PIECEWISE-LINEAR HOMOTOPY ALGORITHMS FOR SPARSE SYSTEMS OF NONLINEAR EQUATIONS

by

Michael J. Todd

*This research was partially supported by a Guggenheim Fellowship, and by National Science Foundation grant ECS-7921279.
ABSTRACT

When piecewise-linear homotopy algorithms are applied to the problem of approximating a zero of a sparse function $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, a large piece of linearity can be traversed in one step by using a suitable linear system. The linear system has $n$ rows and $n+1$ columns, but is subject to a number of inequalities depending on the sparsity pattern of $f$. We show how an algorithm can be implemented using these large pieces; in particular, we demonstrate how to update the linear system corresponding to one large piece to obtain the appropriate system for an adjacent large piece. One measure of the complexity of such an implementation is the number of inequalities that may be required for any one piece. We prove that there can be no more than $O(n^{3/2})$ such inequalities, and that this bound is essentially tight; the argument is purely combinatorial. Finally, we provide guidelines on when such a "large-piece implementation" should be used instead of much simpler "small-piece implementations" for piecewise-linear homotopy algorithms.

Revised version of TR 525

1. Introduction

We are concerned with piecewise-linear homotopy algorithms for approximating a zero of a function \( f: \mathbb{R}^n \to \mathbb{R}^n \); see [1,2,4,9]. Our interest is in the case where \( f \) is sparse, i.e. each component function of \( f \) depends on only a few components of the argument. As in [11], we confine ourselves to Merrill's restart algorithm [7], though our approach can also be applied to other such algorithms, in particular those of Eaves-Saigal [3] and van der Laan-Talman [5,6]. Merrill's algorithm starts by choosing a simple function \( f^0: \mathbb{R}^n \to \mathbb{R}^n \)--we will always take \( f^0(x) = G(x-x^0) \) where \( G \) is \( n \times n \) and nonsingular--and defining \( h: \mathbb{R}^n \times [0,1] \to \mathbb{R}^n \) by \( h(x,t) = tf(x) + (1-t)f^0(x) \). Next we choose a triangulation \( T \) of \( \mathbb{R}^n \times [0,1] \) and let \( \lambda \) be the piecewise-linear approximation to \( h \) with respect to \( T \). One major cycle of the algorithm generates a sequence of simplices of \( T \) that contains the piecewise-linear path that is the connected component of \( \lambda^{-1}(0) \) containing \( (x^0,0) \) (perturbation may be necessary). If this sequence is finite, we obtain a point \( (x^1,1) \in \lambda^{-1}(0) \); either \( x^1 \) is accepted as an approximate zero or another major cycle is initiated after updating \( T \) and \( f^0 \).

Within each major cycle, each simplex that meets \( \lambda^{-1}(0) \) is associated with a certain linear system with \( n+1 \) rows and \( n+2 \) columns subject to \( n+2 \) nonnegativities. Moving from one simplex to its successor requires the evaluation of \( f \) or \( f^0 \) at the new vertex and a linear programming pivot step in the linear system.

In [10] we showed that, because of the linearity of \( f^0 \), the pieces of linearity of \( \lambda \) were much larger than individual simplices for
commonly-used triangulations $T$; moreover, if $f$ enjoyed special structure, these pieces were larger still. However, [10] proposed only a "local" method of exploiting this property - whenever it was determined that the next simplex lay in the same piece of linearity as the current one, one function evaluation was saved and the linear programming pivot step could be executed trivially. Later, we derived in [11] linear systems that allowed pieces of linearity of $x$ to be traversed in one step, when $f$ was general, separable or partially separable. These systems again had $n+1$ rows and $n+2$ columns, with up to $2n+1$ inequalities. However, the important case of a sparse function was not treated in [11]. Finally, [12] showed how these accelerated methods could use basis factorizations rather than inverses for numerical stability and preservation of sparsity and suggested that for the case of a sparse function $f$, the global idea of [11] for general $f$ could be combined with the local method of [10] for sparsity.

In this paper we give in Section 2 a linear system that allows the whole piece of linearity of $x$ to be traversed in one step when $f$ is sparse. This linear system has $n$ rows and $n+1$ columns. However, the number of inequalities depends on the sparsity structure of $f$. Section 3 indicates how a piecewise-linear homotopy algorithm can be implemented using these linear systems. One measure of the complexity of such an implementation is the number of inequalities that may be required for any one piece. Section 4 shows that there can be no more than $O(n^{3/2})$ such inequalities, and gives an example demonstrating that $O(n^{3/2})$ may be necessary. Because of the large number of inequalities possible, it may not always be advisable to use this linear system. We discuss this issue in Section 5. We conclude that if $f$ is very sparse, use of the new system is
worthwhile, and describe an alternative method of implementation for use in other cases.

Since this paper was written, Saigal [8] has proposed another piecewise-linear homotopy algorithm that exploits sparsity.

2. The Large Pieces

In this section we describe the pieces of linearity of the homotopy \( \lambda \) when \( f \) is sparse and show how, when such a piece \( \tilde{\omega} \) meets \( \lambda^{-1}(0) \) in a line segment, this line segment can be traversed by considering a linear system of the form \( Aw = b \), \( Cw \geq d \). Here \( A, b, C \) and \( d \) depend on the piece \( \tilde{\omega} \). \( A \) has \( n \) rows and \( n+1 \) columns, and the solutions to \( Aw = b \) form a line in \( \mathbb{R}^{n+1} \) whose intersection with \( \tilde{\omega} = \{ w \in \mathbb{R}^{n+1} : Cw \geq d \} \) is the desired line segment. The matrix \( C \) has \( n+1 \) columns, but the number of rows (equal to the number of facets of the piece \( \tilde{\omega} \)) depends on the sparsity pattern of \( f \). We trace the segment numerically by generating a particular solution \( \tilde{w} \) to \( Aw = b, Cw \geq d \) (corresponding to where the piece \( \tilde{\omega} \) is entered) and a vector \( z \) in the null space of \( A \). Then \( \{ w : Aw = b \} = \{ \tilde{w} + \lambda z \} \). Then we find where the segment leaves the piece by finding the range of \( \lambda \) for which \( C(\tilde{w} + \lambda z) \geq d \); this corresponds to a minimum ratio test in linear programming. While each inequality in \( Cw \geq d \) is very simple (involving at most two components of \( w \)) the complexity of the algorithm clearly depends on the number of rows of \( C \).

