Higher Order Methods for Nonlinear Equations

Trond Steihaug

Department of Informatics
University of Bergen, Norway

Wednesday January 27, 2010
4th Winter School in eScience
Geilo, Norway

Joint work with Geir Gundersen and Shahadat Hossain

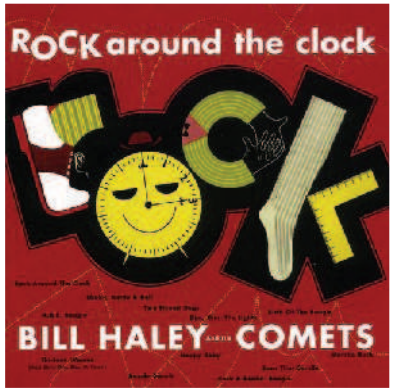
Overview

1. Higher Order Methods
   - Nonlinear systems of equations and the Halley Class.
   - Newton v.s. Halley.
   - Unconstrained Optimization
   - The effect of Sparsity
2. Computing a sparse Jacobian
   - Direct Determination
   - Symmetry

Sir Isaac Newton (1643 - 1727).

Sir Edmond Halley (1656 - 1742)

Halley and the Comet NOT Bill Haley and his Comets



Dead End

Newton's Method

Given a nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$: Solve $F(x) = 0$.

Given $x^0 \in \mathbb{R}^n$
while not converged do
    Compute $F'(x_k)$
    Solve $F'(x_k)s_k = -F(x_k)$
    Update $x_{k+1} = x_k + s_k$
end-while

Element i, j of the Jacobian matrix $F'(x)$ at x

$$F'_{i,j} = \frac{\partial}{\partial x_j} F_i(x)$$

The column j of F' can be computed by AD (or approximated by a finite difference).

$$\frac{\partial}{\partial x_j} F(x) = F'(x)e_j \approx \frac{1}{\varepsilon}\{F(x + \varepsilon e_j) - F(x)\}$$

Newton's method has under suitable assumptions Q order 2 rate of convergence:

$$\|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2)$$

Why we NOT should look at Higher Order Methods (1)

From *Numerical analysis* by L. N. Trefethen in Princeton Companion to Mathematics, Princeton U. Press. 2008.

$$\|x^{(k+1)} - x_*\| = O(\|x^{(k)} - x_*\|^2). \qquad (1)$$

Students often think it might be a good idea to develop formulas to enhance the exponent in this estimate to 3 or 4. However, this is an illusion. Taking two steps at a time of a quadratically convergent algorithm yields a quartically convergent one, so the difference in efficiency between quadratic and quartic is at best a constant factor. The same goes if the exponent 2, 3, or 4 is replaced by any other number greater than 1.

Why we NOT should look at Higher Order Methods(2)

(Ortega and Rheinboldt 1970): Methods which require second and higher order derivatives, are rather cumbersome from a computational view point. Note that, while computation of F' involves only n^2 partial derivatives $\partial_j F_i$, computation of F'' requires n^3 second partial derivatives $\partial_j \partial_k F_i$, in general exorbiant amount of work indeed.

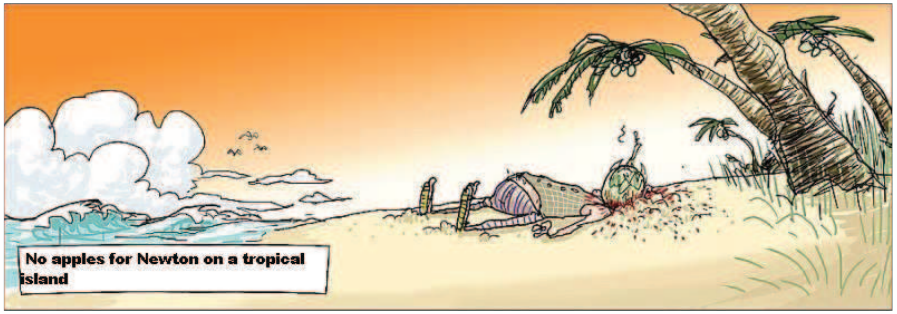
(Rheinboldt 1974): Clearly, comparisons of this type turn out to be even worse for methods with derivatives of order larger than two. Except in the case $n = 1$, where all derivatives require only one function evaluation, the practical value of methods involving more than the first derivative of F is therefore very questionable.

(Rheinboldt 1998): Clearly, for increasing dimension n the required computational work soon outweighs the advantage of the higher-order convergence.

Dead End ? ?

Nonlinear system of equations

- Computing derivatives are not difficult or expensive.
- Newton hitting "Bull's eye" is an interesting observation.
- Sparsity in the second derivative is more dominant than in the Jacobian.
- A constant factor independent of number of variables is very attractive.
- Very high accuracy (200 digits) is possible.



