On solving trust-region and other regularised subproblems in optimization[1]

H. Sue Dollar[2], Nicholas I. M. Gould[2,3] and Daniel P. Robinson[3]

ABSTRACT

The solution of trust-region and regularisation subproblems which arise in unconstrained optimization is considered. Building on the pioneering work of Gay, Moré and Sorensen, methods which 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 in 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.

---

[2] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, EU.
Email: sue.dollar@stfc.ac.uk, nick.gould@stfc.ac.uk .
Current reports available from "http://www.numerical.rl.ac.uk/reports/".

[3] Oxford University Computing Laboratory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, EU.
Email: nick.gould@comlab.ox.ac.uk, daniel.robinson@comlab.ox.ac.uk .
Current reports available from "http://web.comlab.ox.ac.uk/oucl/publications/natr/".

Computational Science and Engineering Department

Atlas Centre

Rutherford Appleton Laboratory

Oxfordshire OX11 0QX

February 11, 2009

1   Introduction

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

$$\underset{x \in \mathrm{I\!R}^n}{\text{minimize}} \quad q(x) \stackrel{\text{def}}{=} c^T x + \tfrac{1}{2} x^T H x \;\; \text{subject to} \;\; \|x\|_M \leq \Delta \tag{1.1}$$

and

$$\underset{x \in \mathrm{I\!R}^n}{\text{minimize}} \quad r(x) \stackrel{\text{def}}{=} c^T x + \tfrac{1}{2} x^T H x + \tfrac{\sigma}{p} \|x\|_M^p, \tag{1.2}$$

where the M-norm of x is $\|x\|_M \stackrel{\text{def}}{=} \sqrt{x^T M x}$. Both problems arise as subproblems in unconstrained optimization; problem (1.1) occurs when computing the step in trust-region methods [8, 33], while (1.2) plays the same role in more recent regularisation approaches [5, 24, 35, 44]; 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 [24].

Although it is now common to try to find approximate solutions to (1.1) and (1.2) using iterative methods [5, 14, 15, 20, 25, 40, 41], 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 Gay-Moré-Sorensen [17, 34] methods in the light of modern sparse factorization.

In §2 we discuss optimality conditions for our two subproblems and see how they lead to a robust framework for their solution. Details are given in §3, and here it is shown that the underlying method may always be made at least superlinearly convergent. The resulting software is outlined in §4, and we follow by describing experiments which indicate the effectiveness of our enhancements. We conclude and suggest future extensions in §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 $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ with associated eigenvectors u_i, $1 \leq i \leq n$, and recall that the generalised *Rayleigh quotient*

$$\rho_M(x) \stackrel{\text{def}}{=} \frac{x^T H x}{x^T M x}$$

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

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

where by convention $\text{gap}(\lambda_i) = \infty$ 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 problems (1.1) and (1.2) become

$$\operatorname*{minimize}_{\overline{x} \in \mathbb{R}^n} \quad \overline{c}^T \overline{x} + \tfrac{1}{2} \overline{x}^T \overline{H} \overline{x} \ \text{ subject to } \ \|\overline{x}\| \leq \Delta$$

and

$$\operatorname*{minimize}_{\overline{x} \in \mathbb{R}^n} \quad \overline{r}(\overline{x}) \equiv \overline{c}^T \overline{x} + \tfrac{1}{2} \overline{x}^T \overline{H} \overline{x} + \tfrac{\sigma}{p} \|\overline{x}\|^p, \tag{2.1}$$

involving data $\overline{c} = R^{-T} c$, $\overline{H} = R^{-T} H R^{-1}$ 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

$$(\overline{H} + \lambda I)\overline{x}(\lambda) = -\overline{c} \tag{2.2}$$

whenever the system (2.2) is consistent; equivalently $x(\lambda) \stackrel{\text{def}}{=} R^{-1}\overline{x}(\lambda)$ satisfies

$$(H + \lambda M)x(\lambda) = -c. \tag{2.3}$$

If $\overline{H}$ has eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ 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

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

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

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

and have the following general result.

Lemma 2.1 Let $\lambda_{\mathrm{S}} = \max(0, -\lambda_1)$, where λ_1 is the leftmost eigenvalue of the pencil (H, M), and suppose that $x(\lambda)$ satisfies (2.3). Then the function

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

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

Proof. The result follows directly from [6, Lem. 2.1] since

$$\|x(\lambda)\|_M = \|\overline{x}(\lambda)\| = \|\overline{y}(\lambda)\| = \sqrt{\sum_{i=1}^{n}\left(\frac{\gamma_i}{\lambda + \lambda_i}\right)^2}. \tag{2.4}$$

□

2.1   The trust-region problem

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

Theorem 2.2 [17: Thm. 2.1, 34: Lem. 2.1]. Any global minimizer x_* of (1.1) satisfies the equation

$$(H + \lambda_* M)x_* = -c, \tag{2.5}$$

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" [34] which we will discuss shortly—the multiplier $\lambda_* > \lambda_S$ and $\|x_*\|_M = \Delta$. In this, by contrast "easy" case, we seek the (unique) root of the scalar nonlinear "secular" equation

$$\pi(\lambda; \beta) \equiv \|x(\lambda)\|_M^{\beta} = \Delta^{\beta} \tag{2.6}$$

in the interval (λ_S, ∞). Our task is helped since Lemma 2.1 shows that $\pi(\lambda; \beta)$ is either strictly convex and decreasing or strictly concave and increasing for $\beta \in [-1, \infty) \setminus \{0\}$. In particular, if we partition the real line into

$$\mathcal{N} \stackrel{\text{def}}{=} \{\lambda \mid \lambda \in (-\infty, \lambda_S]\}, \quad \mathcal{L} \stackrel{\text{def}}{=} \{\lambda \mid \lambda \in (\lambda_S, \lambda_*]\} \quad \text{and} \quad \mathcal{G} \stackrel{\text{def}}{=} \{\lambda \mid \lambda \in (\lambda_*, \infty)\}$$

and denote $\mathcal{F} \stackrel{\text{def}}{=} \mathcal{L} \cup \mathcal{G}$ (see Figure 2.1 on the following page), both Newton's and the secant method for (2.6) are guaranteed to converge globally (monotonically, linearly and ultimately at least superlinearly) to the required root if started from any value(s) in $\mathcal{L}$ [6, Lem. A.1]. Moreover, since this is true for all $\beta \in [-1, \infty) \setminus \{0\}$, we are at liberty to choose the β for which the Newton correction gives the best correction, and it can be shown that this occurs when $\beta = -1$ [6, §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
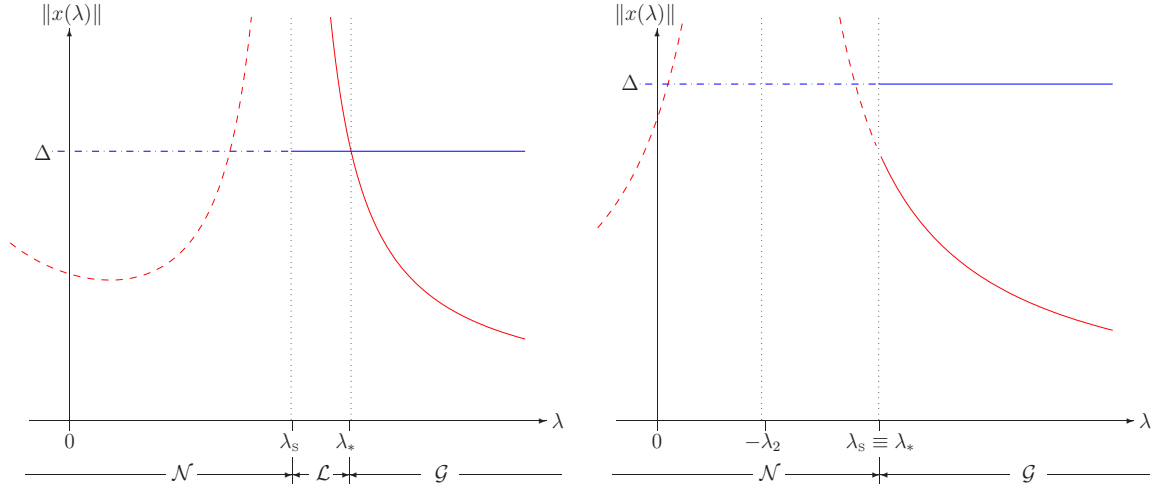
Figure 2.1: The sets $\mathcal{N}$, $\mathcal{L}$ and $\mathcal{G}$ and $\|x(\lambda)\|$ for the problem of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within a ℓ_2-norm trust region of radius 4 ("easy" case, left) and those for $-\frac{1}{4}x_1^2 - \frac{1}{8}x_2^2 + x_2$ within a trust region of radius 5 ("hard" case, right).

the poles present at $\lambda = -\lambda_i$ when $\beta > 0$ [26,38]. 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 §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, (2.3) will be consistent when $\lambda = \lambda_\mathrm{s}$—denote the minimum norm solution in this case as x_s—but if $\|x_\mathrm{s}\| < \Delta$, there is no solution to the secular equation in $[\lambda_\mathrm{s}, \infty)$ or equivalently $\mathcal{L}$ is empty (again see Figure 2.1). In fact the required solution in this case is $x_\mathrm{s} + \alpha_\mathrm{s} u_1$, where the scalar α_s is chosen so that $\|x_\mathrm{s} + \alpha_\mathrm{s} u_1\|_M = \Delta$. 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

$$x_\mathrm{s} = \lim_{\lambda \to \lambda_\mathrm{s}} x(\lambda)$$

from (2.3).

2.2   The regularisation problem

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

Theorem 2.3 [5: Thm. 3.1, 35: Thm. 10 when $p = 3$]. Any global minimizer x_* of (1.2) when $p > 2$ satisfies the equation

$$(H + \lambda_* M)x_* = -c, \tag{2.7}$$

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.

Proof. Since (1.2) is equivalent to (2.1), we establish the result for the latter; the required result then follows using the given linear transformation of variables. We may express the derivatives of the regularisation function $\overline{r}(\overline{x})$ in (2.1) as

$$\nabla_{\overline{x}}\overline{r}(\overline{x}) = \overline{c} + \overline{H}\overline{x} + \lambda\overline{x} \text{ and } \nabla_{\overline{x}\overline{x}}\overline{r}(\overline{x}) = \overline{H} + \lambda I + \lambda(p-2)\left(\frac{\overline{x}}{\|\overline{x}\|}\right)\left(\frac{\overline{x}}{\|\overline{x}\|}\right)^T, \tag{2.8}$$

where $\lambda = \sigma\|\overline{x}\|^{p-2}$.

Firstly, let $\overline{x}_*$ be a global minimizer of $\overline{r}(\overline{x})$. Since $p > 2$, $\overline{r}(\overline{x})$ approaches $+\infty$ as $\overline{x}$ approaches infinity; thus $\overline{x}_*$ is bounded. It follows from (2.8) and the first- and second-order necessary optimality conditions at $\overline{x}_*$ that

$$\overline{c} + (\overline{H} + \lambda_* I)\overline{x}_* = 0,$$

and hence transforming variables that (2.7) holds, and that

$$\overline{w}^T\left(\overline{H} + \lambda_* I + \lambda_*(p-2)\left(\frac{\overline{x}_*}{\|\overline{x}_*\|}\right)\left(\frac{\overline{x}_*}{\|\overline{x}_*\|}\right)^T\right)\overline{w} \geq 0 \tag{2.9}$$

for all vectors $\overline{w}$.

If $\overline{x}_* = 0$, then by definition $\lambda_* = 0$ and (2.9) is equivalent to $\overline{H}$ being positive semi-definite, which immediately gives the required result. Thus we need only consider $\overline{x}_* \neq 0$.