To make this section self-contained, we start by describing the triangulation \( \tilde{\mathcal{J}}_1 = \{ J_1(v, \pi, s) \} \) of \( \mathbb{R}^n \times [0,1] \) on which our large pieces are based. It was shown in [10,11] that this triangulation induces large pieces of linearity for several types of structure. We suppress dependence throughout on the grid size \( \epsilon \). Each individual simplex
\( \sigma = j_1(v, \pi, s) \) depends on the starting vertex \( v \in \mathbb{R}^n \times \{1\} \), the permutation \( \pi \) of \( \{1, 2, \ldots, n+1\} \) and the vector \( s \in \mathbb{R}^n \times \{-1\} \), where each component \( v_i \) of \( v \) is an odd multiple of \( \epsilon \) for \( i = 1, 2, \ldots, n \) and each component of \( s \) is \( \pm 1 \). Define \( j \) by \( \pi(j) = n+1 \) and let \( e^p \) denote the \( p \)-th unit vector of appropriate dimension. Then the vertices of \( \sigma \) are \( v^0, v^1, \ldots, v^{n+1} \), where

\[
v^0 = v, \quad v^i = v^{i-1} + \epsilon s_{\pi(i)} e^{\pi(i)}, \quad 1 \leq i < j,
v^j = v^{j-1} - e^{n+1}, \quad v^k = v^{k-1} + \epsilon s_{\pi(k)} e^{\pi(k)}, \quad j < k \leq n+1.
\] (1)

Alternatively, \( \sigma \) can be described by its facets: it is the set of all \( w \in \mathbb{R}^{n+1} \) satisfying

\[
\epsilon \geq s_{\pi(1)}(w_{\pi(1)} - v_{\pi(1)}) \geq \cdots \geq s_{\pi(j-1)}(w_{\pi(j-1)} - v_{\pi(j-1)}) \geq \epsilon(1-w_{n+1}) \geq \]

\[
s_{\pi(j+1)}(w_{\pi(j+1)} - v_{\pi(j+1)}) \geq \cdots \geq s_{\pi(n+1)}(w_{\pi(n+1)} - v_{\pi(n+1)}) \geq 0.
\] (2)

Henceforth, \( \lambda \) is the piecewise-linear approximation to \( h \) with respect to \( \hat{J}_1 \). However, because \( f^0 \) is affine—recall that we chose \( f^0(x) = G(x-x^0) \)—even for general \( f \) the pieces of linearity of \( \lambda \) are larger than the simplices of \( \hat{J}_1 \). In [11] we showed that these pieces formed a (polyhedral) subdivision \( \hat{J}_1 = \{\hat{j}_1(v, \pi, s)\} \) of \( \mathbb{R}^n \times [0,1] \). The individual piece \( \hat{j}_1(v, \pi, s) \) contains the simplex \( j_1(v, \pi, s) \) and several other simplices; thus several triples \( (v, \pi, s) \) yield the same piece \( \hat{j}_1 \). Suppose again that \( \pi(j) = n+1 \). Then \( \hat{j}_1 \) is the set of all \( w \in \mathbb{R}^{n+1} \) satisfying
\[ \epsilon \geq s_{\pi(1)}(w_{\pi(1)} - v_{\pi(1)}) \geq \cdots \geq s_{\pi(j-1)}(w_{\pi(j-1)} - v_{\pi(j-1)}) \geq \epsilon(1 - w_{n+1}); \]

\[ \epsilon(1 - w_{n+1}) \geq |w_{\pi(k)} - v_{\pi(k)}|, \quad j < k \leq n+1; \]

\[ \epsilon(1 - w_{n+1}) \geq 0 \quad \text{if} \quad j = n+1. \] (3)

Note that \( \sigma \) has \( 2n-j+2 \) facets if \( j \leq n \), and \( n+2 \) if \( j = n+1 \).

Since we will be much concerned with systems of inequalities similar to those in (3), we now introduce some very useful notation. The inequalities in (3) relate certain fundamental affine functions of \( w \). Suppressing the dependence on \( v \) and \( s \), we denote these by \( \gamma_{w-\delta_0}^0 \equiv \epsilon; \)

\[ \gamma_{w-\delta_0}^p \equiv s_p(w - v_p) \quad \text{for} \quad 1 \leq p \leq n; \quad \gamma_{n+1}^{w-\delta_{n+1}} \equiv \epsilon(1 - w_{n+1}); \quad \text{and} \quad \gamma_{n+2}^{w-\delta_{n+2}} \equiv -\epsilon(1 - w_{n+1}). \]

For \( p, q \in N_+ \equiv \{0, 1, \ldots, n+1, n+2\} \) define \( c_{pq}^p = \gamma_{-\gamma^q}^p \) and

\[ d_{pq} = \delta_{p} - \delta_{q}, \] and note that each inequality of (3) is of the form \( c_{pq}^n \geq d_{pq} \) for certain \( p, q \). The appropriate pairs \((p, q)\) are most easily identified by defining a subset \( \Gamma'_{\pi} \) of \( N_+ \times N_+ \). It is convenient to identify such subsets \( \Gamma \subseteq N_+ \times N_+ \) with the corresponding directed graphs \((N_+, \Gamma)\). Then \( \Gamma'_{\pi} \) consists of a path from 0 through \( \pi(1), \pi(2), \ldots, \pi(j-1) \) to \( n+1 \), together with edges \((n+1, \pi(k))\) and \((\pi(k), n+2)\) for \( j < k \leq n+1 \); if \( j = n+1 \), \( \Gamma'_{\pi} \) also contains the edge \((n+1, n+2)\).

