Sparsity in Higher Order Methods in Optimization*

Geir Gundersen

Department of Informatics

University of Bergen, Norway

June. 16, 2006

Sparse Days Meeting 2006 at CERFACS.
June 15th-16th, 2006

*Joint with Trond Steihaug

Motivation

Developing algorithms for solving unconstrained optimization problems using a cubic model with exact third derivatives, which are competitive with quadratic models.

Computing higher order derivatives is made possible with Automatic Differentiation (AD).

Competitiveness is achieved by utilizing the problem structure.

Overview

- Higher-Order *Local* Methods.
- Unconstrained Optimization using a Cubic Model.
- A Higher-Order *Global* Method.
- How to Utilize Structure in the Problem.
- The Cost from Quadratic to Cubic Model.

Methods for Solving Nonlinear Equations

One of the central problems of scientific computation is the efficient numerical solution of the system of n equations in n unknowns

$$F(x) = 0$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is sufficiently smooth.

Consider the Halley class of iterations (Guiterrez and Hernandez 2001).

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

where

$$L(x) = (F^{'}(x))^{-1}F^{''}(x)(F^{'}(x))^{-1}F(x), \;\; x \in X$$

Methods for Solving Nonlinear Equations

The Halley class contains the following classical methods:

Chebyshev's method $(\alpha = 0)$

Halley's method $(\alpha = \frac{1}{2})$

Super Halley's method $(\alpha = 1)$.

All members in the Halley class are cubically convergent.

Methods for Solving Nonlinear Equations

The formulation by (Guiterrez and Hernandez 2001) is not suitable for implementation. By rewriting the equation we get the following iterative method for $k = 0, 1, \ldots$

Solve for $s_k^{(1)}$:

$$F'(x_k)s_k^{(1)} = -F(x_k)$$

Solve for $s_k^{(2)}$:

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

The new step is

$$x_{k+1} = x_k + s_k^{(1)} + s_k^{(2)}$$

Methods for Solving Nonlinear Equations

These methods also apply for algorithms for the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

for

$$F(x) = \nabla f(x),\ F^{'} = \nabla^2 f(x) \text{ and } F^{''} = \nabla^3 f(x)$$

Third order methods have a superior rate of convergence compared to Newton's method to reach the same accuracy.

Using Higher Order Methods?

The following statements show that higher order methods is not considered practical.

(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.

