Reversing the Row Order for the Row-by-Row Frontal Method

J K Reid and J A Scott

22nd June 1999
Reversing the row order for the row-by-row frontal method

by

J. K. Reid and J. A. Scott

Abstract

The efficiency of the row-by-row frontal method is dependent on the row ordering used. Numerical experience has shown us that it can be advantageous to reverse a given row ordering. We present two results on invariances under the reversal of the ordering and use real applications to illustrate the variations that can take place.

Keywords: ordering rows, frontal method.

Computational Science and Engineering Department, 
Atlas Centre, Rutherford Appleton Laboratory, 
Oxon OX11 0QX, England. 
1 Introduction

The row-by-row frontal method (see, for example, Duff, Erisman and Reid, 1986, Section 10.6) solves a large sparse unsymmetric set of $n$ linear equations by Gaussian elimination with the help of a full rectangular matrix held in memory and known as the frontal matrix. The size of the frontal matrix varies during the elimination. Rows are assembled (added) into the frontal matrix one by one. Whenever a column becomes fully summed, that is, the last row in which it has an entry is assembled, a pivot is chosen in the column and is used to eliminate the column and the row containing the pivot. The eliminated row and column are stored for use in the back-substitution or in the solution of further systems of equations.

Since an elimination can only take place after a column becomes fully summed, the order in which the rows are assembled will determine both how long a variable remains in the front and the order in which the variables are eliminated (apart from the order among columns that become fully summed at the same assembly step, which has no effect on the computational requirements). For efficiency, in terms of both storage and arithmetic operations, the rows need to be assembled in an order that keeps the size of the frontal matrix as small as possible. Scott (1998) has considered a number of strategies for determining such an ordering and has found that the results of using the different orderings on practical problems can vary enormously. While developing row ordering software for the Harwell Subroutine Library (1995), Scott tried experimenting with reversing any given row order. She found that this makes no difference to some of the usual measures of the quality of the ordering while it can make enormous changes to others.

These numerical results motivated us to prove the results on invariances under the reversal of the ordering that we present in Section 2. In Section 3, we illustrate the variations that can take place in real applications.

2 Two theorems on invariance

We follow Scott (1998) and use the terms row frontsizes and column frontsizes for the number of rows and columns in the front. Our invariance results pertain to the column frontsizes.

The row and column frontsizes after assembly refer to the frontsizes after a row has been added to the front. When a row is added, more than one column may become fully summed; for each such column, a pivot is chosen and an elimination is performed. The row and column frontsizes before elimination refer to the frontsizes immediately prior to an elimination. The frontal matrix size refers to the product of the row and column frontsizes.

We use the term forward row order to refer to the row ordering $1, 2, \ldots, n$ (that is, to the given ordering), while the reverse ordering is $n, n-1, \ldots, 1$. We assume that the matrix (in the forward order) is $A = \{a_{ij}\}$ and that a reference to 'row i' refers to the row $a_{i\ast}$. 
Before presenting our results, we introduce the small illustrative example

$$\begin{pmatrix} x \\ x & x & x \\ x & x & x & x \\ x & x \\ x & x \end{pmatrix}.$$  

With the forward order we have:

Assembly 1: Frontal matrix increases to $1 \times 1$. No eliminations.

Assembly 2: Frontal matrix increases to $2 \times 3$. No eliminations.

Assembly 3: Frontal matrix increases to $3 \times 4$. Columns 1 and 3 eliminated.
Frontal matrix reduces to $1 \times 2$.

Assembly 4: Frontal matrix increases to $2 \times 2$. Column 4 eliminated.
Frontal matrix reduces to $1 \times 1$.

Assembly 5: Frontal matrix increases to $2 \times 2$. Columns 2 and 5 eliminated.

Thus the row frontsize, column frontsize, and the frontal matrix size before each of the five eliminations and their mean values are

- row frontsize: 3 2 2 2 1 2.0
- column frontsize: 4 3 2 2 1 2.4
- frontal matrix size: 12 6 4 4 1 5.4

When the order is reversed, the corresponding statistics are

- row frontsize: 1 2 2 1 1 1.4
- column frontsize: 2 4 3 2 1 2.4
- frontal matrix size: 2 8 6 2 1 3.8

The first result is about the column frontsize after each assembly. This is important since the size of the frontal matrix after assembly determines how much memory is needed by the frontal solver.

**Theorem 1** If a given row order is reversed, the sequence of $n$ column frontsizes after the assemblies is reversed.

**Proof** After assembly of row $i$, in the forward or reverse row order, the column frontsize is the number of columns with first entry at row $i$ or earlier and last entry at row $i$ or later, that is, the number of columns $j$ for which $\min\{k : a_{kj} \neq 0\} \leq i$ and $\max\{k : a_{kj} \neq 0\} \geq i$.

