Efficient Evaluation of Sparse Jacobians by Matrix Compression Part II: Implementation and Experiments

NIKOLAI STROGIES
Humboldt-Universitt zu Berlin
Department of Applied Mathematics
Berlin, Germany
strogies@math.hu-berlin.de

ANDREAS GRIEWANK
Humboldt-Universitt zu Berlin
Department of Applied Mathematics
Berlin, Germany
griewank@math.hu-berlin.de

Abstract: The accurate and efficient calculations of Jacobians matrices at a sequence of arguments is a key ingredient of numerical methods for nonlinear problems in scientific computing. It has been known since the seminal work of Curtis Powell and Reid [1] that once their sparsity pattern is known Jacobians can be estimated on the basis of divided differences for a set carefully chosen directions. The number p of such seed directions and thus extra function evaluations can often be limited a priori to a smallish number, which is typically much smaller than the number of independent variables and unaffected by grid sizes and other discretization parameters. The cost factor p is bounded below by the maximal number of nonzeros per row, which is actually sufficient for Jacobian estimation using Newsam-Ramsdell compression. This NR approach is numerically less stable than the CPR method, which was therefore preferred in practice as divided differences are strongly affected by truncation and round off errors. However now, using automatic or algorithmic differentiation, one obtains directional derivatives with working accuracy and can thus utilize the more economical NR approach.

Key–Words: Matrix Compression, Algorithmic Differentiation, Graph Coloring, Traveling Salesman

1 Introduction

In [9] we introduced a matrix compression method based on seed matrices of generalized Vandermonde form

$$S = \left[P_k(\lambda_j)\right]_{k=1\dots\rho}^{j=1\dots n} \in \mathbb{R}^{n\times\rho}. \qquad (1)$$

where ρ denotes the maximal number of nonzeros per row of the Matrix to be compressed. Let $\chi = n \geq \rho$ represent the chromatic number of the column incidence graph. We define the *abscissa values* $\{\lambda_i\}_{i=1}^{\chi} \in [-1,1]$ and the *Lagrange polynomials*

$$P_k(\lambda) = \prod_{k\neq q=1\dots\rho} \frac{\lambda - \lambda_{d(q)}}{\lambda_{d(k)} - \lambda_{d(q)}}. \qquad (2)$$

Here the mapping $d : [1..\rho] \mapsto [1..\chi]$ selects a certain subset of *Cartesian colors.* By this technique we can reconstruct the nonzero entries of a sparse Jacobian $F' \in \mathbb{R}^{m\times n}$. Utilizing finite differences or, which is more preferable, algorithmic differentiation (e.g. [9, Section 3]), we can evaluate the Matrix

$$B = F'S \qquad (3)$$

at a cost proportional to ρ times the complexity of F itself. Then one can reconstruct the entries of the original Jacobian according to

$$\bar{a}_i = \left(e_i^\top F'(x)e_j\right)_{j\in\bar{\mathcal{J}}_i} = b_i\bar{S}_i^{-1} \qquad (4)$$

with

$$\bar{S}_i = \left[P_k\left(\lambda_{c(j)}\right)\right]_{k=1\dots\rho}^{j\in\bar{\mathcal{J}}_i} \in \mathbb{R}^{\rho\times\rho}.$$

Here $\mathcal{J}_i$, $i = 1..m$, denotes the index set of nonzero entries in row i with cardinality $|\mathcal{J}_i| = \rho_i$. $\bar{\mathcal{J}}_i$, $i = 1..m$, denotes the augmented index sets where each $\mathcal{J}_i$ is enlarged by $\rho - \rho_i$ indices from $\{1, ..., n\}$ such that $|\bar{\mathcal{J}}_i| = \rho$ holds for all i. Finally, $c : [1, ..., n] \mapsto [1, ..., \chi]$ represents the mapping assigning the columns to the corresponding color. This special choice of the seeding matrix allows for a direct representation of the inverse matrices

$$\bar{S}_i^{-1} = \left[\prod_{j\neq q=1\dots\rho} \frac{\lambda_{d(k)} - \lambda_{c(q)}}{\lambda_{c(j)} - \lambda_{c(q)}}\right]_{j\in\bar{\mathcal{J}}_i}^{k=1\dots\rho}.$$