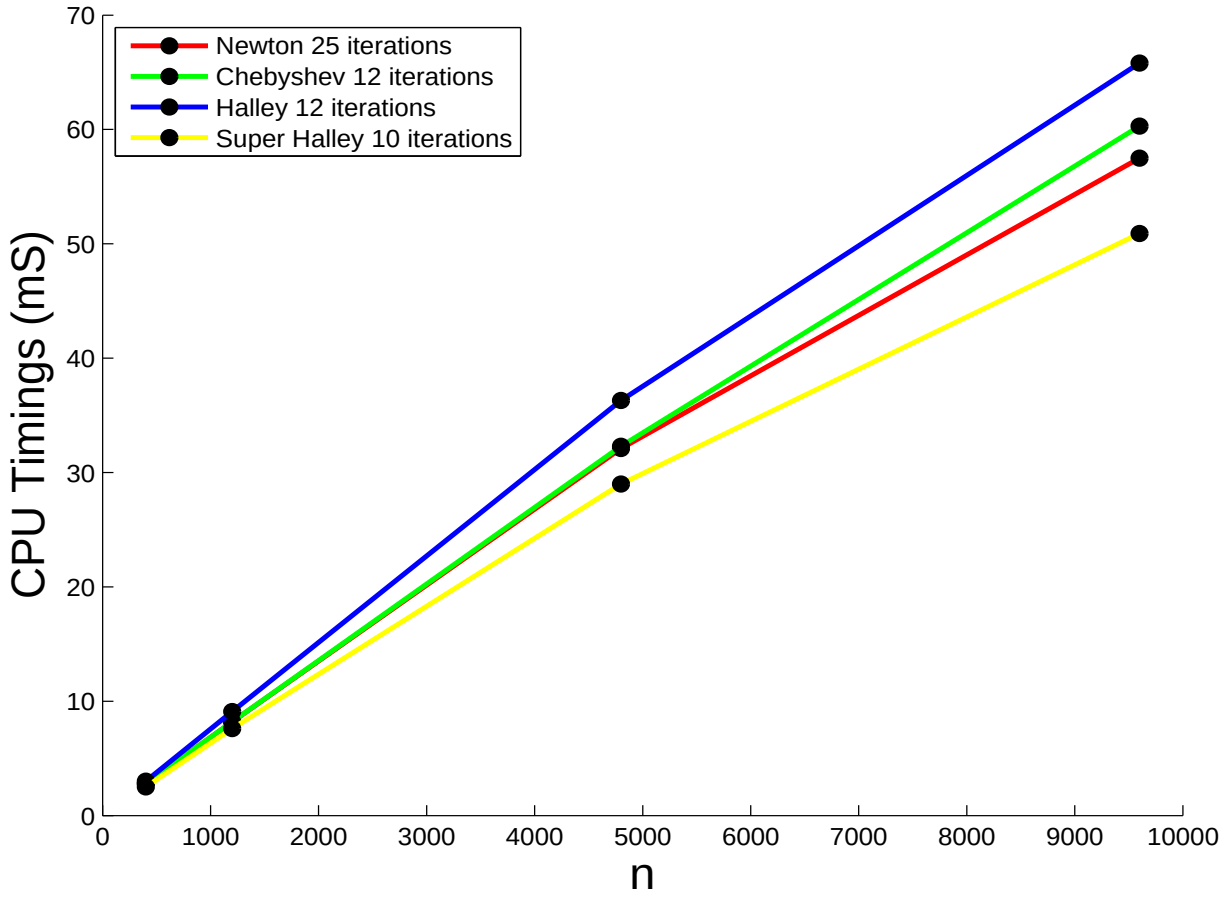
(Rheinboldt 1974): Clearly, comparisons of this type turn out to be even worse for methods with derivatives of order larger than two. Except in the case $n = 1$, where all derivatives require only one function evaluation, the practical value of methods involving more than the first derivative of F is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

(Rheinboldt 1998): Clearly, for increasing dimension n the required computational work soon outweighs the advantage of the higher-order convergence. From this view point there is hardly any any justification to favor the Chebyshev method for large n.

When structure and sparsity is utilized the picture is somewhat different. Sparsity is more predominant in higher derivatives.

Test Case: Generalized Rosenbrock (Schwefel 1981) [7]

x_O=(1.3003,122,...,1.3003,122) and x*=(1.0,...,1.0)



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

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

$$g \in \mathbb{R}^n,\ H \in \mathbb{R}^{n\times n},\ \text{and}\ \mathcal{T} \in \mathbb{R}^{n\times n\times n}$$

H is a symmetric matrix

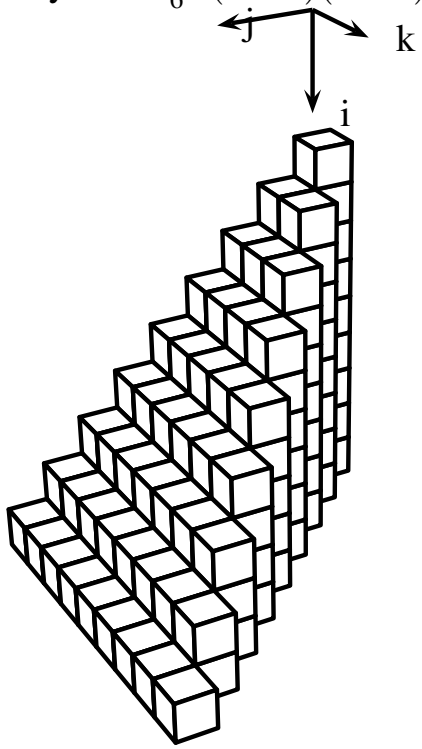
$$H_{ij} = H_{ji},\ \ i \neq j$$

$\mathcal{T}$ is a super-symmetric tensor,

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

Super-Symmetric Tensor

The tensor is super-symmetric and we only store $\frac{1}{6}n(n+1)(n+2)$ elements $\mathcal{T}_{ijk}$ for $1 \leq k \leq j \leq i \leq n$. This is illustrated as follows



The stored elements of a dense super-symmetric tensor, $n = 9$.

