Parallel Communication Analysis for Sparse Cholesky Factorization Algorithms

Victor Amelkin, CRD@LBNL

September 20, 2012

This work is an attempt to analyze parallel communication for parallel sparse Cholesky factorization algorithms. The current immediate goal is to compute the volume of parallel communication as well as the number of messages sent during parallel communication for the right-looking Cholesky factorization algorithm. The notation used in this work is mostly consistent with [1].

1 Problem Statement

We focus on linear systems stemming from discretization of PDEs. The non-zero structure of matrices of such systems depends on the discretized domain and the stencil in use. Analyzing parallel communication for an arbitraty problem seems unfeasible. Thus, we are dealing with a *model problem*: a square k-by-k mesh and a 5-point stencil. Presumably, the results for other stencils using the same mesh will differ from the results for the 5-point stencil only by a lower-order term, which is acceptable, since we are primarily interested in an asymptotic behavior of parallel Cholesky.

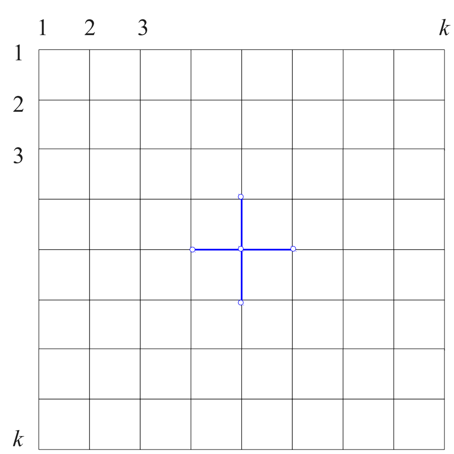

Figure 1: A square k-by-k mesh and a 5-point stencil.

The size k of the mesh is assumed to be large enough, so that we can consider the mesh perfectly partitionable, thereby, achieving perfect balancing of data accross processes. The nodes in the mesh and, equivalently, the unknowns in the corresponding linear system, will be ordered by nested dissection using cross-separators, similarly to the way it is done in [1]. Nested dissection is a good choice because, for one thing, it is one of the standard “good” orderings that reduce fill and improve parallelism and, for another thing, analyzing a recursively ordered mesh is easier than analyzing a naturally-ordered one. The cross-separators are chosen over bisecting separators due to the symmetry of the former, which, again, makes the analysis easier. Further, under a *separator* we understand a cross-separator unless specified otherwise.

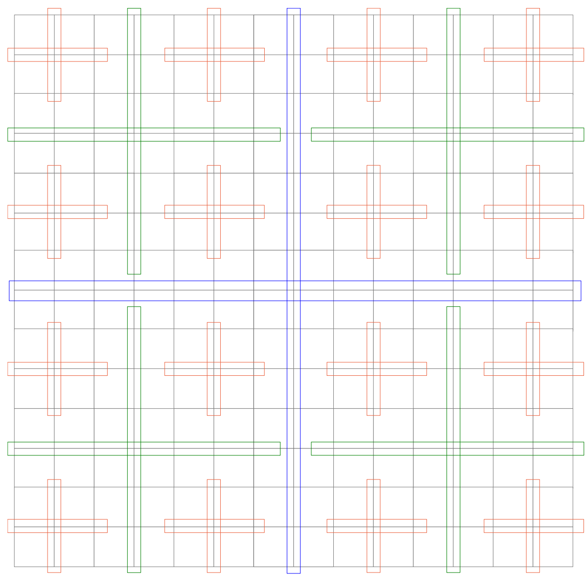

Figure 2: A 15-by-15 mesh partitioned by nested dissection with cross-separators.

Separators will be denoted with S. When we want to specify the level i of a separator in the elimination tree $\mathrm{T}(\boldsymbol{A})$ (where $\boldsymbol{A}$ is the linear system's matrix) and the position q of this separator within its level, we will refer to such a separator as S^i_q. If we want to target different parts of a separator, we shall write S^i_{Lq}, S^i_{Vq}, and $S^i_{R_q}$, respectively, for the horizontal left, vertical, and horizontal right parts of a separator S^i_q. If the subscripts L, V, or R are not present, then we are addressing the entire separator.