The Halley class of methods

Consider the nonlinear system of equations

$$F(x) = 0$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is sufficiently smooth.
The Halley class: Given starting value x_0 compute

$$x_{k+1} = x_k - \left\{ I + \frac{1}{2} L(x_k)[I - \alpha L(x_k)]^{-1} \right\} (F'(x_k))^{-1} F(x_k), \quad k = 0, 1, \ldots,$$

where

$$L(x) = (F'(x))^{-1} F''(x)(F'(x))^{-1} F(x), \quad x \in \mathbb{R}^n.$$

1. Chebyshev's method ($\alpha = 0$).
2. Halley's method ($\alpha = \frac{1}{2}$).
3. Super Halley's method ($\alpha = 1$).

The Halley class for $0 \leq \alpha \leq 1$ was proposed by Gutiérrez and Hernandez (1997) in a Banach space setting and for nonlinear system of equations by Schwetlick (1967)

One step Halley

By rewriting the iteration we get the following two-step method
For $k = 0, 1, \ldots$

Solve for $s_k^{(1)}$: $F'(x_k)s_k^{(1)} = -F(x_k)$
Solve for $s_k^{(2)}$: $\left(F'(x_k) + \alpha F''(x_k)s_k^{(1)}\right) s_k^{(2)} = -\frac{1}{2}F''(x_k)s_k^{(1)}s_k^{(1)}$
Update the iterate: $x_{k+1} = x_k + s_k^{(1)} + s_k^{(2)}$

Define the linear function $L_k(s) = F(x_k) + F'(x_k)s$. In Newton's method we solve for s_k in

$$L_k(s) = 0, \text{ and update } x_{k+1} = x_k + s_k$$

Define the quadratic function

$$T_k(s) = F(x_k) + F'(x_k)s + \frac{1}{2}F''(x_k)ss$$

Question:

Is Halley's method related to solving $T_k(s_k) = 0$ and update $x_{k+1} = x_k + s_k$?

One step Halley is two step Newton

For $k = 0, 1, \ldots$

Solve for $s_k^{(1)}$: $F'(x_k)s_k^{(1)} = -F(x_k)$
Solve for $s_k^{(2)}$: $(F'(x_k) + \alpha F''(x_k)s_k^{(1)})s_k^{(2)} = -\frac{1}{2}F''(x_k)s_k^{(1)}s_k^{(1)}$
Update the iterate: $x_{k+1} = x_k + s_k^{(1)} + s_k^{(2)}$

Note that $T_k(0) = F(x_k)$ and $T_k'(0) = F'(x_k)$

$$T_k(s_k^{(1)}) = F(x_k) + F'(x_k)s_k^{(1)} + \frac{1}{2}F''(x_k)s_k^{(1)}s_k^{(1)} = \frac{1}{2}F''(x_k)s_k^{(1)}s_k^{(1)}.$$

For $k = 0, 1, \ldots$

Solve for $s_k^{(1)}$: $T_k'(0)s_k^{(1)} = -T_k(0)$
Solve for $s_k^{(2)}$: $T_k'(s_k^{(1)})s_k^{(2)} = -T_k(s_k^{(1)})$
Update the iterate: $x_{k+1} = x_k + s_k^{(1)} + s_k^{(2)}$

Key observation

One step super Halley ($\alpha = 1$) is two steps of Newton's on a quadratic function T_k.

Local Convergence

For $k = 0, 1, \ldots$

Solve for $s_k^{(1)}$: $F'(x_k)s_k^{(1)} = -F(x_k)$
Solve for $s_k^{(2)}$: $[F'(x_k) + \alpha F''(x_k)s_k^{(1)}]s_k^{(2)} = -\frac{1}{2}F''(x_k)s_k^{(1)}s_k^{(1)}$
Update the iterate: $x_{k+1} = x_k + s_k^{(1)} + s_k^{(2)}$

Theorem

Assume that $F : \mathbb{R}^n \to \mathbb{R}^n$ is two times continuously differentiable and F'' is Lipschitz continuous in a neighborhood $\mathcal{N}$ of a point x^* where $F(x^*) = 0$ and $F'(x^*)$ is nonsingular. For each α there exists $\varepsilon > 0$ so that for all x_0 so that $\|x_0 - x^*\| \leq \varepsilon$ and $x_0 \in \mathcal{N}$, the iterates $\{x_k\}$ in the Halley class are well defined, $\|x_k - x^*\| \leq \varepsilon$ converges to x^* with at least Q-order 3.

The proof is to show the Schwetlick (1979) class is equivalent to the Halley class (G.Gundersen and T.Steihaug(2007)) proposed by Gutiérrez and Hernandez (1997) and recall the convergence result of Schwetlick (1979).

Computational Cost

For $k = 0, 1, \ldots$

Solve for $s_k^{(1)}$: $F'(x_k)s_k^{(1)} = -F(x_k)$
Solve for $s_k^{(2)}$: $[F'(x_k) + \alpha F''(x_k)s_k^{(1)}]s_k^{(2)} = -\frac{1}{2}F''(x_k)s_k^{(1)}s_k^{(1)}$
Update the iterate: $x_{k+1} = x_k + s_k^{(1)} + s_k^{(2)}$

Solving two linear systems of equations using LU factorization and computing $F''(x_k)s_k^{(1)}$ constitute the major linear algebra cost for each iteration. The cost of solving a linear system and computing $F''(x)s$ are of $O(n^3)$ in terms of arithmetic operations. This means

$$\frac{\text{Work(One step Halley)}}{\text{Work(One step Newton)}} = O(1)$$

A key issue

Can efficient use of sparsity change this ratio?