Induced Sparsity

(Griewank and Toint 1978) define sparsity of the Hessian matrix to be

$$\frac{\partial^2}{\partial x_i \partial x_j} f(x) = 0, \ \ \forall x \in \mathbb{R}^n, \ \ j \leq i.$$

Then

$$\mathcal{T}_{ijk} = \mathcal{T}_{ikj} = \mathcal{T}_{kij} = \mathcal{T}_{jik} = \mathcal{T}_{jki} = \mathcal{T}_{kji} = 0$$

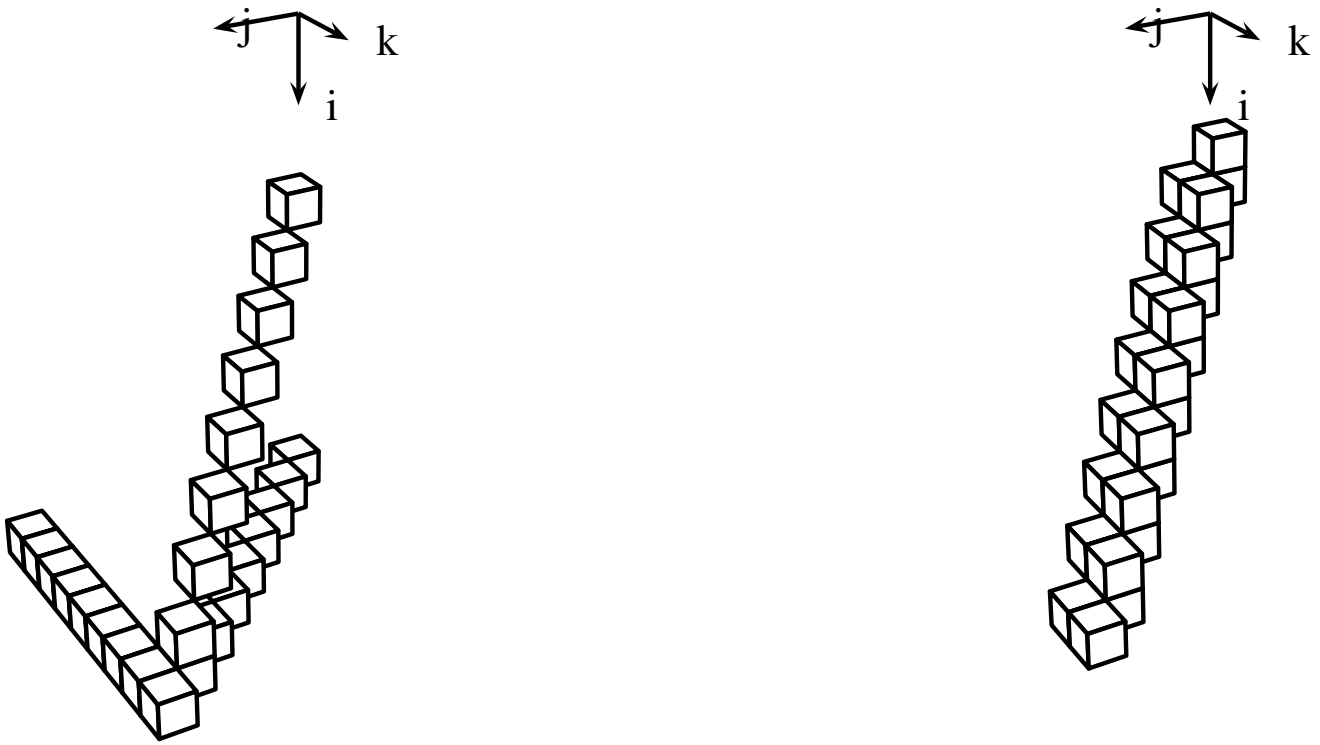
We say that sparsity structure of the tensor is induced by the sparsity structure of the Hessian matrix.

