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

Joint work with Geir Gundersen and Shahadat Hossain

Sir Isaac Newton (1643 - 1727).
Sir Edmond Halley (1656 - 1742)
Newton’s Method

Given a nonlinear function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$: 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 $O$ order 2 rate of convergence:

$$ \| x_{k+1} - x^* \| = O(\| x_k - x^* \|^2) $$
Why we **NOT** should look at Higher Order Methods (1)


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

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 that 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 viewpoint. 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 exorbitant 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.

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

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

where

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

- Chebyshev’s method (\( \alpha = 0 \)).
- Halley’s method (\( \alpha = \frac{1}{2} \)).
- 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 $k = 0, 1, \ldots$

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

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

$L_k(s) = 0$, 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 $x_{k+1} = x_k + s_k$?
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 $\alpha$ 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.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a three times continuously differentiable function. For a given $x \in \mathbb{R}^n$ let

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

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 $T$ is super-symmetric

$T_{ijk} = T_{ikj} = T_{jki} = T_{jik} = T_{kji} = T_{kij}$, $i \neq j, j \neq k, i \neq k$

$f(x)$, $g = \nabla f(x)$, $H = \nabla^2 f(x)$ and $T = \nabla^3 f(x)$.

### 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

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

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).
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
\[
\mathcal{N} = \{(i,j) | 1 \leq i, j \leq n \} \setminus \mathcal{Z}
\]
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

$$(i,j) \in \mathcal{N}, \ (j,k) \in \mathcal{N} \text{ and } (i,k) \in \mathcal{N}.$$
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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 Newton’s method satisfies

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

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 $T$ and Hessian matrix $H$ are stored. \textit{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):

\[ f(x) = \sum_{i=1}^{n-1} [(x_0 - x_i^2)^2 + (x_i - 1)^2]. \]

Broyden Banded (Broyden 1971),

\[ f(x) = \sum_{i=1}^{n} [x_i(2 + 15x_i^3) + 1 - \sum_{j \in J} x_j(1 + x_j)^2]. \]

where

\[ J_i = \{ j : j \neq i, \max\{1, i - m_i\} \leq j \leq \min\{n, i + m_u\} \} \]

and \( m_i = 5 \) and \( m_u = 1 \).

---

<table>
<thead>
<tr>
<th>Chained Rosenbrock</th>
<th>Induced incl. func (1)</th>
<th>( \frac{\text{Ratio of elapsed time}}{\text{Number of unknowns (n)}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Step Halley</td>
<td>One Step Newton</td>
<td>( \frac{\text{Ratio of elapsed time}}{\text{Number of unknowns (n)}} )</td>
</tr>
</tbody>
</table>
Generalized Rosenbrock

Ratio of elapsed time

Broyden Banded

Ratio of elapsed time
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 \( \alpha_i, \beta_i, \beta_j \) and \( c_{ij} \) are constants.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>( n )</th>
<th>( \text{nnz}(P) )</th>
<th>( \text{nnz}(L + D) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>nos4</td>
<td>100</td>
<td>347</td>
<td>632</td>
</tr>
<tr>
<td>nos5</td>
<td>466</td>
<td>2620</td>
<td>16437</td>
</tr>
<tr>
<td>gr3030</td>
<td>900</td>
<td>4122</td>
<td>16348</td>
</tr>
<tr>
<td>nos3</td>
<td>900</td>
<td>8402</td>
<td>31314</td>
</tr>
<tr>
<td>nasa2930</td>
<td>2910</td>
<td>88603</td>
<td>202248</td>
</tr>
<tr>
<td>s3rmt3m3</td>
<td>5357</td>
<td>106526</td>
<td>429359</td>
</tr>
<tr>
<td>Pres1Poisson</td>
<td>14822</td>
<td>365313</td>
<td>2507325</td>
</tr>
</tbody>
</table>


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.

