A Survey of the Trust Region Subproblem Within a Semidefinite Programming Framework

Charles Fortin and Henry Wolkowicz

Department of Combinatorics & Optimization
University of Waterloo

OUTLINE

- Background, Duality
- MS (1983) and GLTR (1999) algorithms within SDP Framework
- RW (1997) Algorithm (Revisited)
- Robustness, Exploiting Sparsity
- Numerics

1 Introduction: Trust Region Subproblem

$$\text{(TRS)} \qquad \begin{array}{rll} q^* = & \min & q(x) := x^T A x - 2a^T x \\ & \text{s.t.} & \|x\| \leq s. \end{array}$$

A is: $n \times n$ symmetric (possibly indefinite) matrix
a is: n-vector
s is: positive scalar (TR radius)
x is: n-vector of unknowns

Many Applications, e.g.:

- forming subproblems for constrained optimization
- regularization of ill-posed problems
- theoretical applications
- etc...
- **AND trust region (TR) methods**

Many Advantages for TR, e.g.:

second order optimality conditions

q-quadratic convergence

BUT: popularity?

sparsity?

hard case?

2 Optimality/Duality

x^* optimal for TRS if and only if

$$\left.\begin{array}{r}(A-\lambda^* I)x^* = a,\\ A-\lambda^* I \succeq 0, \lambda^* \leq 0\end{array}\right\} \quad \text{dual feasibility}$$

$$\|x^*\|^2 \leq s^2 \quad \text{primal feasibility} \tag{2.1}$$

$$\lambda^*(s^2 - \|x^*\|) = 0 \quad \text{complementary slackness}$$

Surprising:

- characterization of optimality of a possibly nonconvex problem
- 2nd order psd necessary conditions hold on all of $\mathbb{R}^n$

2.1 (Dual) Algorithmic Theme

If $A - \lambda I \succ 0$, define

$$x(\lambda) = (A - \lambda I)^{-1} a \quad \text{(stationarity)} \tag{2.2}$$

If optimum is on the boundary and $A - \lambda^* I \succ 0$, then solve

$$\psi(\lambda) := \|x(\lambda)\| - s = 0 \quad \text{(feasibility)}$$

(maintain $A - \lambda I \succ 0, \ \lambda \leq 0$)

2.2 The Hard Case

$x(\lambda) = (A - \lambda I)^\dagger a$; $\mathcal{N}(\cdot)$ null space; $\mathcal{R}(\cdot)$ range space

1. **Easy Case**: $a \notin \mathcal{R}(A - \lambda_1(A)I)$, implies $\lambda^* < \lambda_1(A)$
2. **Hard Case**: $a \in \mathcal{R}(A - \lambda_1(A)I)$
   (a) **Hard Case (case 1)**: $\lambda^* < \lambda_1(A)$
   (b) **Hard Case (case 2)**: $\lambda^* = \lambda_1(A)$ (singularity)
      i. $\|(A - \lambda^* I)^\dagger a\| = s$ or $\lambda^* = 0$ (then the pair $x^* = x(\lambda^*), \lambda^*$ satisfy the optimality conditions)
      ii. $x(\lambda^*) = \|(A - \lambda^* I)^\dagger a\| < s, \lambda^* < 0$ (choose an eigenvector $z \in \mathcal{N}(A - \lambda^* I)$ with $s = \|x(\lambda^*) + \tau z\|$; nonunique optimum)

1. Easy case	2.(a) Hard case (case 1)	2.(b) Hard case (case 2)
$a \notin \mathcal{R}(A - \lambda_1(A)I)$ (implies $\lambda^* < \lambda_1(A)$)	$a \perp \mathcal{N}(A - \lambda_1(A)I)$ and $\lambda^* < \lambda_1(A)$	$a \perp \mathcal{N}(A - \lambda_1(A)I)$ and $\lambda^* = \lambda_1(A)$ (i) $\|(A - \lambda^* I)^\dagger a\| = s$ or $\lambda^* = 0$ (ii) $\|(A - \lambda^* I)^\dagger a\| < s$, $\lambda^* < 0$

Table 2.1: The three different cases for the trust region subproblem. We include two subcases (i) and (ii) for the hard case (case 2).

Lemma 2.1. *(**Handling the Hard Case.**) spectral decomposition of A: $A = \sum_{i=1}^{n} \lambda_i(A) v_i v_i^T = P \Lambda P^T$, v_i othonormal eigenvectors; $\bar{a} = P^T a$.*
$A_i := \sum_{i \in S_i}^{n} \lambda_i(A) v_i v_i^T, \; i = 1, 2, 3$
$S_1 = \{i : \bar{a}_i \neq 0\}, \; S_2 = \{i : \bar{a}_i = 0, \; \lambda_i(A) = \lambda_1(A)\}$
$S_3 = \{i : \bar{a}_i = 0, \lambda_i(A) > \lambda_1(A)\}$.
Define projections $P_i := \sum_{i \in S_i}^{n} v_i v_i^T, \; i = 1, 2, 3$. Then
x^* solves TRS **if and only if**

$$P_3 x^* = 0, \; \textit{and } x^* \textit{ solves TRS when } A \textit{ is replaced by } A_1.$$

Moreover, x^* solves TRS implies: $P_2 x^* = 0$ unless the hard case (case 2(ii)) holds.

2.2.1 Hard Case (case 2); Ill-Posed?

Note: $A - \lambda^* I$ is singular.

Is near hard case ill-conditioned?

Lemma 2.2. *Suppose that the hard case (case 2 (ii)) holds for TRS. Then $\lambda_1(A) < 0$ and TRS is equivalent to the following stable convex program*

$$(TRS_s) \qquad \begin{array}{rll} q_s^* := & \min & q_s(x) := x^T(A - \lambda_1(A))x - 2a^T x + s^2 \lambda_1(A) \\ & s.t. & \|x\| \leq s. \end{array}$$

The equivalence is in the sense that the optimal values satisy $q^* = q_s^*$; and x^* solves TRS if and only if $x^* = x_s^* + z$ with x_s^* optimal for TRS_s and $z \in \mathcal{N}(A - \lambda_1(A))$. ■

3 The Moré-Sorensen (MS) Algorithm 1983

Outline:

(i) Use safeguarding/updating: reduce interval of uncertainty $[\lambda_L, \lambda_U]$ for λ^* and improve upper bound λ_S for $\lambda_1(A)$

(ii) Take a Newton step to implicitly solve $\|(A - \lambda I)^{-1} a\| = s$ for λ.

(iii) If the possibility of the hard case (case 2) is detected ($\|x(\lambda)\| < s$), then take a *primal step* to the boundary while simultaneously reducing the objective function.

(iv) Check if the (unique) unconstrained minimizer exists and is in the strict interior of the trust region.

For $\psi(\lambda) = \|x(\lambda)\| - s$. Use orthogonal diagonalization of A:

$$\psi(\lambda) = \|Q(\Lambda - \lambda I)^{-1} Q^T a\| - s \approx \frac{c_1}{\lambda_1(A) - \lambda} + d,$$

for some constants $c_1 > 0, d$. This function is highly nonlinear for values of λ near $\lambda_1(A)$. convergence for Newton's method.
MS use so-called secular equation

$$\phi(\lambda) := \frac{1}{s} - \frac{1}{\|x(\lambda)\|} = 0. \tag{3.1}$$

(Reinsch and Hebden) since rational structure of $\|x(\lambda)\|^2$, shows that this function is less nonlinear

$$\phi(\lambda) \approx \frac{1}{s} - \frac{\lambda_1(A) - \lambda}{c_2},$$

for some $c_2 > 0$. Also $\phi(\lambda)$ is a convex function strictly increasing on $(-\infty, \lambda_1(A))$.

compute the Newton direction

Algorithm 3.1. *Assume* $\lambda_k \leq 0$ *and* $A - \lambda_k I \succ 0$ *(i.e.* $\lambda_k < \lambda_1(A)$*).*

1. *Factor* $A - \lambda_k I = R^T R$ *(Cholesky factorization).*

2. *Solve, for* x, $R^T R x = a$ $\;(\, x = x(\lambda_k))$.

3. *Solve, for* y, $R^T y = x$.

4. *Let* $\lambda_{k+1} = \lambda_k - \left[\frac{||x||}{||y||}\right]^2 \left[\frac{(||x|| - s)}{s}\right]$ *(Newton step).*

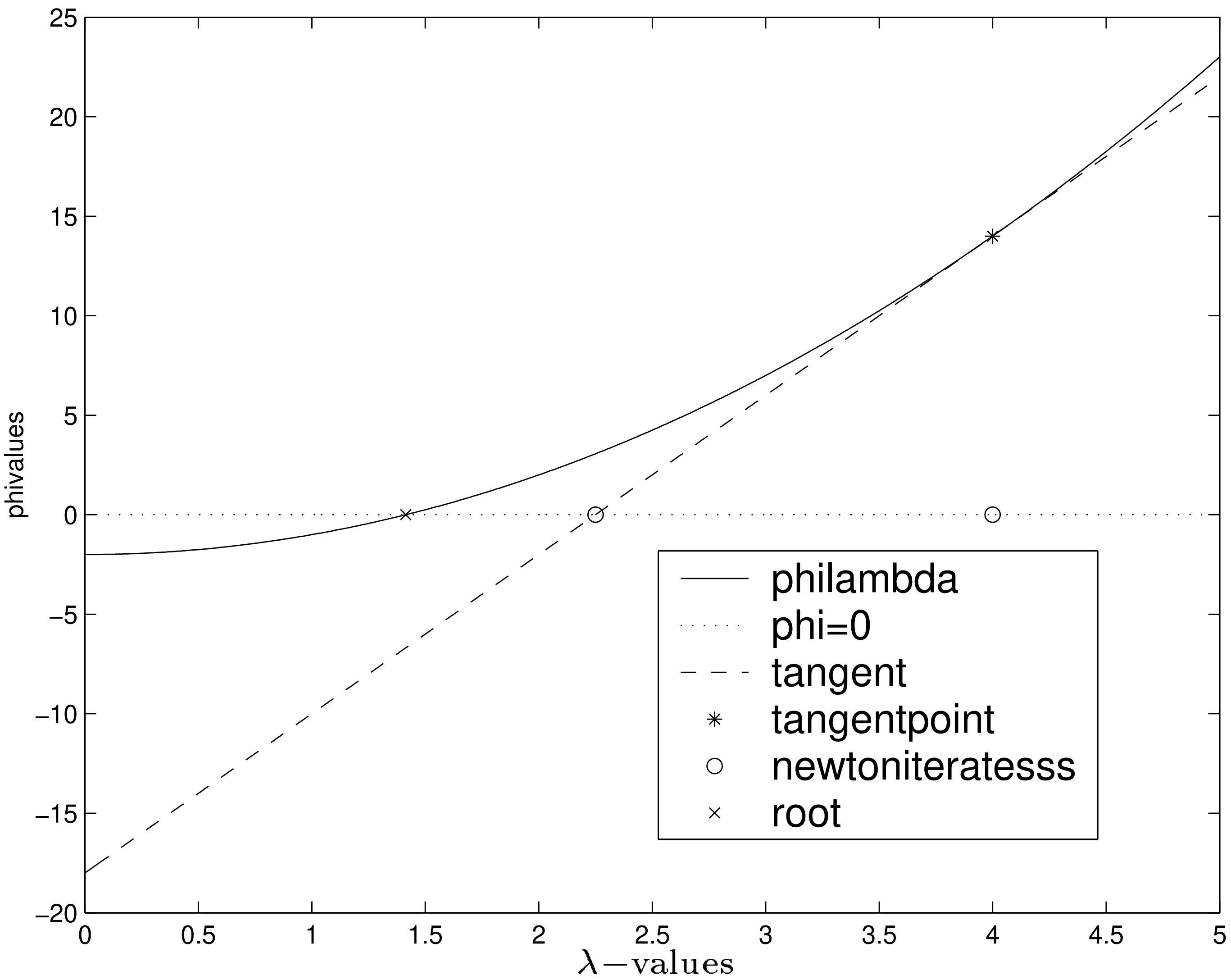


Figure 3.1: Newton's method with the secular function, $\phi(\lambda)$.