Moreover, partitioning every $\bar{\mathcal{J}}_i$ into $\hat{\mathcal{J}}_i$ and $\check{\mathcal{J}}_i$ with $\hat{\mathcal{J}}_i$ containing the noncartesian indices of $\bar{\mathcal{J}}_i$ we obtain the following. For recovering the ith row of the Jacobian we may reorder the columns of the compressed matrix such that the first $|\hat{\mathcal{J}}_i|$ entries of $\bar{\mathcal{J}}_i$ are the noncartesian ones. Then $\bar{S}_i$ and $\bar{S}_i^{-1}$ take the block triangular form

$$
\begin{aligned}
\bar{S}_i &= \begin{bmatrix} \hat{S}_i & \check{S}_i \\ 0 & I \end{bmatrix} \\
\bar{S}_i^{-1} &= \begin{bmatrix} \hat{S}_i^{-1} & -\hat{S}_i^{-1}\check{S}_i \\ 0 & I \end{bmatrix}.
\end{aligned} \tag{5}
$$

with $\hat{S}_i \in \mathbb{R}^{|\hat{\mathcal{J}}_i| \times |\hat{\mathcal{J}}_i|}$.

This paper is organized as follows. In Section 2 we illustrate the method introduced in the first part [9] by an example. Section 3 provides the algorithm used by the authors for determining the ordering of the colors and selecting the Cartesian ones. This procedure is based on several heuristics, which are discussed in detail. In Section 4 we provide examples from [3] and report on the performance of the algorithm of Section 3 in several stages. The paper closes with remarks on extensions to two-sided compression and Hessian matrices in section 5.

2 An Illustrative Example

Consider a small example where $\chi = n$, a situation, which is always arrived at after the rows of an original $m \times n$ matrix have been compressed by the CPR scheme (cf. equation (6) and the explanation in [9]). Then we may have the 7×6 matrix displayed in Figure 1. Its column incidence graph is complete because no two columns are structurally orthogonal as one can check easily. However, the maximal number of nonzeros per row is clearly $\rho = 4$, and the Newsam-Ramsdell approach allows the reduction of the evaluation effort from six to four directional derivatives. Since each color is associated with

$$
\begin{bmatrix}
0 & 0 & 0 & \times & \times & \times \\
0 & \times & \times & \times & 0 & 0 \\
\times & 0 & 0 & \times & 0 & 0 \\
0 & \times & \times & 0 & \times & \times \\
\times & 0 & \times & 0 & 0 & 0 \\
\times & \times & 0 & 0 & \times & 0 \\
\times & 0 & 0 & 0 & 0 & \times
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\times & \times & \times & \times \\
0 & 1 & 0 & 0 \\
\times & \times & \times & \times \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
=
\begin{bmatrix}
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\hline
\times & \times & 0 & 0 \\
\hline
\times & \times & \times & \times \\
\hline
\times & 0 & 0 & \times \\
\hline
\end{bmatrix}
$$

Figure 1: Sparse System with $m = 7$, $n = 6 = \chi$, $\rho = 4$; Boxed Entries are Identifiable

a single column we may number them in the given order from $1 \ldots 6$. Therefore the sparsity is described by the index sets $J_1 = \{4, 5, 6\}$, $J_2 = \{2, 3, 4\}$, $J_3 = \{1, 4\}$, $J_4 = \{2, 3, 5, 6\}$, $J_5 = \{1, 3\}$, $J_6 = \{1, 2, 5\}$, $J_7 = \{1, 6\}$. Suppose we pick the first, third, fifth and sixth color as Cartesian , so that only the second and forth are noncartesian . Then the seed matrix takes the form displayed in the center of Figure 1 irrespective of the actual abscissa values chosen. The resulting compressed Jacobian on the right has only two special rows, namely the fifth and the seventh. They are CPR style compressions of the corresponding rows in the Jacobian whose two nonzeros are simply transferred to the boxed entries without any modifications. In contrast, all other rows are linearly combined to dense 4-vectors.

They can be reconstructed by solving linear systems of the form $b_i = \bar{a}_i \bar{S}_i$ with S_i displayed in Figure 2. For the sixth row the system has the same structure as for $i = 1$ where there are also there nonzero entries of which only one noncartesian column is involved. The augmentations below the dashed lines are somewhat arbitrary as one could pick any one of the remaining rows in S. To reduce the number of operations it makes sense to augment only by Cartesian colors. For conditioning it is important that the four resulting colors are more or less nicely spread amongst the four Cartesian ones, namely 1, 3, 5, and 6, which we have selected. For the first row that is a little bit of a problem as the naturally occurring colors 4, 5, 6 are all on the right end of the range. The matrices $\bar{S}_i$ represent polynomial extrapolation from the values at the Cartesian abscissas to those at the λ_i with $j \in \bar{J}_i$.

