU
1000	3	0.00	3	0.00
3162	3	0.02	3	0.02
10000	3	0.14	3	0.13
31622	2	1.04	3	1.04
100000	3	0.39	3	0.29
316228	2	1.03	3	1.01
1000000	3	4.04	2	2.56
3162278	3	12.92	2	8.10
10000000	1	24.58	2	28.82

Since both `TRS` and `RQS` are designed to cope with large problems, we illustrate their performance on bigger problems from the CUTEr test set. In Table 2, we consider the problem `BOX` as we increase its dimension n from a thousand to ten million; the Hessian has non-zeros along the diagonal, and in positions $(1, i)$, $(i, 1)$, (n, i), (i, n), $(n/2, i)$ and $(i, n/2)$, for all $1 \leq i \leq n$. Here and elsewhere the experiments were performed on a single CPU of a Dell Precision T3400, single Core2 Quad Q9550 processor(2.83 GHz, 1333MHz FSB, 12MB L2 Cache) with 4GB RAM; the code is in double precision and compiled with the g95 compiler using default (-O) optimization. For `TRS` we use the radius $\Delta = 1$, while for `RQS`, cubic ($p = 3$) regularisation with a weight $\sigma = 10$ is used.

The dominant cost here, as might be expected, is for the ordering (particularly for the larger examples) and factorization of $H + \lambda I$, although for the largest problem the cost of the Rayleigh-quotient iteration also starts to become significant.

Table 3 The numbers of factorizations ("facts") and the CPU time (in seconds) required to solve a variety of large CUTEr problems in the trust-region (TRS) and cubic regularisation (RQS) cases

Problem	n	nnz (H)	nnz (fact)	TRS		RQS	
				Facts	CPU	Facts	CPU
SCURLY10	100000	1099945	1849840	14	2.41	12	1.40
SCOSINE	100000	199999	949984	20	1.34	18	0.91
NONCVXUN	100000	399984	11946824	2	287.17	2	299.98
INDEF	100000	299997	1049968	5	0.65	2	0.31
FLETCBV2	100000	199999	949984	4	0.61	3	0.31
DIXMAANA	90000	269999	959911	3	0.42	3	0.38
FMINSRF2	90000	448202	6317096	5	15.05	4	12.07

The number of nonzeros in H ("nnz(H)") and its factors ("nnz(fact)") are also given

In Table 3, we illustrate the behaviour on other large CUTEr examples. Although the actual behaviour clearly depends on sparsity, and particularly on fill-in—the problem NONCVXUN is an example which fills in significantly during factorization—the main message is that both TRS and RQS are capable of solving large problems, and thus often provide good alternatives to iterative methods. We leave a more general comparison between direct and iterative approaches for solving the subproblems to follow-up work, in which we plan to investigate such subproblems in the context of general methods for unconstrained optimization.

5 Comments and conclusions

Our aim has been to revisit the popular Gay-Moré-Sorensen [21,38] algorithm(s) for the direct solution of the trust-region subproblem and to provide flexible modern software for this and the related regularized quadratic subproblem. We have provided enhancements so that the method is both globally and superlinearly convergent in all ("easy" and "hard") cases. The resulting software is freely available as the packages TRS and RQS as part of the GALAHAD optimization library[27].

Our next goal will be to investigate the use of these subproblem solvers as part of general trust-region/regularisation methods for unconstrained and constrained optimization methods. Of particular importance here is whether it pays off to solve the subproblems more accurately than is currently done, and whether these methods are competitive with iterative methods [7,18,19,25,31,45,46] for large problems. We are encouraged here as the sparse-matrix factorization technology has advanced rapidly of late, and both parallel/multi-core and out-of-core factorizations are now available and capable of coping with matrices of high (in the millions) order [3,34,35,42,44].

Some iterative methods [7,25] for the solution of (1.1) and (1.2) solve sequences of problems of the same form, albeit now with simpler tridiagonal matrices H. Clearly the improvements suggested in Sect. 3 are equally appropriate in this case. We plan to update the relevant GALAHAD packages GLTR and GLRT to take account of this.

Problems involving linear equality constraints may be dealt with in essentially the same way. For example, if we add the restrictions $Ax = 0$ to (1.1) or (1.2), the essential difference is that the required $x(\lambda)$, together with some auxiliary $y(\lambda)$, satisfies

$$\begin{pmatrix} H + \lambda M & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x(\lambda) \\ y(\lambda) \end{pmatrix} = - \begin{pmatrix} c \\ 0 \end{pmatrix} \tag{5.1}$$

rather than (2.2). Thus rather than assessing whether a given $\lambda \in \mathcal{F}$ by the success of the Cholesky factorization of $H + \lambda M$ as we do in the unconstrained case, here instead we note that $\lambda \in \mathcal{F}$ if and only if the leading block of the matrix

$$\begin{pmatrix} H + \lambda M & A^T \\ A & 0 \end{pmatrix} \tag{5.2}$$

is positive definite in the null-space of A, or equivalently that (5.2) is non-singular and has precisely rank(A) negative eigenvalues [9,23]. To verify the latter condition and then to solve (5.1), any inertia-revealing symmetric, indefinite factorization package is appropriate (see Sect. 3.1), although now numerical pivoting will be required for stability, and thus non-static data structures for the factors may be required. All other aspects are essentially as for the unconstrained cases covered in Sect. 3.3, although non-trivial initial values (*cf.* Sect. 3.3.2) for λ_{L} and λ_{U} are not obvious.