We need to look at a smaller class of problems from optimization.

Unconstrained Optimization Terminology

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a three times continuously differentiable function. For a given $x \in \mathbb{R}^n$ let

$$g_i = \frac{\partial f(x)}{\partial x_i}, \; H_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \; \mathcal{T}_{ijk} = \frac{\partial^3 f(x)}{\partial x_i \partial x_j \partial x_k} \; .$$

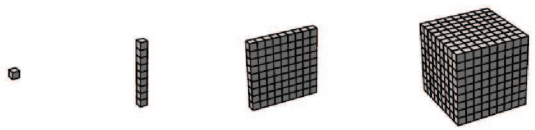
The unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) \Rightarrow \nabla f(x) = 0$$

The $n \times n$ matrix H is symmetric $H_{ij} = H_{ji}, \;\; i \neq j$.
The $n \times n \times n$ tensor $\mathcal{T}$ is super-symmetric

$$\mathcal{T}_{ijk} = \mathcal{T}_{ikj} = \mathcal{T}_{jik} = \mathcal{T}_{jki} = \mathcal{T}_{kij} = \mathcal{T}_{kji}, \qquad i \neq j, j \neq k, i \neq k$$

$$f(x), \quad g = \nabla f(x), \; H = \nabla^2 f(x) \text{ and } \mathcal{T} = \nabla^3 f(x).$$



Induced sparsity (1)

If an element of the Hessian matrix

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = 0, \text{ for all } x,$$

then for this (i, j) we have that

$$\mathcal{T}_{ijk} = \frac{\partial^3 f(x)}{\partial x_i \partial x_j \partial x_k} = 0,$$

and all the permutations of (i, j, k).

Definition

We say that the sparsity structure of the third derivative (tensor) is induced by the sparsity structure of the second derivative (Hessian).

Induced sparsity (2)

Let $\mathcal{Z}$ be the set of indices (i, j) where,

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = 0, \text{ for all } x.$$

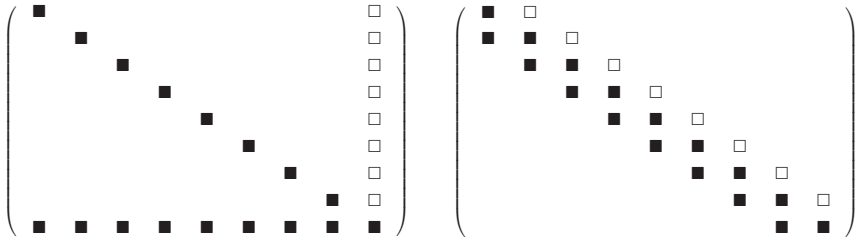
Define

$$\mathcal{N} = \{(i, j) | 1 \leq i, j \leq n\} \backslash \mathcal{Z}$$

and $\mathcal{N}$ will be the set of indices for which the elements in the Hessian matrix at x in general will be nonzero. Assume $(i, i) \in \mathcal{N}$. It follows that we only need to consider the elements (i, j, k) in the tensor, for which

$$(i, j) \in \mathcal{N},\ (j, k) \in \mathcal{N} \text{ and } (i, k) \in \mathcal{N}.$$

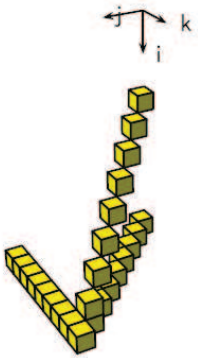
Stored elements of structured Hessian matrices

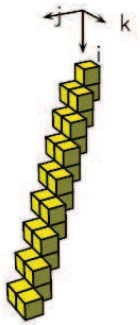


9×9 arrowhead and tridiagonal symmetric matrix. ■ stored element and □ a non-zero element not stored due to symmetry.

Sparsity structure of the tensors

Stored elements of tensors induced by an arrowhead and tridiagonal symmetric matrix where $n = 9$.





Ratio of Newton's and Halley's method

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a three times continuously differentiable and let x^* be a local minimizer where the Hessian matrix is positive definite.

Theorem

Assume that x_0 is close to x^* so that the methods converge. The ratio of the cost in number of arithmetic operations of one step of Halley's method and one step of Newtons method satisfies

$$2 \leq \frac{\textit{flaop(One Step Halley)}}{\textit{flaop(One Step Newton)}} \leq 5$$

when the linear system is solved using LDL^T and **not** *including the cost of computing the function and its derivatives.*

Note the following implicit assumptions: only the nonzero elements in the symmetric part of the tensor $\mathcal{T}$ and Hessian matrix H are stored. **flaop** - number of floating point arithmetic operations $(+, -, /, *)$.