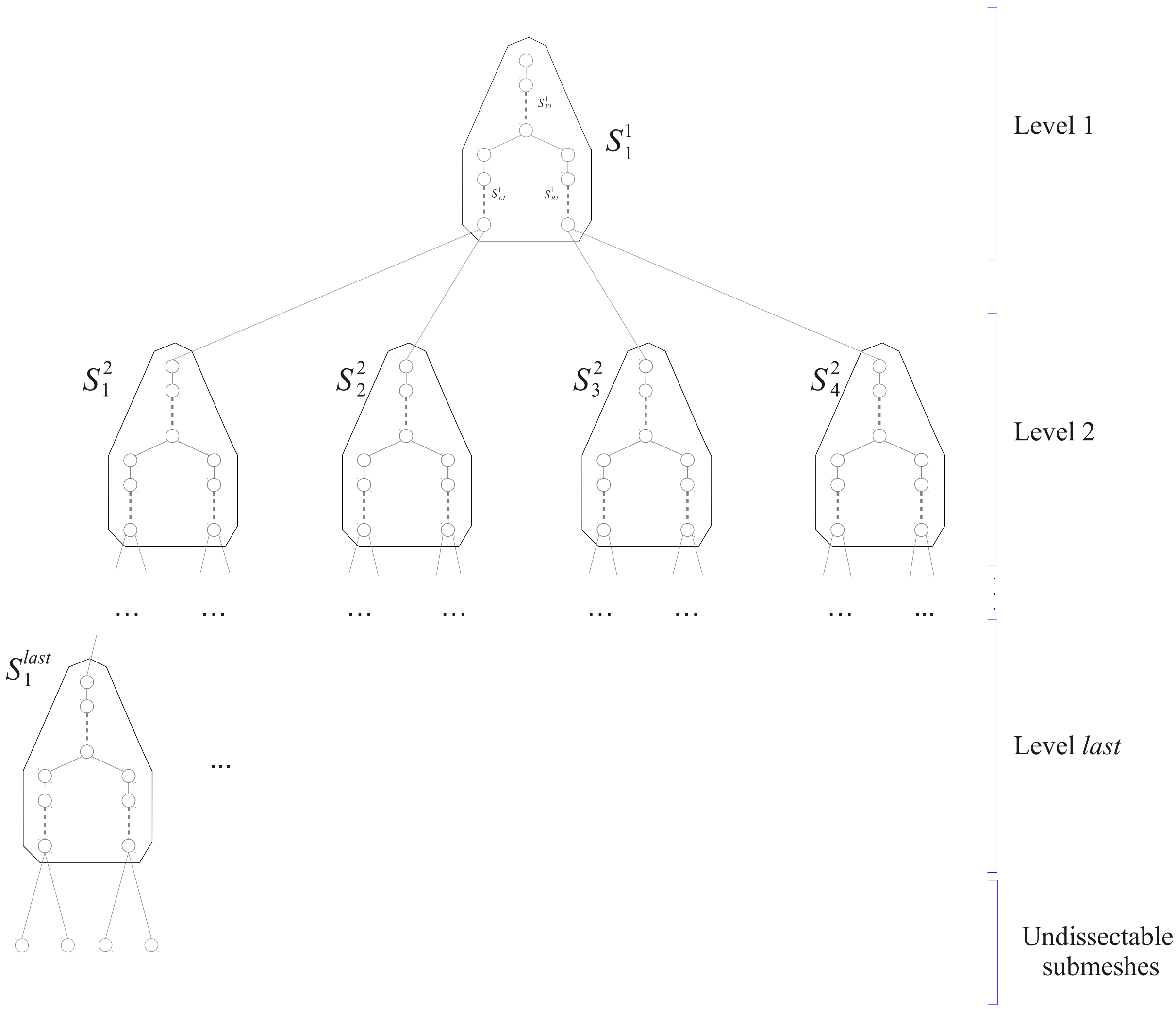

Figure 3: The elimination tree $\mathrm{T}(\boldsymbol{A})$ corresponding to a mesh ordered by nested dissection with cross-separators. Nested dissection has stopped at 1-by-1 undissectable submeshes.