There are two cases to consider. Firstly, suppose that $\overline{w}^T\overline{x}_* = 0$. In this case, it immediately follows from (2.9) that

$$\overline{w}^T(\overline{H} + \lambda_* I)\overline{w} \geq 0 \text{ for all } \overline{w} \text{ for which } \overline{w}^T\overline{x}_* = 0. \tag{2.10}$$

It thus remains to consider vectors $\overline{w}$ for which $\overline{w}^T\overline{x}_* \neq 0$. Since $\overline{w}$ and $\overline{x}_*$ are not orthogonal, the line $\overline{x}_* + \alpha\overline{w}$ intersects the ball of radius $\|\overline{x}_*\|$ at two points, $\overline{x}_*$ and $\overline{u}_* \neq \overline{x}_*$, say, and thus

$$\|\overline{u}_*\| = \|\overline{x}_*\|. \tag{2.11}$$

We let $\overline{w}_* = \overline{u}_* - \overline{x}_*$, and note that $\overline{w}_*$ is parallel to $\overline{w}$. Since $\overline{x}_*$ is a global minimizer, we immediately have that

$$
\begin{aligned}
0 \leq \ & \overline{r}(\overline{u}_*) - \overline{r}(\overline{x}_*) = \overline{c}^T(\overline{u}_* - \overline{x}_*) + \tfrac{1}{2}\overline{u}_*^T \overline{H}\overline{u}_* - \tfrac{1}{2}\overline{x}_*^T \overline{H}\overline{x}_* + \tfrac{\sigma}{p}(\|\overline{u}_*\|^p - \|\overline{x}_*\|^p) \\
= \ & \overline{c}^T(\overline{u}_* - \overline{x}_*) + \tfrac{1}{2}\overline{u}_*^T \overline{H}\overline{u}_* - \tfrac{1}{2}\overline{x}_*^T \overline{H}\overline{x}_*,
\end{aligned}
\tag{2.12}
$$

where the last equality follows from (2.11). But (2.7) gives that

$$
\overline{c}^T(\overline{u}_* - \overline{x}_*) = (\overline{x}_* - \overline{u}_*)^T \overline{H}\overline{x}_* + \lambda_*(\overline{x}_* - \overline{u}_*)^T \overline{x}_*. \tag{2.13}
$$

In addition, (2.11) shows that

$$
(\overline{x}_* - \overline{u}_*)^T \overline{x}_* = \tfrac{1}{2}\overline{x}_*^T \overline{x}_* + \tfrac{1}{2}\overline{u}_*^T \overline{u}_* - \overline{u}_*^T \overline{x}_* = \tfrac{1}{2}\overline{w}_*^T \overline{w}_*. \tag{2.14}
$$

Thus combining (2.12)–(2.14), we find that

$$
0 \leq \tfrac{1}{2}\lambda_* \overline{w}_*^T \overline{w}_* + \tfrac{1}{2}\overline{u}_*^T \overline{H}\overline{u}_* - \tfrac{1}{2}\overline{x}_*^T H\overline{x}_* + \overline{x}_*^T \overline{H}\overline{x}_* - \overline{u}_*^T H\overline{x}_* = \tfrac{1}{2}\overline{w}_*^T(\overline{H} + \lambda_* I)\overline{w}_*
$$

from which we deduce that

$$
\overline{w}^T(\overline{H} + \lambda_* I)\overline{w} \geq 0 \ \text{ for all } \ \overline{w} \ \text{ for which } \ \overline{w}^T \overline{x}_* \neq 0. \tag{2.15}
$$

Hence (2.10) and (2.15) together show that $\overline{H} + \lambda_* I$ and thus $H + \lambda_* M$ are positive semi-definite.

Now suppose that λ_* is a root of the nonlinear equation

$$
\|x(\lambda)\|_M^{p-2} - \lambda/\sigma = 0 \tag{2.16}
$$

for which (2.7) holds and that $H + \lambda_* M$ is positive definite and thus $\lambda_* > \lambda_{\mathrm{S}}$. But it follows from Lemma 2.1 that $\|x(\lambda)\|_M^{p-2}$ strictly decreases from $\|x(\lambda_{\mathrm{S}})\|_M^{p-2}$ to zero on $(\lambda_{\mathrm{S}}, \infty)$ as $p > 2$, while clearly λ/σ strictly increases from $\lambda_{\mathrm{S}}/\sigma$ to infinity on the same interval. Thus (2.16) has at most one, and hence precisely one, root λ_* there. The unique solution x_* from (2.7) for this λ_* is thus the unique global minimizer of $r(x)$. $\square$

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 $\lambda_* > \lambda_{\mathrm{S}}$ of the scalar nonlinear "secular" equation

$$
\pi(\lambda; \beta) \equiv \|x(\lambda)\|_M^{\beta} = (\lambda/\sigma)^{\beta/(p-2)}. \tag{2.17}
$$

We illustrate this in Figure 2.2. As before, Lemma 2.1 shows that $\|x(\lambda)\|_M^{\beta} - (\lambda/\sigma)^{\beta/(p-2)}$ is either strictly convex and decreasing or strictly concave and increasing for $\beta \in [-1, p-2] \setminus \{0\}$, and thus again both Newton's and the secant method for (2.17) are guaranteed to converge globally (monotonically, linearly and ultimately at least superlinearly) to the required root if started from any value(s) in $\mathcal{L}$ [6, Lem. A.1]. Now,
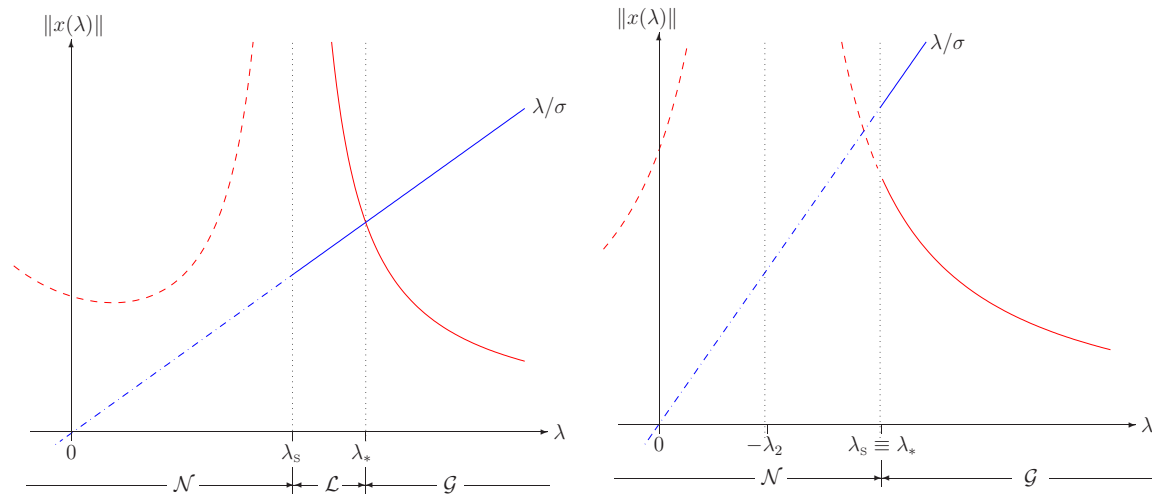
Figure 2.2: The sets $\mathcal{N}$, $\mathcal{L}$ and $\mathcal{G}$ and $\|x(\lambda)\|$ for the problem of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2 + \frac{\sigma}{3}\|x\|^3$ when $\sigma = 0.2$ ("easy" case, left) and those for $-\frac{1}{4}x_1^2 - \frac{1}{8}x_2^2 + x_2 + \frac{\sigma}{3}\|x\|^3$ when $\sigma = 0.1$ ("hard" case, right).

however, the value of β which gives the best step is not known *a priori* although it may be calculated very efficiently by uni-variate maximization [6, §3.3]. Moreover finding the required root by linearizing $\pi(\lambda; \beta)$ but not $(\lambda/\sigma)^{\beta/(p-2)}$ is known to out-perform Newton's method when $\beta = -1$ and $p \in (2, 3]$ [6, Lem. 3.3]. Thus again fast ultimate convergence is assured in the easy case, and the main task is to find an initial value in $\mathcal{L}$.

The hard case occurs under similar circumstances as occur for the trust-region problem, but when $\|x(\lambda_\text{s})\|_M^{p-2} < \lambda_\text{s}/\sigma$ and thus there is no solution to the secular equation in $[\lambda_\text{s}, \infty)$ (see Figure 2.2). The remedy is the same, namely that the required solution is then $x_\text{s} + \alpha_\text{s}u_1$, where the scalar α_s is chosen so that $\|x_\text{s} + \alpha_\text{s}u_1\|_M^{p-2} = \lambda_\text{s}/\sigma$.

3   Algorithmic considerations

3.1   Matrix factorization

We aim to solve (2.3) 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 [19, 23]. We use the commercial package[1] MA57 [12] but provide a slightly-less effective alternative SILS (based on the earlier MA27 [13]) for those unable to access MA57.

3.2   The secular function and its properties

Suppose that $x(\lambda)$ satisfies (2.3). We consider properties of the *secular* function

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

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 3.1 Suppose that $H + \lambda M$ is positive definite, that $x(\lambda)$ satisfies (2.3), 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 $\pi(\lambda) = x^T(\lambda)Mx(\lambda)$ 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 (2.3) that $(H+\lambda M)x^{(1)}(\lambda) = -Mx(\lambda)$ and then by induction by 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

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

---
[1]MA57 is available without charge to academics.

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 3.2 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 $L^T x(\lambda) = z(\lambda)$. 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

$$
Ly^{(k+1)}(\lambda) = -(k+1)Mx^{(k)}(\lambda), \ \ Dz^{(k+1)}(\lambda) = y^{(k+1)}(\lambda) \text{ and } L^T x^{(k+1)}(\lambda) = z^{(k+1)}(\lambda).
$$

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

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

where

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

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 definition of $\pi^{(2k+1)}(\lambda)$ follows from the identity

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

The remainder of the result follows immediately from Theorem 3.1. □

Notice how each odd-power derivative 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 [16, 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 $\pi(\lambda, 2) = \pi(\lambda)$.

Lemma 3.3 Let $\pi(\lambda) = \|x(\lambda)\|_M^2$, where $x(\lambda)$ satisfies (2.3). Suppose that $\lambda_\mathrm{C} > \lambda_\mathrm{S}$ and let $\pi_k(\delta)$ be the k-th order Taylor series approximation to $\pi(\lambda_\mathrm{C} + \delta)$. Then

$$\pi(\lambda_\mathrm{C} + \delta) \leq \pi_k(\delta) \ \text{ for even } k > 0 \text{ and } \ \pi(\lambda_\mathrm{C} + \delta) \geq \pi_k(\delta) \ \text{ for odd } k > 0 \qquad (3.7)$$

when $\delta > 0$, while

$$\pi(\lambda_\mathrm{C} + \delta) \geq \pi_{k+1}(\delta) \geq \pi_k(\delta) \ \text{ for all } k > 0 \qquad (3.8)$$

when $\lambda_s - \lambda_\mathrm{C} < \delta < 0$. The inequalities in (3.7)/(3.8) are strict whenever $c \neq 0$.

Proof.   It follows trivially from (2.4) that the j-th derivative, $\pi^{(j)}(\lambda)$, of $\pi(\lambda)$ is

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

Thus if we define the k-th order Taylor approximation

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

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

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

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

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

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

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

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 due to (3.9) in the above arguments are strict.   □

Now suppose that $\tau(\lambda)$ is a given monotonically non-decreasing function on (λ_S, ∞) and that $\pi(\lambda_S) > \tau(\lambda_S)$. For example $\tau(\lambda)$ might be constant, e.g. Δ^2 as in (2.6), or increasing, e.g. $(\lambda/\sigma)^{2/(p-2)}$ as in (2.17). In this case, Lemma 2.1 implies that there is a unique root, say $\lambda_* > \lambda_S$ of the equation

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

or equivalently

$$\pi(\lambda_C + \delta) = \tau(\lambda_C + \delta). \tag{3.13}$$

for the correction δ to λ_C.

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, Lemma 3.3 implies that $\pi_k(\delta_*) \le \pi(\lambda_C + \delta_*) = \tau(\lambda_C + \delta_*)$. 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 $\tau(\lambda_C + \delta) > \pi(\lambda_C + \delta) \ge \pi_k(\delta)$ for $\lambda_C + \delta > \lambda_*$, all positive roots of (3.14) give under-estimators. By contrast, if k is even, $\pi_k(\delta_*) \ge \pi(\lambda_C + \delta_*) = \tau(\lambda_C + \delta_*)$ 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 $\delta_* < 0$. Lemma 3.3 gives that $\tau(\lambda_C + \delta) > \pi(\lambda_C + \delta) \ge \pi_k(\delta)$ 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 Figure 3.1 on the following page. Notice that, since the secular function is not 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 $\pi(\lambda_B)$. 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 approximation 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 [42, 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.
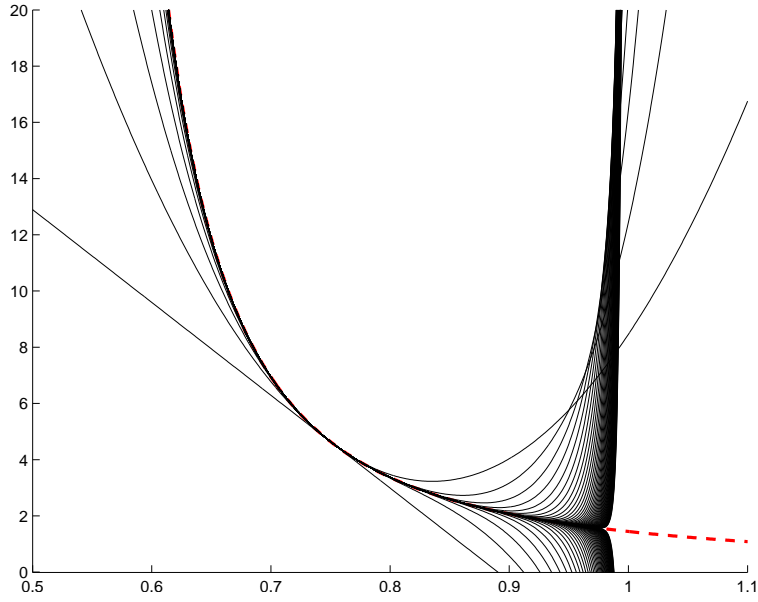
Figure 3.1: The secular function for the problem of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within a ℓ_2-norm trust region along (dashed line) with the Taylor series approximants of degrees 1 to 100 about $\lambda_{\mathrm{C}} = 0.75$ (solid lines). Observe the threshold $\lambda_{\mathrm{B}} \approx 0.98$ above which no approximant is good.

