SHORT COMMUNICATION

PIECEWISE LINEAR PATHS TO MINIMIZE
CONVEX FUNCTIONS MAY NOT BE MONOTONIC

Michael J. TODD*
Cornell University, Ithaca, New York, U.S.A.

Received 28 August 1978
Revised manuscript received 13 February 1979

The exact homotopy path when seeking the minimum of a convex function is monotonic in
the homotopy parameter. This monotonicity is not inherited by the piecewise linear ap-
proximations to such paths produced by fixed-point algorithms.

Key words: Fixed-point Algorithms, Unconstrained Minimization, Monotonic Paths.

1. Introduction

Fixed-point algorithms provide a powerful computational tool in mathematical
economics, game theory and optimization—see for example [1, 3, 9]. While the
asymptotic behavior of such algorithms applied to smooth problems is known
[6, 7], their initial behavior even on very special problems is still not well
understood. For computing the zero (or fixed point) of an affine function, Saigal
[5] has described some properties of the path as the grid size of the triangulation
approaches zero. For larger grid sizes, the behavior depends critically on the
particular triangulation; one example of the complexity of the real piecewise
linear path was given in [8] to illustrate the importance of a good Jacobian
approximation.

Given a $C^2$ function $l: \mathbb{R}^n \to \mathbb{R}^n$ and a one-to-one affine function $r: \mathbb{R}^n \to \mathbb{R}^n$ we
may consider the zeroes of $F(x, t) = tr(x) + (1 - t)l(x)$; these zeroes usually form
a 1-dimensional differentiable manifold $(x(\theta), t(\theta))$ with $x(0)$ the zero of $r$,
t(0) = 1. If $t(\theta) = 0$, $x(\theta)$ is a zero of the function $l$. (There may be other
connected components of zeroes—we ignore these.) The fixed-point algorithm of
Eaves and Saigal [2] approximates this path by considering the zeroes of a
piecewise linear function $\tilde{F}(x, t)$ generated by a triangulation of $\mathbb{R}^n \times (0, 1]$ with

*This research was partially supported by National Science Foundation Grant ENG76-08749.
continuous refinement of grid size. For a vertex \((v, t)\) we set \(\hat{F}(v, t) = l(v)\) if \(t < 1\), \(= r(v)\) if \(t = 1\) and then extend \(\hat{F}\) linearly on each simplex.

Recent investigation of the degree-theoretic properties of these paths has shown that \(t(\theta)\) is monotonic in \(\theta\) for \(0 \leq t \leq 1\) if the Jacobian of \(l\) never has negative eigenvalues and \(r(x) = x - c\) for some \(c\). See Garcia and Gould [4]; for related results for piecewise linear paths, see [1, 3]. If \(l\) is the gradient of a convex function \(f\), the Jacobian of \(l\) is the Hessian of \(f\) which has no negative eigenvalues. The question arises as to whether the monotonicity of \(t\) is inherited by the piecewise linear path generated by the Eaves–Saigal homotopy algorithm. In this note we show that it is not.

To describe the example we presume the reader to be familiar with the homotopy algorithm [2] and the triangulation \(J_3\) [9]. For \(n = 2\) the algorithm will generate a sequence of triangles in \(\mathbb{R}^2 \times (0, 1]\), each containing a zero of \(\hat{F}\) and each consecutive pair lying in a tetrahedron of \(J_3\). The regression in the example is caused solely by the piecewise linear approximation to \(\nabla f\); neither the interaction with \(r\) nor the specific triangulation is responsible.

The example

Consider the convex function \(f'\) on \(\mathbb{R}^2\) defined by

\[
f'(x) = \max\{-4x_1 - 3x_2, -x_2 - 1, 2x_1 + 3x_2 - 6\}.
\]

Let \(f\) be a convex function obtained by smoothing \(f'\) so that \(f\) is \(C^\infty\) and

\[
l(x) = \nabla f(x) = \begin{cases} 
(0, -1) & \text{if } x = (1, 0), (0, 1) \text{ or } (\frac{1}{2}, \frac{1}{2}), \\
(2, 3) & \text{if } x = (1, 1), \\
(-4, -3) & \text{if } x = (0, 0).
\end{cases}
\]

We use the artificial function \(r(x) = x - (\frac{4}{3}, \frac{2}{3})\) and the triangulation \(J_3\) with initial grid size 2.

Consider the vertices of \(J_3\) that are listed with their labels in Table 1.

<table>
<thead>
<tr>
<th>(i)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(v^i)</td>
<td>(2, 2, 1)</td>
<td>(2, 0, 1)</td>
<td>(0, 0, 1)</td>
<td>(1, 1, \frac{1}{2})</td>
<td>(1, 0, \frac{1}{2})</td>
<td>(0, 0, \frac{1}{2})</td>
<td>(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})</td>
<td>(0, 1, \frac{1}{2})</td>
</tr>
<tr>
<td>(\hat{F}(v^i))</td>
<td>(\frac{1}{3}, \frac{1}{3})</td>
<td>(\frac{1}{3}, \frac{1}{3})</td>
<td>(\frac{1}{3}, \frac{1}{3})</td>
<td>(2, 3)</td>
<td>(0, -1)</td>
<td>(-4, -3)</td>
<td>(0, -1)</td>
<td>(0, -1)</td>
</tr>
</tbody>
</table>

Then the triangles generated by the algorithm are

\[
\tau_1 = \langle v^1, v^2, v^3\rangle, \quad \tau_2 = \langle v^2, v^3, v^4\rangle, \quad \tau_3 = \langle v^3, v^4, v^5\rangle, \\
\tau_4 = \langle v^4, v^5, v^6\rangle, \quad \tau_5 = \langle v^4, v^6, v^7\rangle, \quad \tau_6 = \langle v^4, v^6, v^8\rangle, \\
\tau_7 = \langle v^3, v^4, v^5\rangle.
\]