The last and, yet, undefined input parameter for out problem is the number of processes $p = |P|$ we are going to use to compute the Cholesky factor, where P stands for the set containing all the processes we have. In general, this number will be chosen to be significantly less than the size k of the mesh. However, to maintain perfect balancing, we can use only the numbers of processes of the form $p = |P| = 4^\sigma$, where $\sigma \in \mathbb{N} \cup \{0\}$. For such numbers of processes, the data are perfectly balanced accross the processes, which makes our analysis easier. In order to determine possible values for σ as well as some other important characteristics of the factorization algorithm, we need to know a few useful properties of the elimination tree $\mathrm{T}(\boldsymbol{A})$.

2 Some combinatorial properties of $\mathbf{T}(\boldsymbol{A})$

All the properties are stated in relation to a mesh ordered by nested dissection with cross-separators.

Lemma 1 (about the level size)**.** *The number of separators at the i'th level of the elimination tree is* $\#_i = 4^{i-1}$*, where* $i = 1, 2, \ldots, ($*last level's index*$)$[1].

Proof. Obviously, each of the two horizontal part of every separator at level i forks 2 new separators at level $(i + 1)$. Thus, $\#_{i+1} = 4\#_i$. From the obtained recurrence and the initial condition $\#_1 = 1$ (– there is only one separator at the topmost level), it follows that $\#_i = 4^{i-1}$. □

Lemma 2 (about the separator size)**.** *The number of columns in* S_*^i *is* $|S_*^i| = \frac{k+1}{2^{i-2}} - 3$*, where* $i = 1, 2, \ldots, ($*last level's index*$)$*. By '*' in* S_*^i *we mean that* $|S_q^i|$ *is the same for every separator* S_q^i *within level* i*.*

Proof. First, let us notice that it is easy to determine the number of columns in a separator S^i if we know the number of columns in this separator's vertical part S_V^i:

$$|S^i| = |S_V^i| + \underbrace{(|S_V^i| - 1)}_{|S_L^i|+|S_R^i|} = 2|S_V^i| - 1.$$

Since we are using nested dissection, the number of columns $|S_V^i|$ can be recurrently characterized as

$$|S_V^{i+1}| = \frac{|S_V^i| - 1}{2}, \ |S_V^1| = 1.$$

Solving the latter recurrence, we have

$$|S_V^i| = \frac{k+1}{2^{i-1}} - 1.$$

Having substituted this result in the expression for $|S^i|$, we obtain

$$|S^i| = 2|S_V^i| - 1 = \frac{k+1}{2^{i-2}} - 3.$$

□

Let us count as *levels* in the elimination tree only those levels consisting of separators. Thereby, we leave out the undissectable 1-by-1 or 4-by-4[2] submeshes at the bottom of the elimination tree or, equivalently, at the highest depth in the mesh. In no way this will affect the distribution of columns accross the processes or the volume of parallel communication.

[1] Will be defined later.

[2] A particular size – 1-by-1 or 4-by-4 – of the undissectable submeshes depends on the size k of the original mesh. It is good to note that both cases are possible even if a mesh is perfectly partitionable, i.e., ideal balancing is achieved.

Lemma 3 (about the number of levels in $\mathrm{T}(\boldsymbol{A})$)**.** *The number of levels in the elimination tree $T(\boldsymbol{A})$ is $N_{Lev} = \log_2{(k+1)} - 1$.*

Proof. In this proof, let us stick with the case when nested dissection stops at 1-by-1 undissectable submeshes. However, using the same techniques for the case of 4-by-4 undissectable submeshes, we obtain the same result.

On the one hand, the number of columns below the last level (the very same columns belonging to the undissectable submeshes) is simply what remains if we subtract the number of columns belonging to all the levels from the number of all the columns in $\mathrm{T}(\boldsymbol{A})$ (once again, by levels we mean only those containing separators):

$$x = k^2 - \sum_{i=1}^{N_{\mathrm{Lev}}} \#_i |S^i| = (\text{from Lemmas 1, 2}) = k^2 - \sum_{i=1}^{N_{\mathrm{Lev}}} 4^{i-1} \left(\frac{k+1}{2^{i-1}} - 3 \right). \tag{1}$$

On the other hand, we know that every separator within the last level of $\mathrm{T}(\boldsymbol{A})$ forks 4 undissectable submeshes of size 1-by-1, and, thus, the number of columns below the last level is

$$x = 4\#_{N_{\mathrm{Lev}}} = (\text{from Lemma 1}) = 4^{N_{\mathrm{Lev}}}. \tag{2}$$