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 k-th order Taylor series approximation, $\pi_k(\delta; \beta)$, to $\pi(\lambda_{\mathrm{C}} + \delta; \beta)$ for modest values of $k \leq 3$; a higher-order analysis is possible but both 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

$$
\begin{aligned}
\pi^{(1)}(\lambda; \beta) &= \tfrac{\beta}{2} \left[\pi(\lambda)\right]^{\frac{\beta}{2}-1} \pi^{(1)}(\lambda), & (3.15)\\[4pt]
\pi^{(2)}(\lambda; \beta) &= \tfrac{\beta}{2} \left[\pi(\lambda)\right]^{\frac{\beta}{2}-1} \pi^{(2)}(\lambda) + \tfrac{\beta}{2}\left(\tfrac{\beta}{2}-1\right)\left[\pi(\lambda)\right]^{\frac{\beta}{2}-2} \left[\pi^{(1)}(\lambda)\right]^2, &\\[4pt]
\pi^{(3)}(\lambda; \beta) &= \tfrac{\beta}{2} \left[\pi(\lambda)\right]^{\frac{\beta}{2}-3} \left(\left[\pi(\lambda)\right]^2 \pi^{(3)}(\lambda) + 3\left(\tfrac{\beta}{2}-1\right)\pi(\lambda)\pi^{(1)}(\lambda)\pi^{(2)}(\lambda) \right. & (3.16)\\[4pt]
&\qquad \left. + \left(\tfrac{\beta}{2}-1\right)\left(\tfrac{\beta}{2}-2\right)\left[\pi^{(1)}(\lambda)\right]^3 \right) &
\end{aligned}
$$

$$
\begin{aligned}
\text{and } \pi^{(4)}(\lambda; \beta) &= \tfrac{\beta}{2} \left[\pi(\lambda)\right]^{\frac{\beta}{2}-4} \left(\left[\pi(\lambda)\right]^3 \pi^{(4)}(\lambda) + 4\left(\tfrac{\beta}{2}-1\right)\left[\pi(\lambda)\right]^2 \pi^{(1)}(\lambda)\pi^{(3)}(\lambda) \right. & (3.17)\\[4pt]
&\quad + 3\left(\tfrac{\beta}{2}-1\right)\left[\pi(\lambda)\right]^2 \left[\pi^{(2)}(\lambda)\right]^2 + 6\left(\tfrac{\beta}{2}-1\right)\left(\tfrac{\beta}{2}-2\right)\pi(\lambda)\left[\pi^{(1)}(\lambda)\right]^2 \pi^{(2)}(\lambda) &\\[4pt]
&\quad \left. + \left(\tfrac{\beta}{2}-1\right)\left(\tfrac{\beta}{2}-2\right)\left(\tfrac{\beta}{2}-3\right)\left[\pi^{(1)}(\lambda)\right]^4 \right). &
\end{aligned}
$$

From this we may deduce the following result.

Lemma 3.4 Let $\pi(\lambda; \beta) = \|x(\lambda)\|_M^\beta$, where $x(\lambda)$ satisfies (2.3). Suppose that $\lambda > \lambda_{\rm s}$. Then

(i) $\pi^{(1)}(\lambda; \beta) \leq 0$ for all $\beta > 0$ while $\pi^{(1)}(\lambda; \beta) \geq 0$ for all $\beta < 0$;

(ii) $\pi^{(2)}(\lambda; \beta) \geq 0$ for all $\beta > 0$ while $\pi^{(2)}(\lambda; \beta) \leq 0$ for all $\beta \in [-1, 0)$;

(iii) $\pi^{(3)}(\lambda; \beta) \leq 0$ for all $\beta > 0$ while $\pi^{(3)}(\lambda; \beta) \geq 0$ for all $\beta \in [-\frac{2}{3}, 0)$; and

(iv) $\pi^{(4)}(\lambda; \beta) \geq 0$ for all $\beta > 0$ while $\pi^{(4)}(\lambda; \beta) \leq 0$ for all $\beta \in [-\frac{2}{5}, 0)$.

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

Proof. Statements (i) follows directly from (3.15) and Theorem 3.1, while (ii) follows from Lemma 2.1. Statements (iii) and (iv) are established in Appendix B as Lemmas B.3 and B.5. $\qquad\square$

Corollary 3.5 Let $\pi(\lambda; \beta) = \|x(\lambda)\|_M^\beta$, where $x(\lambda)$ satisfies (2.3). Suppose that $\lambda_{\rm c} > \lambda_{\rm s}$ and let $\pi_k(\delta; \beta)$ be the k-th order Taylor series approximation to $\pi(\lambda_{\rm c} + \delta; \beta)$. Then,

(i) for $\beta > 0$,

$$\pi(\lambda_{\rm c} + \delta; \beta) \leq \pi_2(\delta; \beta) \text{ and } \pi(\lambda_{\rm c} + \delta; \beta) \geq \pi_k(\delta; \beta) \text{ for } k = 1, 3 \qquad (3.18)$$

when $\delta > 0$, while

$$\pi(\lambda_{\rm c} + \delta; \beta) \geq \pi_3(\delta; \beta) \geq \pi_2(\delta; \beta) \geq \pi_1(\delta; \beta) \qquad (3.19)$$

when $\lambda_s - \lambda_{\rm c} < \delta < 0$; and

(ii) otherwise
$$\begin{array}{ll} \pi(\lambda_{\rm c} + \delta; \beta) \leq \pi_1(\delta; \beta) & \text{for } \beta \in [-1, 0) \\ \pi(\lambda_{\rm c} + \delta; \beta) \geq \pi_2(\delta; \beta) & \text{for } \beta \in [-\frac{2}{3}, 0) \\ \pi(\lambda_{\rm c} + \delta; \beta) \leq \pi_3(\delta; \beta) & \text{for } \beta \in [-\frac{2}{5}, 0) \end{array} \qquad (3.20)$$

when $\delta > 0$, while

$$\pi(\lambda_{\rm C} + \delta; \beta) \le \pi_3(\delta; \beta) \le \pi_2(\delta; \beta) \le \pi_1(\delta; \beta) \ \text{ for } \ \beta \in [-\tfrac{2}{5}, 0)$$
$$\pi(\lambda_{\rm C} + \delta; \beta) \le \pi_2(\delta; \beta) \le \pi_1(\delta; \beta) \ \text{ for } \ \beta \in [-\tfrac{2}{3}, 0) \qquad (3.21)$$
$$\pi(\lambda_{\rm C} + \delta; \beta) \le \pi_1(\delta; \beta) \ \text{ for } \ \beta \in [-1, 0)$$

when $\lambda_s - \lambda_{\rm C} < \delta < 0$.

The inequalities in (3.18)–(3.21) are strict whenever $c \ne 0$.

Proof. These results follow directly from the relationship $\pi(\lambda_{\rm C} + \delta; \beta) = \pi_k(\delta; \beta) + \pi^{(k+1)}(\lambda_{\rm C} + \xi_k; \beta)\delta^{k+1}/(k+1)!$ for some ξ_k between 0 and δ (Taylors' theorem), $\pi_{k+1}(\delta; \beta) = \pi_k(\delta; \beta) + \pi^{(k+1)}(\lambda_{\rm C}; \beta)\delta^{k+1}/(k+1)!$ (Taylor series) and Lemma 3.4. □

The limiting ranges on negative β in (3.20) and (3.21) may seem inconvenient, but as we see in Figure 3.2 they may be necessary to ensure the Taylor polynomials over/underestimate $\pi(\lambda; \beta)$.
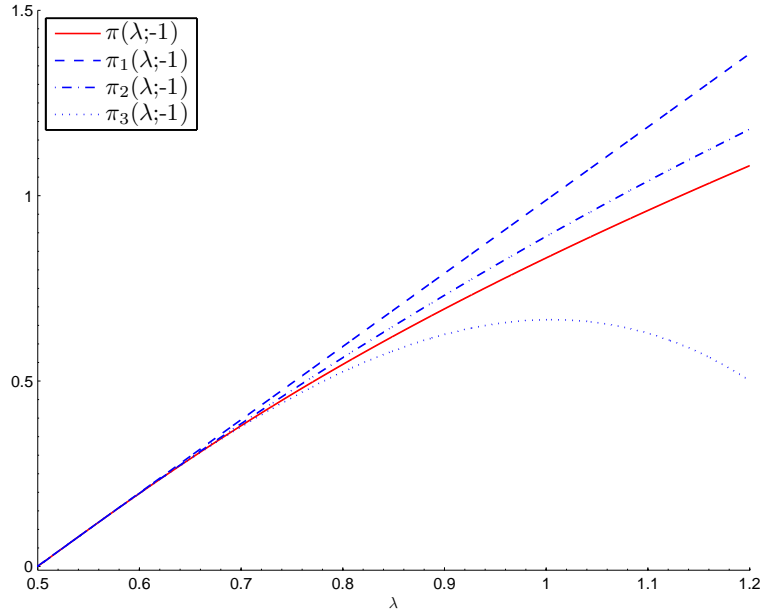
Figure 3.2: The function $\pi(\lambda, -1)$ for the problem of minimizing $-\tfrac{1}{4}x_1^2 + \tfrac{1}{4}x_2^2 + \tfrac{1}{2}x_1 + x_2$ within a ℓ_2-norm trust region along with the Taylor series approximants $\pi_1(\lambda; -1)$ to $\pi_3(\lambda; -1)$ of degrees 1 to 3 about $\lambda_{\rm C} = 0.55$. Observe that $\pi_2(\lambda; \beta)$ and $\pi_3(\lambda; \beta)$ do not obey (3.20); magnifying the figure shows that they also violate (3.21).

We can repeat the discussion following Lemma 3.3 concerning $\pi(\lambda)$ for the more general function $\pi(\lambda; \beta)$. For positive β, all we said about solving (3.12) remains true for

$$\pi(\lambda; \beta) = \tau(\lambda; \beta). \qquad (3.22)$$

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

$$\pi_k(\delta, \beta) = \tau(\lambda_{\rm C} + \delta; \beta) \qquad (3.23)$$

Figure 3.3: 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 $\pi(\lambda; \beta) \equiv \|x(\lambda)\|^{\beta}$ abo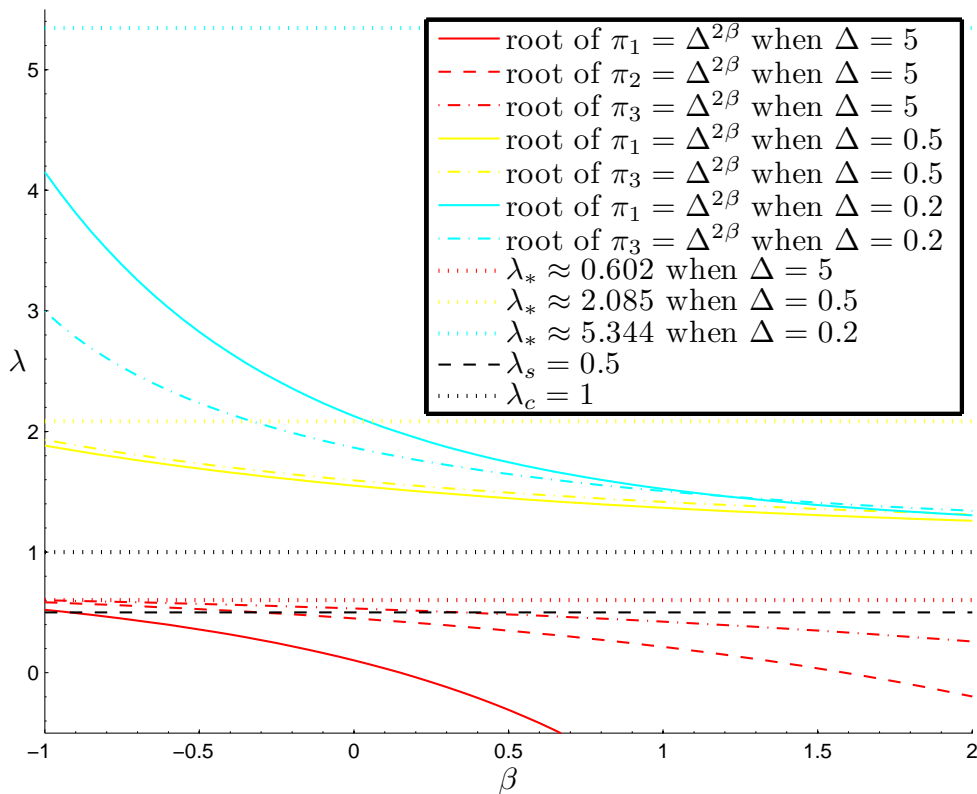ut $\lambda_C = 1$ for the problems of minimizing $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$ within ℓ_2-norm trust regions of radii $\Delta = 5$, 0.5 and 0.2.

for $k = 1, 3$ lead to under-estimators of λ_*, while if $\lambda_* < \lambda_C$, the least negative roots of (3.23) for $k = 1, 2, 3$ 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, $-\frac{2}{3}$ and $-\frac{2}{5}$ respectively for the linear, quadratic and cubic Taylor approximants. Finding the root of (3.23) 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 Corollary 3.5, $\mathcal{B}_1 = [-1, \infty)$ and $\mathcal{B}_3 = [-\frac{2}{5}, \infty)$, while $\mathcal{B}_2 = [-\frac{2}{3}, \infty)$ 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 Figure 3.3 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 [6, §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.