Structured Hessian Matrices

$$\begin{pmatrix} X & & & & & & & & \\ & X & & & & & & & \\ & & X & & & & & & \\ & & & X & & & & & \\ & & & & X & & & & \\ & & & & & X & & & \\ & & & & & & X & & \\ & & & & & & & X & \\ X & X & X & X & X & X & X & X & X \end{pmatrix} \qquad \begin{pmatrix} X & & & & & & & & \\ X & X & & & & & & & \\ & X & X & & & & & & \\ & & X & X & & & & & \\ & & & X & X & & & & \\ & & & & X & X & & & \\ & & & & & X & X & & \\ & & & & & & X & X & \\ & & & & & & & X & X \end{pmatrix}$$

Stored elements of a 9×9 arrowhead and tridiagonal matrix. X means a non-zero element while the rest is zero.

Sparsity Structure of the Tensors



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

Unconstrained Optimization using a Cubic Model

Algorithms for the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

generates a sequence of iterates x_k where at every iterate x_k we generate a model function $m(p)$ that approximates the function. The new iterate is $x_{k+1} = x_k + p$ provided that we have a sufficiently good model.

A third order Taylor approximation of $f(x+p)$ evaluated at x is

$$m(p) = f + g^T p + \frac{1}{2} p^T H p + \frac{1}{6} p^T (p\mathcal{T}) p$$

where $f = f(x)$ and we can expect the model $m(p)$ to be a good approximation for $\|p\|$ small.

Most methods will require the value, the gradient and the Hessian matrix

$$m(p), \quad \nabla m(p) = g + Hp + \frac{1}{2}(p\mathcal{T})p, \quad \nabla^2 m(p) = H + (p\mathcal{T})$$

of the cubic model.

Trust-Region Iterations for Quadratic and Cubic Model

The trust-region method is a global method.

The test functions are tested for different n and starting points. Chained Rosenbrock $x_0 = \{(-1.2, 1.0, \ldots, -1.2, 1.0), (-1.0, -1.0, \ldots, -1.0, -1.0)\}$ and $x^* = (1.0 \ldots, 1.0)$, Generalized Rosenbrock $x_0 = (-1.2, 1.0, \ldots, -1.2, 1.0)$ and $x^* = (1.0 \ldots, 1.0)$, BroydenTridiagonal $x_0 = (-1.0, -1.0, \ldots, -1.0, -1.0)$ and $x^* = (-0.57, \ldots, -0.42)$, and Beale $x_0 = \{(4,4), (3,3), (2,2)\}$ and $x^* = (3.5, 0.5)$

The trust-region method with a cubic method (C) uses fewer iterations then with a quadratic model (Q), to get to a solution.

Trust-Region Iterations for Quadratic and Cubic Model

Chained Rosenbrock			Chained Rosenbrock			Generalized Rosenbrock			BroydenTridiagonal			Beale		
n	Q	C	n	Q	C	n	Q	C	n	Q	C	n	Q	C
2	22	15	4	14	12	4	12	6	4	6	4	2	18	12
6	29	16	8	14	10	8	10	6	20	6	4	2	16	9
18	27	19	10	15	12	10	10	6	30	6	4	2	14	11

Computations of the Cubic Model

The cubic value term $p^T(p\mathcal{T})p \in \mathbb{R}$ is

$$p^T(p\mathcal{T})p = \sum_{i=1}^{n} p_i \sum_{j=1}^{n} p_j \sum_{k=1}^{n} p_k \mathcal{T}_{ijk}$$

The cubic gradient term $(p\mathcal{T})p \in \mathbb{R}^n$ is

$$((p\mathcal{T})p)_i = \sum_{j=1}^{n} \sum_{k=1}^{n} p_j p_k \mathcal{T}_{ijk}, \;\; 1 \leq i \leq n$$

The cubic Hessian term $(p\mathcal{T}) \in \mathbb{R}^{n \times n}$ is

$$(p\mathcal{T})_{ij} = \sum_{k=1}^{n} p_k \mathcal{T}_{ijk}, \;\; 1 \leq i, j \leq n$$