For any directed graph \( \Gamma \) on \( N_+ \), we define the \(|\Gamma| \times (n+1)\) matrix \( C(\Gamma) \) to have as rows the vectors \( c_{pq}^p \) corresponding to \((p, q)\) \( \epsilon \) \( \Gamma \) and the \(|\Gamma|\)-vector \( d(\Gamma) \) similarly. It is then easy to verify that the inequalities (3) can be rewritten as \( C(\Gamma'_{\pi})w \geq d(\Gamma'_{\pi}) \).
Note that the same piece $\hat{\sigma}$ is defined by $C(\hat{\tau}_\pi)w \geq d(\hat{\tau}_\pi)$, where, in addition to the edges in $\Gamma_{\pi}'$, $\Gamma_{\pi}$ contains $(0,\pi(i))$ and $(\pi(i),n+1)$ for all $1 \leq i < j$ and $(\pi(i),\pi(i'))$ for all $1 \leq i < i' \leq j$. Clearly this new system contains a number of redundant inequalities, and the unique minimal set is as given in (3), with at most $2n+1$ inequalities.

Summarizing, we have

Lemma 1. The piece $\sigma$ is exactly \{w $\in \mathbb{R}^{n+1}$: $Cw \geq d$\}, where either $C = C(\Gamma_{\pi}')$ and $d = d(\Gamma_{\pi}')$ or $C = C(\Gamma_{\pi})$ and $d = d(\Gamma_{\pi})$.

Let $A_{\sigma}$ denote the derivative matrix of the affine function from $\mathbb{R}^{n+1}$ to $\mathbb{R}^n$ that agrees with $\lambda$ on $\hat{\sigma}$. Then clearly, for any $w$, $\overline{w} \in \hat{\sigma}$,

$$\lambda(w) = \lambda(\overline{w}) + A_{\sigma}(w-\overline{w}). \tag{4}$$

Note that, because of the form of $\hat{\sigma}$, $A_{\sigma}$ can be obtained very simply from the function values of $f$ and $f^0$ at the vertices of $\hat{\sigma}$. Let us write $y^i$ for the projection of $v^i$ $\in \mathbb{R}^n \times [0,1]$ on $\mathbb{R}^n$, so that $\lambda(v^i) = f(y^i)$ for $0 \leq i < j$, and $y^j = y^{j-1}$. If $a^i$ denotes the ith column of $A_{\sigma}$, we find

$$a^\pi(i) = (f(y^i) - f(y^{i-1}))/\varepsilon_{\pi(i)}, \quad 1 \leq i < j;$$

$$a^\pi(k) = g^\pi(k), \quad j < k \leq n+1; \tag{5}$$

$$a^{n+1} = f(y^j) - f^0(y^j),$$

with $g^k$ the kth column of the matrix $G$ used to define $f^0$. From lemma 1, (4) and (5) we obtain:
Theorem 1. The set of \( w \) lying in \( \mathcal{Q}^1(0) \cap \sigma \) is the set of solutions to \( Aw = b, \ Cw \geq d \), where \( A = A_{\sigma} \) is given by (5), \( b = A\bar{w} - \ell(\bar{w}) \) for any \( \bar{w} \in \sigma \) and \( C = C(\Gamma_\pi') \), \( d = d(\Gamma_\pi') \).

Now we introduce the sparsity of \( f \). We say that coordinates \( p \) and \( q \) interact if there is some component of \( f \) that depends on both \( x_p \) and \( x_q \). If \( f \) is differentiable, \( p \) and \( q \) interact if the \( p \)th and \( q \)th columns of the derivative matrix \( Df(x) \) have nonzeros in some common row. Naturally, we choose \( f^0 \) to have the same sparsity pattern as \( f \). Note that, except for its final dense column, the matrix \( A_{\sigma} \) has the same sparsity pattern as \( G \) and \( Df(x) \).

Suppose \( p = \pi(i) \) and \( q = \pi(i+1) \), \( 1 \leq i < j-1 \), do not interact. Then if \( 0' \) denotes \( \pi \) with the positions of \( p \) and \( q \) interchanged, we find that \( \sigma = j_1(v, 0', s) \) differs only slightly from \( \sigma \); indeed, just one vertex \( v^1 \) of \( \sigma \) changes, and \( A_{\sigma} \) coincides with \( A_{\sigma} \) except possibly in its \( p \)th and \( q \)th columns, which become

\[
\frac{f(y^1 + \epsilon e_q e^q) - f(y^1 - \epsilon e_q e^q)}{\epsilon e_p} \quad \text{and} \quad \frac{f(y^1 + \epsilon e_p e^p) - f(y^1 - \epsilon e_p e^p)}{\epsilon e_q}
\]

respectively. But because \( p \) and \( q \) do not interact, these vectors are precisely the same if the terms \( +\epsilon e_q e^q \) and \( -\epsilon e_p e^p \) are deleted everywhere. Thus \( A_{\sigma} = A_{\sigma} \). It follows that \( \ell \) is linear in \( \sigma \cup 0' \). Continuing in this way we may obtain a much larger piece of linearity than \( \sigma \) if \( f \) is sparse. To describe this piece, which we denote \( \sigma = j_1(v, \pi, s) \), we define certain subgraphs of \( \Gamma_\pi \). Let \( \Sigma \) (corresponding to sparsity) be the subgraph consisting of all edges \((0,p), (p,n+1), (n+1,p) \) and \((p,n+2) \) for \( 1 \leq p \leq n \), as well as \((0,n+1) \) and \((n+1,n+2) \), together with edges \((p,q) \) and \((q,p) \) for \( 1 \leq p,q \leq n \) if \( p \) and \( q \) interact.
Next let $\Delta_\pi = \Gamma_\pi \cap \Sigma$. We can then define $\bar{\Delta}$ as