**Corollary 1** The maximum column frontsize is invariant.

**Corollary 2** The mean column frontsize after assembly is invariant.
Corollary 3 The sum of the column frontsizes after assembly is invariant.

Note that the sum of the column frontsizes after assembly is equal to the sum of the lifetimes, where the lifetime of a variable is defined to be the number of assembly steps for which the variable is in the front. The lifetime of variable \( j \) is also the length of column \( j \), that is, \( \max\{k : a_{kj} \neq 0\} - \min\{k : a_{kj} \neq 0\} \). The sum of the lifetimes is used by Camarda (1997) to compare the quality of different row orderings.

Our second result concerns the column frontsize before each elimination. This is important since the row and column frontsizes before an elimination determine how much work is associated with the elimination and how much storage is required for the pivotal column and row.

Theorem 2 If a given row order is reversed, the sequence of \( n \) column frontsizes before the eliminations is permuted.

Proof At assembly step \( i \) in the forward order, the column frontsize increases by one for each column with its first entry in row \( i \), that is, each \( j \) such that \( \min\{k : a_{kj} \neq 0\} = i \). Following this, it decreases by one for each column with its last entry in row \( i \), that is, each \( j \) such that \( \max\{k : a_{kj} \neq 0\} = i \), as that column is eliminated. There are \( n \) occasions when it increases by one and \( n \) occasions when it decreases by one. These \( 2n \) events begin and end with the column frontsize of zero. For each increase from \( f - 1 \) to \( f \) there is a corresponding later decrease from \( f \) to \( f - 1 \), and to establish a one-to-one correspondence, we take the first such decrease if there is more than one. Each decrease corresponds to an elimination with column frontsizes \( f \) using the forward row order. Each increase corresponds to an elimination with column frontsizes \( f \) using the reverse row order. The result follows.

For our small example, the sequence of column frontsizes for the forward order is \( 0, 1, 2, 3, 4, 3, 2, 1, 2, 1, 0 \). Note that there are two increases from 1 to 2 and two decreases from 2 to 1; the first increase is taken to correspond with the first decrease and the second increase to correspond with the second decrease.

Corollary 4 The mean column frontsize before elimination is invariant.

Note that the mean values in Corollaries 2 and 4 usually differ, while there is only one maximum value.

3 Numerical experience

We now present some numerical results that illustrate our theoretical results. All the frontsizes quoted in this section are frontsizes before elimination. The test problems are those used by Scott (1998). Each problem arises from a real engineering or industrial application. Our results are presented in Tables 3.1
3 NUMERICAL EXPERIENCE

Table 3.1: The maximum (max.) and mean row and column frontsizes before elimination for the forward and reverse row ordering.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Forward</th>
<th></th>
<th></th>
<th>Reverse</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>max.</td>
<td>max.</td>
<td>mean</td>
<td>mean</td>
<td>max.</td>
<td>max.</td>
</tr>
<tr>
<td></td>
<td>row</td>
<td>col.</td>
<td>row</td>
<td>col.</td>
<td>row</td>
<td>col.</td>
</tr>
<tr>
<td>bayer04</td>
<td>349</td>
<td>501</td>
<td>174</td>
<td>275</td>
<td>263</td>
<td>501</td>
</tr>
<tr>
<td>bayer09</td>
<td>83</td>
<td>153</td>
<td>38</td>
<td>62</td>
<td>78</td>
<td>153</td>
</tr>
<tr>
<td>bp1600</td>
<td>242</td>
<td>324</td>
<td>91</td>
<td>142</td>
<td>143</td>
<td>324</td>
</tr>
<tr>
<td>extr1</td>
<td>32</td>
<td>46</td>
<td>15</td>
<td>27</td>
<td>21</td>
<td>46</td>
</tr>
<tr>
<td>gre1107</td>
<td>95</td>
<td>208</td>
<td>55</td>
<td>125</td>
<td>123</td>
<td>208</td>
</tr>
<tr>
<td>hydr1</td>
<td>45</td>
<td>86</td>
<td>22</td>
<td>43</td>
<td>42</td>
<td>86</td>
</tr>
<tr>
<td>lhr07c</td>
<td>169</td>
<td>225</td>
<td>51</td>
<td>111</td>
<td>127</td>
<td>225</td>
</tr>
<tr>
<td>lhr14c</td>
<td>164</td>
<td>306</td>
<td>69</td>
<td>188</td>
<td>255</td>
<td>306</td>
</tr>
<tr>
<td>meg1</td>
<td>700</td>
<td>1150</td>
<td>368</td>
<td>639</td>
<td>517</td>
<td>1150</td>
</tr>
<tr>
<td>onetone2</td>
<td>297</td>
<td>658</td>
<td>198</td>
<td>405</td>
<td>365</td>
<td>658</td>
</tr>
<tr>
<td>orani678</td>
<td>1333</td>
<td>1576</td>
<td>604</td>
<td>771</td>
<td>320</td>
<td>1576</td>
</tr>
<tr>
<td>rdist1</td>
<td>40</td>
<td>81</td>
<td>28</td>
<td>60</td>
<td>49</td>
<td>81</td>
</tr>
<tr>
<td>west2021</td>
<td>38</td>
<td>52</td>
<td>18</td>
<td>29</td>
<td>22</td>
<td>52</td>
</tr>
</tbody>
</table>