3.1 Handling the Hard Case

Lemma 3.1 (Primal step to the boundary). *Let* $0 < \sigma < 1$ *be given and suppose that*

$$A - \lambda I = R^T R, \quad (A - \lambda I)x = a, \quad \lambda \leq 0. \tag{3.2}$$

Let $z \in \mathbb{R}^n$ *satisfy*

$$\|x + z\|^2 = s^2 \tag{3.3}$$

and

$$\|Rz\|^2 \leq \sigma(\|Rx\|^2 - \lambda s^2). \tag{3.4}$$

Then

$$|q(x + z) - q(x^*)| \leq \sigma |q(x^*)|. \tag{3.5}$$

where x^* *is optimal for TRS.* ■

4 Generalized Lanczos Trust Region 1999 (GLTR) Algorithm

GLTR (or GLRT) method of Gould, Lucidi, Roma and Toint
focus on exploiting sparsity
Lanczos tridiagonalization of the matrix A
solve a *sequence* of restricted problems

$$\begin{array}{ll} \min & q(x) \\ \text{s.t.} & \|x\| \le s \\ & x \in S, \quad \text{Krylov subspace} \end{array} \tag{4.1}$$

Outline:

(i) Apply the conjugate-gradient method to $q(\cdot)$ until the boundary is reached; or until a global minimizer is found; or a direction of nonpositive curvature is found to move to the boundary.

(ii) Solve the TRS subproblem

$$(TRS_{\text{sub}}) \quad \begin{array}{ll} \min & q(x) \\ \text{s.t.} & \|x\| \leq s \\ & x \in S \equiv \mathcal{K}_k, \end{array} \tag{4.2}$$

where $\mathcal{K}_k := \text{span}\{a, Aa, A^2a, A^3a, \ldots, A^ka\}$

(iii) Increase k, the size of the Krylov subspace using the conjugate gradient method.

Efficiency based on efficiently finding approximate solutions of $\mathrm{TRS}_{\mathrm{sub}}$ (4.2), or equivalently the following tridiagonal TRS

$$\begin{aligned} \min \quad & h^T T_k h - 2\gamma_0 e_1^T h \\ \text{s.t.} \quad & \|h\| \le s, \end{aligned} \tag{4.3}$$

where

$$T_k = \begin{bmatrix} \delta_0 & \gamma_1 & & & \\ \gamma_1 & \delta_1 & . & & \\ & . & . & . & \\ & & . & \delta_{k-1} & \gamma_k \\ & & & \gamma_k & \delta_k \end{bmatrix},$$

$\delta_k = q_k^T A q_k$, $\gamma_0 = \|a\|$, $q_{-1} = 0$, $q_0 = a/\|a\|$,
$\gamma_k = \|(A - q_{k-1}^T A q_{k-1} I) q_{k-1} - \gamma_{k-1} q_{k-2}\|$, and
$q_k = ((A - q_{k-1}^T A q_{k-1} I) q_{k-1} - \gamma_{k-1} q_{k-2})/\gamma_k$, for $k \ge 1$.

5 Duality; an SDP Framework for TRS

For simplicity, use equality TRS:

$$
(\mathrm{TRS}_=) \qquad \begin{array}{rll} q^* = & \min & q(x) \\ & \text{s.t.} & \|x\| = s, \end{array} \tag{5.1}
$$

strong Lagrangian duality holds for $\mathrm{TRS}_=$, i.e.

$$
q^* = \nu^* := \max_{\lambda} \min_{x} L(x, \lambda), \tag{5.2}
$$

where the Lagrangian is $L(x, \lambda) := x^T(A - \lambda I)x - 2a^T x + \lambda s^2$.

Define

$$
h(\lambda) := \lambda s^2 - a^T (A - \lambda I)^{\dagger} a.
$$

Wolfe dual implies

Theorem 5.1. *The Lagrangian dual for $TRS_=$ is*

$$(D) \qquad q^* = \sup_{A-\lambda I \succ 0} h(\lambda). \tag{5.3}$$

In the easy case and hard case (case 1), the sup *can be replaced by a* max.

5.1 Unconstrained and Linear Duals

Homogenizing $\text{TRS}_=$

$$
\begin{array}{rll}
q^* = & \min & x^T A x - 2 y_0 a^T x \\
& \text{s.t.} & \|x\|^2 = s^2, \\
& & y_0^2 = 1.
\end{array}
\tag{5.4}
$$

Therefore

$$
\begin{aligned}
& q^* \text{ is} \\
= \quad & \max_t \min_{\|x\|^2=s^2, y_0^2=1} x^T A x - 2y_0 a^T x + t(y_0^2 - 1) \\
\geq \quad & \max_t \min_{\|x\|^2+y_0^2=s^2+1} x^T A x - 2y_0 a^T x + t(y_0^2 - 1) \\
\geq \quad & \sup_{t,\lambda} \min_{x,y_0} x^T A x - 2y_0 a^T x + t(y_0^2 - 1) + \lambda(\|x\|^2 + y_0^2 - s^2 - 1) \\
= \quad & \sup_{r,\lambda} \min_{x,y_0} x^T A x - 2y_0 a^T x + r(y_0^2 - 1) + \lambda(\|x\|^2 - s^2) \\
= \quad & \sup_{\lambda} \left(\sup_{r} \min_{x,y_0} x^T A x - 2y_0 a^T x + r(y_0^2 - 1) + \lambda(\|x\|^2 - s^2) \right)
\end{aligned}
$$

where $r = t + \lambda$.

By Strong Duality

$$
\begin{aligned}
q^* &= \sup_{\lambda} \left(\min_{x, y_0} \sup_{r} x^T A x - 2 y_0 a^T x + r(y_0^2 - 1) + \lambda(\|x\|^2 - s^2) \right) \\
&= \sup_{\lambda} \min_{x, y_0^2 = 1} x^T A x - 2 y_0 a^T x + \lambda(\|x\|^2 - s^2) \\
&= \min_{x, y_0^2 = 1} \sup_{\lambda} x^T A x - 2 y_0 a^T x + \lambda(\|x\|^2 - s^2) \\
&= \min_{\|x\|^2 = s^2, y_0^2 = 1} x^T A x - 2 y_0 a^T x \\
&= q^*.
\end{aligned}
$$

All of the above are equal, and

$$q^* = \max_t \min_{||x||^2+y_0^2=s^2+1} x^T A x - 2y_0 a^T x + t(y_0^2 - 1)$$

$$= \max_t \min_{||z||^2=s^2+1} z^T D(t) z - t = \max_t (s^2+1)\lambda_1(D(t)) - t$$

where $z = \begin{pmatrix} y_0 \\ x \end{pmatrix}$ and $D(t) = \begin{pmatrix} t & -a^T \\ -a & A \end{pmatrix}$.

Define

$$k(t) := (s^2+1)\lambda_1(D(t)) - t, \tag{5.5}$$

An unconstrained dual problem for TRS:

$$\max_t \; k(t), \quad \text{(concave, coercive)} \tag{5.6}$$

rewrite into a linear semidefinite program:

$$\max_{D(t)\succeq\lambda I} (s^2+1)\lambda - t. \tag{5.7}$$

Equivalently,

$$\begin{array}{rll} q^* = & \max & (s^2+1)\lambda - t \\ & \text{s.t.} & \lambda I - tE_{00} \preceq D(0), \end{array} \tag{5.8}$$

and its dual

$$\begin{array}{rll} q^* = & \min & \operatorname{trace} D(0)Y \\ & \text{s.t.} & \operatorname{trace} Y = s^2+1 \\ & & -Y_{00} = -1 \\ & & Y \succeq 0. \end{array} \tag{5.9}$$

(Slater's holds for both - stable problems!)

6 The Rendl-Wolkowicz 1997 (RW) Algorithm, Revisited

6.1 Three Useful Functions

6.1.1 $k(t) = (s^2+1)\lambda_1(D(t)) - t \longrightarrow \max$

$$\lim_{t\to\infty} \lambda_1(D(t)) = \lambda_1(A) \text{ and } \lim_{t\to-\infty} (\lambda_1(D(t)) - t) = 0,$$

asymptotic behavior of $k(t)$ is

$$\begin{aligned} k(t) &\sim (s^2+1)\lambda_1(A) - t, && \text{as } t \to \infty && \text{(linear with slope } -1) \\ k(t) &\sim s^2 t, && \text{as } t \to -\infty && \text{(linear with slope } s^2) \end{aligned}$$