$\{w \in \mathbb{R}^{n+1} : Cw \geq d\}$ for $C = C(\Delta_\pi)$, $d = d(\Delta_\pi)$. However, just as the system $C(\Gamma_\pi)w \geq d(\Gamma_\pi)$ contained many superfluous inequalities, so does this new system. We therefore define the "minimum cover" of $\Delta_\pi$ as $\Delta_\pi' = \{(p, r) \in \Delta_\pi : \text{there is no } q \text{ with } (p, q) \in \Delta_\pi \text{ and } (q, r) \in \Delta_\pi\}$. By considering a piecewise-linear path joining any two of the points in $\bar{\Delta}$ we can then easily prove

**Lemma 2.** $\bar{\Delta}$ is linear on the polyhedron $\bar{\Delta}$, which is the set of all $w \in \mathbb{R}^{n+1}$ satisfying $Cw \geq d$, where $C = C(\Delta_\pi')$ and $d = d(\Delta_\pi')$ or $C = C(\Delta_\pi)$ and $d = d(\Delta_\pi)$.

While $\Delta_\pi$ is always smaller than $\Gamma_\pi$, the same is not always true of $\Delta_\pi'$ with respect to $\Gamma_\pi'$. This implies that the number of inequalities defining $\bar{\Delta}$, or, geometrically, the number of its facets, may be larger than $2n+1$. We shall investigate this further in section 4. However, we immediately have

**Theorem 2.** $\bar{\Delta} \cap \mathbb{Z}^n(0)$ is $\{w \in \mathbb{Z}^{n+1} : Aw = b \text{ and } Cw \geq d\}$, where $A$ and $b$ are as in theorem 1, $C = C(\Delta_\pi')$, and $d = d(\Delta_\pi')$.

In the next section, we discuss how a piecewise-linear homotopy algorithm using the large pieces $\bar{\Delta}$ could be implemented.

3. **Implementation**

We have already given, at the beginning of section 2, an outline of how a result such as theorem 2 can be used to traverse a given piece $\bar{\Delta}$. Here we are concerned with some details of an implementation of a piecewise-linear homotopy algorithm using these large pieces.
First, we describe what must be stored in such an algorithm. We maintain a point \( \overline{w} \) with \( \mathbb{Z}(\overline{w}) = 0 \), which is the point at which the current piece \( \hat{c} \) was entered. We keep the vectors \( y \) (the projection of \( v \) onto \( R^m \)), \( f(y) \) and \( s \), the matrix \( A = A_{\hat{c}} \) and some representation of its null space. In order to generate new columns of \( A \), we also store, for each \( p \), a vector \( t^p \) of 0's and ±1's such that, if \( 1 \leq \pi^{-1}(p) < j \),
\[
a^p = (f(y+\varepsilon t^p + \varepsilon_0 e^p) - f(y + \varepsilon t^p))/\varepsilon_0 \quad \text{and} \quad a^{n+1} = f(y + \varepsilon t^{n+1}) - f(y + \varepsilon t^{n+1}).
\]
Finally the graph \( \Delta'_{\pi} \) is stored. Note that this information is not enough to recover the permutation \( \pi \), nor even its first \( j-1 \) elements, although \( j \) can easily be obtained by counting the edges emanating from node \( n+1 \).

In order to be able to generate vectors in the null space of \( A \), we will maintain an LU factorization of some permutation of the columns of \( A \). Thus \( L^{-1}AP = U = [\overline{U}, u] \), where \( L^{-1} \) is a product of permutations and lower triangular elementary matrices, \( P \) is a permutation matrix and \( U \) is upper triangular, with \( \overline{U} \) nonsingular. Thus
\[
z = P \begin{pmatrix} \overline{U}^{-1} u \\ -1 \end{pmatrix}
\]
is in the null space of \( A \).

The graph \( \Delta' = \Delta'_{\pi} \) is stored by maintaining with each \( q \in N' \) the set \( P(q) = \{ p \in N' : (p, q) \in \Delta \} \) of predecessors and the set \( S(q) = \{ r \in N' : (q, r) \in \Delta \} \) of successors of \( q \). We also store \( I = \{0, 1, 2, \ldots, n+1\} \setminus S(n+1) \) in a doubly-linked list so that if \( (p, q) \in \Delta n(I \setminus I) \), \( p \) is before \( q \) in the list. By "add (delete) \( (p, q) \) from \( \Delta' \) we mean make the appropriate changes to \( S(p) \) and \( P(q) \).
Initially, we have \( \bar{w} = (x^0, 0) \), \( y_i \) the nearest odd integer multiple of \( \epsilon \) to \( x_i^0 \) and \( s_i \) arbitrary for \( i = 1, 2, \ldots, n \).

We evaluate \( f(y) \) and thus obtain \( A = [G, f(y) - f^0(y)] \); if an LU factorization of \( G \) is available or computed, we obtain easily a corresponding LU factorization of \( A \). We set each \( t^0 = 0 \). The graph \( \Delta_n' \) is the graph with edges \((0, n+1)\) and \((n+1, p), (p, n+2)\) for \( 1 \leq p \leq n \).

Each iteration is performed as follows. We use the factorization of \( A \) to obtain a vector \( z \) in its null space. Ignoring degeneracy, we find that \( \bar{w} + \lambda z \) lies in \( \bar{\sigma} \) for \( 0 \leq \lambda \leq \bar{\lambda} \), some \( \bar{\lambda} > 0 \), or for \( \bar{\lambda} \leq \lambda \leq 0 \), some \( \bar{\lambda} < 0 \). We will choose the sign of \( z \) so that the former case occurs. Then for \( \lambda \) just larger than \( \bar{\lambda} \), \( \bar{w} + \lambda z \) lies in a new piece \( \bar{\sigma} \) (or has last component greater than one) and we must update everything stored.