$$
\begin{array}{ccccc}
i=1 & 3: & 5: & 6: & 1: \\
4: & \boxed{\times} & \times & \times & \times \\
5: & 0 & 1 & 0 & 0 \\
6: & 0 & 0 & 1 & 0 \\
\hdashline
1: & 0 & 0 & 0 & 1
\end{array}
\qquad
\begin{array}{ccccc}
i=2 & 1: & 5: & 3: & 6: \\
2: & \boxed{\times} & \boxed{\times} & \times & \times \\
4: & \boxed{\times} & \boxed{\times} & \times & \times \\
3: & 0 & 0 & 1 & 0 \\
\hdashline
6: & 0 & 0 & 0 & 1
\end{array}
$$

$$
\begin{array}{ccccc}
i=3 & 3: & 1: & 5: & 6: \\
4: & \boxed{\times} & \times & \times & \times \\
1: & 0 & 1 & 0 & 0 \\
\hdashline
5: & 0 & 0 & 1 & 0 \\
6: & 0 & 0 & 0 & 1
\end{array}
\qquad
\begin{array}{ccccc}
i=4 & 1: & 3: & 5: & 6: \\
2: & \boxed{\times} & \times & \times & \times \\
3: & 0 & 1 & 0 & 0 \\
5: & 0 & 0 & 1 & 0 \\
6: & 0 & 0 & 0 & 1
\end{array}
$$

Figure 2: Permuted Submatrices $\bar{S}_i$ for $i = 1, 2, 3, 4$

Conversely, the inverses $\bar{S}_i^{-1}$ represent the extrapolation process in the opposite direction. Therefore, we have picked the free color $i = 1$ to augment the set $J_1 = \{4, 5, 6\}$. In general we will strive to include the endpoints λ_1 and λ_χ into the augmented sets $\bar{J}_i$ whenever that is possible at all. The boxed entries in Figure 2 represent the individual entries and the 2×2 matrix that need to be inverted in the sense of linear equation solving.

Their conditioning is essential for the conditioning of the whole scheme. Then we choose the six abscissas as the Chebyshev points

$$-\cos((k-0.5)\pi/6) \text{ for } k = 1\ldots6.$$

As one can see they can attain values greater than one and that is even more so true for the sum of their absolute values. These sums are called Legendre functions, which are well understood when their support grid is uniform or otherwise regular. In contrast we choose them as ρ elements amongst χ Chebyshev points, because other choices were less successful in our experience. On our little example the two noncartesian rows in the seed matrix of our small example take the values

$$\begin{array}{ll} 2: & (0.464,\ 0.732,\ -0.464,\ 0.268) \\ 4: & (-0.072,\ 0.464,\ 1.071,\ -0.464) \end{array} \tag{6}$$

The only comparatively small entry -0.072 would become a pivot if we had augmented $J_1 = \{4, 5, 6\}$ with 3 rather than 1. With our choice division is required by pivots with the same absolute value 0.464 for rows 1, 3, and 4. For row 2 one has to solve a little 2×2 system whose determinant is also given by $(0.464 \cdot 1.079 - 0.072 \cdot 0.464) = 0.464$ with its largest entry of size 1.071.

Also, the remaining off-diagonal entries are all small, so that the linear equations solving provides no numerical difficulties whatsoever. Notice that the parts beyond the dashed line may be left out of the computation. The total number of numerical operations is $17 = 3 \cdot 3 + 8$, which must be compared with the effort for solving one 4×4, three 3×3, and two 2×2 Vandermonde system in the classical Newsam-Ramsdell variant. Obviously, this effort would be much higher, including in particular quite a few divisions. Naturally, the condition numbers and operations counts are extremely small on this toy example.

They are bound to be significantly larger on real problems. However, we should keep in mind that the dense linear sub-systems that need to be solved are maximally of dimension $\chi - \rho$. If this discrepancy is significant the little extra solving effort will quite likely pay off by avoiding the evaluation of $\chi - \rho$ directional derivatives.

3 Algorithmic Approach

In the following discussion we will again assume without loss of generality that the Jacobian has been precompressed by CPR grouping so that now $n = \chi$. Moreover, the sparsity pattern of the precompressed Jacobian is given by the sets $\{\mathcal{J}_i\}_{i=1}^m$. There is a rather well-developed theory on the conditioning of polynomial interpolation schemes starting with ground breaking papers by Erdös and Gautschi (see e.g. [4], [5] and [6]). However, none of these results seem directly applicable to our situation because they concern a single linear system for which the abscissas may be chosen appropriately. Here we first have to choose abscissas $(\lambda_j)_{j=1\ldots\chi}$, and then select ρ of them as *Cartesian* denoted by $(\hat{\lambda}_j)_{j=1..\rho}$ such that these interpolate well with the m subsets $(\lambda_j)_{j\in\bar{\mathcal{J}}_i}$. By *interpolate well* we mean that the values of a polynomial of order ρ can be transformed back and forth stably between the sets of abscissas $(\hat{\lambda}_j)_{j=1..\rho}$ and the m subsets $(\lambda_j)_{j\in\bar{\mathcal{J}}_i}$.