Combining (1) and (2), we get the following equation for N_{Lev}:

$$k^2 - \sum_{i=1}^{N_{\mathrm{Lev}}} 4^{i-1} \left(\frac{k+1}{2^{i-1}} - 3 \right) = 4^{N_{\mathrm{Lev}}}.$$

It is left to the reader to make sure that the following expression for N_{Lev} is the solution of the obtained equation:

$$N_{\mathrm{Lev}} = \log_2{(k+1)} - 1.$$

If the mesh size k corresponds to a perfectly partitionable mesh, i.e., $k = 2^i - 1,\ i \in \mathbb{N}$, but nested dissection stops at 4-by-4 undissectable submeshes, then the obtained expression for N_{Lev} should be wrapped with a lower bound opearator $\lfloor \dots \rfloor$. $\square$

3 Column Assignment to Processes

First of all, it is worthy to complete specification for the number of processes. Previously, we mentioned that we will use $p = |P| = 4^\sigma$ processs, where $\sigma = 1, 2, \dots$ − such a number guarantees perfect load balancing, which simplifies our analysis. The maximum value of σ corresponds to the situation when every separator $S_q^{N_{\mathrm{Lev}}}$ of the last level along with a few[3] columns below it are assigned to an individual process. In such a case, the number of processes is $4^{\sigma_{max}} = 4^{\log_2{(k+1)}-1} = (k+1)^2/4$.

[3] There are 4 columns below each last-level separator if nested dissection stops at 1-by-1 undissectable submeshes and 16 columns in the case with 4-by-4 undissectable submeshes.

However, this maximum σ_{max} is unlikely to be feasible in practice. Thus, though we will keep in mind the theoretical maximum for the number of processes, in practice, we will think about the number of processes simply as some power of 4:

$$p = |P| = 4^{\sigma} \ll k, \ \sigma \in \mathbb{N} \cup \{0\}.$$

Since we do not have enough processes to assign one to every column, we are going to assign large subtrees of $\mathrm{T}(\boldsymbol{A})$ to individual processes and, then, proceed up to the root of $\mathrm{T}(\boldsymbol{A})$ using the wrap-around assignment scheme. Such an assignment strategy is known as *subtree-to-subcube* assignment scheme [2].

The number of processes was specifically chosen to maintain perfect data balancing accross processes. Thus, the number of processes corresponds to the number of separators at a certain level of $\mathrm{T}(\boldsymbol{A})$ or, equivalently, the number of subtrees of $\mathrm{T}(\boldsymbol{A})$ rooted at these separators. Since $p = |P| = 4^{\sigma}$, this certain level of $\mathrm{T}(\boldsymbol{A})$ is $(\sigma + 1)$, which is clear from the following two figures.