The critical value \( \bar{\lambda} \) is found by examining the inequalities \( C(\Delta)(\bar{w} + \lambda z) \geq d(\Delta) \) and (again under a nondegeneracy assumption) exactly one of these inequalities will be tight for \( \lambda = \bar{\lambda} \). (Otherwise we make a perturbation, replacing \( \bar{w} \) by \( \bar{w}_\varepsilon = \bar{w} + d_\varepsilon \), where \( Ad_\varepsilon = (\varepsilon, \varepsilon^2, \ldots, \varepsilon^n)^T \) for small positive \( \varepsilon \).) Each inequality corresponds to an edge of \( \Delta \), and we analyze the update by considering each possibility. In each case \( \bar{w} + \bar{w} = \bar{w} + \lambda z \).

**Case 1.** The critical edge (leading to the tight inequality) is \((n+1, n+2)\). Then \( \bar{w} = (x^1, 1) \) for some \( x^1 \), and a major cycle is completed. We also have an LU factorization of a finite difference approximation to \( Df(x^1) \), or at least an "LH" factorization. (I.e., \( L^{-1}DP = H \) where \( D \approx Df(x^1) \) and \( H \) is upper Hessenberg so that \( h_{ij} = 0 \) for \( i > j+1 \). It is easy to obtain from this an LU factorization of \( D \).
Alternatively, we can maintain the permutation \( P \) so that \( a_{n+1} \) always corresponds to the last or penultimate column of \( U \), in which case an LU factorization of \( D \) is immediate.)

Case 2. The critical edge is \((n+1, p)\) for some \( 1 \leq p \leq n \). Define \( y^{j-1} = y + \varepsilon \sum_{i \in I} s_i e^i \) and calculate \( f(y^{j-1}) = f(y) + \varepsilon \sum_{i \in I} s_i a^i \). Then set \( s_p + 1 \) and evaluate \( f(y^{j-1} + \varepsilon s_p e^p) \) and replace \( a^p = g^p \) by

\[
(f(y^{j-1} + \varepsilon s_p e^p) - f(y^{j-1}))/\varepsilon s_p \quad \text{and then} \quad a_{n+1} \quad \text{by}
\]

\[
a_{n+1} + \varepsilon s_p (a^p - g^p). \quad \text{The LU factorization of} \quad A \quad \text{is updated}
\]

using a standard technique, see, e.g., [12]. (We may, if desired, keep \( a_{n+1} \) corresponding to one of the last two columns of \( U \); then the second replacement is trivial.) The vectors \( y \) and \( f(y) \) are unchanged, while \( s \) is only possibly changed in its \( p \)th component; we set \( t^p = \sum_i s_i e^i \) and \( t^{n+1} = t^p + s_p e^p \). Finally, we update \( \Delta \) (i.e. the doubly-linked list \( I \) and the sets \( S(q), P(q) \) for \( q \in N_+ \)) as follows.

Delete \((n+1, p), (p, n+2)\) and (if present) \((0, n+1)\) from \( \Delta \).

Add \((0, p), (p, n+1)\), and, if \( S(n+1) = \emptyset, (n+1, n+2) \) to \( \Delta \).

Set \( B = \emptyset \). (\( B \) will represent the set of nodes in \( I \) with a path to \( p \).)

For \( i \in I, \) working backwards through the list from \( n+1 \) to \( 0 \):

i) if \( S(i) \cap B \neq \emptyset \) or \( (i, p) \in \sum_i, B + B \cup \{i\} \);

ii) if \( S(i) \cap B = \emptyset \) and \( (i, p) \in \sum_i \), add \( (i, p) \) to \( \Delta \); delete \((0, p)\) from \( \Delta \).

Set \( I + I \cup \{p\} \) and add \( p \) to the doubly-linked list just before \( n+1 \).

Case 3. The critical edge is \((p, n+2)\) for some \( 1 \leq p \leq n \). Proceed exactly as in case 2 but with \( s_p - 1 \).
Case 4. The critical edge is $(p, n+1)$. This is the reverse of case 2 or 3. We replace $a^{n+1}$ by $a^{n+1} - e_p(a^p - g^p)$ and $a^p$ by $g^p$ and update the factorization of $A$. The vectors $y, f(y)$ and $s$ remain unchanged; we set $t^{n+1} + t^{n+1} - s_p e^p$. $\Delta$ is updated as follows. Set $I = I \setminus \{p\}$ and remove $p$ from the doubly-linked list. For $i \in p(p)$, delete $(i, p)$ from $\Delta$; if $S(i) = \{p\}$, add $(i, n+1)$ to $\Delta$. Remove $(p, n+1)$ and add $(n+1, p)$ and $(p, n+2)$. If $S(0) = \emptyset$, add $(0, n+1)$ to $\Delta$; if $(n+1, n+2)$ was present, delete it from $\Delta$.

Case 5. The critical edge is $(0, p)$ for some $1 \leq p \leq n$. Then set $f_{old} + f(y), y + y + 2e_p e^p$ and $s_p + e_p$. Evaluate $f(y)$ and set $a^p + a^p + (f(y) - f_{old})/e_p$; update the factorization of $A$. For each $q$, set $t^q + t^q - e_p e^p$. $\Delta$ is unchanged.

Case 6. The critical edge is $(p, q)$ for some $1 \leq p, q \leq n$. Calculate $f(y + e^p) = f(y) + eA^p$ ($A$ is $A$ with its final column $a^{n+1}$ removed). Evaluate $f(y + e^p + e^q e^q)$. Set $a_{old}^q + a^q, a^q + (f(y + e^p + e^q e^q) - f(y + e^p))/e^q$ and $a^p + a^p + (a_{old}^q - a^q)s_q/s_p$. Update the factorization of $A$. Set $t^q + t^p$ and $t^p + t^p + s_q e^q$. The vectors $y, f(y)$ and $s$ are unchanged. Finally update $\Delta$ as follows.

1) Set $B_q \rightarrow \{q\}$. ($B_q$ will represent the nodes before $q$, and not because of $p$.)

For $i \in I$, working backwards through the list from $q$ to $p$ (not including $q$ or $p$)

if $S(i) \cap B_q \neq \emptyset$, $B_q \rightarrow B_q \cup \{i\}$. 