Finally, as an alternative to (3.23), suppose that $\beta > 0$, that $\lambda_C < \lambda_*$, that $\pi_k(\delta; \beta) \leq \pi(\lambda_C + \delta; \beta)$ for $\delta \in [0, \lambda_* - \lambda_C]$, and that we can find a function $\zeta(\lambda; \beta) \geq \tau(\lambda; \beta)$ for $\lambda \in [\lambda_C, \lambda_*]$. Now consider instead the equation

$$\pi_k(\delta; \beta) = \zeta(\lambda_C + \delta; \beta). \tag{3.24}$$

Then if $\delta_r > 0$ is a root of (3.24), we have $\pi(\lambda_C + \delta_r; \beta) \geq \pi_k(\delta_k; \beta) = \zeta(\lambda_C + \delta_r; \beta) \geq \tau(\lambda_C + \delta_r; \beta)$, and hence positive roots of (3.24) will under-estimate those of both (3.22) and (3.23). The same is true when $\lambda_* < \lambda_C$ if $\pi_k(\delta; \beta)$ and $\zeta(\lambda; \beta)$ satisfy the given relationships when $\delta \in [\lambda_* - \lambda_C, 0]$ and $\lambda \in [\lambda_*, \lambda_C]$, and we are seeking negative roots of (3.24). When $\beta < 0$, the above remains true so long as we reverse the sense of the inequalities on π_k and ζ.

This might be of value when $\beta > 0$ if τ is concave but complicated to evaluate, for then picking ζ to be its first-order Taylor approximant is appropriate and the overall system (3.24) will be polynomial; the same is true for more general τ when we pick ζ to be the ℓ-th order Taylor approximant if $\lambda_C < \lambda_*$ and the $\ell + 1$-st derivative of τ is negative throughout (λ_C, λ_*) or if $\lambda_C > \lambda_*$ and the $\ell + 1$-st derivative of τ has the same sign as $(-1)^\ell$ throughout (λ_*, λ_C). For negative β, the ℓ-th order Taylor approximant is appropriate if $\lambda_C < \lambda_*$ and the $\ell + 1$-st derivative of τ is positive throughout (λ_C, λ_*) or if $\lambda_C > \lambda_*$ and the $\ell + 1$-st derivative of τ has the opposite sign as $(-1)^\ell$ throughout (λ_*, λ_C).

For example, for the regularisation problem $\tau(\lambda; \beta) = (\lambda/\sigma)^{\beta/(p-2)}$, which is concave so long as $\beta \in [0, p-2]$ and convex for all negative β. More generally for non-integer $\beta/(p-2) > 0$, the $\ell + 1$-st derivative of $(\lambda/\sigma)^{\beta/(p-2)}$ first becomes negative when $\ell = \lceil \beta/(p-2) \rceil$, and oscillates in sign thereafter. Thus Taylor approximants of degree $\lceil \beta/(p-2) \rceil + 2j$, $j \geq 0$ will be possible for $\lambda_C < \lambda_*$, while if $\lambda_C > \lambda_*$ approximants (of every order) are possible if and only if $\lceil \beta/(p-2) \rceil$ is even. For negative β, Taylor approximants of every degree suffice if $\lambda_C > \lambda_*$ while only odd-degree approximants are appropriate when $\lambda_C < \lambda_*$.

Of course other generalisations of (3.12) are possible. In particular, one might consider

$$\phi(\pi(\lambda)) = \phi(\tau(\lambda))$$

for a (carefully chosen) nonlinear function ϕ [16, §6] or a nonlinear transformation of the variables

$$\pi(\lambda(\eta)) = \tau(\lambda(\eta))$$

by regarding λ as a function of the scalar η [29–32]. We have refrained from doing so as in general it does not seem clear how to retain the useful convexity/concavity properties that we have exploited in this section.

3.3 The trust-region problem

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

The standard method [17,34] for finding an initial point in $\mathcal{L}$ (if one exists) is to determine bounds $\lambda_{\text{L}} \geq 0$ and λ_{U} for which $\lambda_* \in [\lambda_{\text{L}}, \lambda_{\text{U}}]$. If $\lambda_{\text{U}} > \lambda_*$, then necessarily $\lambda_{\text{U}} \in \mathcal{G}$, while $\lambda_{\text{L}} \in \mathcal{N} \cup \mathcal{L}$. The bounds λ_{L} and λ_{U} are adjusted by iteration so that the length of $[\lambda_{\text{L}}, \lambda_{\text{U}}]$ 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 $[\lambda_{\text{L}}, \lambda_{\text{U}}]$ surrounding λ_*, but that the current estimate $\lambda_{\text{C}} \in [\lambda_{\text{L}}, \lambda_{\text{U}}]$ of λ_* is not in $\mathcal{L}$. If $\lambda_{\text{C}} \in \mathcal{G}$, an improvement λ_+ may be sought by applying one or more iterations of a suitable root finder—Newton's method applied to (2.6) 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 $\lambda_{\text{C}} \in \mathcal{N}$, 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 bi-product of the root-finding when $\lambda_{\text{C}} \in \mathcal{G}$ or from some auxiliary calculation, for instance from a suitably chosen Rayleigh-quotient, when $\lambda_{\text{C}} \in \mathcal{N}$—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 [8, §7.3.6], usually involving a convex and/or geometric combination of the current bounds.

We formalize this discussion as Algorithm 3.1 on the next page. The proposed formula for computing the next λ ensures that the ratio of successive bounding intervals is at most

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

for some $\theta \in (\theta_{\text{L}}, \theta_{\text{U}}) \subset (0, 1)$ [8, §7.3.6], and thus that the algorithm has the desired effect of ensuring finite convergence if $\mathcal{L}$ is non-empty.

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 (2.3) give

$$(\lambda_* + \lambda_1)^2 \leq \frac{\overline{x}_*^T (\overline{H} + \lambda_* I)^2 \overline{x}_*}{\overline{x}_*^T \overline{x}_*} = \frac{\|\overline{c}\|^2}{\Delta^2} = \frac{\|c\|_{M^{-1}}^2}{\Delta^2} \leq (\lambda_* + \lambda_n)^2 \tag{3.25}$$

Algorithm 3.1: Find $\lambda \in \mathcal{L}$

Given initial $\lambda_\mathrm{L} \leq \lambda_* \leq \lambda_\mathrm{U}$ and λ_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 $\lambda_\mathrm{L} \leftarrow \max(\lambda_\mathrm{L}, \lambda_\mathrm{C})$.
    Else (i.e., $\lambda_\mathrm{C} \in \mathcal{G}$)
        set $\lambda_\mathrm{U} \leftarrow \min(\lambda_\mathrm{U}, \lambda_\mathrm{C})$.
        Compute a new estimate λ_+ of λ_*.
        If $\lambda_+ \in \mathcal{L}$
            exit loop with $\lambda \leftarrow \lambda_+$.
        Else if $\lambda_+ \in \mathcal{N}$
            set $\lambda_\mathrm{L} \leftarrow \max(\lambda_\mathrm{L}, \lambda_+)$.
        Else (i.e., $\lambda_+ \in \mathcal{G}$)
            set $\lambda_\mathrm{U} \leftarrow \min(\lambda_\mathrm{U}, \lambda_+)$.
    Possibly compute an estimate $\lambda_\mathrm{E} \geq \lambda_1$, and in this case 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 $\lambda_\mathrm{C} \leftarrow \max(\gamma\sqrt{\lambda_\mathrm{L}\lambda_\mathrm{U}}, \lambda_\mathrm{L} + \theta(\lambda_\mathrm{U} - \lambda_\mathrm{L}))$.

---

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

$$\frac{\|c\|_{M^{-1}}}{\Delta} - \lambda_n \leq \lambda_* \leq \frac{\|c\|_{M^{-1}}}{\Delta} - \lambda_1. \tag{3.26}$$

For (3.26) 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 $\|H\|_F$ [8: §7.3.8, 17, 34]. 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

$$|h_{k,k} - \lambda m_{k,k}| \leq \left| \sum_{i=1, i\neq k}^{n} (h_{k,i} - \lambda m_{k,i}) \right| \leq o_k^\mathrm{H} + |\lambda|o_k^\mathrm{M}, \tag{3.27}$$

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

$$|h_{k,k} - \lambda m_{k,k}| \leq o_k^{\mathrm{H}} + |\lambda| o_k^{\mathrm{M}}, \ k = 1, \ldots, n; \tag{3.28}$$

it is easy to see that each region (3.28) 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 (3.28) 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 (3.27) 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 [17], 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 $\lambda_{\mathrm{s}1}$ is the leftmost eigenvalue of the pencil $(H_{\mathrm{s}}, M_{\mathrm{s}})$ with associated eigenvector u_{s}, then appropriately padding u_{s} with zeros to obtain a vector $u \in \mathrm{I\!R}^n$, we have that $\lambda_1 \leq \rho_M(u) = u_{\mathrm{s}}^T H_{\mathrm{s}} u_{\mathrm{s}} / u_{\mathrm{s}}^T M_{\mathrm{s}} u_{\mathrm{s}} = \lambda_{\mathrm{s}1}$. 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 discovered $\lambda_{\mathrm{C}} \in \mathcal{G}$ 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 §3.2.3 is to estimate the root of (2.6) by replacing $\pi(\lambda; \beta)$ by its k-th order Taylor approximant $\pi_k(\delta; \beta)$ for some suitable β. As we mentioned in §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,

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

In practice, the computationally viable under-estimate $\lambda_+ = \lambda_{\mathrm{T}}$ for

$$\lambda_{\mathrm{T}} = \max \left(\lambda_1(-1), \lambda_2(-\tfrac{2}{3}), \lambda_3(-\tfrac{2}{5}) \right) \tag{3.29}$$

usually suffices. 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

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

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 §3.3.6.

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 [17] 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 $D_k = \text{diag}(d_{i,i})$, $1 \le i \le k$, with $d_{k,k} \le 0$. If y_k satisfies $L_k^T y_k = e_k$ and $w_k = (y_k^T \ 0)^T$, it follows that

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

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 $H + \lambda_{\text{C}} M$—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 $\lambda_{\text{C}} \in \mathcal{L}$, fast (quadratic) convergence is assured using Newton's method. However, as we discussed in §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 $\pi(\lambda_{\text{C}} + \delta) \equiv \pi(\lambda_{\text{C}} + \delta; 2)$, and $\lambda_{2k+1}(2)$ 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.1);

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

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

will give quartic global convergence, which suffices for all practical purposes. 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 3.2 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 $\theta_k \in [\theta_\varepsilon, 1]$ and $\gamma_k \in [1, 2n_k - \gamma_\varepsilon]$ are now assigned; the restrictions imposed by θ_ε and $2n_k - \gamma_\varepsilon$ are required to guarantee (ultimately) that $\lambda_k^A > -\lambda_1$, while the restrictions $\theta_k \leq 1$ and $\gamma_k \geq 1$ are needed to ensure that the

---

Algorithm 3.2: Potential hard case

Given $\lambda_0^A > -\lambda_1$ and z_0 such that $\|z_0\|_M = 1$.
Set real constants $0 < \theta_\varepsilon \leq 1$, $0 < \gamma_\varepsilon \leq 1$, 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 $\theta_k \in [\theta_\varepsilon, 1]$ and $\gamma_k \in [1, 2n_k - \gamma_\varepsilon]$.
    Set $\lambda_{k+1}^A = -\rho_M(z_{k+1}) + \theta_k\left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\gamma_k}$.

sequence $\{\lambda_k^A\}$ is monotonically decreasing (see Lemma 3.8). 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. This process is then repeated.

Numerous standard results relating to both inverse iteration and the Rayleigh quotient may be found in [9,36,43]; for simplicity, these results typically assume that the eigenvalue for which convergence occurs is simple and that $M = I$. Lemmas 3.6 and 3.7 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 3.6 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 [43, see p.204] to the transformed problem in the "bar" variables and then transforming back.   □

Lemma 3.7 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 $\mathrm{gap}(\lambda_1) < \infty$. 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 $|\lambda_1 - \mu| < \mathrm{gap}(\lambda_1)/2$. If $\{z_k\}$ denotes the sequence of inverse iterates, then

$$|z_k - {\scriptstyle(-/+)}u| = O\left(\left|\frac{\mu - \lambda_1}{\mu - \lambda_J}\right|^k\right) \quad \text{and} \quad |\rho_M(z_k) - \lambda_1| = O\left(\left|\frac{\mu - \lambda_1}{\mu - \lambda_J}\right|^{2k}\right)$$

as $k \to \infty$, where λ_J is defined by $|\lambda_J - \lambda_1| = \mathrm{gap}(\lambda_1)$ 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.

