Derivative Accumulation on Compressed Row Storage of Extended Jacobians

Ebadollah Varnik and Uwe Naumann
RWTH Aachen University
Department of Computer Science
Seffenter Weg 23, D-52056 Aachen
Germany
[varnik|naumann]@stce.rwth-aachen.de

1 Introduction

The context of this paper is *automatic differentiation* of numerical programs [2]. We consider vector functions

$$F : I\!R^n \supseteq D \to I\!R^m, \quad \mathbf{y} = F(\mathbf{x}) \quad , \qquad (1)$$

that map a vector $\mathbf{x} \equiv (x_i)_{i=1,\ldots,n}$ of *independent* variables onto a vector $\mathbf{y} \equiv (y_j)_{j=1,\ldots,m}$ of *dependent* variables. We assume that F has been implemented as a computer program. We refer to C' as the *extended Jacobian* and G as the *linearized computational graph* of F, respectively [2]. Rows and columns of C' are enumerated as $j, i = 1, \ldots, n + p, \ldots, n + p + m$. Row j of C' corresponds to vertex j of G and contains the partial derivatives $c_{j,k}$ of vertex j w.r.t. all of its predecessors $k \in P_j$. In the following we refer to a row [vertex] i as *independent* for $i \in \{1, \ldots, n\}$, as *intermediate* for $i \in \{n + 1, \ldots, n + p\}$, and as *dependent* if $i \in \{n+p+1, \ldots, n+p+m\}$. Our aim is to compute Jacobian efficiently on the *Compressed Row Storage* CRS (α, κ, ρ) representation of C' [1] by trading *fill-out* for *fill-in* during elimination [3].

2 Problem Description

Consider a topological ordering of the vertices

$$top : \{v_{n+1}, \ldots, v_q\} \to \{n + 1, \ldots, q\},$$

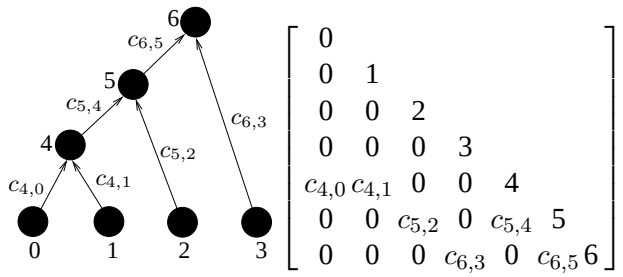
with $top(v_i) = k$ is the topological index of v_i. We aim to find such a topological ordering of the non-independent vertices of G such that the number of edges (i, j) with $top(j) = top(i) + 1$ is maximal. In this case, the fill-out generated by elimination of row i can be reused to store fill-in of row j. On CRS such fill-out exploitation can destroy the order of κ entries which makes a linear search for column indices unavoidable.

3 Example

As an example we consider the following implementation of a function f.

```
y=x[0];
for(int i=1; i<n; i++) y = y*x[i];
```

For $n = 4$ the corresponding computational graph G and extended Jacobian of f are shown in Figure 1 (a) and (b), respectively. The following illustrates the accumulation of Jacobian of f by reverse elimination



$$\begin{bmatrix} 0 & & & & & & \\ 0 & 1 & & & & & \\ 0 & 0 & 2 & & & & \\ 0 & 0 & 0 & 3 & & & \\ c_{4,0} & c_{4,1} & 0 & 0 & 4 & & \\ 0 & 0 & c_{5,2} & 0 & c_{5,4} & 5 & \\ 0 & 0 & 0 & c_{6,3} & 0 & c_{6,5} & 6 \end{bmatrix}$$

(a) (b)

Figure 1: Linearized computational graph G (a) and extended Jacobian (b) of f.

on CRS.

$$\begin{aligned} \alpha =& [c_{4,0}, c_{4,1}, c_{5,2}, c_{5,4}, c_{6,3}, c_{6,5}] \\ \kappa =& [0, 1, 2, 4, 3, 5] \\ \rho =& [0, 0, 0, 0, 0, 2, 4, 6] \end{aligned}$$

Elimination of 5 yields :

$$\begin{aligned} \alpha =& [c_{4,0}, c_{4,1}, c_{6,5}.c_{5,2}, c_{6,5}.c_{5,4}, c_{6,3}, \mathbf{0}] \\ \kappa =& [0, 1, 2, 4, 3, 5] \\ \rho =& [0, 0, 0, 0, 0, 2, 2, 6] \end{aligned}$$

Elimination of 4 yields :

$$\begin{aligned} \alpha =& [c_{6,5}.c_{5,4}.c_{4,0}, c_{6,5}.c_{5,4}.c_{4,1}, c_{6,5}.c_{5,2}, \mathbf{0}, c_{6,3}, \mathbf{0}] \\ \kappa =& [0, 1, 2, 4, 3, 5] \\ \rho =& [0, 0, 0, 0, 0, 0, 0, 6] \end{aligned}$$

Bold **0**'s in α represent fill-out.

4 Results and Outlook

Figure 2 illustrates the runtime analysis of Jacobian accumulation in reverse order on CRS (RElimOnCRS) against forward mode on CRS (FElimOnCRS), the centered finite approximation (CenteredFinite) and reverse elimination on the computational graph (RElimOnDAG). Ongoing work is focused on finding an elimination ordering that maximizes the number of immediate successors dynamically during the elimination process. Thus we expect to make even better use of fill-out eventually leading to even faster Jacobian codes.

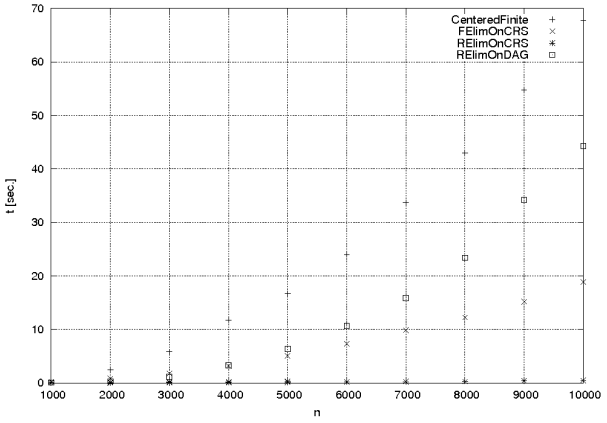


Figure 2: Runtime comparison of different Jacobian accumulation methods.

References:

[1] A. Lyons E. Varnik, U. Naumann. Toward low static memory Jacobian accumulation. *WSEAS Transactions on Mathematics*, 5:909–917, July 2006.

[2] A. Griewank. *Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 2000.

[3] U. Naumann. Optimal accumulation of Jacobian matrices by elimination methods on the dual computational graph. *Math. Prog.*, 99(3):399–421, April 2004.