Covering the Extremes:
Geometric intuition strongly suggests that this will only be possible if the two abscissa sets are nicely intertwined rather than occupying disparate parts of the real line. For the color ordering of the columns

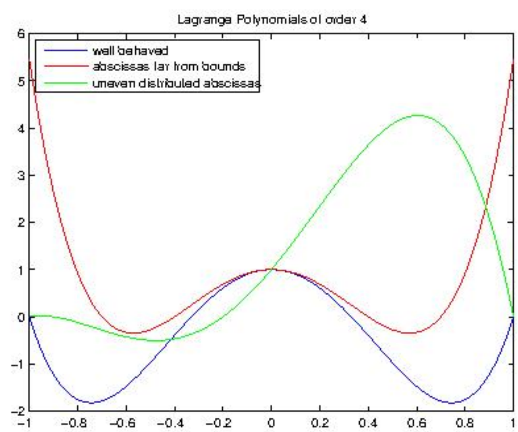

Figure 3: Lagrange polynomials for various roots

of the matrix to compress, the following step is crucial. Figure 3 indicates, that ideally a common pair of indices should be contained in all the sets $\bar{\mathcal{J}}_i$. Since we are forced to augment the index sets $\mathcal{J}_i$ by at most $\rho - \rho_i \geq 2$ indices, the above task can be per domed as follows.

Given m subsets $\mathcal{J}_i \subset \{1, ..., \chi\}$ with $|\mathcal{J}_i| = \rho_i \leq \rho$ we are looking for a *covering pair* $\{j, k\}$ of indices such that $\{j, k\} \subset \mathcal{J}_i$ for all i referring to so called *maximal rows* ($\rho_i = \rho$) and $\{j, k\} \cap \mathcal{J}_i \neq \emptyset$ for all i referring to so called *long rows* ($\rho_i = \rho - 1$). After identifying these elements, all index sets $\mathcal{J}_i$ are joined with $\{j, k\}$ (and probably more Cartesian abscissas) to obtain the sets $\bar{\mathcal{J}}_i$, each containing $\{j, k\}$. These indices are included in the set of Cartesian colors and further identified with the left-most and right-most abscissa value. Thus we stabilize the behavior of the Lagrangian polynomials. However, the existence of a covering pair is not guaranteed at all which leads to the consideration of *covering sets* which are disjoint k tuples $\mathcal{K}_1$ and $\mathcal{K}_2$ satisfying $|\mathcal{K}_1| \leq |\mathcal{K}_2| = k$ and

$$\mathcal{K}_1 \cap \mathcal{J}_i \neq \emptyset \text{ and } \mathcal{K}_2 \cap \mathcal{J}_i \neq \emptyset$$

if i refers to a maximal row and

$$\mathcal{K}_1 \cap \mathcal{J}_i \neq \emptyset \text{ or } \mathcal{K}_2 \cap \mathcal{J}_i \neq \emptyset$$

if i refers to a long row. Determining the covering sets is equivalent to the Hitting Set Problem which is known to be NP complete [10]. Consequently, we use a greedy strategy for the determination of this set which works as follows.

First, set $\mathcal{K}_1 = \mathcal{K}_2 = \emptyset$. Then both are filled alternately by the index which occurs most frequently in all maximal and long rows. Afterward, this index will not longer be considered and we look for the next most frequent index. As soon as two indices from a maximal row or one from a long row was selected, the corresponding row is not longer considered. The k left-most and right-most colors are related to the elements in the covering sets according to the frequency with with the corresponding indices occur among the maximal and long rows. Thus we cover as many rows, maximal or long, as possible with values as close as possible to -1 and 1. At this point we have to realize, that there are examples of sparse matrices where the cardinality of the covering set exceeds ρ which is the number of available Cartesian colors. In section 5, examples of matrices with this property are given.