Note: the statement $z_0 \not\perp \mathrm{eig}(\lambda_1)$ should be interpreted in the inner-product $\langle x, y \rangle_M = x^T M y$.

Proof. Essentially, the proof follows from [43, see p.204–207]. Specifically, we may apply [43, Thm 27.2]—taking multiple eigenvalues into account—to the transformed problem in "bar" variables and then transform back. $\square$

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 IR^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 3.2.

Lemma 3.8 Let λ_1 be the left-most eigenvalue of the pencil (H, M). Then the sequence $\{\lambda_k^A\}$ generated by Algorithm 3.2 satisfies

  (i) $\lambda_{k+1}^A \le \lambda_k^A$ and

  (ii) $\lambda_{k+1}^A > -\lambda_1$

for all $k \ge 0$ provided λ_0^A is sufficiently close to $-\lambda_1$. Moreover, *strict* inequality holds in part (i) if either $\theta_k < 1$ or $\gamma_k > 1$.

Proof. The proof is by induction. If λ_0^A is sufficiently close to $-\lambda_1$ then Lemma 3.7 implies

$$0 < \lambda_0^A + \rho_M(z_1) = (\lambda_0^A + \lambda_1) + (-\lambda_1 + \rho_M(z_1)) < 1.$$

Using this fact, $\theta_0 \le 1$, and $\gamma_0 \ge 1$, we may write

$$\lambda_1^A = -\rho_M(z_1) + \theta_0 (\lambda_0^A + \rho_M(z_1))^{\gamma_0} \le -\rho_M(z_1) + \lambda_0^A + \rho_M(z_1) = \lambda_0^A,$$

so that part (i) holds for $k = 0$. We now show that part (ii) is true for $k = 0$, but first note that part (ii) is true for $k = -1$ by construction of the algorithm. Also, the constants that are implicitly defined by the $O(\cdot)$ notation in Lemma 3.7 are uniformly bounded over all iterations of the algorithm, provided that we choose λ_0^A sufficiently close to $-\lambda_1$; we denote this uniform bound by $\hat{c}$. We then assume that λ_0^A is sufficiently

close to $-\lambda_1$ so that

$$\frac{\hat{c}(\lambda_0^A + \lambda_1)^{\gamma_\varepsilon}}{\kappa} \leq \theta_\varepsilon/2 \quad \text{and} \quad |\lambda_0^A + \lambda_1| < 1, \quad \text{where } \kappa \stackrel{\text{def}}{=} \min\left(1, \text{gap}(\lambda_1)^{2n_u}\right). \quad (3.30)$$

We then have

$$\lambda_1^A + \lambda_1 = -\rho_M(z_1) + \lambda_1 + \theta_0(\lambda_0^A + \rho_M(z_1))^{\gamma_0} \quad (3.31)$$

$$\geq -\hat{c}\left(\frac{\lambda_0^A + \lambda_1}{\text{gap}(\lambda_1)}\right)^{2n_0} + \theta_0(\lambda_0^A + \lambda_1)^{\gamma_0} \quad (3.32)$$

$$= (\lambda_0^A + \lambda_1)^{\gamma_0}\left(\theta_0 - \frac{\hat{c}(\lambda_0^A + \lambda_1)^{2n_0 - \gamma_0}}{\text{gap}(\lambda_1)^{2n_0}}\right) \quad (3.33)$$

$$\geq (\lambda_0^A + \lambda_1)^{\gamma_0}\left(\theta_\varepsilon - \frac{\hat{c}(\lambda_0^A + \lambda_1)^{\gamma_\varepsilon}}{\kappa}\right) \quad (3.34)$$

$$\geq \frac{\theta_\varepsilon(\lambda_0^A + \lambda_1)^{\gamma_0}}{2} > 0. \quad (3.35)$$

Equation (3.31) follows from the definition of λ_1^A, while equation (3.32) follows from Lemma 3.7 and the inequality $\lambda_0^A + \rho_M(z_1) \geq \lambda_0^A + \lambda_1$. Equations (3.33)–(3.35) follow from factorization, the restrictions that θ_ε and γ_ε place on θ_0 and γ_0, and (3.30). Therefore, part (ii) holds for $k = 0$.

Next assume that $\lambda_{j+1}^A \leq \lambda_j^A$ and $\lambda_{j+1}^A > -\lambda_1$ for all $0 \leq j \leq k - 1$. We now show that it is true for $j = k$. The inductive hypothesis implies that λ_j^A is greater than $-\lambda_1$ and that it approximates $-\lambda_1$ at least as well as λ_0^A does for $j = 0, \ldots, k$. It then follows from Lemma 3.7 that

$$0 < \lambda_k^A + \rho_M(z_{k+1}) = (\lambda_k^A + \lambda_1) + (-\lambda_1 + \rho_M(z_{k+1})) < 1. \quad (3.36)$$

Using this fact, $\theta_k \leq 1$, and $\gamma_k \geq 1$, we may write

$$\lambda_{k+1}^A = -\rho_M(z_{k+1}) + \theta_k(\lambda_k^A + \rho_M(z_{k+1}))^{\gamma_k} \leq -\rho_M(z_{k+1}) + \lambda_k^A + \rho_M(z_{k+1}) = \lambda_k^A,$$

so that part (i) holds for $j = k$. Using essentially the same argument as for equations (3.31)–(3.35), we have

$$\lambda_{k+1}^A + \lambda_1 = -\rho_M(z_{k+1}) + \lambda_1 + \theta_k(\lambda_k^A + \rho_M(z_{k+1}))^{\gamma_k}$$

$$\geq -\hat{c}\left(\frac{\lambda_k^A + \lambda_1}{\text{gap}(\lambda_1)}\right)^{2n_k} + \theta_k(\lambda_k^A + \lambda_1)^{\gamma_k}$$

$$= (\lambda_k^A + \lambda_1)^{\gamma_k}\left(\theta_k - \frac{\hat{c}(\lambda_k^A + \lambda_1)^{2n_k - \gamma_k}}{\text{gap}(\lambda_1)^{2n_k}}\right)$$

$$\geq (\lambda_k^A + \lambda_1)^{\gamma_k}\left(\theta_\varepsilon - \frac{\hat{c}(\lambda_k^A + \lambda_1)^{\gamma_\varepsilon}}{\kappa}\right)$$

$$\geq \frac{\theta_\varepsilon(\lambda_k^A + \lambda_1)^{\gamma_k}}{2} > 0,$$

so that part (ii) holds for $j = k$. In addition, it is easily verified that strict inequality holds in part (i) if either $\theta_k < 1$ or $\gamma_k > 1$. $\qquad\square$

Theorem 3.9 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 3.2 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

$$|\lambda_{k+1}^A + \lambda_1| = |-\rho_M(z_{k+1}) + \lambda_1 + \theta_k\left(\lambda_k^A + \rho_M(z_{k+1})\right)^{\bar{\gamma}}| \tag{3.37}$$

$$\leq |-\rho_M(z_{k+1}) + \lambda_1| + \theta_k|\lambda_k^A + \rho_M(z_{k+1})|^{\bar{\gamma}} \tag{3.38}$$

$$\leq |-\rho_M(z_{k+1}) + \lambda_1| + 2^{\bar{\gamma}}\theta_k|\lambda_k^A + \lambda_1|^{\bar{\gamma}} \tag{3.39}$$

$$\leq \hat{c}\left|\frac{\lambda_k^A + \lambda_1}{\text{gap}(\lambda_1)}\right|^{2\bar{n}} + 2^{\bar{\gamma}}|\lambda_k^A + \lambda_1|^{\bar{\gamma}} \tag{3.40}$$

$$= \left[\frac{\hat{c}(\lambda_k^A + \lambda_1)^{2\bar{n}-\bar{\gamma}}}{\text{gap}(\lambda_1)^{2\bar{n}}} + 2^{\bar{\gamma}}\right]|\lambda_k^A + \lambda_1|^{\bar{\gamma}} \tag{3.41}$$

$$\leq \left[\frac{\hat{c}(\lambda_0^A + \lambda_1)^{\gamma_\varepsilon}}{\text{gap}(\lambda_1)^{2\bar{n}}} + 2^{\bar{\gamma}}\right]|\lambda_k^A + \lambda_1|^{\bar{\gamma}} = c|\lambda_k^A + \lambda_1|^{\bar{\gamma}}, \tag{3.42}$$

where

$$c = \frac{\hat{c}(\lambda_0^A + \lambda_1)^{\gamma_\varepsilon}}{\text{gap}(\lambda_1)^{2\bar{n}}} + 2^{\bar{\gamma}} \quad \text{and } \hat{c} \text{ was defined in the proof of Lemma 3.8.}$$

Equations (3.37) and (3.38) follow from the definition of λ_{k+1}^A and the triangle inequality. Equation (3.39) follows from the inequality $\lambda_k^A + \rho_M(z_{k+1}) \leq 2(\lambda_k^A + \lambda_1)$, which follows from Lemma 3.7 for λ_0^A sufficiently close to $-\lambda_1$. Equations (3.40) and (3.41) follow from Lemma 3.7, the definition of $\text{gap}(\lambda_1)$, the inequality $\theta_k \leq 1$, and factorization. Finally, equation (3.42) follows from the properties of $\{\lambda_k^A\}$ described in Lemma 3.8 and definition of γ_k. $\qquad\square$

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 $2n_k - \gamma_\varepsilon > \bar{\gamma}$ (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 3.2 once $\lambda_\text{U} - \lambda_\text{L}$ is relatively small. In the easy case, Algorithm 3.2 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 problems (1.1) and (1.2) 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 § 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 $\lambda_C \in \mathcal{G}$, we compute a new estimate λ_+ as follows. Let λ_T be the Taylor series under-estimate (3.29), let λ_{SL} be the best under-estimate of λ_S 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 $(0, 1)$—for example, in practice we use $\theta_g = 0.5$ and $\theta_H = 0.9$. Then whenever $|\lambda_C - \lambda_{SL}| \leq \theta_H \lambda_C$, we suspect we might be in the hard case, so use the Rayleigh quotient $-\rho_M(z_{k+1})$ to try to improve λ_{SL} and subsequently set

$$\lambda_+ = \min\left(\lambda_L + \theta_G(\lambda_C - \lambda_L), \max\left(\lambda_T, \lambda_L, \lambda_{SL} + \left(\frac{\lambda_C - \lambda_{SL}}{\lambda_C}\right)^{\bar{\gamma}}\right)\right).$$

Otherwise, if $\lambda_T \geq \lambda_L$, the Taylor estimate gives the best estimate in $\mathcal{F}$ found so far, and we assign $\lambda_+ = \lambda_T$. If neither happens, we simply revert to improving the interval of uncertainty by setting $\lambda_+ = \lambda_L + \theta_G(\lambda_C - \lambda_L)$.

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 $\lambda_L > 0$, so any investigation should be delayed until the initial λ_L has been computed [34].

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 $\lambda_- \in \mathcal{L} \cup \{\lambda_S\}$, 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 $\lambda_L = 0$, and as $\max(\gamma\sqrt{\lambda_L \lambda_U}, \lambda_L + \theta(\lambda_U - \lambda_L))$ for some $\theta \in (0, 1)$ and $\gamma \in \{0, 1\}$ (c.f., Algorithm 3.1) otherwise.

3.4   The regularisation problem

Much of what we described in the previous section may easily be adapted to solve the regularisation problem (1.2). The main differences result from our needing to solve the secular equation (2.16) rather than $\|x(\lambda)\|_M = \Delta$; there is no "interior solution" case to worry about.

To find a point in $\mathcal{L}$ for the regularisation problem, Algorithm 3.1 is still applicable. Finding initial values for λ_{L} and λ_{U} is slightly different. The Rayleigh-quotient bound (3.25) becomes

$$(\lambda_* + \lambda_1)^2 \leq \frac{\overline{x}_*^T (\overline{H} + \lambda_* I)^2 \overline{x}_*}{\overline{x}_*^T \overline{x}_*} = \frac{\sigma^{2/(p-2)} \|\overline{c}\|^2}{\lambda_*^{2/(p-2)}} = \frac{\sigma^{2/(p-2)} \|c\|_{M^{-1}}^2}{\lambda_*^{2/(p-2)}} \leq (\lambda_* + \lambda_n)^2,$$

leading to the inequalities

$$(\lambda_* + \lambda_1)\lambda_*^{1/(p-2)} \leq \sigma^{1/(p-2)} \|c\|_{M^{-1}} \leq (\lambda_* + \lambda_n)\lambda_*^{1/(p-2)}. \tag{3.43}$$

Replacing λ_1 and λ_n by outer bounds in (3.43) gives simple nonlinear inequalities from which bounds on λ_* may be deduced; when $p = 3$, this involves solving quadratic equations, while for other p iteration may be required. Outer bounds, such as that provided by the generalization of the Gershgorin theorems described in §3.3.2, may be used. In addition, as before, the lower bound $\lambda_* \geq \lambda_{\text{S}} = \max(0, -\lambda_1)$ may be employed, so any known upper bound on λ_1 is useful. Inverse iteration/Rayleigh-quotient-based estimates as described in §3.3.3, 3.3.4 and 3.3.6 are appropriate. The only real issue is in how to compute improvements from a given λ_{C} in $\mathcal{G}$ or $\mathcal{L}$.