Computing the Cubic Model utilizing Super-Symmetry

The cubic value term $p^T(p\mathcal{T})p \in \mathbb{R}$ is

$$p^T(p\mathcal{T})p = \sum_{i=1}^{n} p_i \left\{ \left[\sum_{j=1}^{i-1} p_j \left(6 \sum_{k=1}^{j-1} p_k \mathcal{T}_{ijk} + 3 p_j \mathcal{T}_{ijj} \right) + 3 p_i \sum_{k=1}^{i-1} p_k \mathcal{T}_{iik} \right] + p_i^2 \mathcal{T}_{iii} \right\}.$$

Let $\mathcal{T} \in \mathbb{R}^{n \times n \times n}$ be a super-symmetric tensor.
Let $p \in \mathbb{R}^n$ be a vector.
Let $c, s, t \in \mathbb{R}$ be a scalar.
for $i = 1$ to n **do**
  $t = 0$
  for $j = 1$ to $i-1$ **do**
    $s = 0$
    for $k = 1$ to $j-1$ **do**
      $s += p_k \mathcal{T}_{ijk}$
    end for
    $t += p_j (6s + 3 p_j \mathcal{T}_{ijj})$
  end for
  $s = 0$
  for $k = 1$ to $i-1$ **do**
    $s += p_k \mathcal{T}_{iik}$
  end for
  $c += p_i (t + p_i (3s + p_i \mathcal{T}_{iii})$
end for

Utilizing super-symmetry $p^T(p\mathcal{T})p$ requires $\frac{1}{3}n^3 + 3n^2 + \frac{11}{3}n$ number of arithmetic operations.

How to Utilize Sparsity in the Problem: A Skyline Matrix

In a symmetric skyline storage mode, all matrix elements from the first nonzero in each row to the diagonal in the row are explicitly stored.

We define β_i to be the (lower) bandwidth of row i,

$$\beta_i = \max\{i - j | \text{ for nonzero } H_{ij} \text{ with } j < i.\}$$

Further we define f_i to be the start index for row i in the Hessian matrix as

$$f_i = i - \beta_i$$

The storage requirement for a symmetric skyline storage (only the lower triangle need to be stored), is $\sum_i \beta_i + n$.

Skyline implementation of: $p^T(p\mathcal{T})p$

Let $\mathcal{T} \in \mathbb{R}^{n\times n\times n}$ be a super-symmetric tensor.
Let $p \in \mathbb{R}^n$ be a vector.
Let $c,s,t \in \mathbb{R}$ be a scalar.
Let $\{f_0,\ldots,fn\}$ be the indices of the first nonzero elements for each row in the Hessian matrix.

for $i = 1$ to n **do**
    $t = 0$
    for $j = f_i$ to $i-1$ **do**
        $s = 0$
        for $k = \max\{f_i, f_j\}$ to $j-1$ **do**
            $s+ = p_k \mathcal{T}_{ijk}$
        end for
        $t+ = p_j(6s + 3p_j\mathcal{T}_{ijj})$
    end for
    $s = 0$
    for $k = f_i$ to $i-1$ **do**
        $s+ = p_k\mathcal{T}_{iik}$
    end for
    $c+ = p_i(t + p_i(3s + p_i\mathcal{T}_{iii}))$
end for

Utilizing sparsity $p^T(p\mathcal{T})p$ requires $2nnz(\mathcal{T}) + 5nnz(H) - n$ number of arithmetic operations.

The Efficiency Ratio Indicator

A measure of the complexity of working with the third derivative (tensor) compared to the second derivative (matrix) will be the ratio of the number of stored nonzero elements in the tensor and Hessian matrix.

Cubic and quadratic value term ratio is

$$\frac{flops(p^T(p\mathcal{T})p)}{flops(p^THp)} = \frac{2nnz(\mathcal{T})+5nnz(H)-n}{2nnz(H)+3n} = \frac{nnz(\mathcal{T})}{nnz(H)} + 2.5 + O(\frac{1}{n})$$