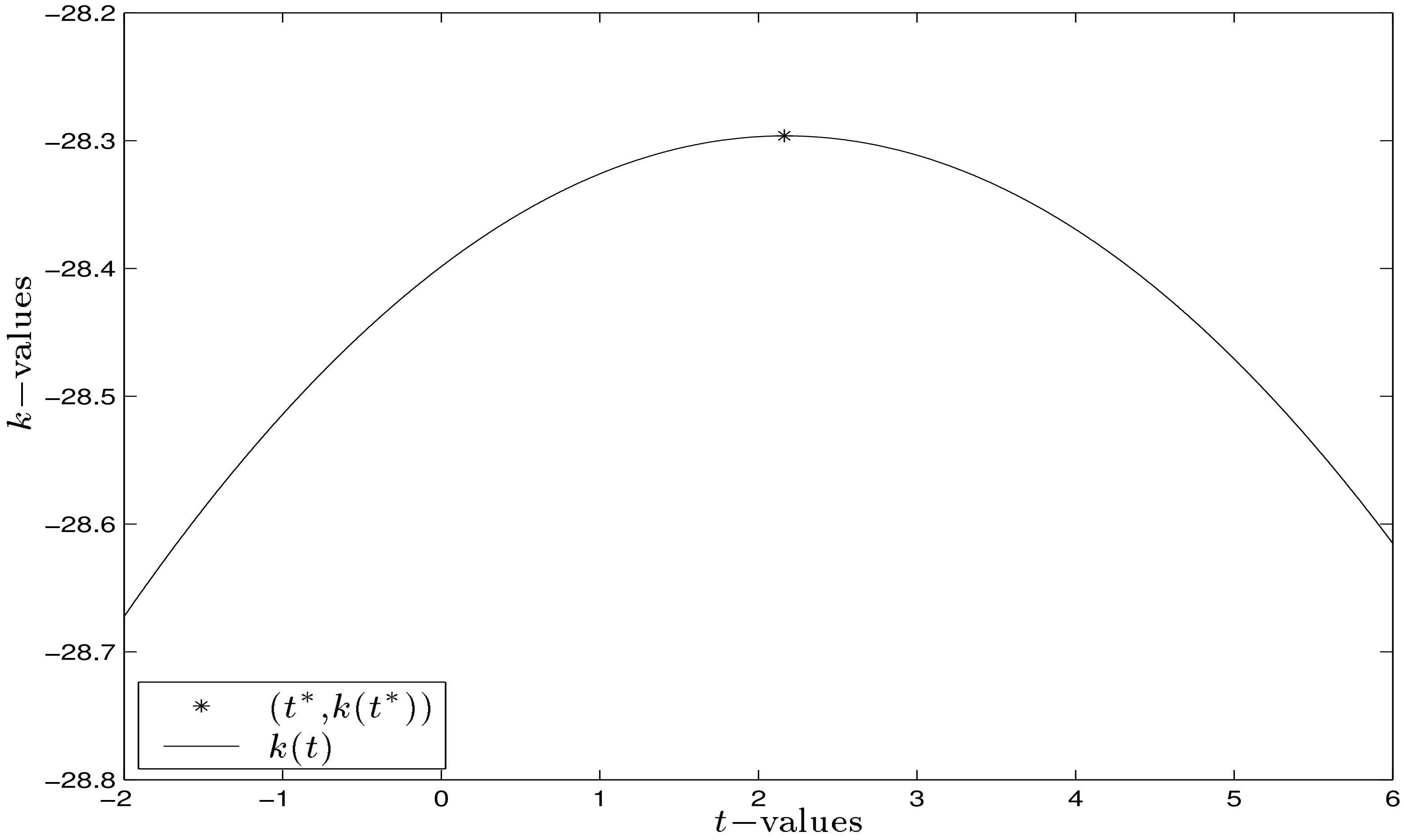


Figure 6.1: $k(t)$ in the easy case

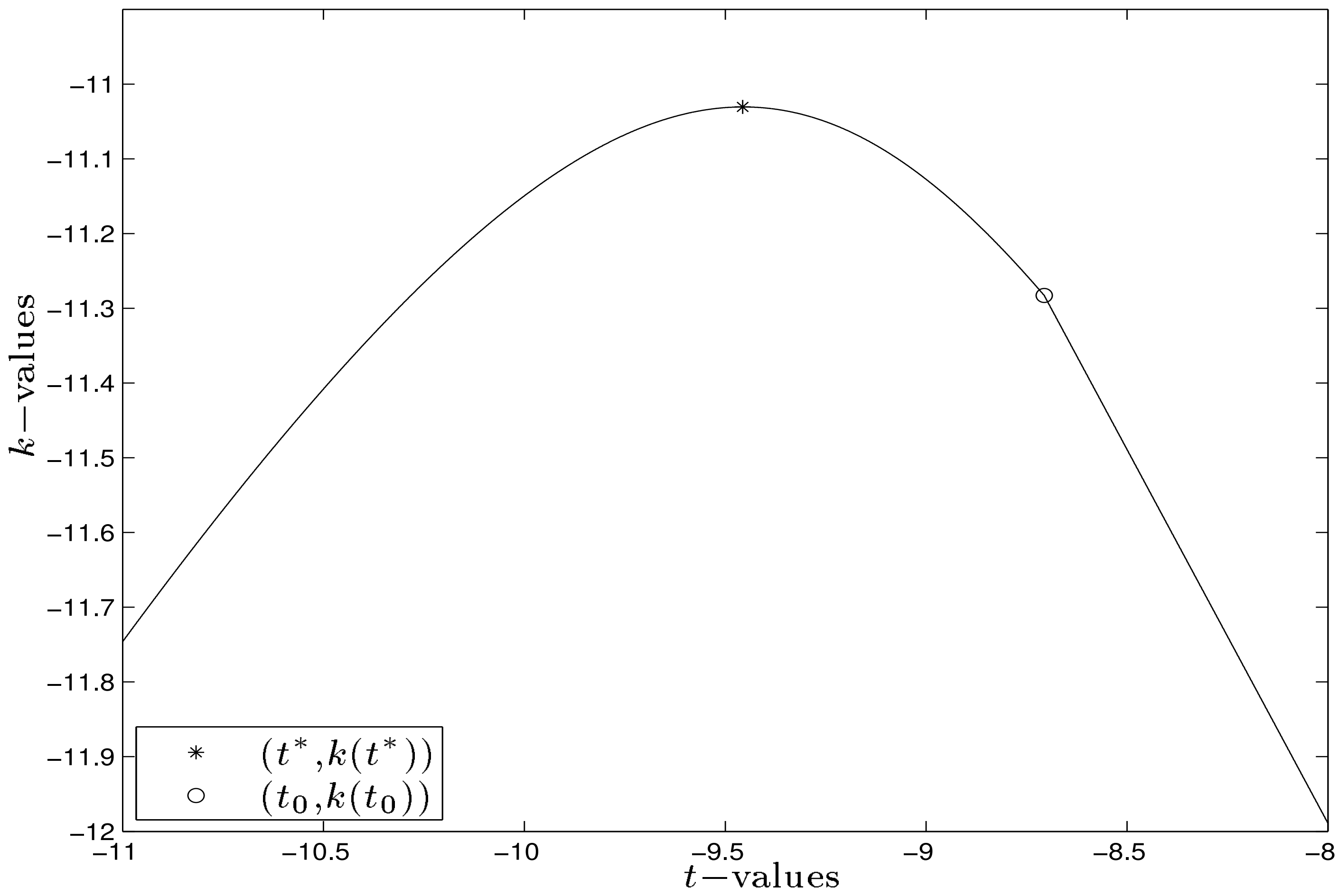


Figure 6.2: $k(t)$ in the hard case (case 1)

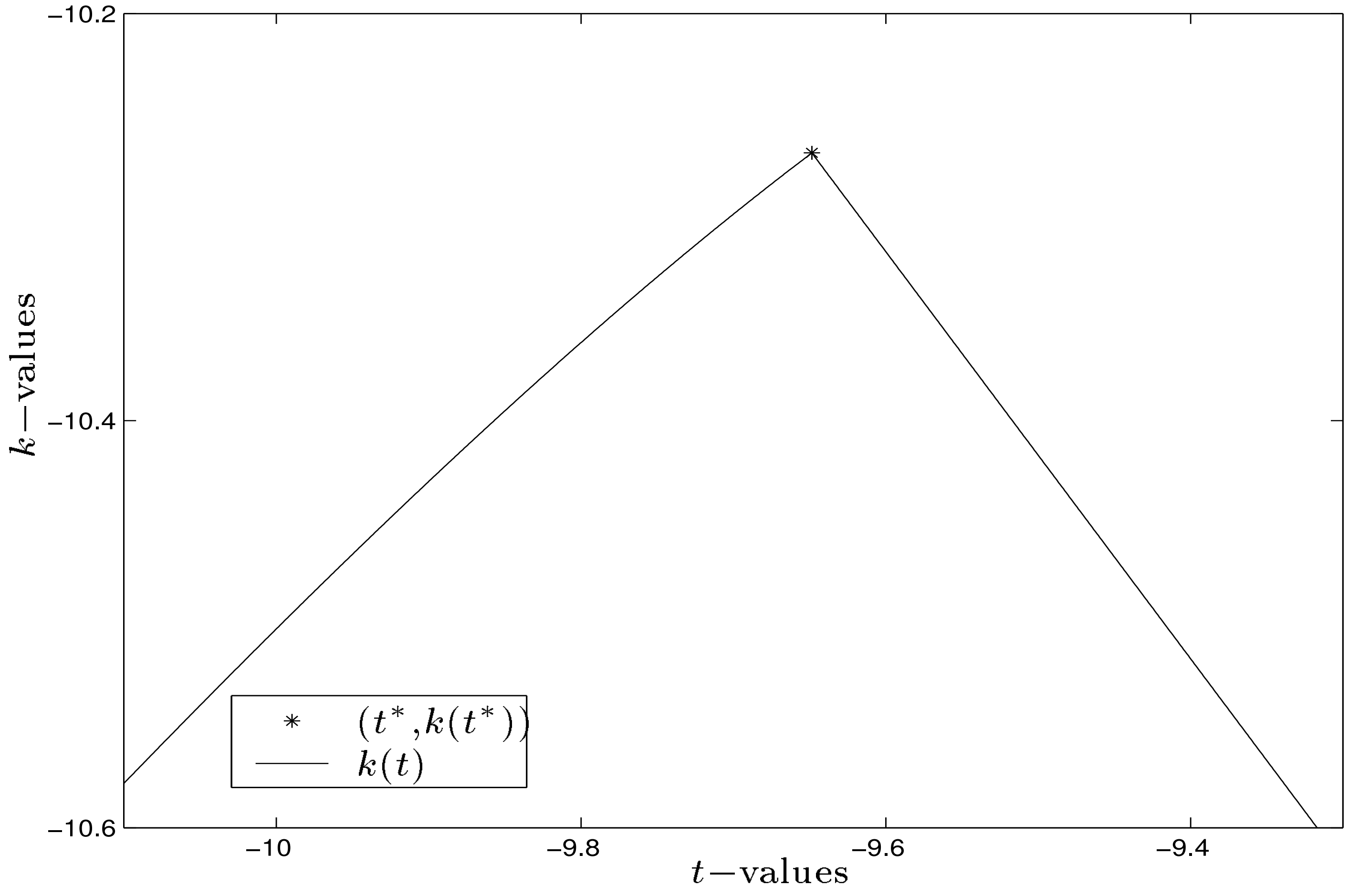


Figure 6.3: $k(t)$ in the hard case (case 2)

6.1.2 $k'(t) = (s^2+1)y_0(t)^2 - 1$

$y(t)$ normalized eigenvector for $\lambda_1(D(t))$

$y(t) = \begin{pmatrix} y_0(t) \\ x(t) \end{pmatrix}$

$$\frac{1}{y_0(t)}\|x(t)\| = s \text{ if and only if } k'(t) = 0$$

Theorem 6.1. *Let $y(t)$ be a normalized eigenvector for $\lambda_1(D(t))$ and let $y_0(t)$ be its first component. Then:*

1. **In the easy case**: *for $t \in \mathbb{R}$, $y_0(t) \neq 0$;*
2. **In the hard case**:
   - (a) **for** $t < t_0$: $y_0(t) \neq 0$;
   - (b) **for** $t > t_0$: $y_0(t) = 0$;
   - (c) **for** $t = t_0$: *there exists a basis of eigenvectors for the eigenspace of $\lambda_1(D(t_0))$, such that one eigenvector of this basis, ω, has a non-zero first component ($\omega_0 \neq 0$) and the other eigenvectors in the basis have a zero first component ($y_0(t) = 0$).*

In the easy case,

$y_0(t)$ is strictly monotonically decreasing,
$y_0(t) \to 1$ as $t \to -\infty$
and $y_0(t) \to 0$ as $t \to \infty$.