Given a nonlinear function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$: 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).

$$\frac{\partial}{\partial x_j} F(x) = F'(x)e_j \approx \frac{1}{\epsilon} \{F(x + \epsilon e_j) - F(x)\}$$
**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 CPR Method [Curtis, Powell, and Reid (1974)]

\[ A = \begin{pmatrix}
0 & \times & 0 & 0 & 0 \\
\times & \times & \cdots & 0 & 0 \\
\times & 0 & \times & \times & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\times & 0 & \cdots & 0 & \times \\
0 & \times & 0 & 0 & \times \\
\end{pmatrix}_{j,k} \]

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

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

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.
Let $C \subseteq \{1, 2, \cdots, n\}$ 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 $j \in \text{index set and } A(:, j) \perp_s \{A(:, k) : k \in C_p\}$
    $C_p := C_p \cup \{j\}$
  end-for
  index_set := index_set \{j\}
end-while

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

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

Formulation of the problem

Obtain vectors \( s_1, \ldots, s_p \) such that the matrix vector product

\[
  b_i \equiv A s_i, \quad i = 1, \ldots, p \quad \text{or} \quad B \equiv A S
\]

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)

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

$$\begin{pmatrix} A \\ i \end{pmatrix} \times \begin{pmatrix} S \\ k_1 \\ k_2 \\ k_3 \end{pmatrix} = \begin{pmatrix} B \end{pmatrix}$$

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

$$\hat{S}_i^T A(i,v)_i^T = B(i,:)^T.$$
Graph Concepts

A \in \mathbb{R}^{m \times n}, \ G(A) = (V, E) \ V = \{A(:,1), \ldots, A(:,n)\}

E = \{\{A(:,i), A(:,j)\} : A(:,i) \nperp A(:,j)\}.

A p-coloring of the vertices of G is a function \phi : V \rightarrow \{1, 2, \ldots, 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, \ldots, 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.

- \phi \text{ is a coloring of } G(A) \text{ if and only if } \phi \text{ 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) \text{ in their given order } 1, \ldots, n.
  for \ k = 1, \ldots, n
  - Assign vertex \ v_k \text{ the smallest possible color}
  Ordering of the vertices affects the coloring.
- Early numerical testing on the DSM code indicated \ p \text{ close to } \rho \text{ or equal largest identified clique.}
Some CPR implementations

<table>
<thead>
<tr>
<th>Matrix</th>
<th>m</th>
<th>n</th>
<th>nnz</th>
<th>ρ</th>
<th>DSM</th>
<th>GMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>at23560</td>
<td>23560</td>
<td>23560</td>
<td>484256</td>
<td>21</td>
<td>41</td>
<td>32</td>
</tr>
<tr>
<td>cage11</td>
<td>39082</td>
<td>39082</td>
<td>559722</td>
<td>31</td>
<td>62</td>
<td>81</td>
</tr>
<tr>
<td>cage12</td>
<td>130228</td>
<td>130228</td>
<td>2032536</td>
<td>33</td>
<td>68</td>
<td>96</td>
</tr>
<tr>
<td>e40r0100</td>
<td>9661</td>
<td>9661</td>
<td>306356</td>
<td>62</td>
<td>70</td>
<td>66</td>
</tr>
<tr>
<td>ihr34</td>
<td>14270</td>
<td>14270</td>
<td>307858</td>
<td>63</td>
<td>63</td>
<td>65</td>
</tr>
<tr>
<td>ihr71c</td>
<td>70304</td>
<td>70304</td>
<td>1528092</td>
<td>63</td>
<td>63</td>
<td>65</td>
</tr>
</tbody>
</table>

DSM: Coleman, Garbow and Moré 1984
GMP: Gebremedhin, Manne and Pothen 2005
All these methods have storage θ(nnz(A)).