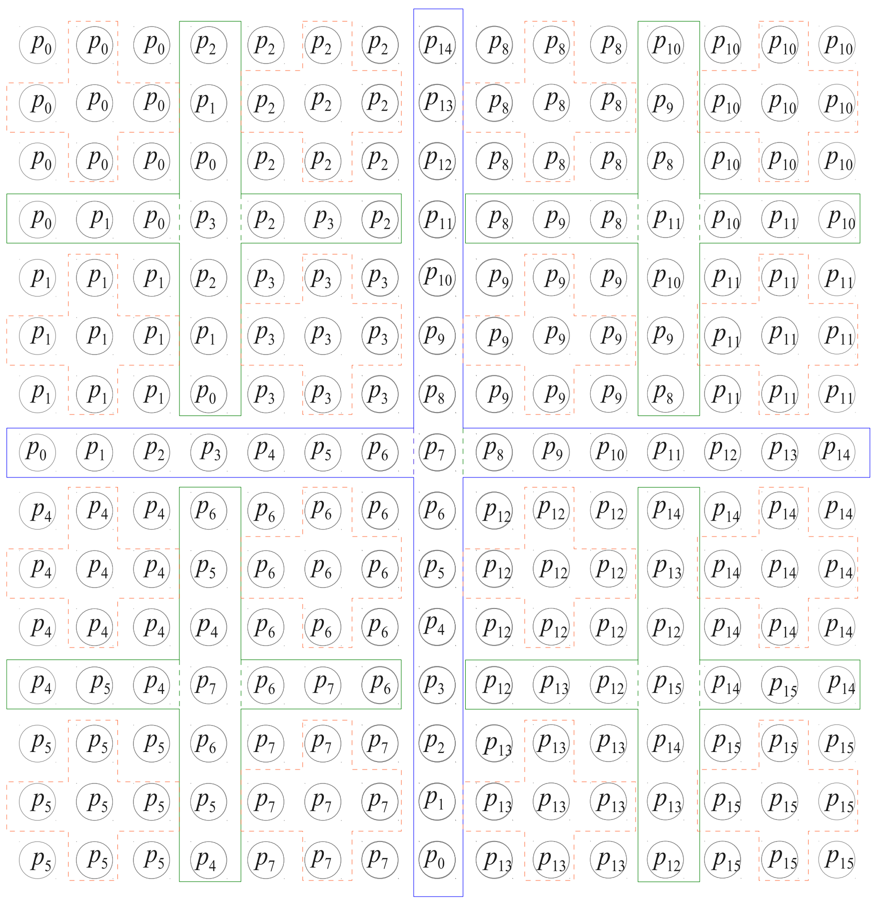

Figure 4: A 15-by-15 mesh ordered by nested dissection with cross-separators with columns assigned to $p = 4^2 = 16$ processes using the subtree-to-subcube assignment strategy.