Acknowledgments This work was supported by the EPSRC grants EP/E053351/1 and EP/F005369/1. We are extremely grateful for the helpful comments made by three referees and an associate editor on our original manuscript.

Appendix A

Let $\phi \in C^{m+1}, \theta \in C^1 : \mathbb{R} \to \mathbb{R}$, and suppose we wish to find a root of the scalar equation

$$\phi(\lambda) = \theta(\lambda). \tag{A.1}$$

Now suppose that a given λ_k is closest to the simple root λ_*, of (A.1)—the root is *simple* if $\theta^{(1)}(\lambda_*) - \phi^{(1)}(\lambda_*) \neq 0$—and let

$$\phi_m(\delta; \lambda_k) \stackrel{\text{def}}{=} \sum_{i=0}^{m} \frac{\phi^{(i)}(\lambda_k)}{i!} \delta^i$$

be the m-th order Taylor approximation to $\phi(\lambda_k + \delta)$. To improve on λ_k, we compute the root δ_k of smallest magnitude to the approximating equation $\phi_m(\delta; \lambda_k) = \theta(\lambda_k + \delta)$, update $\lambda_{k+1} = \lambda_k + \delta_k$, increment k by 1, and repeat.

Theorem A.8 *Suppose that $\phi \in C^{m+1}, \theta \in C^1 : \mathbb{R} \to \mathbb{R}$, that λ_* is a simple root of $\phi(\lambda) = \theta(\lambda)$. Then for all λ_k sufficiently close to λ_*,*

$$|\lambda_k + \delta_k - \lambda_*| = O(|\lambda_k - \lambda_*|^{m+1}),$$

where δ_k is the root of smallest magnitude of $\phi_m(\delta;\lambda_k)=\theta(\lambda_k+\delta)$ and $\phi_m(\delta;\lambda_k)$ is the m-th order Taylor approximation to $\phi(\lambda_k+\delta)$.

Proof We first show that δ_k is small when λ_k is close to λ_*. Define the function

$$F(\delta,\lambda)=\sum_{i=0}^{m}\frac{\phi^{(i)}(\lambda)}{i!}\delta^i-\theta(\lambda+\delta).$$

From the assumptions of this theorem and the fact that

$$F(0,\lambda_*)=\phi(\lambda_*)-\theta(\lambda_*)=0 \quad\text{and}\quad F'(0,\lambda_*)=\phi'(\lambda_*)-\theta'(\lambda_*)\neq 0,$$

it follows by the implicit function theorem [4, Theorem 13.7] that there exists an open neighborhood T of λ_* such that $\lambda_*\in T\subseteq\mathbb{R}$ and a unique continuously differentiable function $G:T\to\mathbb{R}$ such that

$$G(\lambda_*)=0 \quad\text{and}\quad F\left(G(\lambda),\lambda\right)=0. \tag{A.2}$$

This implies that for λ_k sufficiently close to λ_*, we have $G(\lambda_k)\equiv\delta_k$ so that

$$\lim_{\lambda_k\to\lambda_*}\delta_k=\lim_{\lambda_k\to\lambda_*}G(\lambda_k)=G(\lambda_*)=0, \tag{A.3}$$

where the last two equalities follow from the continuity of G and Eq. (A.2). Therefore, δ_k converges to zero as λ_k approaches λ_*.

Now let $\epsilon_k=\lambda_*-\lambda_k$. Taylor's theorem and the fact that λ_* is a root give

$$\phi(\lambda_*)=\phi_m(\epsilon_k;\lambda_k)+\frac{\phi^{(m+1)}(\zeta_k)}{(m+1)!}\epsilon_k^{m+1}=\theta(\lambda_*) \tag{A.4}$$

for some ζ_k between λ_k and λ_*, while the definition of δ_k and Taylor's theorem give

$$\phi_m(\delta_k;\lambda_k)=\theta(\lambda_{k+1})=\theta(\lambda_*)+\theta^{(1)}(\chi_k)(\lambda_{k+1}-\lambda_*) \tag{A.5}$$

for some other χ_k between λ_{k+1} ($=\lambda_k+\delta_k$) and λ_*. Hence, combining (A.4) and (A.5),

$$\phi_m(\delta_k;\lambda_k)-\phi_m(\epsilon_k;\lambda_k)-\theta^{(1)}(\chi_k)(\lambda_{k+1}-\lambda_*)=\frac{\phi^{(m+1)}(\zeta_k)}{(m+1)!}\epsilon_k^{m+1}. \tag{A.6}$$