Reordering the interior columns:
Next we discuss the reordering procedure for the uncovered columns. Let $\mathcal{C} = \{d(k)\}_{k=1\ldots\rho}$ denote the so far not specified set of Cartesian indices. Then we have for each i the partition

$$\bar{\mathcal{J}}_i = \hat{\mathcal{J}}_i \cup \check{\mathcal{J}}_i \text{ with } \check{\mathcal{J}}_i = \bar{\mathcal{J}}_i \cap \mathcal{C}.$$

The linear transformations described by the matrix $\bar{S}_i$ and its inverse $\bar{S}_i^{-1}$ represent polynomial interpolation from the nodes $\mathcal{C}$ to $\bar{\mathcal{J}}_i$ and back. Large entries in some $\bar{S}_i$ and thus S itself are undesirable because, as we have seen in [9, Section 3], they cause comparatively large errors in the evaluation or approximation of the corresponding directional derivatives by AD or differencing, respectively. Large entries in $\bar{S}_i^{-1}$ open the chance that these errors are enlarged during the reconstruction of the Jacobian rows a_i from the compressed rows b_i. The more specific analysis in [9, Section 3] showed that what matters for either differentiation method in the end are the l_1 norms of the inverses of the pre-scaled versions of the matrices $\bar{S}_i$.
Their columns $s_k^{(i)}$ must be rescaled either by the l_∞ norms of the $s_k^{(i)}$ themselves or the norm of the underlying full column s_k. Again we see, that the entries of the matrices $\bar{S}_i$ have to be small which depends on the distribution of the Cartesian abscissa values. Ideally we would like to minimize the following dissimilarity measure. For each index set $\mathcal{J}_i$ we may find an injective mapping $\varphi : \mathcal{J}_i \to \mathcal{C}$ that minimizes the distance

$$\sigma(\mathcal{C}, \mathcal{J}_i) \equiv \max_{j \in \mathcal{J}_i} \big|\varphi_i(j) - j\big| \tag{7}$$

over all such assignments. Striving to make all linear subsystems well conditioned we could try to minimize

$$\sigma\big(\mathcal{C}, (\mathcal{J}_i)_{i=1\ldots m}\big) \equiv \max_{1 \leq i \leq m} \sigma(\mathcal{C}, \mathcal{J}_i) \tag{8}$$

by a suitable ordering of the colors and selection of the Cartesian subset $\mathcal{C}$. Of course, a direct attack on this combinatorial problem seems rather hopeless. Therefore, we will make the assumption that the ρ Cartesian colors should in any case be spread with an approximately even distance of χ/ρ amongst all χ colors. Consequently, reasonable assignments $\varphi_i : \mathcal{J}_i \to \mathcal{C}$ can be found provided the index sets $\mathcal{J}_i$ are nowhere densely clustered. Ideally, the distances between its element should also be about χ/ρ or possibly larger. However, we have already fixed the order of $|\mathcal{K}_1| + |\mathcal{K}_2|$ columns of the Jacobian according to the previous step and have to fix the remaining ones.

The TSP heuristic:
We aim to avoid having adjacent colors in each of the index sets $\mathcal{J}_i$ as described above. This problem can be regarded as an instance of the symmetric Traveling Salesman Problem (TSP) with fixed start- and endpoint. Starting point is the right-most index of $\mathcal{K}_1$ and endpoint is the left-most index in $\mathcal{K}_2$ while the objects to order are the remaining $\chi - |\mathcal{K}_1| - |\mathcal{K}_2|$ columns of the Jacobian.

The distance between two columns is the number of rows, where both columns have a nonzero. One easily checks, that this metric does not fulfill the triangle inequality. The TSP is NP hard but can nevertheless be solved quite reliably for tens of thousands of nodes. For small numbers of free columns we have implemented a brute force (BF) and a greedy strategy which is also used for moderate and large numbers of free columns since the effort of BF grows exponentially in this number.

Complete and partial Chebyshev seeds:
Since our S always contains exactly ρ distinct Cartesian rows it has in each column at least one 1. To keep the columns of S at a similar size we have to watch its largest elements. It was shown in [11] that the l_∞ norm of S is proportional to $2\log(\rho)/\pi$ when

the Cartesian abscissas $(\lambda_k)_{k\in\mathcal{C}}$ are the roots of the Chebyshev polynomial of degree ρ, i.e.