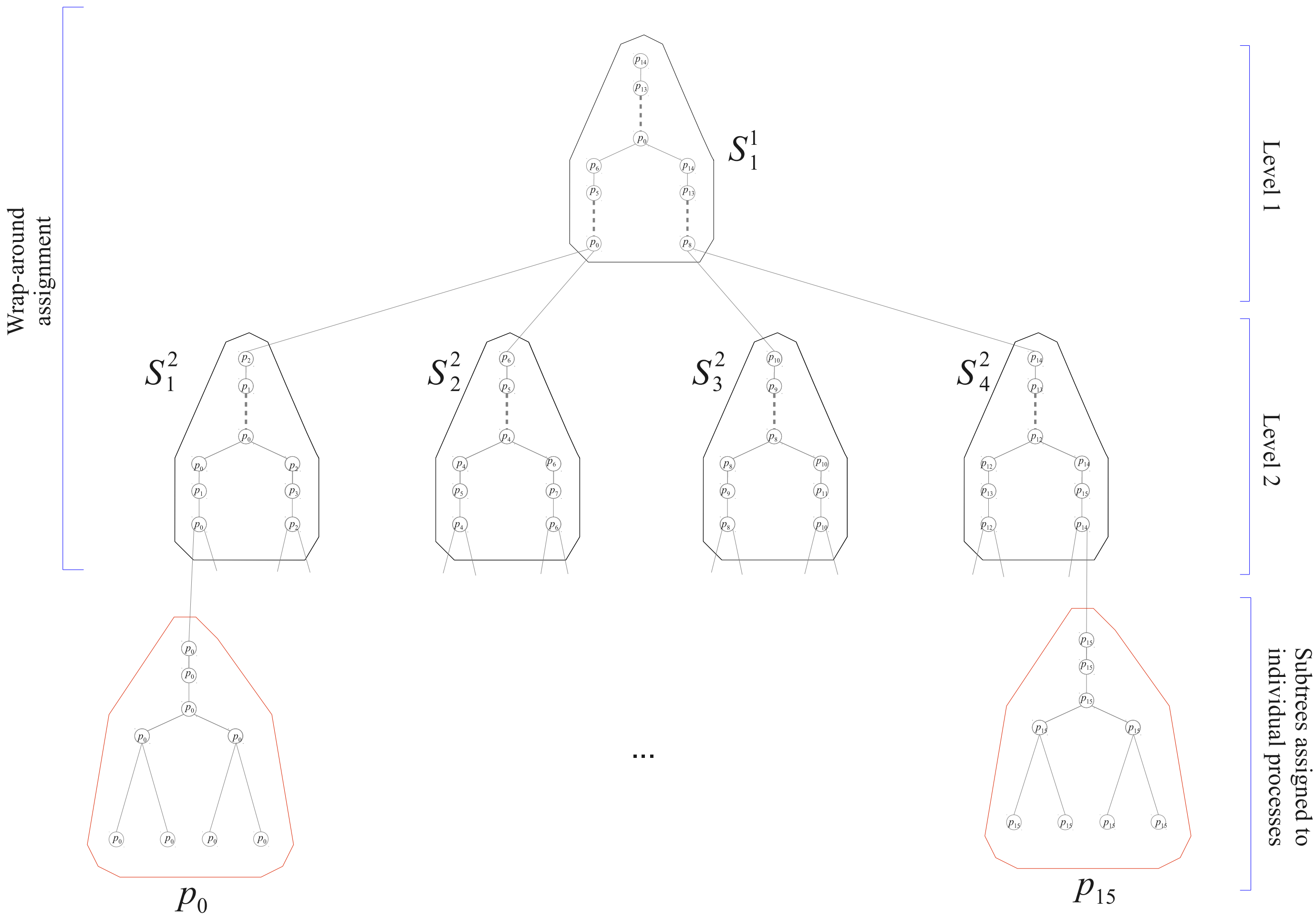

Figure 5: The elimination tree for a 15-by-15 mesh ordered by nested dissection with cross-separators with columns assigned to $p = 4^2 = 16$ processes using the subtree-to-subcube assignment strategy. $\sigma = 2$, and, thus, columns at Level 1 and 2 are assigned using the wrap-around assignment scheme, while the subtrees starting with the vertical separators of Level 3 are assigned to processes one-to-one.

It is easy to see that there is no parallel communication happening when the columns within the above mentioned subtrees are computed, i.e. those columns within the separators of level $(\sigma + 1)$ and below. However, at level σ, parallel communication starts, and its volume increases while we proceed up towards the root of $\mathrm{T}(\boldsymbol{A})$.

4 Parallel Communication Analysis for Right-Looking Cholesky

The parallel sparse right-looking (fan-out) Cholesky factorization algorithm looks as follows:

Algorithm 1 Parallel sparse right-looking (fan-out) Cholesky

for $j \in mycols$ **do**
  # while column j is a leaf in $\mathrm{T}(\boldsymbol{A})$ (i.e., has no dependency columns)
  # or all its dependency columns have already been computed
  if j is a leaf node in $\mathrm{T}(\boldsymbol{A})$ **then**
    # finalize column j and send it to the processes who need it
    $cdiv(j)$
    send $L^{\blacktriangle}_{*j}$ to processes $map(\Omega(L^{\triangle}_{*j}))$
    $mycols = mycols \setminus \{j\}$
  end if
  # while I still have some columns that have not been computed
  # (since they require updates from other processes)
  while $mycols \neq \varnothing$ **do**
    # receive an update
    receive any column, say, L_{*k}
    # apply the update to all my columns who need it
    for $j \in mycols \cap \Omega(L^{\triangle}_{*k})$ **do**
      $cmod(j \leftarrow k)$
      # if column j requires no more updates,
      # finalize and send it to those processes who need it
      if column j requires no more *cmod*s **then**
        $cdiv(j)$
        send $L^{\blacktriangle}_{*j}$ to processes in $map(\Omega(L^{\triangle}_{*j}))$
        $mycols = mycols \setminus \{j\}$
      end if
    end for
  end while
end for

To measure the volume of parallel communication, let us analyze when and how much data processes send to each other. It is clear that parallel communication during processing of a column j occures when this column j has received all its updates ($cmod(j \leftarrow *)$) and been finalized ($cdiv(j)$). As soon as such a situation happens, the column j is sent to those processes who hold columns corresponding to the positions of non-zeros in column j ($map(\Omega(L^{\triangle}_{*j}))$).

Consequently, the expression for the volume of parallel communication should look as follows:

$$comm = \sum_{l=1}^{n} \Big(\eta(L^{\triangle}_{*l}) \times \text{“number of receivers”} \Big).$$

In this expression as it is stated, it is quite challenging to compute both the number of non-zeros in every column l and the “number of receivers” – the number of processes who need column l to *cmod* their columns. The reason is that both these quantities differ accross different separators S^i_q.