In particular, $y_0(t) \neq 0$ for all $t \in \mathbb{R}$.

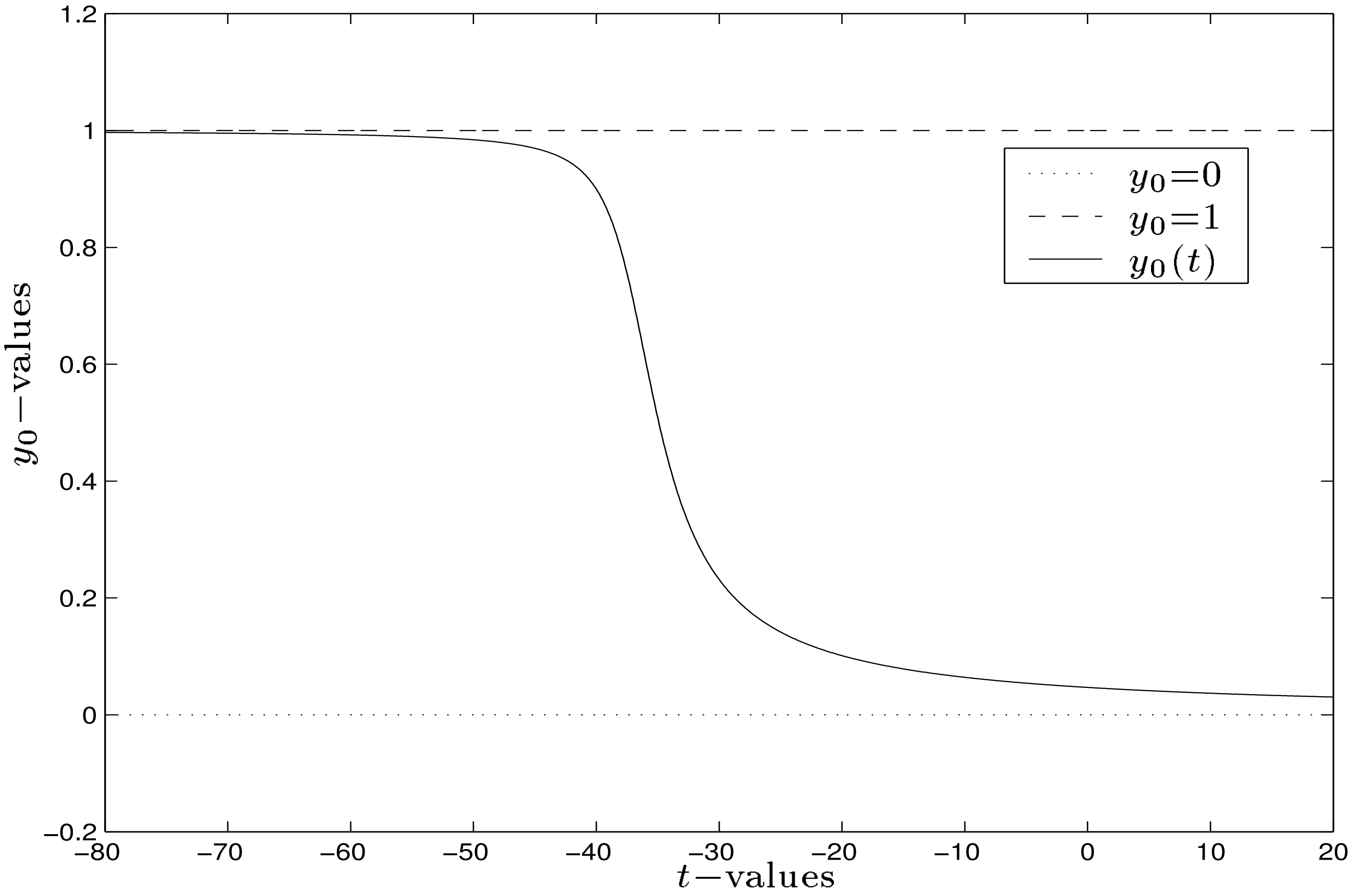


Figure 6.4: $y_0(t)$ in the easy case

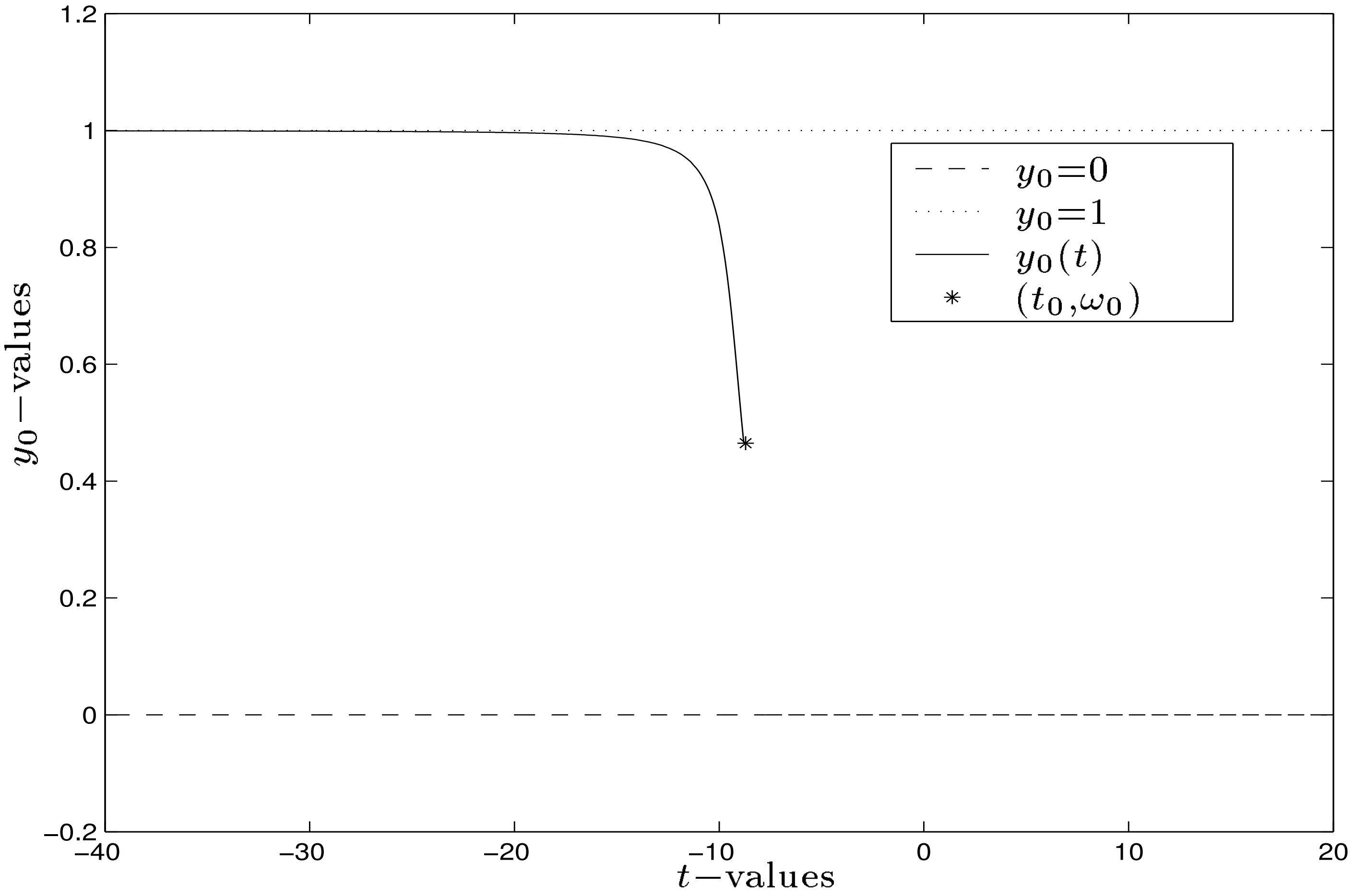


Figure 6.5: $y_0(t)$ in the hard case

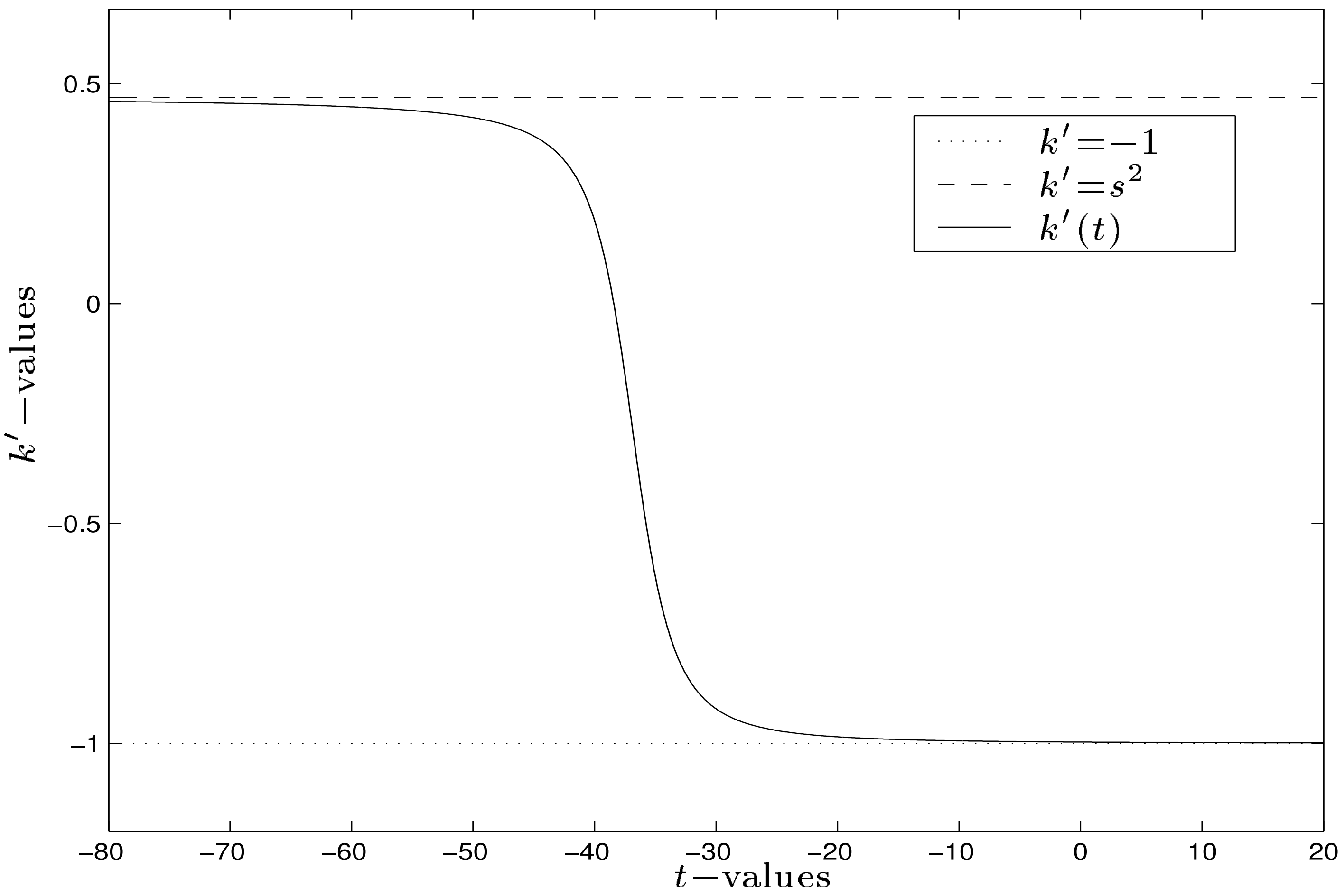