Cubic and quadratic gradient term ratio is

$$\frac{flops((p\mathcal{T})p)}{flops(Hp)} = \frac{4nnz(\mathcal{T})+8nnz(H)-6n}{4nnz(H)-n} = \frac{nnz(\mathcal{T})}{nnz(H)} + 2 + O(\frac{1}{n})$$

where

$$\lim_{n\to\infty} O(\frac{1}{n}) \to 0$$

The Efficiency Ratio Indicator

The ratio of number of nonzero elements of the tensor and the banded Hessian matrix is

$$nnz(\mathcal{T})/nnz(H) = \frac{1}{2}(\beta+1)(\beta+2)(n-\frac{2}{3}\beta)/(\beta+1)(n-\frac{\beta}{2}) = \frac{1}{2}(\beta+2)(n-\frac{2}{3}\beta)/(n-\frac{\beta}{2}) \leq \frac{\beta+2}{2}$$

Then the ratio of number of nonzero elements of the tensor and the band Hessian matrix is

$$\frac{\beta+2}{2}$$

where β and the number of nonzero elements of the tensor is of the same order as for the band Hessian matrix.

The Efficiency Ratio Indicator for the Halley Class

The ratio indicators shows the cost of using third order method's compared to a Newton's method for one iteration.

$$\frac{flops(Chebyshev)}{flops(Newton)} = \frac{3n\beta^2 - \frac{4}{3}\beta^3 + 20n\beta - 8\beta^2 - \frac{20}{3}\beta + 2n}{n(\beta^2 + 8\beta + 2)} = \frac{3\beta^2 + 20\beta + 2}{\beta^2 + 8\beta + 2} + O(\frac{\beta}{n})$$

$$\frac{flops(Halley)}{flops(Newton)} = \frac{5n\beta^2 - 2\beta^3 + 24n\beta - 4\beta^2 - 2\beta + 12n}{n(\beta^2 + 8\beta + 2)} = \frac{5\beta^2 + 24\beta + 12}{\beta^2 + 8\beta + 2} + O(\frac{\beta}{n})$$

$$\frac{flops(SuperHalley)}{flops(Newton)} = \frac{5n\beta^2 - 2\beta^3 + 23n\beta - 4\beta^2 - 2\beta + 12n}{n(\beta^2 + 8\beta + 2)} = \frac{5\beta^2 + 23\beta + 12}{\beta^2 + 8\beta + 2} + O(\frac{\beta}{n})$$

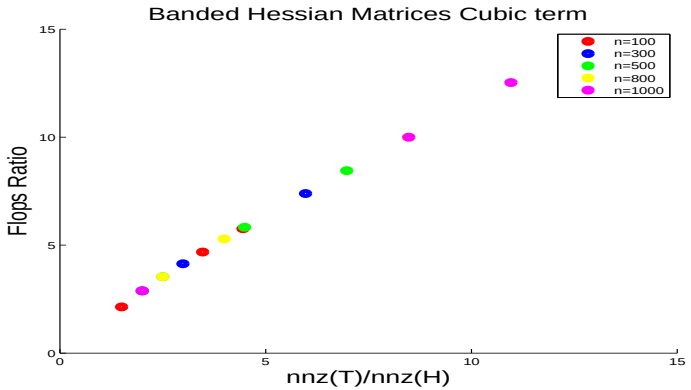
where

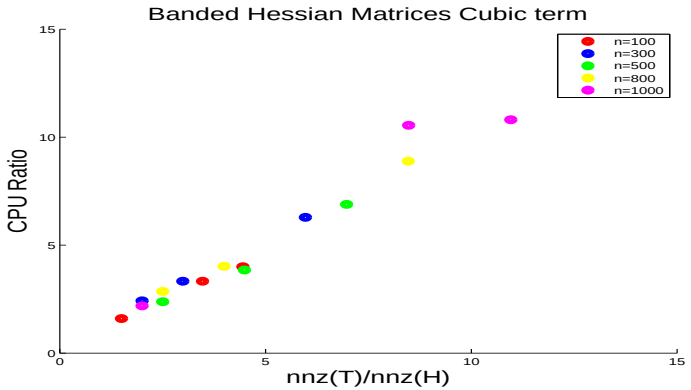
$$\lim_{n \to \infty} O(\frac{\beta}{n}) \to 0$$