Table 3.2: The maximum and mean frontal matrix size and the sum of lifetimes (+10⁵) for the forward and reverse row ordering.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Forward</th>
<th></th>
<th></th>
<th>Reverse</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max.</td>
<td>Mean</td>
<td>Sum of</td>
<td>Max.</td>
<td>Mean</td>
<td>Sum of</td>
</tr>
<tr>
<td></td>
<td>frontal</td>
<td>frontal</td>
<td>matrix</td>
<td>frontal</td>
<td>frontal</td>
<td>matrix</td>
</tr>
<tr>
<td></td>
<td>matrix</td>
<td>size</td>
<td>lifetimes</td>
<td>size</td>
<td>lifetimes</td>
<td>size</td>
</tr>
<tr>
<td>bayer04</td>
<td>17365</td>
<td>5341</td>
<td>5708</td>
<td>12387</td>
<td>3340</td>
<td>5708</td>
</tr>
<tr>
<td>bayer09</td>
<td>1237</td>
<td>283</td>
<td>194</td>
<td>1193</td>
<td>197</td>
<td>194</td>
</tr>
<tr>
<td>bp1600</td>
<td>6607</td>
<td>1833</td>
<td>100</td>
<td>4633</td>
<td>659</td>
<td>100</td>
</tr>
<tr>
<td>extr1</td>
<td>147</td>
<td>43</td>
<td>76</td>
<td>84</td>
<td>37</td>
<td>76</td>
</tr>
<tr>
<td>gre1107</td>
<td>1925</td>
<td>814</td>
<td>140</td>
<td>2489</td>
<td>1055</td>
<td>140</td>
</tr>
<tr>
<td>hydr1</td>
<td>387</td>
<td>104</td>
<td>226</td>
<td>361</td>
<td>104</td>
<td>226</td>
</tr>
<tr>
<td>lhr07c</td>
<td>3489</td>
<td>622</td>
<td>858</td>
<td>2331</td>
<td>791</td>
<td>858</td>
</tr>
<tr>
<td>lhr14c</td>
<td>5953</td>
<td>1530</td>
<td>2755</td>
<td>8992</td>
<td>2849</td>
<td>2755</td>
</tr>
<tr>
<td>meg1</td>
<td>77970</td>
<td>28129</td>
<td>1750</td>
<td>54395</td>
<td>18375</td>
<td>1750</td>
</tr>
<tr>
<td>onetone2</td>
<td>19483</td>
<td>8646</td>
<td>14637</td>
<td>23980</td>
<td>9264</td>
<td>14637</td>
</tr>
<tr>
<td>orani678</td>
<td>210080</td>
<td>64517</td>
<td>1704</td>
<td>48247</td>
<td>8679</td>
<td>1704</td>
</tr>
<tr>
<td>rdist1</td>
<td>275</td>
<td>169</td>
<td>242</td>
<td>397</td>
<td>202</td>
<td>242</td>
</tr>
<tr>
<td>west2021</td>
<td>198</td>
<td>57</td>
<td>58</td>
<td>86</td>
<td>36</td>
<td>58</td>
</tr>
</tbody>
</table>
and 3.2. In each case, the problem is reordered using our new code MC62 (Scott, 1999), with default values for all control variables.

From the tables, we see that for some problems, including bp1600 and orani678, it can be extremely advantageous to reverse the row order. For other problems, such as hydri, reversal has little effect. Note also that reversing the order can reduce the maximum row frontal size while increasing the mean row frontal size or the mean frontal matrix size. This is illustrated by problem 1hr07d. In this case, which is the better ordering depends upon whether the prime concern is to reduce main memory requirements (choose the smaller maximum frontal matrix size), to minimise factor storage (choose the smaller mean row frontal size), or to minimise the operation count (choose the smaller mean frontal matrix size). Since the cost of computing the frontal sizes for the forward and reverse orders is negligible compared with the cost of using a frontal solver, MC62 computes a new row ordering and then automatically reverses it and selects the better of the two orderings. By default, MC62 chooses the ordering for which the mean frontal matrix size is smaller.

References