Set $B_p + \{p\}$. ($B_p$ will represent the nodes before $p$ and not necessarily before $q$.)

For $i \in I$, working backwards from $p$ to $0$ (not including $p$ or $0$)

if $S(i) \cap B_q \neq \emptyset$, $B_q + B_q \cup \{i\}$,
    if $p \in S(i)$, delete $(i,p)$ from $\Delta$;
else if $S(i) \cap B_p \neq \emptyset$ and $(i,q) \in \Sigma$,
    $B_q + B_q \cup \{i\}$,
    add $(i,q)$ to $\Delta$,
    if $p \in S(i)$, delete $(i,p)$ from $\Delta$;
else if $S(i) \cap B_p \neq \emptyset$, $B_p + B_p \cup \{i\}$.

Add $(q,p)$ to and delete $(p,q)$ from $\Delta$.

If $(0,p)$ is present in $\Delta$, delete it.

If $P(q) = \emptyset$, add $(0,q)$ to $\Delta$.

2) Set $A_p + \{p\}$. ($A_p$ will represent the nodes after $p$, and not because of $q$.)

For $i \in I$, working forwards through the list from $p$ to $q$ (not including $p$ or $q$)

if $P(i) \cap A_p \neq \emptyset$, $A_p + A_p \cup \{i\}$.

Set $A_q + \{q\}$. ($A_q$ will represent the nodes after $q$ and not necessarily after $p$.)

For $i \in I$, working forwards through the list from $q$ to $n+1$ (not including $q$ or $n+1$)

if $P(i) \cap A_p \neq \emptyset$, $A_p + A_p \cup \{i\}$,
    if $q \in P(i)$, delete $(q,i)$ from $\Delta$;
else if $P(i) \cap A_q \neq \emptyset$ and $(p,i) \in \Sigma$,
    $A_p + A_p \cup \{i\}$,
    add $(p,i)$ to $\Delta$,
    if $q \in P(i)$, delete $(q,i)$ from $\Delta$;
else if \( P(i) \cap A_q \neq \emptyset \), \( A_q + A_q \cup \{i\} \).

If \((q, n+1)\) is present in \( \Delta \), delete it.

If \( S(p) = \emptyset \), add \((p, n+1)\) to \( \Delta \).

This concludes our discussion of the implementation of the large-piece algorithm. Note that, in each case, updating the factorization of \( A \) requires the replacement of one or two columns of \( A \) and \( U \); but that only in the final case are two general column exchanges required. The sparsity of \( A \) and its factors is likely to compensate for the possible increase in work over other implementations requiring only single column exchanges.

Note also that the sparsity information in the graph \( \Delta \) is used in performing the minimum ratio test in determining \( \bar{\lambda} \). The work involved in this test, and also in the rather complicated updating of \( \Delta \), is proportional to the cardinality of \( \Delta \), which we study in the next section.

### 4. A Bound on the Number of Facets of a Large Piece \( \check{c} \)

We have seen that the complexity of our large-piece implementation depends on the number of facets of such a piece \( \check{c}, \) or equivalently on the number of edges of a graph \( \Delta_{\pi}^i \). This section investigates how large this number can be. We can assume that \( \pi = (1, 2, \ldots, j-1, n+1, j, \ldots, n) \) for some \( 1 \leq j \leq n+1 \). Now consider the following combinatorial problem.

Let \( m = j+1 \) and let \( S_1, S_2, \ldots, S_m \) be arbitrary subsets of \( N = \{1, 2, \ldots, n\} \). Write \( p \preceq q \) if \( 1 \leq p < q \leq m \) and \( S_p \cap S_q \neq \emptyset \). We wish to find an upper bound \( \phi(m, n) \) on the cardinality of

\[
E = \{(p, r): p \preceq r \text{ and for no } q \text{ is } p \prec q \prec r\}.
\] (6)

The relationship with bounding \( |\Delta_{\pi}^i| \) is as follows. Define \( S_1 = S_m = N \) and \( S_{i+1} = \{p: f_p(x) \text{ depends on } x_i\}, \)
1 ≤ i < j. Then \( c_{pq} \) is a row of \( C(\Delta)w > d(\Delta) \)
with \( \min\{p,q\} < j \) iff \( p+1 < q'+1 \) where \( q' = j \) if \( q = n+1 \) and \( q' = q \)
otherwise; moreover inequalities of \( C(\Delta')w > d(\Delta') \) with
\( \min\{p,q\} < j \) correspond in this way to pairs in the set \( E \). Hence
\( \max\{1,2(n+1-j)\} + \phi(j+1,n) \) gives an upper
bound on \( |\Delta'| \). (The first term comes from counting edges of the form
\((n+1,p)\) or \((p,n+2)\).)

**Theorem 3.** \( \phi(m,n) = \min\{m(m-1)/2, 2m^2(n+1)/3\} \) is an upper bound on \( |E| \).

**Proof.** Since \( p < q \) implies \( p < q \), \( |E| \leq m(m-1)/2 \). So assume that
\( 2m^2(n+1)/3 < m(m-1)/2 \), so that \( \sqrt{3(n+1)} < m-1 \).

Define variables \( x_{pq} \), \( 1 \leq p < q \leq m \), to be 1 or 0 according as
the pair \((p,q)\) lies in \( E \) or not. Define

\[
y_s = \sum_{p=1}^{m-s} x_{p,p+s} \quad \text{for} \quad 1 \leq s < m.
\]

Then

\[
|E| = \sum_{s=1}^{m-1} y_s.
\]

Now \( x_{pr} = 1 \) implies that there is some index \( i \in N \) with
\( i \in S_p \cap S_r \), \( i \not\in S_q \) for \( p < q < r \). Thus in the \( m \times n \) matrix
of incidences of the sets \( S_p \) with \( N \), there is a column with the pattern
\((...,1,0,...,0,1,...)\) with the ones in rows \( p \) and \( r \). The sequence of
zeroes followed by the one in row \( r \) occupies \( r-p \) positions, and cannot be
associated with any other pair \((p,r)\) in \( E \). Thus

\[
\sum_{s=1}^{m-1} sy_s \leq mn,
\]