Figure 6.6: $k'(t)$ in the easy case

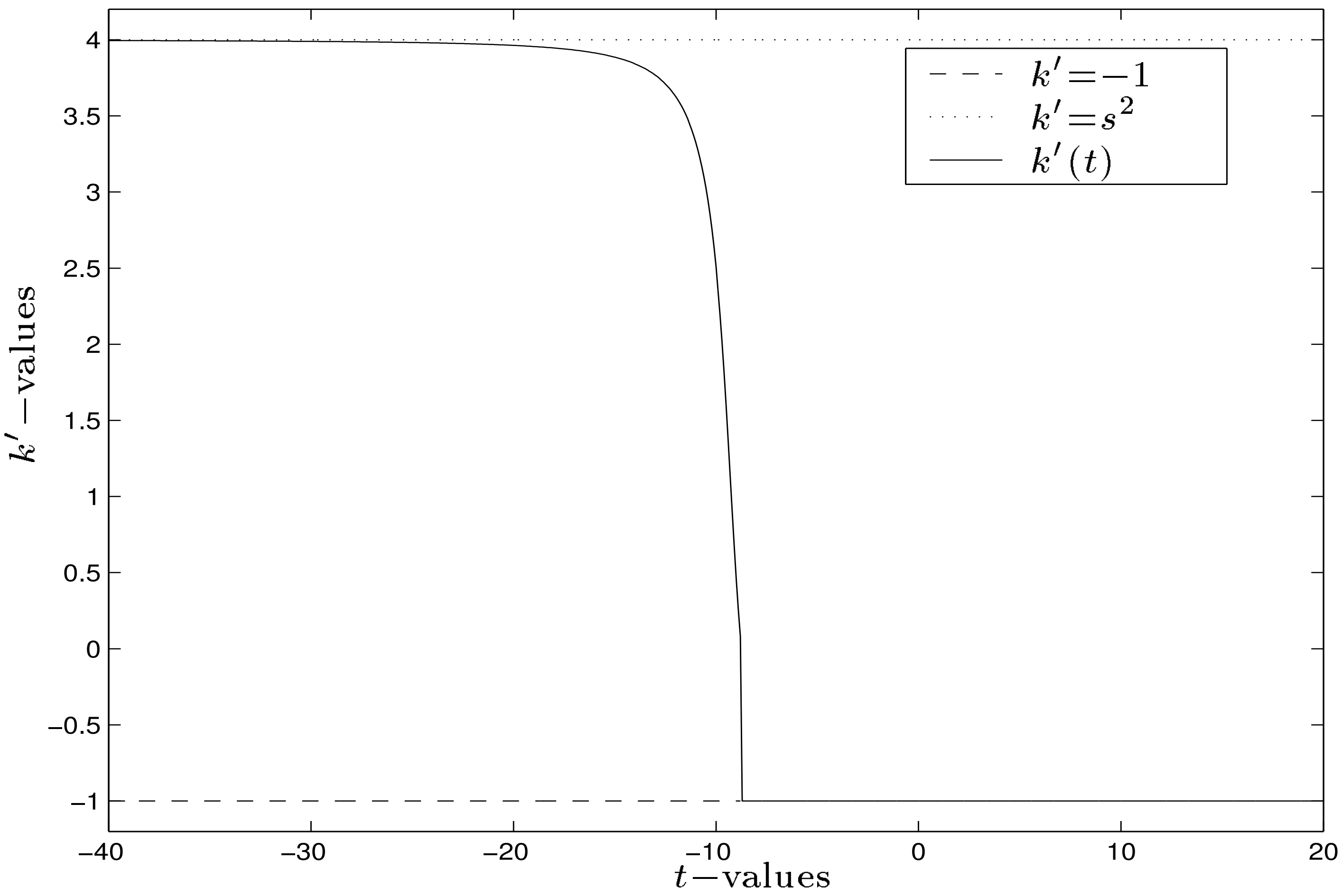


Figure 6.7: $k'(t)$ in the hard case (*case 1*)

6.1.3 $\psi(t) = \sqrt{s^2+1} - \frac{1}{y_0(t)}$

Solving $\psi(t) = 0$ or $k'(t) = 0$ is equivalent.