Test functions

Chained Rosenbrock (Toint 1982):

$$f(x) = \sum_{i=2}^{n} [6.4(x_{i-1} - x_i^2)^2 + (1 - x_i)^2].$$

Generalized Rosenbrock (Schwefel 1977):

$$f(x) = \sum_{i=1}^{n-1} [(x_n - x_i^2)^2 + (x_i - 1)^2].$$

Broyden Banded (Broyden 1971),

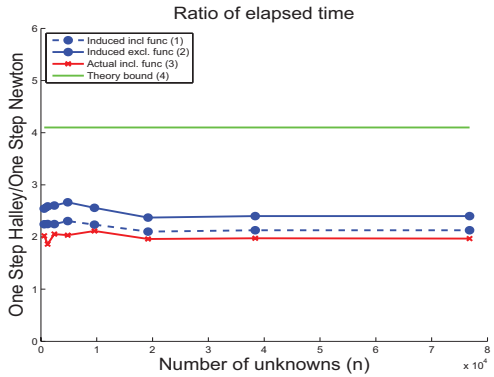
$$f(x) = \sum_{i=1}^{n} [x_i(2 + 15x_i^2) + 1 - \sum_{j \in J_i} x_j(1 + x_j)]^2.$$

where

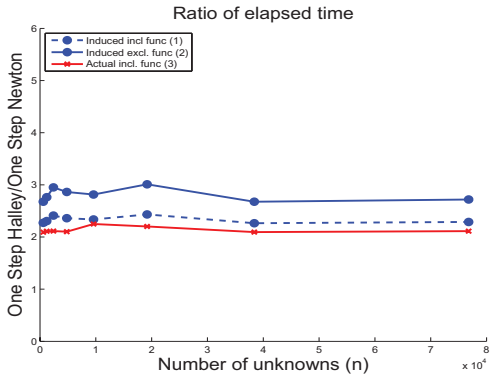
$$J_i = \{j : j \neq i, max\{1, i - m_l\} \leq j \leq min\{n, i + m_u\}\}$$

and $m_l = 5$ and $m_u = 1$.

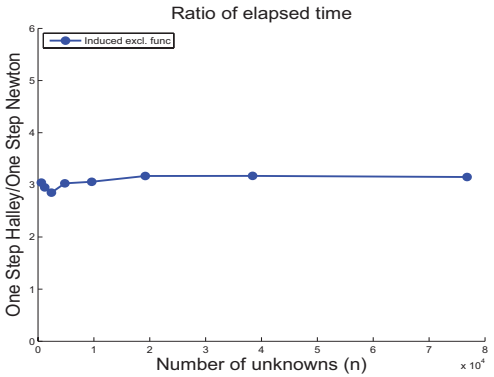
Chained Rosenbrock



Generalized Rosenbrock



Broyden Banded



Test functions:General sparsity

A Trigonometric Function (Toint 1978), the sparsity pattern is given by $\mathcal{N}$.

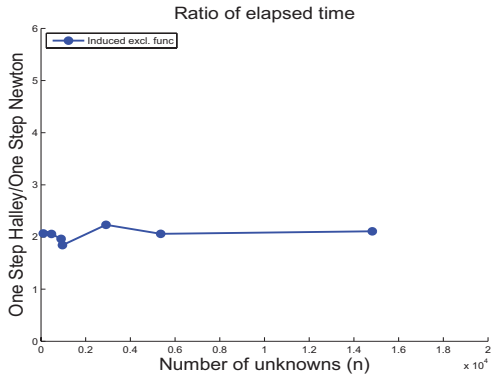
$$f(x) = \sum_{(i,j)\in\mathcal{N}} \alpha_{ij} sin(\beta_i x_i + \beta_j x_j + c_{ij}),$$

where the α_{ij}, β_i, β_j and c_{ij} are constants.

Matrix Properties			
Matrix	n	nnz(H)	$nnz(L+D)^*$
nos4	100	347	632
nos5	468	2820	18437
gr3030	900	4322	16348
nos3	960	8402	31314
nasa2910	2910	88603	202248
s3rmt3m3	5357	106526	429359
PresPoisson	14822	365313	2507325

*Symbolic phase: Approximate Minimum Degree ordering AMD Version 2.2 by P. R. Amestoy, T. A. Davis, I. S. Duff 2007.

Trigonometric Function



Concluding remarks first part

1. Higher order methods suitable when
   - Very high accuracy needed for the solution
   - Need 'bull's eye' for a noisy function
   - AD is available or taking derivatives are no pain
2. In most global methods *Newton* can be substituted by *Halley*.
3. Preliminary numerical results indicate that super Halley is as efficient as Newton's method.

Newton's Method

Given a nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$: Solve $F(x) = 0$.