$$\lambda_{d(k)} = -\cos((k-0.5)\pi/\rho) \tag{9}$$

for $k = 1\ldots\rho$. Moreover, the asymptotic bound $2\log(\rho)/\pi$ cannot be undercut by any other system of abscissas. Especially in view of difference quotient calculations one may wish to keep the seed matrix S nicely balanced. Also, large entries in the seed matrix might be disconcerting to the user. Hence we may chose the seed matrix S by first selecting the Cartesian abscissa as Chebyshev points and then placing the remaining $\chi-\rho$ ones more or less evenly amongst them. This strategy, which we will refer to as *complete Chebyshev seed*, may be especially useful in the context of differencing where it should allow the use of the same δ in all test directions. However, in our preliminary tests we had even more success with the following strategy, which we will refer to as *partial Chebyshev seed*. Define the full set of abscissas $(\lambda_k)_{k=1\ldots\chi}$ as the Chebyshev points of degree χ and then select the Cartesian ones as roughly every (χ/ρ)-th of them.
More specifically, we set

$$d(k) = \left\lceil \left(k - \tfrac{1}{2}\right)\chi/\rho \right\rceil \tag{10}$$

for $k = 1, 2, \ldots, \rho$.
For the example given in section 2 the latter distribution of abscissas would provide the symmetric distribution $d(1) = 1$, $d(2) = 3$, $d(3) = 4$, and $d(4) = 0$ rather than the slightly more interesting choice $d(1) = 1$, $d(2) = 3$, $d(3) = 5$, and $d(4) = 6$ we used for this example.
Summing up, we arrive at the following procedure given the sparsity pattern $\{\mathcal{J}_i\}_{i=1\ldots m}$.

1. Precompression:
   Color the column incidence graph by some heuristic using χ colors yielding the grouping matrix G. Then determine the weighted reduced column incidence graph of the precompressed Jacobian $F'(x)G$.

2. Covering Sets:
   Determine maximal and long rows, compute the covering set according to the discussion above, join the long rows with a suitable element from the covering set.

3. Reordering:
   Order the colors by approximately solving the traveling salesman problem on the weighted reduced column incidence graph.

4. Fixing Abscissa Values:
   Select the Cartesian colors $d(j)$ according to the rule (10) and compute the compression matrix C as defined in (1). Finally obtain the seed matrix $S = GC$.

5. Augmentation:
   Augment all remaining index sets $\mathcal{J}_i$ with $|\mathcal{J}_i| < \rho$ by Cartesian indices.

The reconstruction of the Jacobian can then be performed according to (4).

4 Experimental results

The coloring number was calculated by ColPack [7] version 1.04 which is distributed under the GNU Lesser General Public License. We have chosen this tool because it directly provides the precompressed Jacobian. In order to justify the proposed approach, we first present several matrices we encountered in the Florida Sparse Matrix Collection [3], where the gap between coloring number and maximal number of nonzeros per row is significant.

Matrix	Row	Column	Nnz
cis-n4c6-b13	6300	25605	88200
cari	400	1200	152800
nug08-3rd	19728	29856	148416
bayer02	13935	13935	63679
gre_1107	1107	1107	5664
ODLIS	2909	2909	18246
cis-n4c6-b14	920	6300	13800
scsd8-2b	5130	35910	112770
scsd8-2r	8650	60550	190210
west0381	381	381	2157
bayer03	6747	6747	56196
bayer04	20545	20545	159082
onetone2	36057	36057	227628
bayer09	3083	3083	21216
west0067	67	67	294
Kemelmacher	28452	9693	100875
graphics	29493	11822	117954

Table 1: Sparsity parameters of example matrices

In Table 1 we show the maximal number of nonzeros per column, and in Table 2 the chromatic number of the column incidence graph, the corresponding difference and ratio.

Matrix	χ	ρ	$\chi-\rho$	χ/ρ
cis-n4c6-b13	33	14	19	2.36
cari	400	382	18	1.05
nug08-3rd	19	8	11	2.38
bayer02	32	22	10	1.45
gre_1107	16	7	9	2.29
ODLIS	49	41	8	1.20
cis-n4c6-b14	22	15	7	1.47
scsd8-2b	30	24	6	1.25
scsd8-2r	30	24	6	1.25
west0381	31	25	6	1.24
bayer03	39	34	5	1.15
bayer04	39	34	5	1.15
onetone2	38	33	5	1.15
bayer09	38	34	4	1.12
west0067	10	6	4	1.67
Kemelmacher	10	7	3	1.43
graphics	6	4	2	1.50

Table 2: Chromatic number and max row lengths

For these matrices, the application of the presented seeding technique is promising since we need significantly less rows in the seeding matrix for most of them. However, most of the matrices we used from this collection satisfy $\chi-\rho=1$ and we therefore did not include them in our experiments.
Note that according to the first part of Section 5 even the consideration of matrices with this small gap pays off. For completeness we mention, that there are in addition several examples (e.g. *baxter, ulevimin* and *pre2*), where the coloring of the column incidence graph yields optimal results, meaning that $\chi=\rho$ equals the lower bound maximal nonzeros per row. For these matrices, the presented approach does not yield any improvement.