(7)

where the left side gives the number of positions in the incidence matrix
required to obtain the elements of \( E \) and the right side is the total
number of entries in the incidence matrix.
Next, we have \( x_{pq} + x_{qr} + x_{pr} \leq 2 \) for all \( 1 \leq p < q < r \leq m \), since if \((p,r) \in E\) we cannot have \((p,q)\) and \((q,r)\) in \(E\). Thus for \(1 \leq s < u < m\)

\[
\sum_{p=1}^{m-u} (x_{p,p+s} + x_{p+s,p+u} + x_{p+p+u}) \leq 2(m-u);
\]

adding to this

\[
\sum_{p=m-u+1}^{m-s} x_{p,p+s} \leq u-s
\]

and

\[
\sum_{p=1-s}^{0} x_{p+s,p+u} \leq s
\]

gives

\[
y_s + y_t + y_u \leq 2m
\]

whenever \(s+t = u, \ 1 \leq s,t,u < m\).

Now for any odd \(s \geq 3\) we deduce from (8)

\[
y_1
\]

\[
y_2
\]

\[
\vdots
\]

\[
y_{s-1} + y_{s+1} + y_s \leq 2m;
\]

summing these inequalities gives
\[ \sum_{r=1}^{s-1} y_r + \frac{s-1}{2} y_s \leq 2m \frac{s-1}{2}. \]  \hspace{1cm} (9)

This inequality is also valid for \( s \geq 3 \) and even, using half of \( 2y_{s/2} + y_s \leq 2m \)

as the final inequality. Then, by induction, (9) yields

\[ \sum_{r=1}^{s} y_r \leq 2ms/3 \]  \hspace{1cm} (10)

for any \( s \geq 3 \). Indeed, (10) is exactly (9) for \( s = 3 \), while \( (t-2)/2 \) times (10) for \( s = t \) added to (9) for \( s = t+1 \) yields \( t/2 \) times (10) for \( s = t+1 \). Further, we have trivially

\[ y_1 \leq m \text{ and } y_1 + y_2 \leq 2m. \]  \hspace{1cm} (11)

Then adding inequality (7) to the inequalities in (11) and inequality (10) for \( s = 3, 4, \ldots, t-1 \) we obtain

\[ t \sum_{s=1}^{t-1} y_s + \sum_{s=t}^{m-1} sy_s \leq mn + m + mt(t-1)/3 \]

so that

\[ |E| = \sum_{s=1}^{m-1} y_s \leq \frac{m(n+1)}{t} + \frac{m(t-1)}{3}. \]  \hspace{1cm} (12)

Now choose \( t = \lceil \sqrt{3(n+1)} \rceil \leq m-1 \) where for any real \( \lambda \), \( \lceil \lambda \rceil \) denotes the
least integer not less than $\lambda$ and $[\lambda]$ the greatest integer not greater
than $\lambda$. Then $m(n+1)/t \leq m\sqrt{(n+1)/3}$ and $m(t-1)/3 \leq m\sqrt{(n+1)/3}$ and the
theorem is proved.

Next we show that the bound in theorem 3 is essentially tight.

**Theorem 4.** The sets $S_1, \ldots, S_m$ can be chosen so that

$$|E| \geq \min\{m^2/4, m\sqrt{n}/2\}.$$ 

**Proof.** Suppose first $n \geq \lfloor m^2/4 \rfloor$. Assume $m$ is even—the construction
is similar with $m$ odd. Then we choose $S_1, \ldots, S_m$ so that

$(p,q) \in E$ iff $p \leq m/2 < q$. Indeed, since $n \geq m^2/4$ we can assign to
each such pair $(p,q)$ a different $i \in \mathbb{N}$, and then let $S_p$ and
$S_q$ but no other $S_r$ contain this $i$. It is clear that

$|E| = m^2/4$ with this construction.

Now suppose $n < \lfloor m^2/4 \rfloor$ so that $v = \lfloor \sqrt{n} \rfloor < m/2$. Write $m = rv + s,$
$0 < s < v$. We then divide $M = \{1, 2, \ldots, m\}$ into $M_1, \ldots, M_r$, the
first, second, ..., $r$th group of $v$ elements, and $M_{r+1}$, the last $s$
elements of $M$. Since $n \geq v^2$ we may associate with each $i \in \mathbb{N}$ a
pair $(t,u)$ with $1 \leq t, u \leq v$. This index $i$ is then put into the $r+1$
sets $S_t, S_v + \{t+u\}, S_{2v} + \{t+2u\}, \ldots, S_{rv} + \{t+ru\}$ where $[h]$
denotes the integer between $1$ and $v$ that equals $h$ modulo $v$. For
certain $t,u$, the final $S$ may have an index greater than $m$—arbitrarily
change it to $m$. It can be checked that the resulting $E$ contains all
pairs $(p,q)$ with $p \in M_k$, $q \in M_{k+1}$, $k = 1, 2, \ldots, r$; thus there
are $(r-1)v^2 + vs = v(rv+s-v) = v(m-v) > mv/2 = m\lfloor \sqrt{n} \rfloor /2$ pairs in $E.$
This completes the proof.
Theorem 3 implies that the number of facets of \( \hat{\sigma} \) can be no more than \( 1 + 2(n+2)\sqrt{(n+1)/3} \). Theorem 4 (by adding sets \( S_0 = S_{m+1} = N \) to the extremes of the sets constructed there) shows that there may be as many as \( 1 + n\lfloor \sqrt{n} \rfloor /2 \). That is, the bound is \( O(n^{3/2}) \) and is tight.

5. Discussion

Section 3 has described how a large-piece implementation can be designed, based on theorem 2. Thus we use the linear system

\[
A_\sigma w = b
\]

\[
C_\Delta w \geq d^\Delta
\]

(13)

where \( C_\Delta = C(\Delta_\pi') \), \( d^\Delta = d(\Delta_\pi') \). Some measure of the complexity of such an implementation is provided by the number of inequalities in (13), \( |\Delta_\pi'| \), which we have seen can be \( O(n^{3/2}) \).