3.4.1   New estimates from $\mathcal{G}$

Consider first a $\lambda_{\text{C}} \in \mathcal{G}$. We may then follow the reasoning in §3.2.3 to compute $\lambda_k(\beta) = \lambda_{\text{C}} + \delta_k(\beta)$, where $\delta_k(\beta)$ is the least-negative root of

$$\pi_k(\delta; \beta) = (\lambda/\sigma)^{\beta/(p-2)}, \tag{3.44}$$

the polynomial $\pi_k(\delta; \beta)$ is the k-th degree Taylor approximant to $\pi(\lambda; \beta)$ at λ_{C}, and $\beta \in \mathcal{B}_k$. If $\beta/(p-2)$ is an integer, the resulting equation is (or may be converted to) a polynomial system, and easily solved. For non-integral $\beta/(p-2)$, we can use a derivative-based root finder. Alternatively, as we saw in §3.2.3, we might instead pick $\lambda_{k,\ell}(\beta) = \lambda_{\text{C}} + \delta_{k,\ell}(\beta)$ from the least-negative root $\delta_{k,\ell}(\beta)$ of

$$\pi_k(\delta; \beta) = \theta_\ell(\delta; \beta), \tag{3.45}$$

where $\theta_\ell(\delta; \beta)$ is the ℓ-th order Taylor approximation to $(\lambda/\sigma)^{\beta/(p-2)}$; if $\beta > 0$ this is only permissible if $\lceil \beta/(p-2) \rceil$ is even and $\ell = \lceil \beta/(p-2) \rceil + 2j$ for some $j \geq 0$. Since such a root is inferior to $\delta_k(\beta)$, this alternative is only worthwhile if solving (3.44) is viewed unreasonable.

Once again, 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,

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

As before, the computationally viable under-estimate

$$\lambda_+ = \max\left(\lambda_1(-1), \lambda_2(-\tfrac{2}{3}), \lambda_3(-\tfrac{2}{5})\right)$$

usually suffices. The less expensive but likely inferior alternative

$$\lambda_+ = \max\left(\lambda_{1,1}(-1), \lambda_{2,2}(-\tfrac{2}{3}), \lambda_{3,3}(-\tfrac{2}{5})\right)$$

is another possibility.

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

Once we have found $\lambda_\mathrm{C} \in \mathcal{L}$, fast monotonic convergence is assured if we compute $\lambda_k(\beta)$ via (3.44) just as in §3.44 so long as k is odd and $\beta_k \in \mathcal{B}_k$ (and thus $\pi_k(\delta; \beta) < \pi(\lambda_\mathrm{C} + \delta)$ for positive δ). Indeed, the convergence will be of asymptotic Q-order $k+1$ (see Theorem A.1). If solving (3.44) is considered unreasonable for particular p and β, the (likely inferior) alternative using (3.45) is available so long as $\ell = \lceil \beta/(p-2) \rceil + 2j$ for some $j \geq 0$ when $\beta > 0$ or $\ell = 2j + 1$ when $\beta < 0$ (see §3.2.3), and then the convergence will be of Q-order $\min(k, \ell) + 1$ (see Theorem A.2).

Once again, the impractical

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

would be ideal, but both

$$\lambda_+ = \max\left(\lambda_1(-1), \lambda_3(2), \lambda_3(-\tfrac{2}{5})\right)$$

and

$$\lambda_+ = \max\left(\lambda_{1,1}(-1), \lambda_{3,2/(p-2)}(2), \lambda_{3,3}(-\tfrac{2}{5})\right)$$

will give quartic global convergence which is sufficient for all practical purposes. Higher-order estimates might be added if factorization of $H + \lambda M$ dominates other costs.

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[2] [22]. The packages provide a number of options.

---

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

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 (2.3) and this is particularly important in the "hard" or "nearly hard" cases. Any *a priori* knowledge of initial λ_L, λ_U and λ may optionally be provided, and this 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 $\lambda_C \in \mathcal{N}$ mentioned in §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[3] dgqt which is an implementation of the Moré-Sorensen [34] approach; we slightly modified this software to record and print required details.

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 §3.3.7. Running TRS when $c = c_1$ and stopping as soon as

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

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 $\left| \|x(\lambda)\| - \Delta \right|$ and the change in λ. By contrast dgqt (started with the same initial λ) yields

---

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

```
     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
```

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.

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

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

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
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 $\lambda_{\rm L}$, 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".

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 [21] 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 4.1 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 Figure 4.1; 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 [10] for a formal definition of performance profiles and a discussion of their properties.
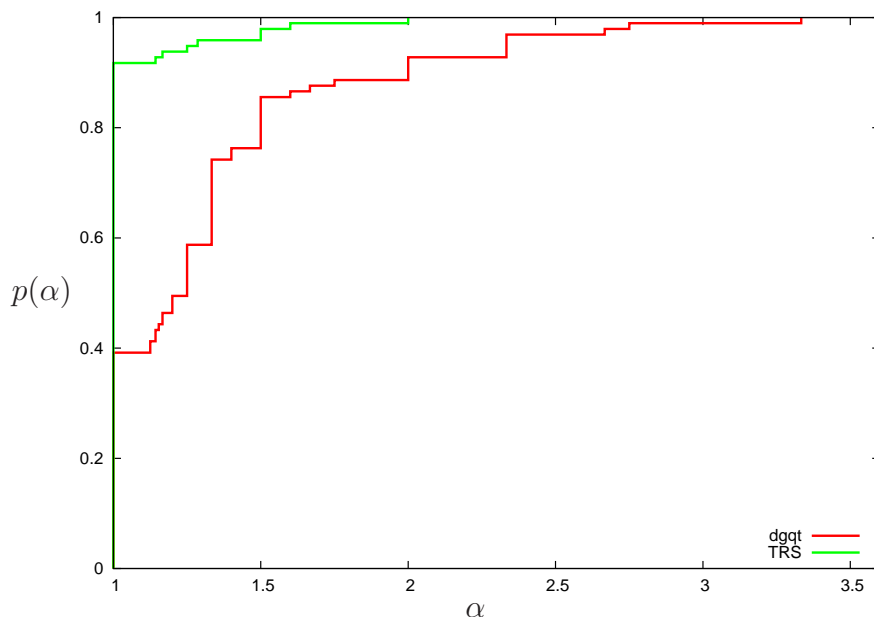
Figure 4.1: Performance profile for the numbers of factorizations required to solve the sample set of `CUTEr` problems using `dgqt` compared to those using `TRS`.

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 worst performance is for problem `GROWTHLS`, and in details 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
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
```

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						

Table 4.1: The numbers of factorizations required to solve the sample set of CUTEr problems using `dgqt` compared to those using TRS.

```
 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 $[\lambda_{\mathrm{L}}, \lambda_{\mathrm{U}}]$ 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.

Since both TRS and RQS are designed to cope with large problems, we illustrate its performance on bigger problems from the CUTEr test set. In Table 4.2, 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 T340, single Core2 Quad Q9550 processor(2.83GHz, 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.

	TRS		RQS	
n	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

Table 4.2: 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.

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

In Table 4.3, we illustrate the behaviour on other large CUTEr examples. Although the actual behaviour clearly depends on sparsity, and particularly on fill-in—the problem NONCVUN 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.

problem	n	TRS		RQS	
		factorizations	CPU	factorizations	CPU
SCURLY10	100000	14	2.41	12	1.40
SCOSINE	100000	20	1.34	18	0.91
NONCVXUN	100000	2	287.17	2	299.98
INDEF	100000	5	0.65	2	0.31
FLETCBV2	100000	4	0.61	3	0.31
DIXMAANA	90000	3	0.42	3	0.38
FMINSRF2	90000	5	15.05	4	12.07

Table 4.3: The numbers of factorizations 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.

5   Comments and conclusions

Our aim has been to revisit the popular Gay-Moré-Sorensen [17, 34] 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 [22].

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 [5, 14, 15, 20, 25, 40, 41] 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, 27, 28, 37, 39].

Some iterative methods [5, 20] 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 §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 (2.3). 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 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 [7, 18]. To verify the latter condition and then to solve (5.1), any inertia-revealing symmetric, indefinite factorization package is appropriate (see §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 §3.3 and §3.4, although non-trivial initial values (*cf.* §3.3.2) for λ_{L} and λ_{U} are not obvious.

References

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, USA, 2008.

[3] P. Amestoy, I. S. Duff, S. Pralet, and C. Voemel. Adapting a parallel sparse direct solver to SMP architectures. *Parallel Computing*, 29(11–12):1645–1668, 2003.

[4] T. M. Apostol. *Mathematical Analysis*. Addison-Wesley Publishing Company, Reading, Massachusetts, USA, second edition, 1974.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic overestimation methods for unconstrained optimization. Technical Report RAL-TR-2007-007, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2007.

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Trust-region and other regularisations of linear least-squares problems. Technical Report RAL-TR-2008-005, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2008.

[7] Y. Chabrillac and J.-P. Crouzeix. Definiteness and semidefiniteness of quadratic forms revisited. *Linear Algebra and its Applications*, 63:283–292, 1984.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, 2000.

[9] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[10] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[11] H. S. Dollar. 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.

[12] I. S. Duff. MA57 - a code for the solution of sparse symmetric definite and indefinite systems. *ACM Transactions on Mathematical Software*, 30(2):118–144, 2004.

[13] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[14] J. B. Erway and P. E. Gill. A subspace minimization method for the trust-region step. Technical Report NA 08-1, Department of Mathematics, University of California, San Diego, 2008.

[15] J. B. Erway, P. E. Gill, and J. D. Griffin. Iterative methods for finding a trust-region step. Technical Report NA 07-2, Department of Mathematics, University of California, San Diego, 2007.

[16] W. Gander. *On the linear least squares problem with a quadratic constraint.* PhD thesis, Computer Science Department, Stanford University, Stanford, California, USA, 1978.

[17] D. M. Gay. Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, 2(2):186–197, 1981.

[18] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem. *Mathematical Programming*, 32(1):90–99, 1985.

[19] N. I. M. Gould, Y. Hu, and J. A. Scott. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. *ACM Transactions on Mathematical Software*, 32(2), 2007. Article 10.

[20] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[21] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[22] N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software*, 29(4):353–372, 2003.

[23] N. I. M. Gould and J. A. Scott. A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations. *ACM Transactions on Mathematical Software*, 30(3):300–325, 2004.

[24] A. Griewank. 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, UK, 1981.

[25] W. W. Hager. Minimizing a quadratic over a sphere. *SIAM Journal on Optimization*, 12(1):188–208, 2001.

[26] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, AERE Harwell Laboratory, Harwell, Oxfordshire, England, 1973.

[27] J. D. Hogg. A DAG-based parallel Cholesky factorization for multicore systems. Technical Report RAL-TR-2008-029, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2008.

[28] J. D. Hogg, J. K. Reid, and J. A. Scott. Sparse symmetric linear solvers for multicore architectures. Technical Report (to appear), Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2009.

[29] A. Melman. Numerical solution of a secular equation. *Numerische Mathematik*, 69(4):483–493, 1995.

[30] A. Melman. A numerical comparison of methods for solving secular equations. *Journal of Computational and Applied Mathematics*, 86(1):237–249, 1997.

[31] A. Melman. A unifying convergence analysis of second-order methods for secular equations. *Mathematics of Computation*, 66(217):333–344, 1997.

[32] A. Melman. Analysis of third-order methods for secular equations. *Mathematics of Computation*, 67(221):271–286, 1998.

[33] J. J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*, pages 258–287, Heidelberg, Berlin, New York, 1983. Springer Verlag.

[34] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[35] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[36] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1980. Reprinted as *Classics in Applied Mathematics 20*, SIAM, Philadelphia, USA, 1998.

[37] J. K. Reid and J. A. Scott. An out-of-core sparse Cholesky solver. Technical Report RAL-TR-2006-013, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2006.

[38] C. Reinsch. Smoothing by spline functions II. *Numerische Mathematik*, 16(5):451–454, 1971.

[39] O. Schenk, M. Christen, and H. Burkhart. Algorithmic performance studies on graphics processing units. *Journal of Parallel Distributed Computation*, 68:1360–1369, 2008.

[40] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[41] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88, London, 1981. Academic Press.

[42] J. F. Traub. *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[43] L. N. Trefethen and D. Bai. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[44] M. Weiser, P. Deuflhard, and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optimization Methods and Software*, 22(3):413–431, 2007.