Next we report on the performance of the greedy algorithm for the determination of the covering sets. The corresponding results are depicted in Table 3 where also the number of maximal and long rows (m_m and m_l) is listed. Here we especially highlight *cis-n4c6-b13* with $|\mathcal{K}_1|+|\mathcal{K}_2|=\rho$ and *nug08-3rd* and *graphics* both satisfying $|\mathcal{K}_1|+|\mathcal{K}_2|>\rho$ as examples for matrices where all special abscissas are located at the ends of the interval $[-1,1]$.
Table 4 provides information about the improvement we could achieve in our tests with the reordering process. We listed most of the matrices from Table 3

Matrix	Row	m_m	m_l	Size
nug08-3rd	19728	10320	9408	8, 8
cis-n4c6-b13	6300	6300	0	7, 7
gre_1107	1107	344	40	5, 5
cis-n4c6-b14	920	920	0	3, 2
graphics	29493	29475	18	3, 3
Kemelmacher	28452	128	147	2, 2
west0655	655	2	59	2, 1
onetone2	36057	240	400	1, 1
sc205-2r	35213	6402	6402	1, 1

Table 3: Size of covering sets

again, where the gap was significant.

Matrix	Average distance		Travel cost	
	before	after	before	after
cis-n4c6-b13	116.37	1028.06	21102	16746
cari	255590.00	262144.00	145200	144800
nug08-3rd	15.43	57.25	120	30
bayer02	147.59	1172.92	26076	12203
gre_1107	7.86	41.76	207	48
ODLIS	28297.60	66224.70	4364	3147
cis-n4c6-b14	6.41	27.87	8583	7721
scsd8-2b	10.70	182.39	97895	74574
scsd8-2r	10.70	182.48	165127	125790
west0381	80.43	606.99	1041	463
bayer03	199.16	6369.84	34871	16409
bayer04	190.56	7077.71	96668	47473
onetone2	20.48	4769.70	126848	40701
Kemelmacher	2.08	9.42	12109	1188
graphics	1.32	3.04	89950	85185

Table 4: Reordering of Uncovered Columns

In the column Average distance we provide the mean value of the products

$$G_i=\prod_{k=1}^{\rho_i-1}(g_i(k+1)-g_i(k))$$

where g_i maps $\{1,...,\rho_i\}$ to the actual colors $\{1,...,\chi\}$ such that $\{g_i(k)\}_{k=1}^{\rho_i}=\mathcal{J}_i$ holds for all $i=1,...,m$. This quantity is given before we reorder the Jacobian with the pair of covering sets and the TSP and after. The values in this column are thus $(\sum_{i=1}^m G_i)/m$ at the two different stages for the construction of S. The Travel cost, we observe at the same stages of the process, are the sum of all column distances our Traveling Salesman model travels in the current column order, i.e. we see the total number of neighboring nonzeros. Without having to pick specific examples, we see that our approach of reordering by covering sets and trying to avoid only directly neighboring nonzeros shows a considerable improvement in the distribution of the nonzeros over the whole sparsity pattern. In this part a run time analysis for the construction of S, the preparation of S_i^{-1} in a suitable form and the solves per Jacobian evaluation remain to be done. Also the numerical conditioning of S_i^{-1} should be computed explicitly rather than deduced from the traveling distance, which is a measure of the size of the interpolation denominators. It is also not yet quite clear what is the best general purpose scheme for effecting the multiplications $\bar{S}_i^{-1}$. One may of course compute and store the inverses explicitly, or apply fast interpolation algorithms from scratch every time, i.e. at each new Jacobian evaluation and reconstruction.

5 Summary and Outlook

In this two part paper we have considered the task of reconstructing Jacobians from a minimal number of directional derivatives, be they estimated by divided differences or evaluated in the forward mode of algorithmic differentiation. The total resulting error is structurally similar but considerably larger due to truncation and round-off in case of divided differences. While CPR coloring typically reduces the column number χ of the precompressed Jacobian to a fraction of the number n of independent variables one needs Newsam-Ramsdell type seeding to reach the absolutely minimal Jacobian width ρ. Based on interpolation at Chebyshev points we have developed a seed matrix S that has a minimal number of nonzero entries according to the total linear independence condition. They are all of moderates size to limit the evaluation and differencing error. Moreover, by suitable augmenting the sparsity pattern without increasing ρ and reordering the columns at least conceptually we can ensure that the relevant square submatrices of S have also comparatively few nonzero entries, which are again all of moderate size. To achieve these goals at least approximately we employ heuristic methods for achieving small coloring numbers short total distances in a traveling salesman variation. We verified the effectiveness of the overall approach on a significant number of test cases from the Florida matrix collection.