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

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

\[ A = \begin{pmatrix}
  a_{11} & 0 & 0 & a_{14} & 0 & 0 \\
  0 & a_{22} & 0 & 0 & a_{25} & 0 \\
  0 & 0 & a_{33} & 0 & 0 & a_{36} \\
  a_{41} & a_{42} & a_{43} & 0 & 0 & 0 \\
  a_{51} & 0 & 0 & a_{55} & a_{56} & 0 \\
  0 & a_{62} & 0 & a_{64} & a_{65} & a_{66} \\
  0 & 0 & a_{73} & a_{74} & a_{75} & 0
\end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \]

\[ G(A) \text{ is complete so } S = I \text{ and } p = 6. \]

OR

1. estimate 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 (Reconstruction)

\[ S^T \alpha = \beta \]

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

\[
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\end{pmatrix}
=
\begin{pmatrix}
a_{51} \\
a_{55} \\
a_{56} \\
\end{pmatrix}
\]

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

\[ E = \{a_{ij} : a_{ij} \text{ is not isolated from } a_{pq}\}. \]


**Theorem**

The minimal number of matrix-vector multiply in any direct determination method is

\[ p = \chi(G_E(A)). \]
Consider row 5. We can reconstruct the matrix with $p = 4$ (v.s. 5).

$$
\begin{pmatrix}
\frac{a_{11}}{a_{14}} + a_{14} & a_{14} & a_{11} & a_{11} \\
a_{25} & a_{25} & a_{22} & a_{22} \\
a_{36} & a_{36} & a_{33} & a_{33} \\
a_{42} & a_{42} & a_{41} & a_{41} + a_{42} \\
\frac{a_{51}}{a_{54}} + \frac{a_{54}}{a_{56}} + a_{56} & \frac{a_{51}}{a_{54}} + \frac{a_{54}}{a_{56}} + a_{56} & \frac{a_{51}}{a_{54}} + \frac{a_{54}}{a_{56}} + a_{56} & a_{55} + a_{56} + a_{56} \\
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\alpha_4 \\
\end{pmatrix}
= \begin{pmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\end{pmatrix}
\begin{pmatrix}
a_{55} \\
a_{56} \\
a_{51} + a_{55} + a_{56} \\
a_{51} \\
\end{pmatrix}
$$
which gives

$$
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\end{pmatrix}
= \begin{pmatrix}
a_{51} \\
a_{55} \\
a_{56} \\
a_{51} \\
\end{pmatrix}
.$$
Nonlinear system of equations
Computing a sparse Jacobian
The CPR method
Direct Determination

Matrices with Optimal DD where DSM is optimal

<table>
<thead>
<tr>
<th>Matrix</th>
<th>DSM</th>
<th>Element Isolation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nodes</td>
<td>Edges</td>
</tr>
<tr>
<td>ash219</td>
<td>85</td>
<td>219</td>
</tr>
<tr>
<td>abb313</td>
<td>176</td>
<td>3206</td>
</tr>
<tr>
<td>ash331</td>
<td>104</td>
<td>331</td>
</tr>
<tr>
<td>will199</td>
<td>199</td>
<td>960</td>
</tr>
<tr>
<td>ash608</td>
<td>188</td>
<td>608</td>
</tr>
<tr>
<td>ash958</td>
<td>292</td>
<td>958</td>
</tr>
</tbody>
</table>

Examples from the Harwell package where CPR gives optimal coloring

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

(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 $\rho$. 
Direct determination of a symmetric matrix (Hessian) Powell and Toint (1979):

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

The Bipartition Problem

Given \( A \in \mathbb{R}^{m \times n} \) 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 \( a_{ij} = b_{il} \), or that there is an index \( l_r \) such that \( a_{ij} = c_{jl} \), where \( W^T A = C^T \) and \( AS = B \) and with \( p = p_c + p_r \) minimized.
Coleman and Verma 1996. Alternate between reverse (determine a group of rows) and forward (determine a group of columns) mode of AD.

\[
\begin{pmatrix}
A_1 & A_2 & A_3 & A_4 \\
\end{pmatrix}
\equiv A' + A^c
\]

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.
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 ($\rho = \max_i \rho_i$)

Symmetry must be utilized
Minimizing $p$ is done once, seeding and harvesting for every iteration.

The End