Given $x^0 \in \mathbb{R}^n$
while not converged do
  Compute $F'(x^k)$
  Solve $F'(x^k)s^k = -F(x^k)$
  Update $x^{k+1} = x^k + s^k$
end-while

Column j of the Jacobian matrix at x

$$F_j' = \frac{\partial}{\partial x_j} F(x)$$

can be computed by AD (or approximated by a finite difference).

$$\frac{\partial}{\partial x_j} F(x) = F'(x)e_j \approx \frac{1}{\varepsilon}\{F(x + \varepsilon e_j) - F(x)\}$$

Assumptions

Basic Assumption

The sparsity pattern of the Jacobian matrix is known a priori and independent of the actual values of x.

- The sparsity pattern of the Jacobian matrix is known a priori and independent of the actual values of x.
- Can be computed as in AD for a neighbourhood of x
- If we need one or more components of F at x we need to compute the whole vector $F(x)$
  - It is more efficient to evaluate the vector $F(x)$ than to evaluate each component of $F(x)$ separately: common sub-expressions are evaluated only once
  - F is a computer subroutine that returns the vector $F(x)$

The Beginning

J. Inst. Maths Applics (1974) **13**, 117–119

On the Estimation of Sparse Jacobian Matrices

A. R. Curtis, M. J. D. Powell and J. K. Reid
Theoretical Physics Division, U.K.A.E.A. Research Group,
Atomic Energy Research Establishment, Harwell, Berks.

[Received 20 March 1972]

We show how to use known constant elements in a Jacobian matrix to reduce the work required to estimate the remaining elements by finite differences.

The CPR Method [Curtis, Powell, and Reid (1974)]

$$A = \begin{pmatrix} 0 & \times & & 0 & 0 & 0 \\ \times & \times & \cdots & 0 & 0 & 0 \\ \times & 0 & & \times & \times & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ \times & 0 & \cdots & 0 & \times & \times \\ 0 & \times & & 0 & 0 & \times \\ & j & & & k & \end{pmatrix}$$

$$F'_j + F'_k \approx A(:,j) + A(:,k) = A(e_j + e_k)\frac{1}{\varepsilon}[F(x + \varepsilon(e_j + e_k)) - F(x)]$$

Two columns are **structurally orthogonal** if they do not contain nonzeros in the same row position.

Partition the columns into structurally orthogonal groups

The Three Steps of the CPR method

1. Obtain the partitioning using a priori structural information
2. Compute the actual elements in the *compressed* matrix using finite differences or AD.
3. Reconstruct the elements in the Jacobian matrix.

The CPR Partitioning: A Greedy Algorithm

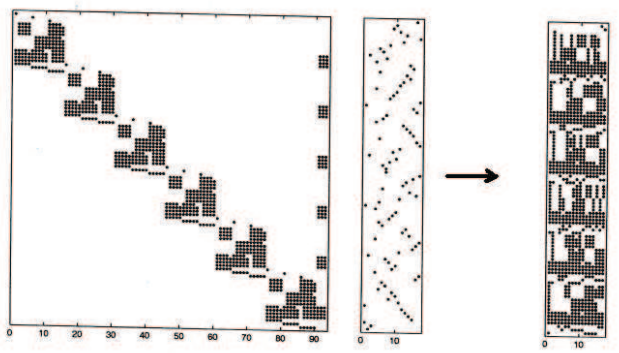
Let $C \subset \{1, 2, \cdots, n\}$ be a set of column indices of A. If $A(:, j)$ is structurally orthogonal to $A(:, k)$, $\forall k \in C$ we write

$$A(:, j) \perp_s \{A(:, k) : k \in C\} \quad (\text{ Notation: } A(:, j) \perp_s \emptyset)$$

index_set $:= \{1, 2, \cdots, n\}$
$p := 0$
while *index_set* $\neq \emptyset$ do
  $p := p + 1$
  $C_p := \emptyset$
  for $j \in$ *index_set* and $A(:, j) \perp_s \{A(:, k) : k \in C_p\}$
    $C_p := C_p \cup \{j\}$
    index_set $:=$ *index_set* $\setminus \{j\}$
  end-for
end-while

Output is structurally orthogonal groups $C_i, i = 1, \cdots, p$

Does it work?



IER from Minpack-2, $n = 93$, $p = 17$ $(= \chi(G(A)))$(Griewank 2000).

The Problem

Formulation of the problem

Obtain vectors $s_1, \cdots, s_p$ such that the matrix vector product

$$b_i \equiv As_i, \ i = 1, \cdots, p \text{ or } B \equiv AS$$

determine the $m \times n$ matrix A uniquely.

The $n \times p$ matrix S is called the *seed* matrix and the $m \times p$ matrix B is called the *compressed* matrix.

This is a too general formulation of the problem to be useful!