When the sparsity pattern is not hundred percent certain or the accumulation of round off is a reason for concern one might add an extra column to S so that the compressed Jacobian contains some redundant information that can be checked for consistency. This could be selected as an extreme and Cartesian color so that the covering problem would become much easier. Like for all augmentations we can then check whether the extra components of b_i are zero up to the error size suggested by our analysis. If not we can conclude that the sparsity pattern was wrongly specified, possibly due to control flow variations in the evaluation procedure for F. Serious implementation with this feature is currently under development.

Besides the Jacobian vector products (3) which corresponds to the forward mode of AD providing a column compression of the Jacobian, one also can aim for a row compression defining matrices

$$C = W^\top F'. \tag{11}$$

The reverse mode of AD provides the matrix C for an arbitrary weight matrix $W \in \mathbb{R}^{q\times m}$. By $C^\top = F'^\top W$ we see, that for this type of compression the nonzeros per column and the chromatic number of the row incidence graph have to be considered. Then the whole procedure from section 4 can be applied to obtain W. Note that the computation of the compressed matrix C cannot be based on divided differences.

When F' has (almost) dense columns the forward mode is more efficient, when it has (almost) dense rows the reverse mode is preferable. If it has both one can apply two-sided compression as suggested by Coleman and Verma [2], now again with minimal complexity in the NR fashion suggested here.

Similarly, two-sided compression can be applied to evaluate sparse Hessians of functions $f : \mathbb{R}^n \to \mathbb{R}$ or the corresponding complete second derivative tensor of vector valued functions $F : \mathbb{R}^n \to \mathbb{R}^m$ [8]. For this gradients and second order adjoints can be evaluated in the reverse mode, each at a complexity comparable to the cost of the underlying partially separable function.

Of course, for second derivatives, approximation by divided differences is even more inaccurate than for first, but a mixed mode with second order adjoints approximated by differencing on gradients obtained via AD or hand coding is certainly feasible. Again the coloring and ordering strategies developed in this paper should be very useful.

To the best of our knowledge third and higher derivative tensors are rarely used in scientific computing, though ADOL-C provides tools for their dense evaluation and the methods employed here could naturally also be generalized to such scenarios. They might occur for example in bifurcation and stability calculations.

Acknowledgements

The authors thank the diploma student Daniel Josh for providing numerical results. They are also indebted to Arun Verma and Shahadat Hossein who contributed significantly to earlier unpublished papers on the Newsam-Ramsdell-Lagrange seeding approach.

References:

[1] A.R. Curtis, M.J.D. Powell, and J.K. Reid, *On the estimation of sparse Jacobian matrices*, J. Inst. Math. Appl. **13** (1974), 117–119.

[2] T. F. Coleman and A. Verma *Structure and efficient Jacobian calculation* Cornell Theory Center Technical Report (1996)

[3] T. A. Davis and Y. Hu *The University of Florida sparse matrix collection* ACM Transactions on Mathematical Software **38** (2011)

[4] P. Erdös, *Problems and Results on the Theory of Interpolation*, Acta Math. Acad. Sci. Hung. **XII** (1955), 235–244.

[5] P. Erdös and P. Turan, *On the Role of the Labesque Functions in the Theory of Lagrange Interpolation*, Acta Math. Acad. Sci. Hung. **VI** (1955), 47–66.

[6] W. Gautschi, *Optimally Conditioned Vandermonde Matrices*, Numer. Math. **24:1** (1975).

[7] A. H. Gebremedhin, D. Nguyen, Md. M. A. Patwary and A. Pothen *ColPack: Software for graph coloring and related problems in scientific computing* AACM Transactions on Mathematical Software **40:1** (2013), 1–31

[8] A. Griewank. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, 2008.

[9] A. Griewank and N. Strogies *Efficient Evaluation of Sparse Jacobians by Matrix Compression, Part I: Motivation and Theory*

[10] R. M. Karp *Reducibility among combinatorial problems*, Springer (1972)

[11] F.W. Luttmann and T. J. Rivlin, *Some Numerical Experiments in the Theory of Polynomial Interpolation*, IBM J. Res. Develop. **9** (1965), 187–191.