Appendix A

Let $\phi \in C^{m+1}, \theta \in C^1 : \mathrm{I\!R} \to \mathrm{I\!R}$, 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.1 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

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

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 equation (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} 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

$$\phi_m(\delta_k; \lambda_k) - \phi_m(\epsilon_k; \lambda_k) = \sum_{i=1}^{m} \frac{\phi^{(i)}(\lambda_k)}{i!} (\delta_k^i - \epsilon_k^i)$$

$$= (\lambda_{k+1} - \lambda_*) \left(\phi^{(1)}(\lambda_k) + \sum_{i=2}^{m} \frac{\phi^{(i)}(\lambda_k)}{i!} \sum_{j=0}^{i-1} \delta_k^j \epsilon_k^{i-j-1} \right)$$

$$(A.7)$$

in which case (A.6) gives

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

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 $\lambda_k - \lambda_*$, $|\kappa_k| \leq 2|\phi^{(m+1)}(\lambda_*)/(\phi^{(1)}(\lambda_*) - \theta^{(1)}(\lambda_*))| < \infty$ because x_* is a simple root, the required convergence estimate follows from (A.8). □

If we replace $\theta(\lambda)$ by a Taylor approximation we have the following simple generalisation.

Theorem A.2 Suppose that $\phi \in C^{m+1}, \theta \in C^{\ell+1} : \mathrm{I\!R} \to \mathrm{I\!R}$ where $\min(m, \ell) \geq 1$, that λ_* is a simple root of $\phi(\lambda) = \theta(\lambda)$. Then for all λ_k sufficiently close to λ_*,

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

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

Proof. The proof follows the same general arguments as its predecessor. Under the current assumptions of this theorem and using an argument similar to the previous theorem, we may again show that

$$\lim_{\lambda_k \to \lambda_*} \delta_k = 0. \qquad (A.9)$$

Taylor's theorem and the fact that λ_* is a root give

$$\phi(\lambda_*) = \phi_m(\epsilon_k; \lambda_k) + \frac{\phi^{(m+1)}(\zeta_k)}{(m+1)!} \epsilon_k^{m+1} = \theta(\lambda_*) = \theta_\ell(\epsilon_k; \lambda_k) + \frac{\theta^{(\ell+1)}(\chi_k)}{(\ell+1)!} \epsilon_k^{\ell+1}$$

for some ζ_k and χ_k between λ_k and λ_*, and hence

$$\phi_m(\delta_k; \lambda_k) - \phi_m(\epsilon_k; \lambda_k) - \theta_\ell(\delta_k; \lambda_k) + \theta_\ell(\epsilon_k; \lambda_k) = \frac{\phi^{(m+1)}(\zeta_k)}{(m+1)!}\epsilon_k^{m+1} - \frac{\theta^{(\ell+1)}(\chi_k)}{(\ell+1)!}\epsilon_k^{\ell+1}.$$
(A.10)

But then, using (A.7) and the analogous expansion for $\theta_\ell(\delta_k; \lambda_k) - \theta_\ell(\epsilon_k; \lambda_k)$, we deduce

$$(\lambda_{k+1} - \lambda_*)\left(\phi^{(1)}(\lambda_k) - \theta^{(1)}(\lambda_k) + \sum_{i=2}^{m}\frac{\phi^{(i)}(\lambda_k)}{i!}\sum_{j=0}^{i-1}\delta_k^j\epsilon_k^{i-j-1} - \sum_{i=2}^{\ell}\frac{\theta^{(i)}(\lambda_k)}{i!}\sum_{j=0}^{i-1}\delta_k^j\epsilon_k^{i-j-1}\right)$$
$$= \frac{\phi^{(m+1)}(\zeta_k)}{(m+1)!}\epsilon_k^{m+1} - \frac{\theta^{(\ell+1)}(\chi_k)}{(\ell+1)!}\epsilon_k^{\ell+1}$$

which leads to the desired estimate for sufficiently small $\lambda_k - \lambda_*$.   □

Appendix B

Let

$$\pi(\lambda) \overset{\text{def}}{=} \|x(\lambda)\|_M^2 = \sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda + \lambda_i)^2},$$

where $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Let

$$\pi(\lambda; \beta) \overset{\text{def}}{=} \|x(\lambda)\|_M^\beta = [\pi(\lambda)]^{\frac{\beta}{2}}.$$

Differentiating $\pi(\lambda; \beta)$ with respect to λ we obtain

$$\begin{aligned}
\pi^{(1)}(\lambda; \beta) &= \tfrac{\beta}{2}[\pi(\lambda)]^{\frac{\beta}{2}-1}\pi^{(1)}(\lambda), \\
\pi^{(2)}(\lambda; \beta) &= \tfrac{\beta}{2}[\pi(\lambda)]^{\frac{\beta}{2}-1}\pi^{(2)}(\lambda) + \tfrac{\beta}{2}\left(\tfrac{\beta}{2}-1\right)[\pi(\lambda)]^{\frac{\beta}{2}-2}\left[\pi^{(1)}(\lambda)\right]^2, \\
\pi^{(3)}(\lambda; \beta) &= \tfrac{\beta}{2}[\pi(\lambda)]^{\frac{\beta}{2}-3}\left([\pi(\lambda)]^2\pi^{(3)}(\lambda) + 3\left(\tfrac{\beta}{2}-1\right)\pi(\lambda)\pi^{(1)}(\lambda)\pi^{(2)}(\lambda)\right. \\
&\qquad\qquad \left. + \left(\tfrac{\beta}{2}-1\right)\left(\tfrac{\beta}{2}-2\right)\left[\pi^{(1)}(\lambda)\right]^3\right)
\end{aligned}$$
(B.1)

$$\begin{aligned}
\text{and } \pi^{(4)}(\lambda; \beta) &= \tfrac{\beta}{2}[\pi(\lambda)]^{\frac{\beta}{2}-4}\left([\pi(\lambda)]^3\pi^{(4)}(\lambda) + 4\left(\tfrac{\beta}{2}-1\right)[\pi(\lambda)]^2\pi^{(1)}(\lambda)\pi^{(3)}(\lambda)\right. \\
&\quad + 3\left(\tfrac{\beta}{2}-1\right)[\pi(\lambda)]^2\left[\pi^{(2)}(\lambda)\right]^2 + 6\left(\tfrac{\beta}{2}-1\right)\left(\tfrac{\beta}{2}-2\right)\pi(\lambda)\left[\pi^{(1)}(\lambda)\right]^2\pi^{(2)}(\lambda) \\
&\quad \left. + \left(\tfrac{\beta}{2}-1\right)\left(\tfrac{\beta}{2}-2\right)\left(\tfrac{\beta}{2}-3\right)\left[\pi^{(1)}(\lambda)\right]^4\right).
\end{aligned}$$
(B.2)

Since $\pi(\lambda) = \sum_{i=1}^{n} \gamma_i^2 / (\lambda + \lambda_i)^2$ it follows that

$$\pi^{(1)}(\lambda) = -2 \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3}, \tag{B.3}$$

$$\pi^{(2)}(\lambda) = 6 \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^4}, \tag{B.4}$$

$$\pi^{(3)}(\lambda) = -24 \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^5}, \tag{B.5}$$

$$\pi^{(4)}(\lambda) = 120 \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^6}. \tag{B.6}$$

Lemma B.3 Let $\beta \geq 0$, then $\pi^{(3)}(\lambda; \beta) \leq 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$; let $\beta \in [-\frac{2}{3}, 0)$, then $\pi^{(3)}(\lambda; \beta) \geq 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$. These inequalities are strict if $\gamma_i \neq 0$ for some i.

Proof. Substituting (B.3)–(B.5) into (B.1) and rearranging we obtain

$$\pi^{(3)}(\lambda; \beta) = \beta \left[\pi(\lambda)\right]^{\beta-6} \eta(\lambda; \beta),$$

where

$$\eta(\lambda; \beta) = (-\beta^2 + 6\beta - 8) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3}\right)^3 - 12 \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2}\right)^2 \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^5}\right)$$

$$+ (18 - 9\beta) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2}\right) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3}\right) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^4}\right).$$

If $\gamma_i = 0$ for $i = 1, \ldots, n$, clearly $\pi^{(3)}(\lambda; \beta) = 0$ for all β. Furthermore, from Lemma B.4 we deduce that $\pi^{(3)}(\lambda; \beta) \leq 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$ and $\beta > 0$; $\pi^{(3)}(\lambda; \beta) \geq 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$ and $\beta \in [-\frac{2}{3}, 0]$. □

Lemma B.4 Let $\sum_{i=1}^{n} \gamma_i^2 > 0$ and

$$
\begin{aligned}
\eta(\lambda; \beta) \;=\; & (-\beta^2 + 6\beta - 8) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right)^3 - 12 \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right)^2 \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^5} \right) \\
& + (18 - 9\beta) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right) \left(\sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda + \lambda_i)^4} \right).
\end{aligned}
$$

The expression $\eta(\lambda; \beta)$ is negative for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$ and $\beta \geq -\frac{2}{3}$.

Proof. The proof is by induction. Let $\gamma_p \neq 0$ and $\gamma_i = 0$ for all $i < p$. Define

$$
\begin{aligned}
\eta(\lambda; \beta; q) \;=\; & (-\beta^2 + 6\beta - 8) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right)^3 - 12 \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right)^2 \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^5} \right) \\
& + (18 - 9\beta) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^4} \right)
\end{aligned}
$$

and note that $\eta(\lambda; \beta) = \eta(\lambda; \beta; n)$.

Now $\eta(\lambda; \beta; p) = (-\beta^2 - 3\beta - 2) \frac{\gamma_p^6}{(\lambda + \lambda_p)^9} < 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$ and $\beta \geq -\frac{2}{3}$.

Assume that $\eta(\lambda; \beta; k - 1) < 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$. Now

$$
\begin{aligned}
\eta(\lambda\, \beta; k) \;=\; & \eta(\lambda; \beta; k - 1) + \left(-\beta^2 - 3\beta - 2 \right) \frac{\gamma_k^6}{(\lambda + \lambda_k)^9} \\
& + \sum_{i=p}^{k-1} \frac{\gamma_k^4 \gamma_i^2 \eta_1(\lambda; \lambda_k; \lambda_i; \beta)}{(\lambda + \lambda_k)^7 (\lambda + \lambda_i)^5} \\
& + \sum_{i,j=p}^{k-1} \frac{\gamma_k^2 \gamma_i^2 \gamma_j^2 \eta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)}{(\lambda + \lambda_k)^5 (\lambda + \lambda_i)^5 (\lambda + \lambda_j)^5},
\end{aligned}
$$

where

$$
\begin{aligned}
\eta_1(\lambda; \lambda_k; \lambda_i; \beta) \;=\; & (-9\beta - 6)(\lambda + \lambda_i)^3 + (-3\beta^2 + 9\beta - 6)(\lambda + \lambda_i)^2 (\lambda + \lambda_k) \\
& + (18 - 9\beta)(\lambda + \lambda_i)(\lambda + \lambda_k)^2 - 12(\lambda + \lambda_k)^3, \\
\eta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta) \;=\; & -12(\lambda + \lambda_i)^3 (\lambda + \lambda_j)^3 - 12(\lambda + \lambda_i)^3 (\lambda + \lambda_k)^3 - 12(\lambda + \lambda_j)^3 (\lambda + \lambda_k)^3 \\
& + \frac{18 - 9\beta}{2} \left((\lambda + \lambda_i)^2 (\lambda + \lambda_j)^3 (\lambda + \lambda_k) + (\lambda + \lambda_i)^3 (\lambda + \lambda_j)^2 (\lambda + \lambda_k) \right) \\
& + \frac{18 - 9\beta}{2} \left((\lambda + \lambda_i)(\lambda + \lambda_j)^3 (\lambda + \lambda_k)^2 + (\lambda + \lambda_i)^3 (\lambda + \lambda_j)(\lambda + \lambda_k)^2 \right) \\
& + \frac{18 - 9\beta}{2} \left((\lambda + \lambda_i)(\lambda + \lambda_j)^2 (\lambda + \lambda_k)^3 + (\lambda + \lambda_i)^2 (\lambda + \lambda_j)(\lambda + \lambda_k)^3 \right) \\
& + (-3\beta^2 + 18\beta - 24)(\lambda + \lambda_i)^2 (\lambda + \lambda_j)^2 (\lambda + \lambda_k)^2
\end{aligned}
$$

The proof is completed by showing that for all $\beta \geq -\frac{2}{3}$, $\eta_1(\lambda; \lambda_k; \lambda_i; \beta)$ is non-positive for all $\lambda \in [\max(0, -\lambda_i), \infty)$ and $\eta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)$ is non-positive for all $\lambda \in [\max(0, -\lambda_i, -\lambda_j), \infty)$. We give a sketch of the methodology for proving these results and refer the reader to [11] for more details.