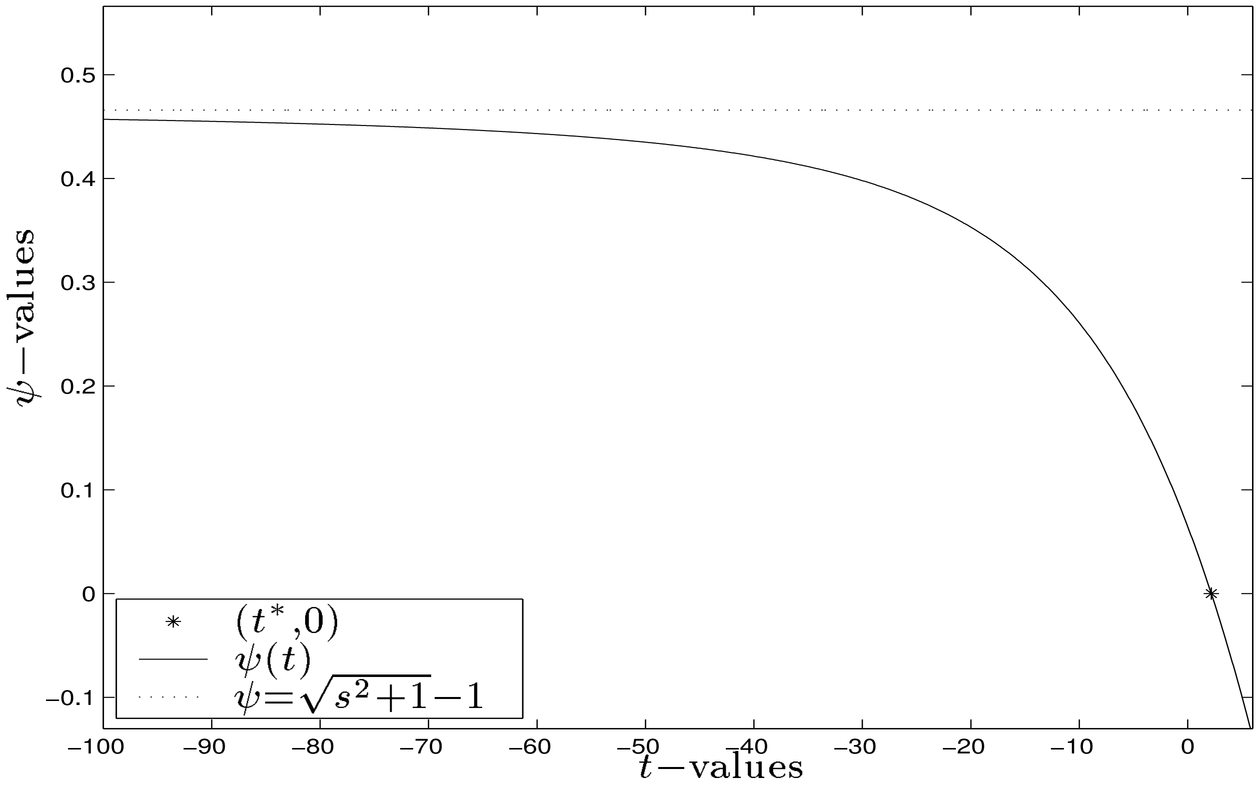


Figure 6.8: $\psi(t)$ in the easy case

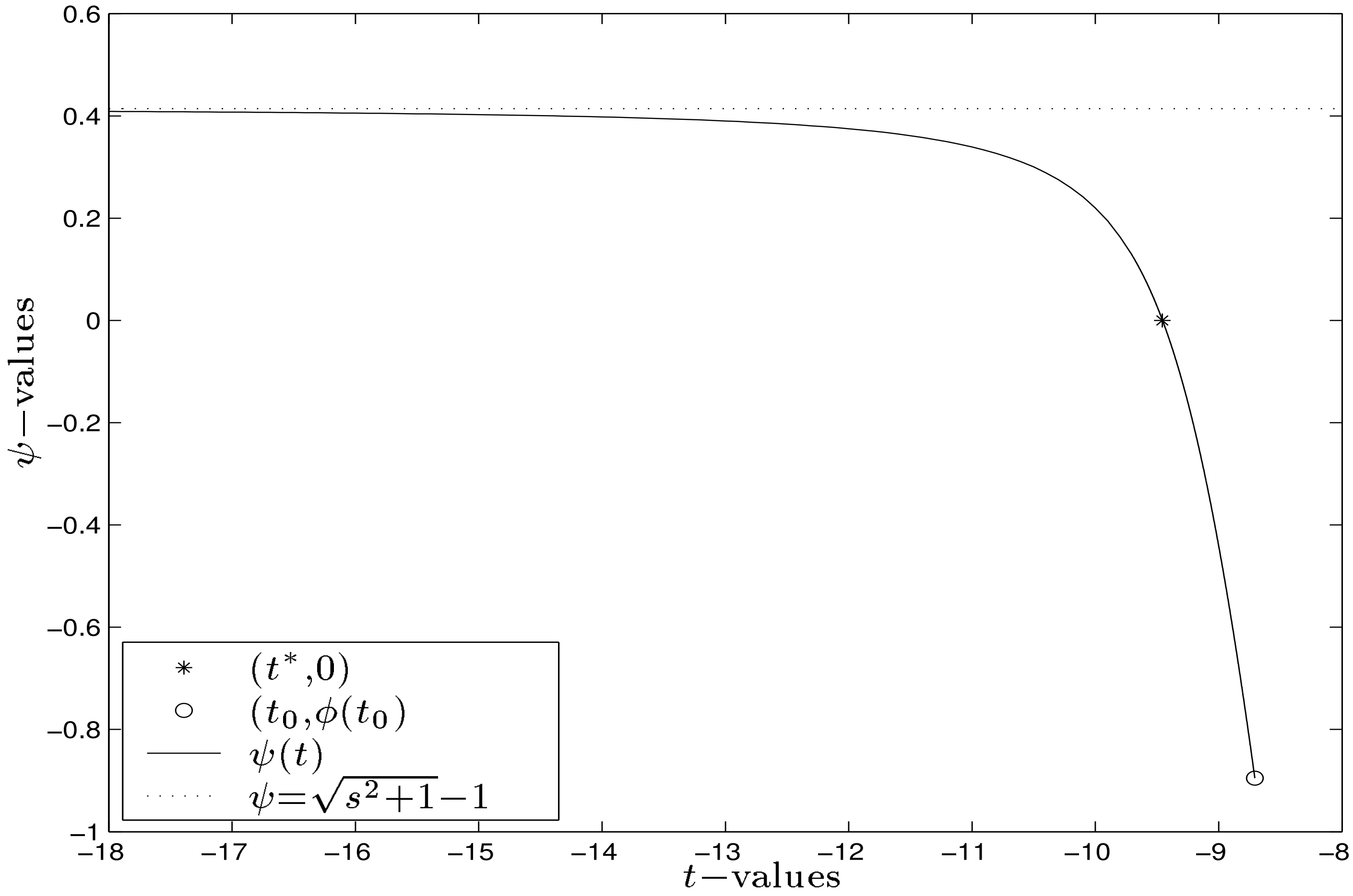


Figure 6.9: $\psi(t)$ in the hard case (*case 1*)

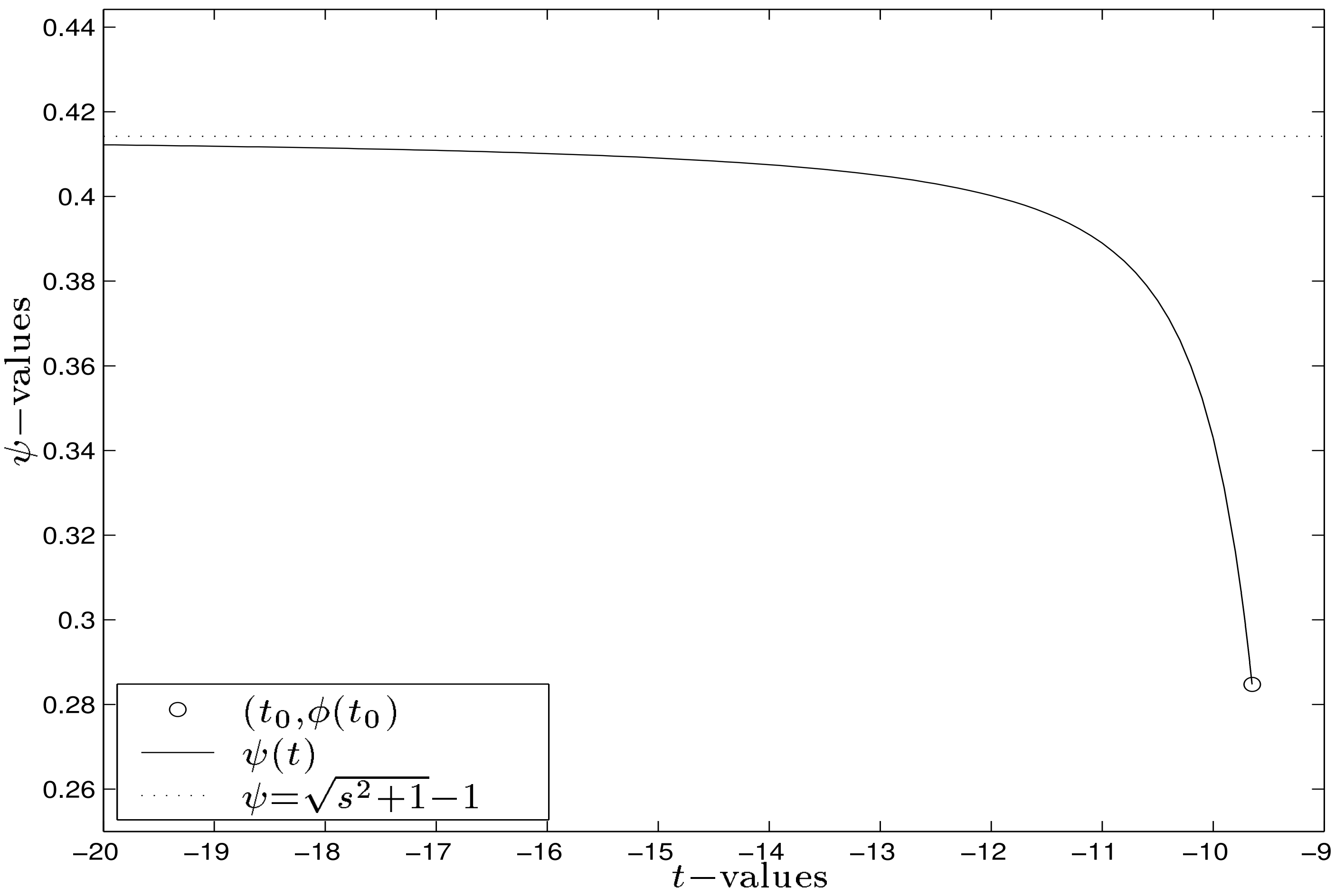


Figure 6.10: $\psi(t)$ in the hard case (*case 2*)

assume that $y_0 > 0$

$$\textbf{Easy Case Indicators:} \quad \left\{ \begin{array}{rcl} \phi(\lambda) & > & 0 \\ k'(t) & < & 0 \quad ((s^2+1)y_0^2 - 1 < 0) \\ y_0 & < & \frac{1}{\sqrt{s^2+1}} \\ t & > & t^* \\ \lambda & > & \lambda^* \\ \|x(\lambda)\| & > & s \\ h'(\lambda) & < & 0. \end{array} \right. \tag{6.1}$$

7 Flowchart (RW Revisited)

INITIALIZATION:

1. Compute $\lambda_1(A)$ and corresponding eigenvector v_1.

   If $a^T v_1$ is small (near hard case), then deflate, i.e. add the vector $y_1 = \begin{pmatrix} 0 \\ v_1 \end{pmatrix}$ to the set $\mathcal{Y}$.

2. Obtain bounds on q^*, λ^* and t^*.

   If $\lambda_1 > 0$, test if the optimum is an unconstrained minimizer. EXIT here if so.

3. Initialize parameters and the stopping criteria based on the optimality conditions, duality gap, and intervals of uncertainty.

ITERATION LOOP: (until convergence to the desired tolerance or until we find the solution is the unconstrained minimizer.)

1. **FIND a NEW VALUE of t.**
   (a) Set t using Newton's method on $k(t) - M_t$ if possible; otherwise set it to the the midpoint (default) of the interval of uncertainty for t.
   (b) If points from the hard and easy side are available:
      i. Do TRIANGLE INTERPOLATION (Update upper bound on q_* and set t, if possible.)
      ii. Do VERTICAL CUT (Update lower or upper bound for interval of uncertainty for t.)
   (c) Do INVERSE INTERPOLATION (Set t, if possible)

UPDATE

2. (a) With new t, compute (with restarts) $\lambda = \lambda_1(D(t))$, corresponding eigenvector y, $y_0 \geq 0$. (y orthogonal to $\mathcal{Y}$)

   (b) $\lambda > 0$ and $y_0 > 1/(s^2 + 1)$ implies solution is unconstrained minimizer. Use CG, EXIT.

   (c) Update bounds on interval of uncertainty of q^*.

   (d) i. If y_0 is small, then deflate, i.e. add y to $\mathcal{Y}$.

      ii. elseif t is on easy side, update parameters. Take primal step to boundary if hard side point exists.

      iii. elseif t is on hard side, update parameters. Take primal step to boundary from this hard side point.

   (e) Save new bounds and update stopping criteria.