But

$$\begin{aligned}\phi_m(\delta_k;\lambda_k)-\phi_m(\epsilon_k;\lambda_k)&=\sum_{i=1}^{m}\frac{\phi^{(i)}(\lambda_k)}{i!}(\delta_k^i-\epsilon_k^i)\\&=(\lambda_{k+1}-\lambda_*)\left(\phi^{(1)}(\lambda_k)+\sum_{i=2}^{m}\frac{\phi^{(i)}(\lambda_k)}{i!}\sum_{j=0}^{i-1}\delta_k^j\epsilon_k^{i-j-1}\right)\end{aligned} \tag{A.7}$$

in which case (A.6) gives

$$\lambda_{k+1} - \lambda_* = \frac{\kappa_k}{(m+1)!}(\lambda_* - \lambda_k)^{m+1} \tag{A.8}$$

where

$$\kappa_k = \frac{\phi^{(m+1)}(\zeta_k)}{\phi^{(1)}(\lambda_k) - \theta^{(1)}(\chi_k) + \sum_{i=2}^{m} \frac{\phi^{(i)}(\lambda_k)}{i!} \sum_{j=0}^{i-1} \delta_k^j \epsilon_k^{i-j-1}}.$$

Then (A.3) implies that for sufficiently small $\lambda_k - \lambda_*$,

$$|\kappa_k| \leq 2|\max[1, \phi^{(m+1)}(\lambda_*)]/(\phi^{(1)}(\lambda_*) - \theta^{(1)}(\lambda_*))| < \infty$$

as x_* is a simple root, and the required convergence estimate follows from (A.8). □

References

1. Absil, P.-A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**(3), 303–330 (2007)
2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
3. Amestoy, P., Duff, I.S., Pralet, S., Voemel, C.: Adapting a parallel sparse direct solver to SMP architectures. Parallel Comput. **29**(11–12), 1645–1668 (2003)
4. Apostol, T.M.: Mathematical Analysis. 2nd edn. Addison-Wesley, Reading (1974)
5. Berkes, P., Wiskott, L.: Analysis and interpretation of quadratic models of receptive fields. Nat. Protoc. **2**(2), 400–407 (2007)
6. Busygin, S., Ag, C., Butenko, S., Pardalos, P.M.: A heuristic for the maximum independent set problem based on optimization of a quadratic over a sphere. J. Comb. Optim. **6**(3), 287–297 (2002)
7. Cartis, C., Gould, N.I.M., Toint, Ph.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. Math. Program. Ser. A 51 pages (2009) doi:10.1007/s10107-009-0286-5
8. Cartis, C., Gould, N.I.M., Toint, Ph.L.: Trust-region and other regularisations of linear least-squares problems. BIT **49**(1), 21–53 (2009)
9. Chabrillac, Y., Crouzeix, J.-P.: Definiteness and semidefiniteness of quadratic forms revisited. Linear Algebra Appl. **63**, 283–292 (1984)
10. Cline, A.K., Moler, C.B., Stewart, G.W., Wilkinson, J.H.: An estimate for the condition number of a matrix. SIAM J. Numer. Anal. **16**(2), 368–375 (1979)
11. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Trust-Region Methods. SIAM, Philadelphia (2000)
12. Demmel, J.W.: Applied Numerical Linear Algebra. SIAM, Philadelphia (1997)
13. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
14. Dollar, H.S.: On Taylor series approximations for trust-region and regularized subproblems in optimization. Internal Technical Report Internal-2009-1, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England (2009)
15. Dollar, H.S., Gould, N.I.M., Robinson, D.P.: On solving trust-region and other regularised subproblems in optimization. Technical Report RAL-TR-2009-003, Rutherford Appleton Laboratory (2009)
16. Duff, I.S.: MA57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Trans. Math. Softw. **30**(2), 118–144 (2004)
17. Duff, I.S., Reid, J.K.: The multifrontal solution of indefinite sparse symmetric linear equations. ACM Trans. Math. Softw. **9**(3), 302–325 (1983)