Main Steps in Computing A (Procedure)

- A: $m \times n$ matrix to be determined
- ρ_i: Number of nonzero entries in row i of A
- v_i: vector of column indices of nonzero entries in row i of A

1. Obtain the $n \times p$ "seed" matrix S.
2. **Seeding or Compression.** Obtain $B \ (= AS)$.
3. **Harvesting or reconstruction.** Determine the nonzero elements of A row-by-row:
   a. Identify the reduced seed matrix $\hat{S}_i \in R^{\rho_i \times p}$ for $A(i, v_i)$

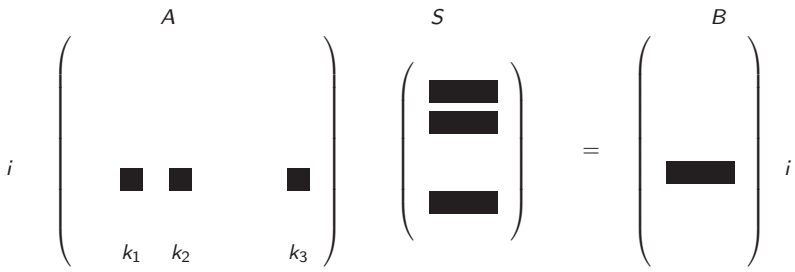
   $$\hat{S}_i = S(v_i, :)$$

   b. Solve for the ρ_i unknown elements $a_{ik} \neq 0$ of $A(i, :)$

   $$\hat{S}_i{}^T A(i, v_i)^T = B(i, :)^T$$

Bottom Line: Obtain a suitable seed matrix $S \in R^{n \times p}$ any square submatrix of which is *numerically well-conditioned* and *easy to solve*

Harvesting row i of A



Harvesting the unknown entries with column indices k_1, k_2, and k_3 in row i of A

$$\hat{S_i}^T A(i, v_i)^T = B(i, :)^T.$$

Classification of Jacobian Matrix Determination Methods (Powell and Toint ,1979)

Let $\alpha = A(i, v_i)^T \in R^{\rho_i}$ and $\beta = B(i, :)^T \in R^p$. Then the unknown elements satisfy the overdetermined system ($\rho_i \leq p$)

$$\hat{S}^T \alpha = \beta$$

Let $\rho_i = p$. If $\hat{S}$

- is a permutation matrix then we have **direct determination**
- can be permuted to a triangular matrix then we have **determination by substitution**
- is a general nonsingular matrix we have **determination by elimination**.

Fact: The minimal number of matrix vector products p for any method is

$$p = \rho \equiv \max_i \rho_i.$$

Intersection Graph

Graph Concepts
$A \in R^{m \times n}$, $G(A) = (V, E)$ $V = \{A(:,1), \cdots, A(:,n)\}$
$E = \{\{A(:,i), A(:,j)\} : A(:,i) \not\perp_S A(:,j)\}$.

A **p-coloring** of the vertices of G is a function $\phi : V \rightarrow \{1, 2, \cdots, p\}$ such that $\{u, v\} \in E \Rightarrow \phi(u) \neq \phi(v)$. The **chromatic number** $\chi(G(A))$ is the smallest p for which $G(A)$ has a p-coloring.

Column partition
A *partition* of the columns of A is a division of columns into groups $C_1, C_2, \cdots, C_p$ such that each column belongs to one and only one group.

Consistent partition

A column partition where each group consists of *structurally orthogonal* columns is called a consistent (with direct determination) partition.

Consistent partitioning and coloring (Coleman and Moré[1983])

- ϕ is a coloring of $G(A)$ if and only if ϕ induces a consistent partition of the columns of A
- Coloring $G(A)$ is as hard as coloring a general graph
- The CPR method is a greedy coloring method
  Consider vertices $v_k = A(:,k)$ in their given order $1, \cdots, n$.

  for $k = 1, \cdots, n$
  Assign vertex v_k the smallest possible color

  Ordering of the vertices affects the coloring.
- Early numerical testing on the DSM code indicated p close to $\rho = \max_i \rho_i$ or equal largest identified clique.

Some CPR implementations

Matrix	m	n	nnz	ρ	DSM	GMP
af23560	23560	23560	484256	21	41	**32**
cage11	39082	39082	559722	31	**62**	81
cage12	130228	130228	2032536	33	**68**	96
e40r0100	9661	9661	306356	62	70	**66**
ihr34	14270	14270	307858	63	**63**	65
ihr71c	70304	70304	1528092	63	**63**	65

DSM: Coleman, Garbow and Moré 1984
GMP: Gebremedhin, Manne and Pothen 2005
All these methods have storage $\theta(\text{nnz}(A))$.

Two issues: Approx v.s. exact (finding $\chi(G(A))$) and is this the best we can do (in terms of p)?

Conjecture: The chromatic number of the intersection graph is the minimal p

COLORING LARGE SPARSE JACOBIANS AND HESSIANS

Thomas F. Coleman and Jorge J. Moré**

Large scale optimization problems usually involve computing or estimating a Jacobian or Hessian matrix. If the matrix is known to be sparse, then estimation by finite differences becomes attractive since the number of function evaluations needed is often small, relative to the order of the matrix. For example, if the matrix is tridiagonal then it is known that, at most, three function evaluations are needed. In this paper we consider the general situation: Given an m×n Jacobian, or n×n Hessian matrix with known sparsity structure, what is the minimum number of function evaluations needed to obtain the nonzero quantities in a direct way? We demonstrate that in both the symmetric and unsymmetric cases this problem is equivalent to a graph coloring problem. Using this equivalence, we derive column and row invariant algorithms to 'color' Jacobian and Hessian matrices. Numerical results are given and complexity questions are considered.