END ITERATION LOOP

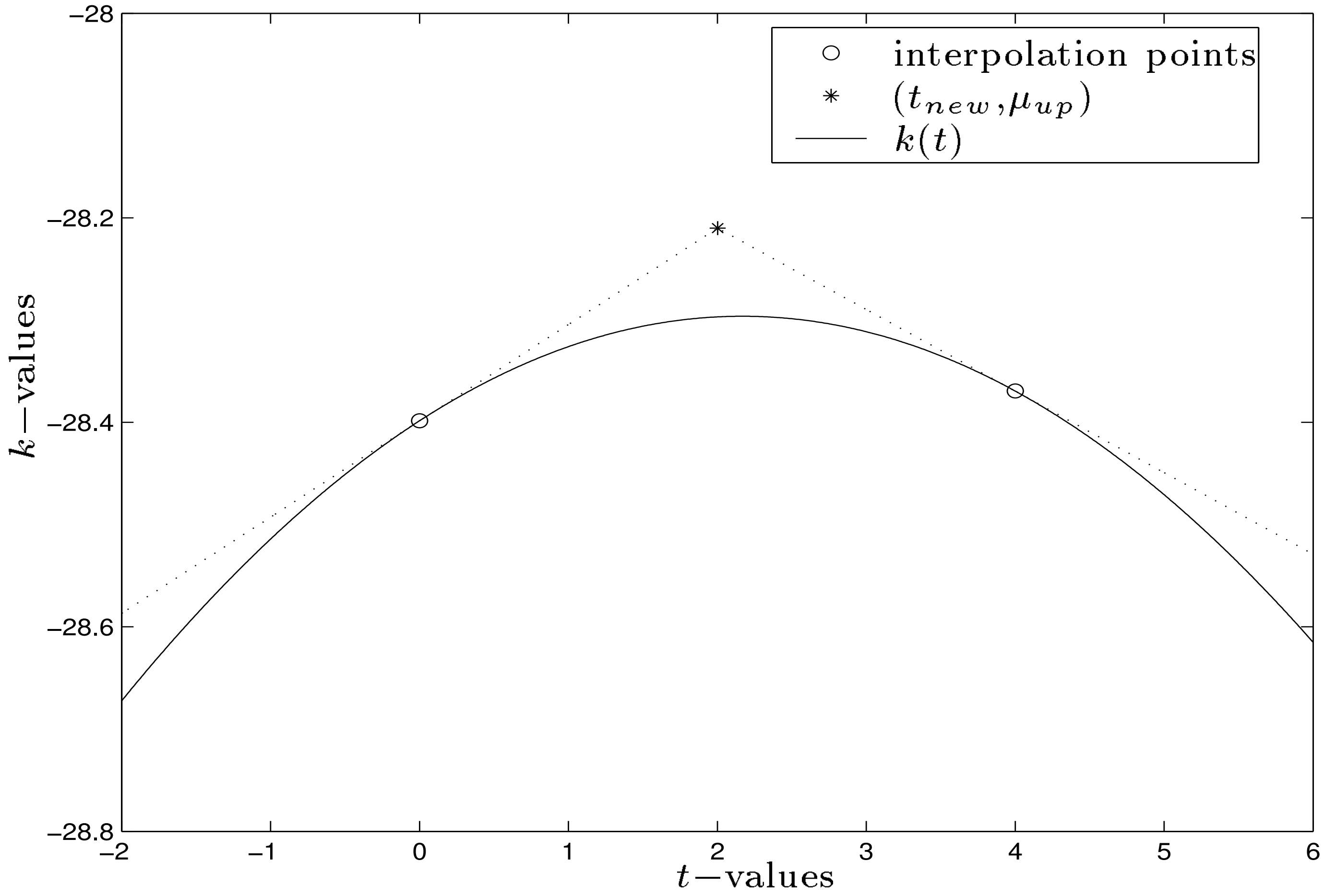
7.1 Techniques used in the algorithm

7.1.1 Newton's Method on $k(t) - M_t = 0$

$$t_+ = t_c - \frac{k(t_c) - M_t}{k'(t_c)} = \frac{(s^2+1)(t_c y_0^2(t_c) - \lambda(D(t_c)) - M_t}{(s^2+1)y_0(t_c)^2 - 1}.$$

7.1.2 Triangle Interpolation using $k(t)$

Reduce interval of uncertainty of t; improve upper bound on q^*.



7.1.3 Vertical Cut

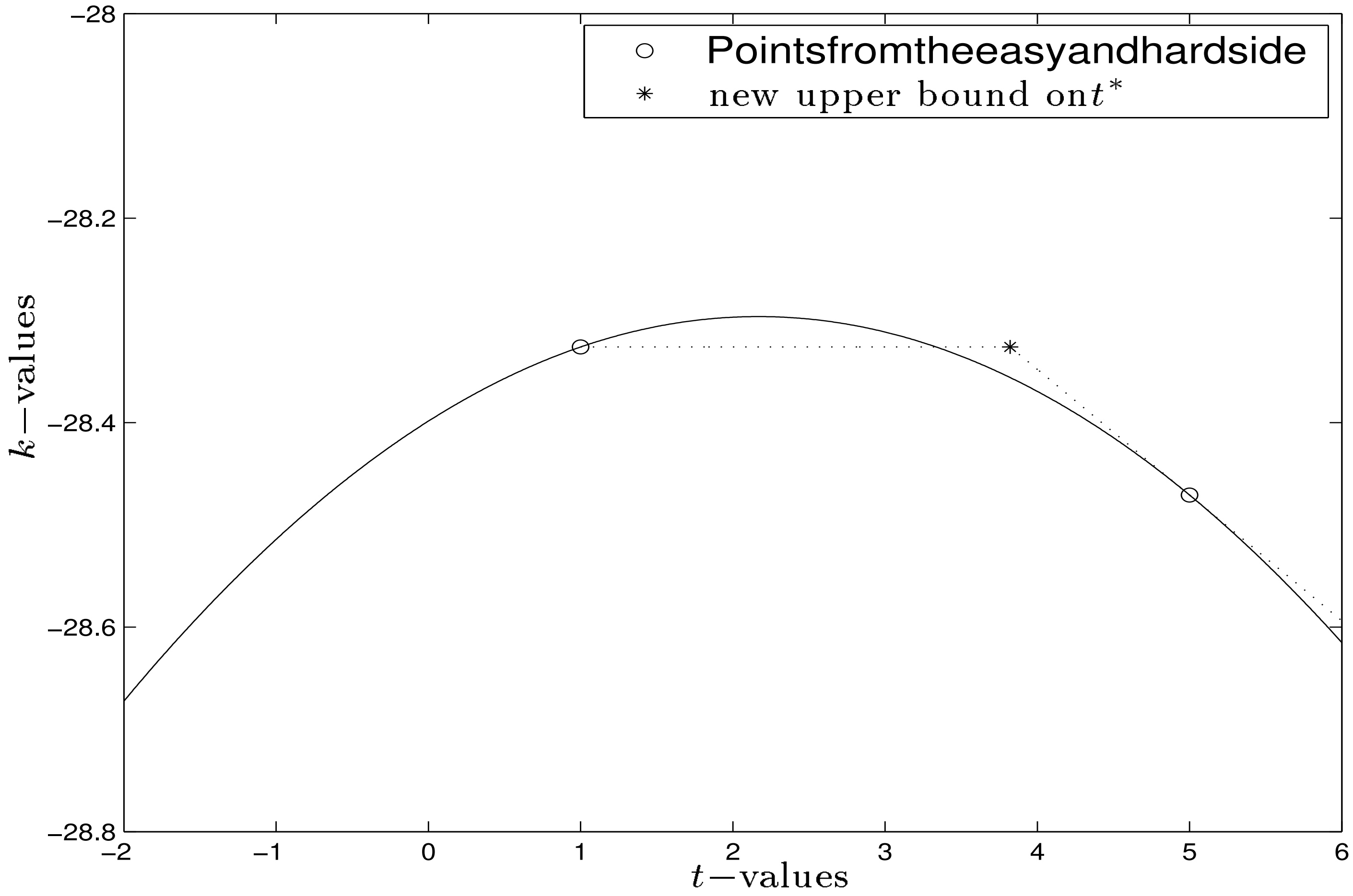


Figure 7.1: Vertical cut

7.1.4 Inverse Interpolation

$$\begin{bmatrix} \psi_1^2 & \psi_1 & 1 \\ \psi_2^2 & \psi_2 & 1 \\ \psi_3^2 & \psi_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ t_{\text{new}} \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

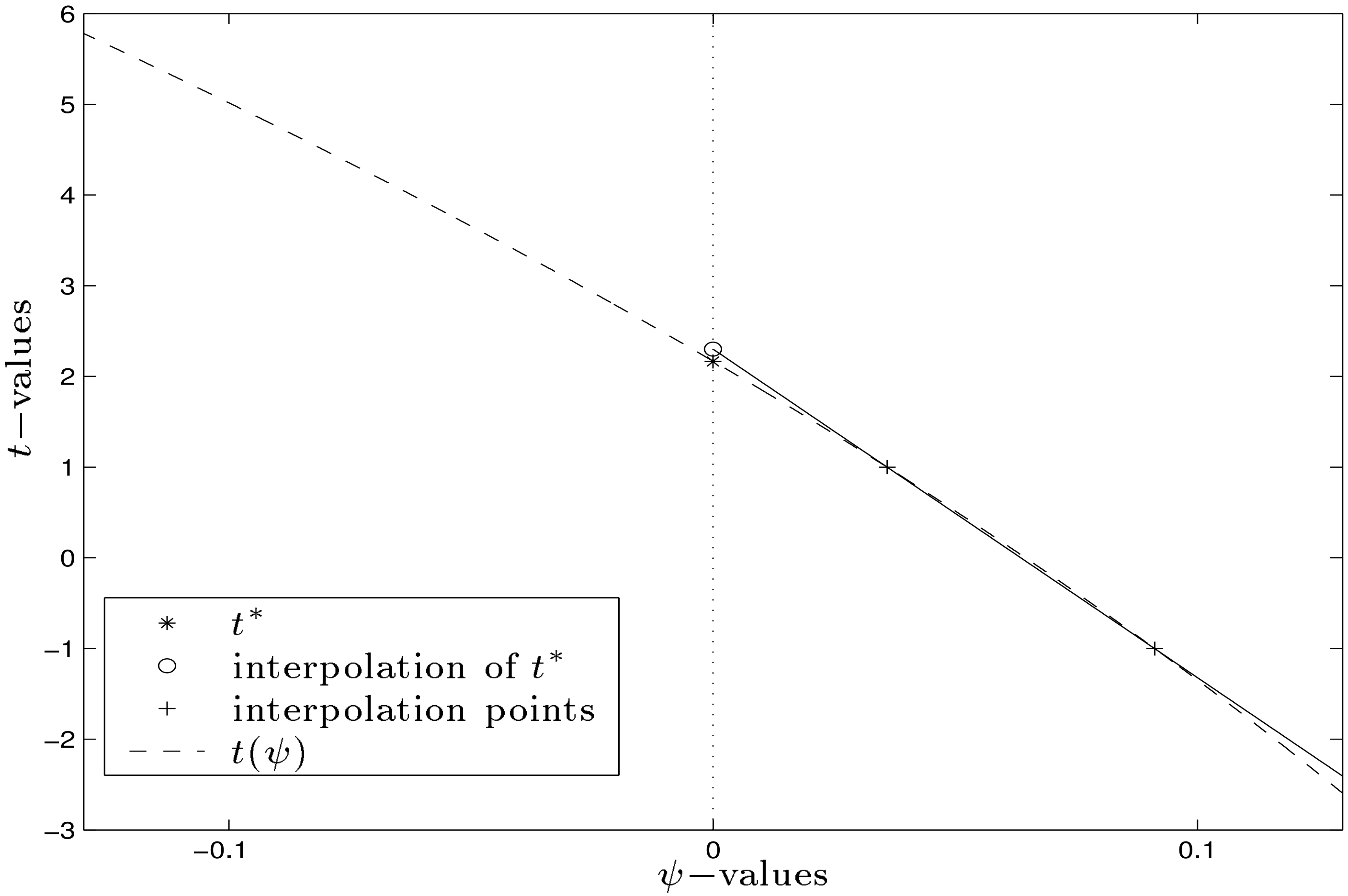


Figure 7.2: Inverse interpolation; easy case & hard case (*case 1*)