18. Erway, J.B., Gill, P.E.: A subspace minimization method for the trust-region step. SIAM J. Optim. **20**(3), 1439–1461 (2009)
19. Erway, J.B., Gill, P.E., Griffin, J.D.: Iterative methods for finding a trust-region step. SIAM J. Optim. **20**(2), 1110–1131 (2009)
20. Gander, W.: On the Linear Least Squares Problem with a Quadratic Constraint. Technical Report STAN-CS-78-697. Computer Science Department, Stanford University, California (1978)
21. Gay, D.M.: Computing optimal locally constrained steps. SIAM J. Sci. Stat. Comput. **2**(2), 186–197 (1981)
22. Gertz, E.M., Gill, P.E.: A primal-dual trust region algorithm for nonlinear optimization. Math. Program. Ser. B **100**(1), 49–94 (2004)
23. Gould, N.I.M.: On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem. Math. Program. **32**(1), 90–99 (1985)
24. Gould, N.I.M., Hu, Y., Scott, J.A.: A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. ACM Trans. Math. Softw. **32**(2), Article 10 (2007)
25. Gould, N.I.M., Lucidi, S., Roma, M., Toint, Ph.L.: Solving the trust-region subproblem using the Lanczos method. SIAM J. Optim. **9**(2), 504–525 (1999)
26. Gould, N.I.M., Orban, D., Toint, Ph.L.: CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited. ACM Trans. Math. Softw. **29**(4), 373–394 (2003)
27. Gould, N.I.M., Orban, D., Toint, Ph.L.: GALAHAD—a library of thread-safe fortran 90 packages for large-scale nonlinear optimization. ACM Trans. Math. Softw. **29**(4), 353–372 (2003)
28. Gould, N.I.M., Scott, J.A.: A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations. ACM Trans. Math. Softw. **30**(3), 300–325 (2004)
29. Griewank, A.: The Modification of Newton's Method for Unconstrained Optimization by Bounding Cubic Terms. Technical Report DAMTP/NA12. Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge (1981)
30. Hager, W.W.: Condition estimates. SIAM J. Sci. Stat. Comput. **5**(2), 311–316 (1984)
31. Hager, W.W.: Minimizing a quadratic over a sphere. SIAM J. Optim. **12**(1), 188–208 (2001)
32. Hebden, M.D.: An Algorithm for Minimization Using Exact Second Derivatives. Technical Report T.P. 515. AERE, Harwell Laboratory, Harwell (1973)
33. Higham, N.J.: Fortran codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. ACM Trans. Math. Softw. **14**(4), 381–396 (1988)
34. Hogg, J.D.: A DAG-Based Parallel Cholesky Factorization for Multicore Systems. Technical Report RAL-TR-2008-029. Rutherford Appleton Laboratory, Chilton (2008)
35. Hogg, J.D., Reid, J.K., Scott, J.A.: A DAG-Based Sparse Cholesky Solver for Multicore Architectures. Technical Report RAL-TR-2009-004. Rutherford Appleton Laboratory, Chilton (2009)
36. Montero, A.: Study of SU(3) vortex-like configurations with a new maximal center gauge fixing method. Phys. Lett. **B467**, 106–111 (1999)
37. Moré, J.J.: Recent developments in algorithms and software for trust region methods. In: Bachem, A., Grötschel, M., Korte, B. (eds.) Mathematical Programming: The State of the Art, pp. 258–287. Springer, Heidelberg (1983)
38. Moré, J.J., Sorensen, D.C.: Computing a trust region step. SIAM J. Sci. Stat. Comput. **4**(3), 553–572 (1983)
39. Nesterov, Yu., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Math. Program. **108**(1), 177–205 (2006)
40. Parlett, B.N.: The Symmetric Eigenvalue Problem. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1980. Reprinted as Classics in Applied Mathematics 20, SIAM, Philadelphia, USA (1998)
41. Poljack, S., Wolkowicz, H.: Convex relaxations of (0,1)–quadratic programming. Math. Oper. Res. **20**(3), 550–561 (1995)
42. Reid, J.K., Scott, J.A.: An Out-of-Core Sparse Cholesky Solver. Technical Report RAL-TR-2006-013. Rutherford Appleton Laboratory, Chilton (2006)
43. Reinsch, C.: Smoothing by spline functions II. Numerische Mathematik **16**(5), 451–454 (1971)
44. Schenk, O., Christen, M., Burkhart, H.: Algorithmic performance studies on graphics processing units. J. Parallel Distrib. Comput. **68**, 1360–1369 (2008)
45. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal. **20**(3), 626–637 (1983)
46. Toint, Ph.L.: Towards an efficient sparsity exploiting Newton method for minimization. In: Duff, I.S. (ed.) Sparse Matrices and Their Uses, pp. 57–88. Academic Press, London (1981)

47. Traub, J.F.: Iterative Methods for the Solution of Equations. Prentice-Hall, Englewood Cliffs (1964)
48. Trefethen, L.N., Bai, D.: Numerical Linear Algebra. SIAM, Philadelphia (1997)
49. Weiser, M., Deuflhard, P., Erdmann, B.: Affine conjugate adaptive Newton methods for nonlinear elastomechanics. Optim. Methods Softw. **22**(3), 413–431 (2007)