The Eisenstat Counter Example

$$A = \left(\begin{array}{cccccc} a_{11} & 0 & 0 & a_{14} & 0 & 0 \\ 0 & a_{22} & 0 & 0 & a_{25} & 0 \\ 0 & 0 & a_{33} & 0 & 0 & a_{36} \\ \hline a_{41} & a_{42} & a_{43} & 0 & 0 & 0 \\ a_{51} & 0 & 0 & 0 & a_{55} & a_{56} \\ 0 & a_{62} & 0 & a_{64} & 0 & a_{66} \\ 0 & 0 & a_{73} & a_{74} & a_{75} & 0 \end{array}\right) = \left(\begin{array}{c} A_1 \\ A_2 \end{array}\right)$$

$G(A)$ is complete so $S = I$ and $p = 6$.

OR

1. estimate the the first 3 rows $(n/2)$ of $A \equiv A_1$ using 2 matrix-vector products
2. estimate the last 4 $(n/2+1)$ rows of $A \equiv A_2$ using 3 $(n/2)$ matrix-vector products

yields $p = 5$ $(n/2+2)$ matrix-vector products to determine A directly v.s. 6 (n).

The Eisenstat Counter Example

$$AS = \left(\begin{array}{cccccc} a_{11} & 0 & 0 & a_{14} & 0 & 0 \\ 0 & a_{22} & 0 & 0 & a_{25} & 0 \\ 0 & 0 & a_{33} & 0 & 0 & a_{36} \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 \\ \boxed{a_{51}} & 0 & 0 & 0 & \boxed{a_{55}} & \boxed{a_{56}} \\ 0 & a_{62} & 0 & a_{64} & 0 & a_{66} \\ 0 & 0 & a_{73} & a_{74} & a_{75} & 0 \end{array}\right) \left(\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right)$$

Compression

$$B = AS = \left(\begin{array}{ccccc} a_{11} & a_{14} & a_{11}+a_{14} & 0 & 0 \\ a_{22} & a_{25} & 0 & a_{22}+a_{25} & 0 \\ a_{33} & a_{36} & 0 & 0 & a_{33}+a_{36} \\ a_{41}+a_{42}+a_{43} & 0 & a_{41} & a_{42} & a_{43} \\ \boxed{a_{51}} & \boxed{a_{55}+a_{56}} & \boxed{a_{51}} & \boxed{a_{55}} & \boxed{a_{56}} \\ a_{62} & a_{64}+a_{66} & a_{64} & a_{62} & a_{66} \\ a_{73} & a_{74}+a_{75} & a_{74} & a_{75} & a_{73} \end{array}\right)$$

The Eisenstat Counter Example (Reconstruction)

$$\hat{S}^T \alpha = \beta$$

For example the nonzero elements in row 5 is determined in the reduced linear system

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix}$$

From Compression $\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix} = \begin{pmatrix} a_{51} \\ a_{55} + a_{56} \\ a_{51} \\ a_{55} \\ a_{56} \end{pmatrix}$ which gives $\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} a_{51} \\ a_{55} \\ a_{56} \end{pmatrix}$.

Optimal Direct Determination

The *element isolation graph* (Newsam and Ramsdell, 1983) associated with $A \in R^{m \times n}$ is denoted $G_{\mathcal{I}} = (V, E)$ where $V = \{a_{ij} \neq 0 : 1 \leq i \leq m, 1 \leq j \leq n\}$ and

$$E = \{\{a_{ij}, a_{pq}\} \mid a_{ij} \text{ is not isolated from } a_{pq}\}.$$

NR'83 called structural orthogonal columns for Variable Isolation.
Characterization of Optimal Direct Determination (Hossain and Steihaug, 2003, 2006)

Theorem

The minimal number of matrix-vector multiply in any direct determination method is $p = \chi(G_{\mathcal{I}}(A))$.

The Eisenstat Example Revisited - Seeding

$$\begin{pmatrix} a_{11} & 0 & 0 & a_{14} & 0 & 0 \\ 0 & a_{22} & 0 & 0 & a_{25} & 0 \\ 0 & 0 & a_{33} & 0 & 0 & a_{36} \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 \\ \boxed{a_{51}} & 0 & 0 & 0 & \boxed{a_{55}} & \boxed{a_{56}} \\ 0 & a_{62} & 0 & a_{64} & 0 & a_{66} \\ 0 & 0 & a_{73} & a_{74} & a_{75} & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} a_{14} & a_{14} & a_{11} & a_{11} & \\ a_{25} & a_{22} & a_{25} & a_{22} & \\ a_{33} & a_{36} & a_{36} & a_{33} & \\ a_{43} & a_{42} & a_{41} & a_{43}+ & a_{41}+a_{42} \\ \boxed{a_{55}} & \boxed{a_{56}} & \boxed{a_{51}+a_{55}+a_{56}} & \boxed{a_{51}} & \\ a_{64} & a_{62}+a_{64}+a_{66} & a_{66} & a_{62} & \\ a_{73}+a_{74}+a_{75} & a_{74} & a_{75} & a_{73} & \end{pmatrix}$$

The Eisenstat Example Revisited - Harvesting

Consider row 5. We can reconstruct the matrix with $p = 4$ (v.s. 5).