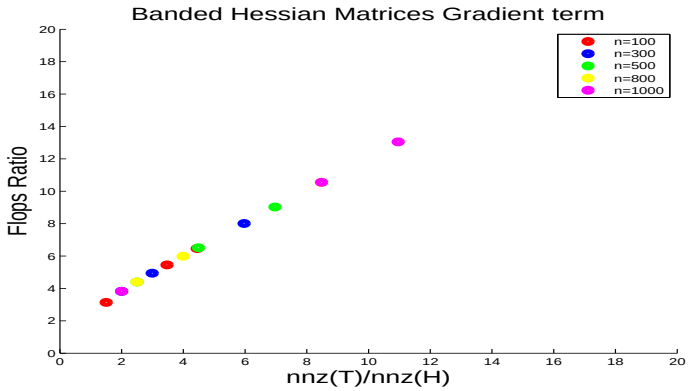
The Efficiency Ratio Indicator

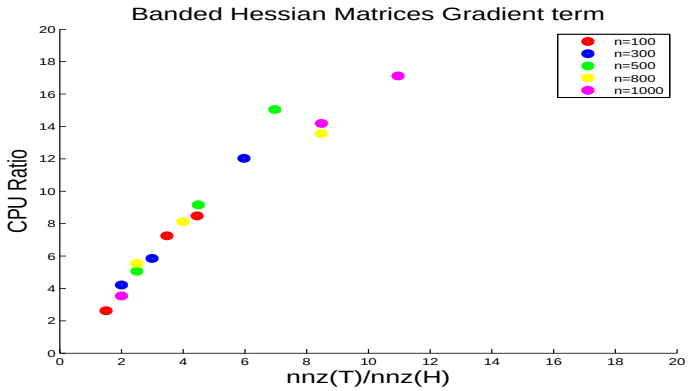
- The ratio $\frac{nnz(\mathcal{T})}{nnz(H)}$ is an indicator for all types of Hessian structure.
- The growth of the ratio $\frac{nnz(\mathcal{T})}{nnz(H)}$ for dense Hessian structure is large for increasing n.
- The ratio $\frac{nnz(\mathcal{T})}{nnz(H)}$ for sparse Matrix Market matrices is usually small.
- The Efficiency Ratio Indicator for the Halley Class for banded Hessian matrices is independent of n.

Banded for the operations $p^T H p$ versus $p^T (p\mathcal{T}) p$ and Hp versus $(p\mathcal{T})p$









Summary

- Third order local methods are competitive with Newton's method.
- The Trust Region algorithm implemented with a cubic model has fewer iterations than with a quadratic model (for our test cases).
- For large and increasing n the difference in efficiency is linear between Newton and third order methods for banded problems.
- Contradicts Rheinboldt's statements about third-order methods.
- The ratio $\frac{nnz(\mathcal{T})}{nnz(H)}$ is a good indicator of the cost from going from a quadratic to a cubic model.

References

[1] A. Griewank and Ph. L. Toint. *On the unconstrained optimization of partially separable functions*. In Michael J. D. Powell, editor, Nonlinear Optimization 1981, pages 301-312. Academic Press, New York, NY, 1982.

[2] J. M. Gutierrez and M. A. Hernandez. *An acceleration of Newton's method: Super-Halley method*. Applied Mathematics and Computation. 25 January 2001, vol. 117, no. 2, pp. 223-239(17).

[3] J. Nocedal and S. J. Wright. *Numerical Optimization. Springer Series in Operations Research*. Springer-Verlag, 1999.

[4] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. New York, Academic Press, 1970.

[5] W. C. Rheinboldt. *Methods for Solving Systems of Equations of Nonlinear Equations*. Reg. Conf. Ser. in Appl. Math, Vol. 14. SIAM Publications, Philadelphia, PA, 1974.

[6] W. C. Rheinboldt. *Methods for Solving Systems of Equations of Nonlinear Equations*. Second edition. Regional Conf. Series in Appl. Math., Vol. 70. SIAM Publications, Philadelphia, PA, 1998.

[7] H. P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley and Sons, Chichester, 1981.

[8] Ph.L. Toint. *Some numerical results using a sparse matrix updating formula in unconstrained optimization*. Mathematics of Computation, Volume 32, Number 143. July 1978, pages 839-851.