Consider $\eta_1(\lambda; \lambda_k; \lambda_i; \beta)$. Let $\lambda_{\mathrm{is}} := \max(0, -\lambda_i)$. Using the assumption that $\lambda_i \leq \lambda_k$ we prove the result by showing that

$$\eta_1(\lambda_{\mathrm{is}}; \lambda_k; \lambda_i; \beta), \eta_1^{(1)}(\lambda_{\mathrm{is}}; \lambda_k; \lambda_i; \beta), \eta_1^{(2)}(\lambda_{\mathrm{is}}; \lambda_k; \lambda_i; \beta), \eta_1^{(3)}(\lambda; \lambda_k; \lambda_i; \beta)$$

are all non-positive, where each derivative is with respect to λ. For $\lambda_i \leq 0$, this is straightforward to do. For $\lambda_i > 0$, each derivative will be a polynomial in terms of λ_k and λ_i (and β). We show that these are all non-positive for all $\lambda_k \geq \lambda_i$ by descending through their own derivatives with respect to λ_k and showing that the value of each of these derivatives is non-positive at $\lambda_k = \lambda_i$ for all $\beta \geq -\frac{2}{3}$.

The proof for $\eta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)$ is similar to that of $\eta_1(\lambda; \lambda_k; \lambda_i; \beta)$. Without loss of generality assume that $\lambda_k \geq \lambda_j \geq \lambda_i$. We show that

$$\eta_2(\lambda_{\mathrm{is}}; \lambda_k; \lambda_j; \lambda_i; \beta), \eta_2^{(1)}(\lambda_{\mathrm{is}}; \lambda_k; \lambda_j; \lambda_i; \beta), \ldots, \eta_2^{(6)}(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)$$

are non-negative for all $\beta \geq -\frac{2}{3}$. For $\lambda_i > 0$, each of these derivates will be polynomials in terms of λ_k, λ_j and λ_j. We show that these polynomials are non-positive by descending through their own derivatives with respect to λ_k and showing that these derivatives are non-positive for $\lambda_k = \lambda_j$. Correspondingly, each of these will be polynomials in terms of λ_i and λ_j which we show to be non-positive for all $\lambda_j \geq \lambda_i$ and $\beta \geq -\frac{2}{3}$.

Hence, $\eta(\lambda, k) < 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$. This completes the proof.

$\square$

Lemma B.5 Let $\beta \geq 0$, then $\pi^{(4)}(\lambda; \beta) \geq 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$; let $\beta \in [-\frac{2}{5}, 0)$, then $\pi^{(4)}(\lambda; \beta) \leq 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$. These inequalities are strict if $\gamma_i \neq 0$ for some i.

Proof. Substituting (B.3)–(B.6) into (B.2) we obtain

$$\pi^{(4)}(\lambda; \beta) = \beta \left[\pi(\lambda)\right]^{\beta-8} \zeta(\lambda; \beta),$$

where

$$
\begin{aligned}
\zeta(\lambda;\beta) \;=\; & (\beta-2)(\beta-4)(\beta-6)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^3}\right)^4 \\
& +60\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)^3\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^6}\right) \\
& +27(\beta-2)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)^2\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^4}\right)^2 \\
& +48(\beta-2)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)^2\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^3}\right)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^5}\right) \\
& +18(\beta-2)(\beta-4)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^3}\right)^2\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^4}\right)
\end{aligned}
$$

If $\gamma_i = 0$ for $i = 1,\ldots,n$, clearly $\pi^{(4)}(\lambda;\beta) = 0$ for all β. Furthermore, from Lemma B.6 we deduce that $\pi^{(4)}(\lambda;\beta) \geq 0$ for all $\lambda \in (\lambda_{\mathrm{s}},\infty)$ and $\beta > 0$; $\pi^{(4)}(\lambda;\beta) \leq 0$ for all $\lambda \in (\lambda_{\mathrm{s}},\infty)$ and $\beta \in [-\frac{2}{5}, 0]$.

$\square$

Lemma B.6 Let $\sum_{i=1}^{n}\gamma_i^2 > 0$ and

$$
\begin{aligned}
\zeta(\lambda;\beta) \;=\; & (\beta-2)(\beta-4)(\beta-6)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^3}\right)^4 \\
& +60\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)^3\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^6}\right) \\
& +27(\beta-2)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)^2\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^4}\right)^2 \\
& +48(\beta-2)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)^2\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^3}\right)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^5}\right) \\
& +18(\beta-2)(\beta-4)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^2}\right)\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^3}\right)^2\left(\sum_{i=1}^{n}\frac{\gamma_i^2}{(\lambda+\lambda_i)^4}\right).
\end{aligned}
$$

The expression $\zeta(\lambda;\beta)$ is positive for all $\lambda \in (\lambda_{\mathrm{s}},\infty)$ and $\beta \geq -\frac{2}{5}$.

Proof.   The proof is by induction. Let $\gamma_p \neq 0$ and $\gamma_i = 0$ for all $i < p$. Define

$$
\begin{aligned}
\zeta(\lambda; \beta; q) &= (\beta - 2)(\beta - 4)(\beta - 6) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right)^4 \\
&+ 60 \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right)^3 \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^6} \right) \\
&+ 27(\beta - 2) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right)^2 \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^4} \right)^2 \\
&+ 48(\beta - 2) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right)^2 \left(\sum_{i=q}^{p} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^5} \right) \\
&+ 18(\beta - 2)(\beta - 4) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^2} \right) \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^3} \right)^2 \left(\sum_{i=p}^{q} \frac{\gamma_i^2}{(\lambda + \lambda_i)^4} \right)
\end{aligned}
$$

and note that $\zeta(\lambda; \beta) = \zeta(\lambda; \beta; n)$.

Now $\zeta(\lambda; \beta; p) = (\beta^3 + 6\beta^2 + 11\beta + 6) \frac{\gamma_p^8}{(\lambda + \lambda_p)^{12}} > 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$ and $\beta \geq -\frac{2}{5}$.

Assume that $\zeta(\lambda; \beta; k - 1) > 0$ for all $\lambda \in (\lambda_{\mathrm{s}}, \infty)$. Now

$$
\begin{aligned}
\zeta(\lambda; \beta; k) &= \zeta(\lambda; \beta; k - 1) + (\beta^3 + 6\beta^2 + 11\beta + 6) \frac{\gamma_1^8}{(\lambda + \lambda_k)^{12}} \\
&+ \sum_{i=p}^{k-1} \frac{\gamma_k^6 \gamma_i^2 \zeta_1(\lambda; \lambda_k; \lambda_i; \beta)}{(\lambda + \lambda_k)^{10} (\lambda + \lambda_i)^6} \\
&+ \sum_{i,j=p}^{k-1} \frac{\gamma_k^4 \gamma_i^2 \gamma_j^2 \zeta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)}{(\lambda + \lambda_k)^8 (\lambda + \lambda_i)^6 (\lambda + \lambda_j)^6} \\
&+ \sum_{i,j,l=p}^{k-1} \frac{\gamma_k^2 \gamma_i^2 \gamma_j^2 \gamma_l^2 \zeta_3(\lambda; \lambda_k; \lambda_l; \lambda_j; \lambda_i; \beta)}{(\lambda + \lambda_k)^6 (\lambda + \lambda_i)^6 (\lambda + \lambda_j)^6} (\lambda + \lambda_l)^6,
\end{aligned}
$$

where

$$\zeta_1(\lambda; \lambda_k; \lambda_i; \beta) = \left(18\beta^2 + 42\beta + 24\right)(\lambda + \lambda_i)^4 + 60\,(\lambda + \lambda_k)^4$$
$$+ \left(4\beta^3 - 12\beta^2 + 8\beta\right)(\lambda + \lambda_i)^3\,(\lambda + \lambda_k) - 48\,(2 - \beta)\,(\lambda + \lambda_i)\,(\lambda + \lambda_k)^3$$
$$+ \left(18\beta^2 - 54\beta + 36\right)(\lambda + \lambda_i)^2\,(\lambda + \lambda_k)^2\,,$$

$$\zeta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta) = (75\beta + 30)\,(\lambda + \lambda_i)^4\,(\lambda + \lambda_j)^4$$
$$+ 90\,(\lambda + \lambda_i)^4\,(\lambda + \lambda_k)^4 + 90\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_k)^4$$
$$+ \left(18\beta^2 - 60\beta + 48\right)(\lambda + \lambda_i)^3\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_k)$$
$$+ \left(18\beta^2 - 60\beta + 48\right)(\lambda + \lambda_i)^4\,(\lambda + \lambda_j)^3\,(\lambda + \lambda_k)$$
$$- 48\,(2 - \beta)\,(\lambda + \lambda_i)\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_k)^3$$
$$- 48\,(2 - \beta)\,(\lambda + \lambda_i)^4\,(\lambda + \lambda_j)\,(\lambda + \lambda_k)^3$$
$$- 24\,(2 - \beta)\,(\lambda + \lambda_i)\,(\lambda + \lambda_j)^3\,(\lambda + \lambda_k)^4$$
$$- 24\,(2 - \beta)\,(\lambda + \lambda_i)^3\,(\lambda + \lambda_j)\,(\lambda + \lambda_k)^4$$
$$- 27\,(2 - \beta)\,(\lambda + \lambda_i)^2\,(\lambda + \lambda_j)^2\,(\lambda + \lambda_k)^4$$
$$+ \left(9\beta^2 - 36\right)(\lambda + \lambda_i)^2\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_k)^2$$
$$+ \left(9\beta^2 - 36\right)(\lambda + \lambda_i)^4\,(\lambda + \lambda_j)^2\,(\lambda + \lambda_k)^2$$
$$+ 18\left(\beta^2 - 6\beta + 8\right)(\lambda + \lambda_i)^2\,(\lambda + \lambda_j)^3\,(\lambda + \lambda_k)^3$$
$$+ 18\left(\beta^2 - 6\beta + 8\right)(\lambda + \lambda_i)^3\,(\lambda + \lambda_j)^2\,(\lambda + \lambda_k)^3$$
$$+ \left(6\beta^3 - 54\beta^2 + 156\beta - 144\right)(\lambda + \lambda_i)^3\,(\lambda + \lambda_j)^3\,(\lambda + \lambda_k)^2\,,$$

and

$$\zeta_3(\lambda; \lambda_k; \lambda_l; \lambda_j; \lambda_i; \beta) = 60\,(\lambda + \lambda_i)^4\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_l)^4$$
$$+ 60\,(\lambda + \lambda_i)^4\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_k)^4$$
$$+ 60\,(\lambda + \lambda_i)^4\,(\lambda + \lambda_k)^4\,(\lambda + \lambda_l)^4$$
$$+ 60\,(\lambda + \lambda_j)^4\,(\lambda + \lambda_k)^4\,(\lambda + \lambda_l)^4$$
$$+ (16\beta - 32) \sum_{\substack{\{n_1,n_2,n_3,n_4\} \\ \in \mathrm{perm}(1,3,4,4)}} (\lambda + \lambda_i)^{n_1}\,(\lambda + \lambda_j)^{n_2}\,(\lambda + \lambda_k)^{n_3}\,(\lambda + \lambda_l)^{n_4}$$
$$+ (18\beta - 36) \sum_{\substack{\{n_1,n_2,n_3,n_4\} \\ \in \mathrm{perm}(2,2,4,4)}} (\lambda + \lambda_i)^{n_1}\,(\lambda + \lambda_j)^{n_2}\,(\lambda + \lambda_k)^{n_3}\,(\lambda + \lambda_l)^{n_4}$$
$$+ \left(6\beta^2 - 36\beta + 48\right) \sum_{\substack{(n_1,n_2,n_3,n_4) \\ \in \mathrm{perm}(2,3,3,4)}} (\lambda + \lambda_i)^{n_1}\,(\lambda + \lambda_j)^{n_2}\,(\lambda + \lambda_k)^{n_3}\,(\lambda + \lambda_l)^{n_4}$$
$$+ \left(4\beta^3 - 48\beta^2 + 176\beta - 192\right)(\lambda + \lambda_k)^3\,(\lambda + \lambda_i)^3\,(\lambda + \lambda_j)^3\,(\lambda + \lambda_l)^3\,,$$

where $\mathrm{perm}(i, j, k, l)$ is defined to be the set of distinct permutations of the arguments. For example,

$$\mathrm{perm}(2, 2, 4, 4) = \{(2, 2, 4, 4), (2, 4, 2, 4), (2, 4, 4, 2), (4, 2, 2, 4), (4, 2, 4, 2), (4, 4, 2, 2)\}.$$

Without loss of generality, we assume that $\lambda_k \geq \lambda_j \geq \lambda_i$ in $\zeta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)$, and $\lambda_k \geq \lambda_l \geq \lambda_j \geq \lambda_i$ in $\zeta_3(\lambda; \lambda_k; \lambda_l; \lambda_j; \lambda_i; \beta)$.

The proof is completed by showing that $\zeta_1(\lambda; \lambda_k; \lambda_i; \beta)$, $\zeta_2(\lambda; \lambda_k; \lambda_j; \lambda_i; \beta)$ and $\zeta_3(\lambda; \lambda_k; \lambda_l; \lambda_j; \lambda_i; \beta)$ are non-negative for all $\beta \geq -\frac{2}{5}$ and $\lambda \in [\max(0, -\lambda_i), \infty)$. The methodology for proving these results is similar to that used in the proof of Lemma B.4. We refer the reader to [11] for further details.

$\square$