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}$$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} a_{55} \\ a_{56} \\ a_{51}+a_{55}+a_{56} \\ a_{51} \end{pmatrix} \text{ which gives } \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} a_{51} \\ a_{55} \\ a_{56} \end{pmatrix}.$$

Matrices with Optimal DD where DSM is optimal

Matrix	DSM			Element Isolation		
	Nodes	*Edges*	$\chi(G(A))$	*Nodes*	*Edges*	$\chi(G_{\mathcal{I}}(A))$
ash219	85	219	4	438	2205	4
abb313	176	3206	10	1557	65390	9
ash331	104	331	6	662	4185	4
will199	199	960	7	701	7065	7
ash608	188	608	6	1216	7844	4
ash958	292	958	6	1916	12506	4

Examples from the Harwell package where CPR gives optimal coloring

$$p = \chi(G(A)) > \rho = \max_{i=1}^{m} \rho_i$$

.

Hard Coloring Instances (2008)

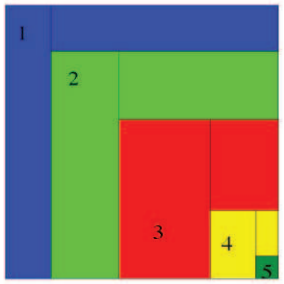
Matrix	$\|G\|$	$\|E\|$	(1)	(2)	(3)	(4)	(5)	(6)	Optimal
abb313	1557	53356		11	10	9			Gap
ash331	662	4185	4	5		4	4	4	Bound
ash608	1216	7844		5		4		4	Bound
ash958	1916	12506		6		4		4	Bound
will199	701	6772	7	7	7	7			Gap

(1) Croitoru, Gheorghies, and Apetrei (2005)
(2) Galinier, Hertz and Zufferey (2005)
(3) Mendez-Diaz Dukanovic (2005)
(4) Bui, Nguyen, Patel, and Phan (2005)
(5) Desrosiers, Galinier, and Hertz (2005)
(6) Mendez-Diaz (2005)

Bound: Lower bound equal p.

Symmetric Matrix

Direct determination of a symmetric matrix (Hessian) Powell and Toint (1979):



Rows and Columns - Forward and Reverse in AD

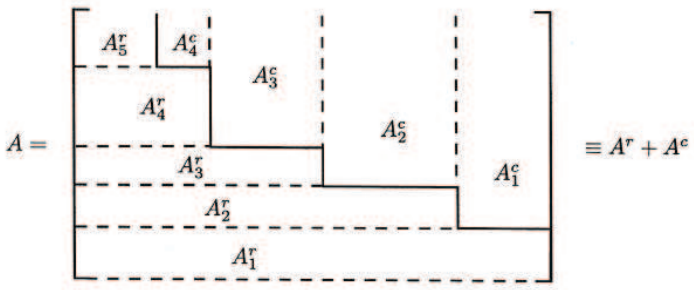
$$A = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 \\ \times & 0 & 0 & \times & 0 \\ \times & 0 & 0 & 0 & \times \end{pmatrix}, W^T = (1\,0\,0\,0\,0), S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

The Bipartition Problem

Given $A \in R^{m \times n}$ obtain vectors $w_1, \ldots, w_r \equiv W$ and $s_1, \ldots, s_p \equiv S$ such that for each $a_{ij} \neq 0$ there is an index l_c such that $a_{ij} = b_{il_c}$ or that there is an index l_r such that $a_{ij} = c_{jl_r}$ where $W^T A = C^T$ and $AS = B$ and with $p = p_c + p_r$ minimized

Method using forward AND Reverse mode of AD

Coleman and Verma 1996. Alternate between reverse (determine a group of rows) and forward (determine a group of columns) mode of AD.



Symmetry II: A "New" Direct Method (Thapa, 1980)

$$A = \begin{pmatrix} \times & \times & * & * & 0 & 0 \\ * & \times & \times & 0 & * & 0 \\ \times & * & \times & 0 & 0 & * \\ \times & 0 & 0 & \times & 0 & 0 \\ 0 & \times & 0 & 0 & \times & 0 \\ 0 & 0 & \times & 0 & 0 & \times \end{pmatrix}, S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Obtain a partitioning with the extension that at most one overlap is allowed for each column group and let one of the overlapping columns be in the new group. Note that the harvesting phase is not purely row-wise. We are now leaving DD.

Concluding remarks second part

1. Direct Determination
   - Optimal DD is very expensive
   - Heuristic Methods often very close to optimal column partitioning
   - The Harvesting is very cheap for DD
   - DD is often far from optimal Jacobian determination ($=\rho = \max_i \rho_i$)
2. Symmetry must be utilized
3. Minimizing p is done once, seeding and harvesting for every iteration.

The End