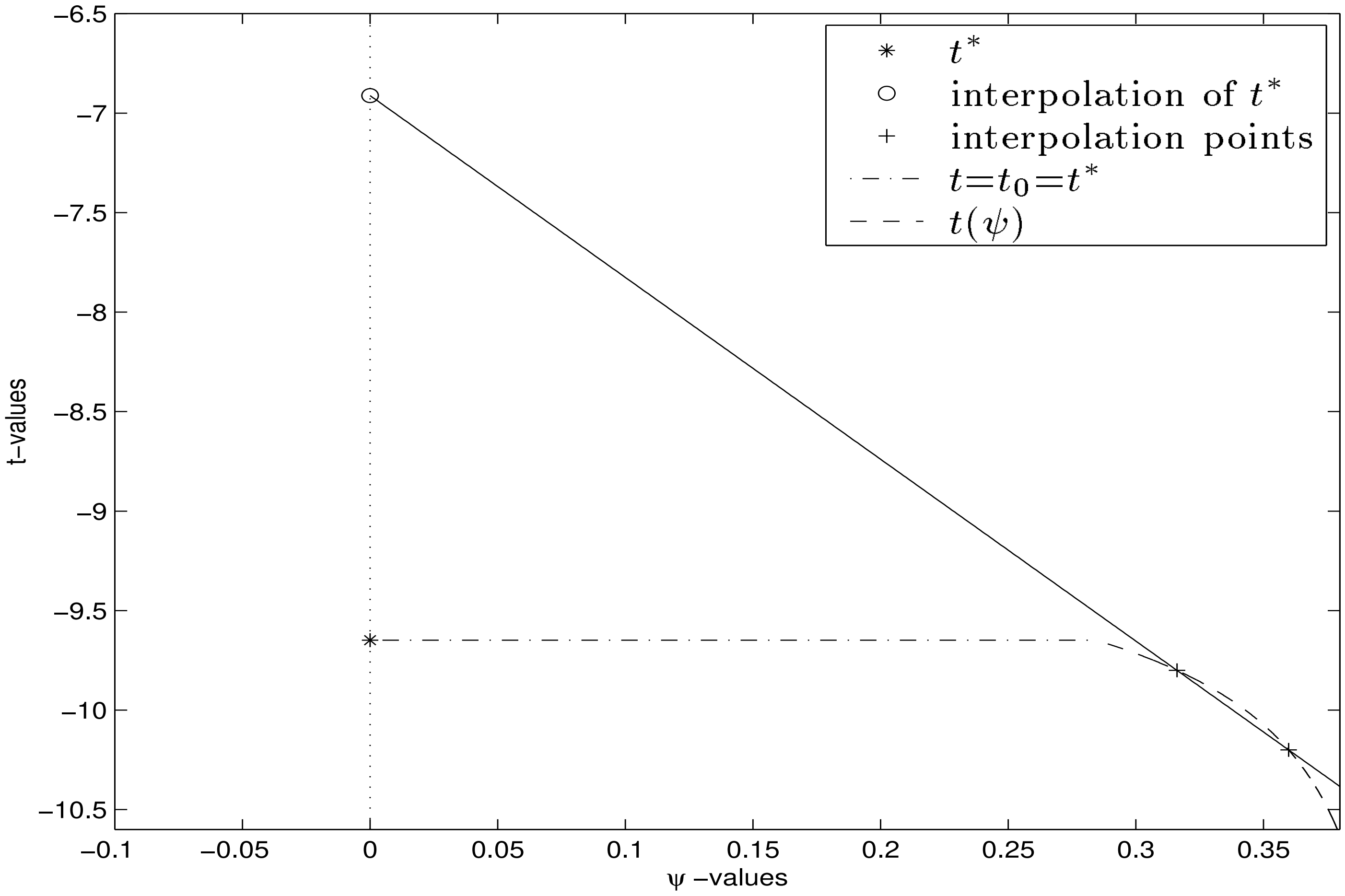


Figure 7.3: Inverse interpolation in the hard case (*case 2*)

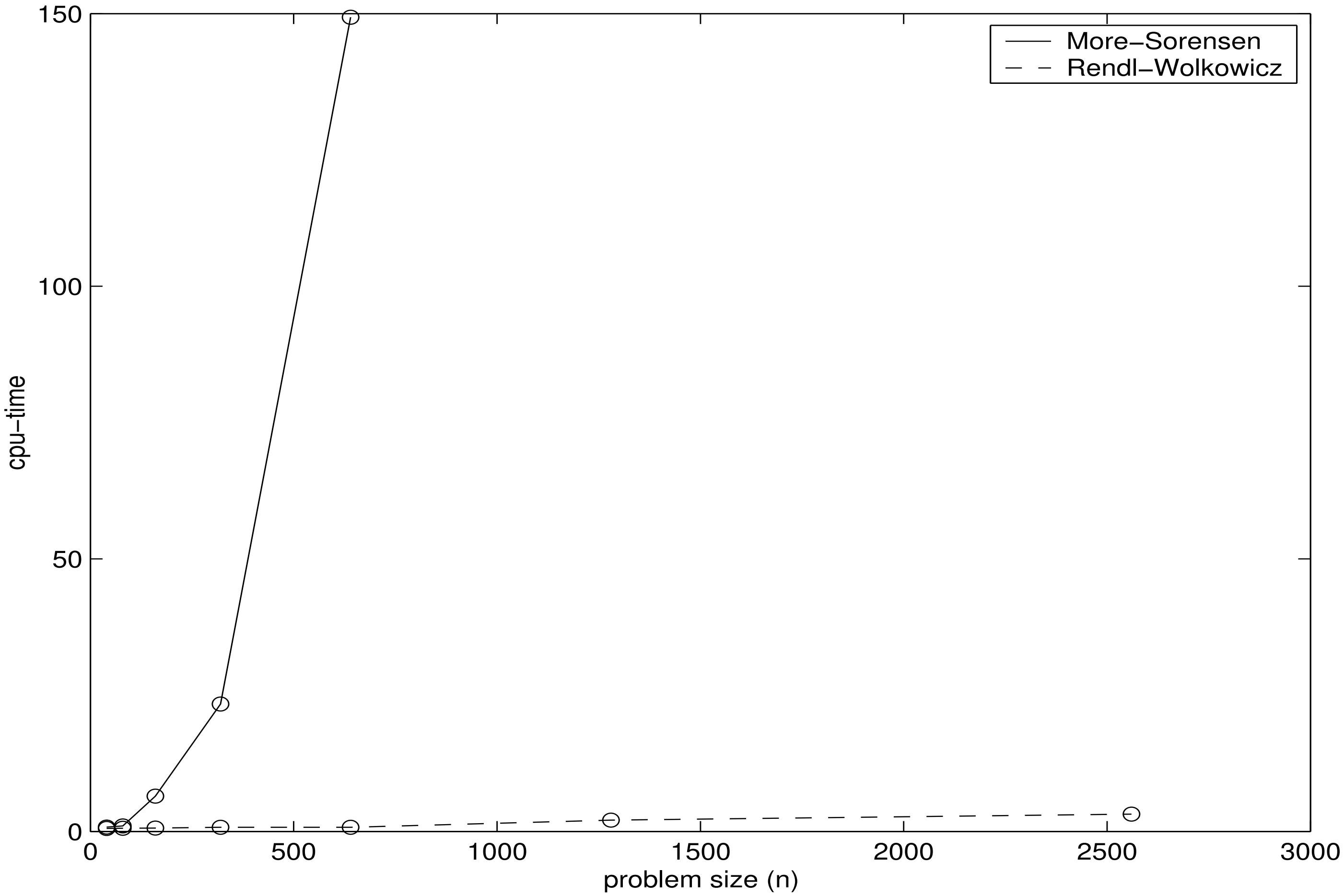


Figure 7.4: cputime in the hard case (*case 2*)

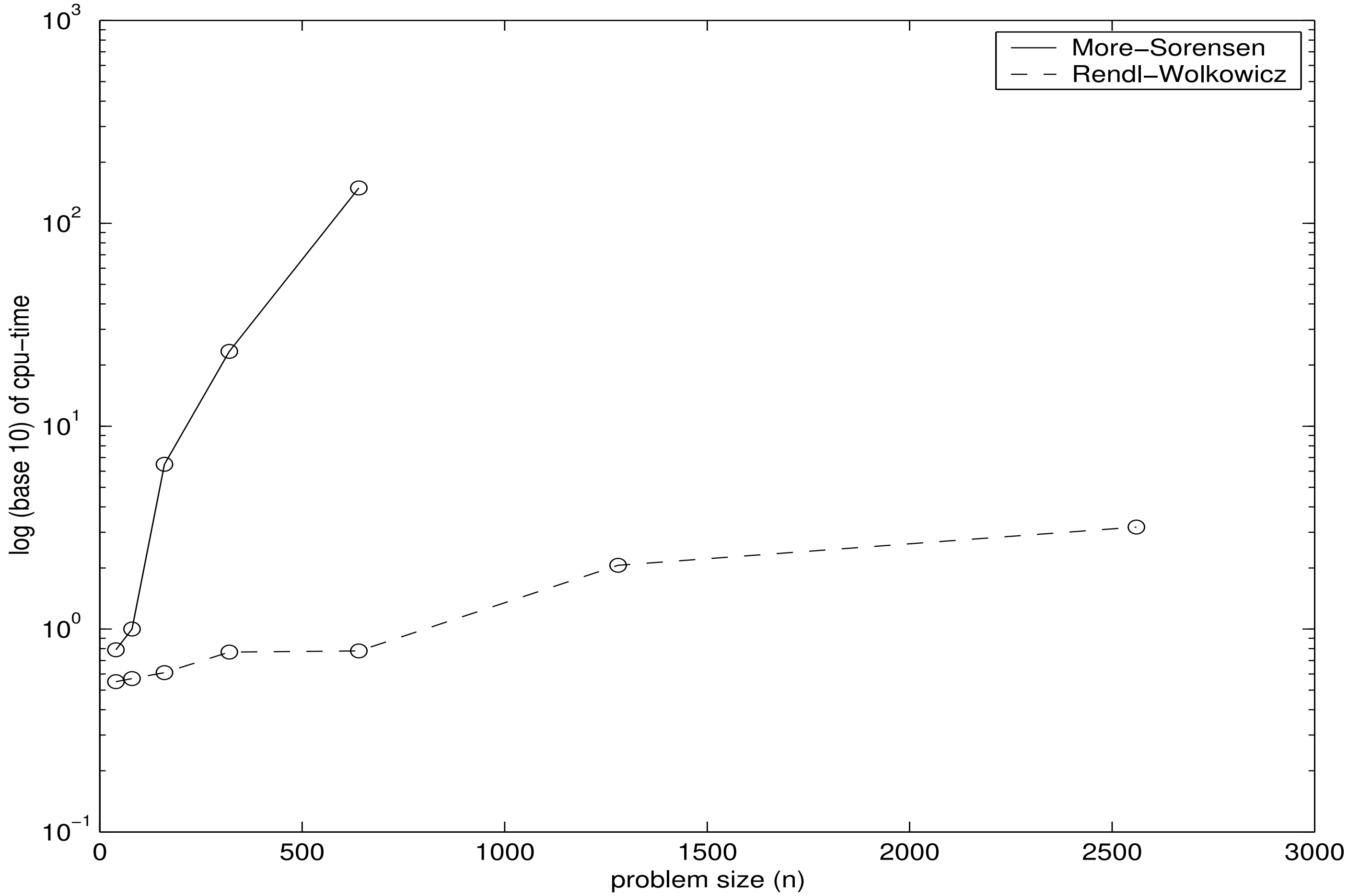


Figure 7.5: log of cputime in the hard case (*case 2*)