An alternative implementation would use instead the smaller pieces \( \hat{\sigma} \) and theorem 1. This would therefore involve the linear system

\[
A_{\hat{\sigma}} w = b
\]

\[
C_{\hat{\sigma}} w \geq d^\Gamma
\]

(14)

where \( C_{\hat{\sigma}} = C(\Gamma_\pi') \), \( d^\Gamma = d(\Gamma_\pi') \). Note that (14) has at most \( 2n+1 \) inequalities. If we move from \( \hat{\sigma} \) into an adjacent piece \( \hat{\sigma}' \) and \( \ell \) is linear on \( \hat{\sigma} \cup \hat{\sigma}' \) (see the discussion below theorem 1), then \( A_{\hat{\sigma}} = A_{\hat{\sigma}'} \) and \( C_\Gamma \) and \( d^\Gamma \) have at most three new rows; only two new ratios need be computed. We can then search for the new minimum
ratio (i.e., traverse \( \hat{o} \)) and continue in this way. If it is likely that several small pieces belonging to one large piece will be encountered, then it may be worthwhile maintaining the ratios in a heap instead of recomputing the minimum afresh each time.

To determine whether it is more efficient to use (13) than (14) and whether it is worthwhile to form a heap in the latter case, we need to answer two questions. Firstly, is it likely that (13) will have more than a small multiple of \( n \) inequalities; secondly, are the pieces \( \hat{o} \) generally composed of many pieces \( \hat{o} \), or, more precisely, can line segments in pieces \( \hat{o} \) meet many pieces \( \hat{o} \)? We give some indication of the answer to these questions.

First note that the number of inequalities in (13) equals the number of facets of the piece \( \hat{o} \). Since, from lemma 1, each piece \( \hat{o} \) has at most \( 2n+1 \) facets, we conclude that whenever (13) has \( O(n^{3/2}) \) inequalities, \( \hat{o} \) consists of at least \( O(n^{1/2}) \) pieces \( \hat{o} \). A much more complete analysis can be made in particular cases.

Let us first consider the example constructed in theorem 4 (we assume \( m = n \)). The number of pieces \( \hat{o} \) contained in one piece \( \hat{o} \) is then the number of permutations that keep the first \( \lfloor \sqrt{n} \rfloor \) positions ahead of the next \( \lfloor \sqrt{n} \rfloor \), and so on. However each subgroup can be arbitrarily permuted, so that there are at least \( (\lfloor \sqrt{n} \rfloor!)^{\lfloor \sqrt{n} \rfloor} > (n/9)^{n/2} \) such pieces. Secondly, it is possible to choose \( w \) and \( z \) so that \( w \) satisfies (14) but \( w + \lambda z \), as \( \lambda \) increases, completely reverses the order of the first \( \lfloor \sqrt{n} \rfloor \) terms, the second \( \lfloor \sqrt{n} \rfloor \) terms, and so on before any of these subgroups cross. This implies that at least \( \lfloor \sqrt{n} \rfloor^2 (\lfloor \sqrt{n} \rfloor - 1)/2 \) inequalities of (14) that do not occur in (13) are violated before encountering one from (13); thus there are line segments in \( \hat{o} \) that meet \( O(n^{3/2}) \) pieces \( \hat{o} \). It is of
course very likely that far fewer pieces \( \tilde{\sigma} \) within a given piece \( \sigma \) will be met by most line segments. For this example it seems reasonable to avoid having \( O(n^{3/2}) \) inequalities by using (14) rather than (13), but to maintain the ratios in a heap to guard against sequences of a large number of pieces \( \tilde{\sigma} \) within one piece \( \sigma \).

Let us now consider a much more reasonable example. Suppose that each component \( f_i \) of the function \( f \) depends on at most \( r \) components \( x_j \) of the argument \( x \) and that each \( x_j \) similarly affects at most \( c \) \( f_i \)'s. Then each individual coordinate can interact with at most \( c(r-1) \) other coordinates. It follows that (13) has at most \( \max\{2, c(r-1)/2\}n \) inequalities. If \( cr \) is relatively small (e.g. \( c < 5, r < 5 \)) then using (13) does not seem to exact too high a price. Let us examine the possible inefficiencies of using (14) when \( Df(x) \) has small band width, i.e. \( f_i \) depends on \( x_j \) only if \( |i-j| < k \) for some \( k \ll n \). In this case \( c = r = 2k-1; \) suppose for simplicity that \( n = (2k-1)t \). Let the permutation \( \pi \) take \( (1,2,...,n+1) \) into \( (1,c+1,...,(t-1)c+1,2,c+2,...,n,n+1) \) and let \( \tilde{\sigma} = \tilde{x}_1(v,\pi,s) \) for some \( v,s \). Then, since the first, second,...,\( c \)th group of \( t \) consecutive elements can be arbitrarily permuted and yet give rise to the same piece \( \sigma \), we see that \( \tilde{\sigma} \) contains exactly \((t!)^c \) pieces \( \tilde{\sigma} \); for example, if \( Df \) is tridiagonal, \( c = r = 3 \) and \((t!)^c = ((n/3)!)^3 \). In addition, by choosing \( w \) and \( z \) appropriately as in the previous example, we find that \( \tilde{\sigma} \) contains line segments meeting at least \( ct(t-1)/2 = n(t-1)/2 \leq n^2/4k \) pieces \( \tilde{\sigma} \). Hence this example suggests the use of (13), which can be far more efficient than using (14) without incurring an excessive number of inequalities.
The conclusion from these examples is that use of the new linear system (13) for traversing the large pieces ø is recommended when the function f is very sparse so that Df has O(n) nonzeroes; otherwise the use of (3) is suggested, with the ratios maintained in a heap.

Acknowledgment

I am grateful to Mike Powell and the referees for several very helpful suggestions concerning this work.
References