Thus, it makes sense to group the summands of the sum above by separators:

$$comm = \sum_{i=1}^{\sigma} \sum_{q=1}^{\#_i} \sum_{l \in S_q^i} \Big(\eta(L_{*l}^{\llcorner}) \times \text{“number of receivers”} \Big).$$

Now we have to count the number of non-zeros whithin a q'th separator at level i of $\mathrm{T}(\boldsymbol{A})$, which can easily be done using the technique from [1] for counting non-zeros in L. To simplify our analysis, let us make the following assumption.

Assumption 1. *Let us think of the original mesh as bordered and surrounded with some separators of levels higher than* 1 *(even though in reality there are no separators higher than* S_1^1 *in* $T(\boldsymbol{A})$*).*

In the context of counting non-zeros, Assumption 1 corresponds to an overapproximation – some non-zeros will correspond to the (non-existing in reality) paths that end at the border of the original mesh, as it becomes clear from the proof of the following lemma.

Lemma 4 (about *nnz* in a separator's columns)**.** *The aggregate number of non-zeros in the columns of a separator* S^i *(– any separator at level* i*) is*

$$\eta(S^i) = 31\left(\frac{k}{2^i}\right)^2 - 11\left(\frac{k}{2^i}\right) + \frac{7}{4}.$$

Proof. Counting non-zeros in the columns of a separator is based on a theorem from [1] stating that non-zeros in the Cholesky factor correpond to the paths in $\mathrm{T}(\boldsymbol{A})$ with the interior columns of these paths having indices less than both path ends. Thus, to count the number of non-zeros in the columns of S^i, we need to count the number of paths starting at the columns of S^i and having the mentioned property.

$$\eta(S^i) = \eta(S_L^i) + \eta(S_V^i) + \eta(S_R^i) = \Big(\eta(S_L^i) = \eta(S_R^i) \text{ in a bordered (sub)mesh}\Big) =$$

$$= 2 \times \eta(S_L^i) + \eta(S_V^i) =$$

$$= 2 \times \left\langle \sum_{l=1}^{|S_L^i|} \left[\left(|S_L^i| - l\right) + 2|S_V^i| + 2|S_L^i| \right] \right\rangle + \left\langle \sum_{l=1}^{|S_V^i|} \left[\left(|S_V^i| - l\right) + 4|S_V^i| \right] \right\rangle =$$

$$= \Big(\text{trivially follows from Lemma 2 about the size of a separator}\Big) =$$

$$= 2 \times \left\langle \sum_{l=1}^{\frac{k}{2^i} - \frac{1}{2}} \left[\left(\left(\frac{k}{2^i} - \frac{1}{2} \right) - l \right) + 2\left(\frac{k}{2^{i-1}} \right) + 2\left(\frac{k}{2^i} - \frac{1}{2} \right) \right] \right\rangle +$$

$$+ \left\langle \sum_{l=1}^{\frac{k}{2^{i-1}}} \left[\left(\left(\frac{k}{2^{i-1}} \right) - l \right) + 4\left(\frac{k}{2^{i-1}} \right) \right] \right\rangle =$$

$$= 2 \times \left\langle \frac{13}{2}\left(\frac{k}{2^i}\right)^2 - 5\left(\frac{k}{2^i}\right) + \frac{7}{8} \right\rangle + \left\langle 18\left(\frac{k}{2^i}\right)^2 - \frac{k}{2^i} \right\rangle =$$

$$= 31\left(\frac{k}{2^i}\right)^2 - 11\left(\frac{k}{2^i}\right) + \frac{7}{4}.$$

□

Counting the number of receivers is still problematic. First, let us note that the number of receivers is the same for all the columns within one part of a separator (S^i_{Lq}, S^i_{Vq}, or S^i_{Rq}). However, different parts of the same separator, in general, have different numbers of receivers, which complicates counting. Thus, we will approximate the number of receivers making the following assumption:

Assumption 2. *Let us assume that the number of receiving processes ("receivers") is the same for all the columns in a cross-separator.*

Clearly, Assumption 2 does not hold in reality. For example, the columns belonging to the left part of S^2 are sent to twice as many (or half as many, depending on which S^2 is chosen) processes as the columns belonging to the right part of the same separator. However, this assumption will allow us to compute the approximate communication volume and, then, possibly, refine the result having removed the assumption.

In order to compute the number of receivers for a particular part of a separator, we need to look at the submesh this part belongs two and find the largest subset of processes assigned to the separators bordering this submesh.

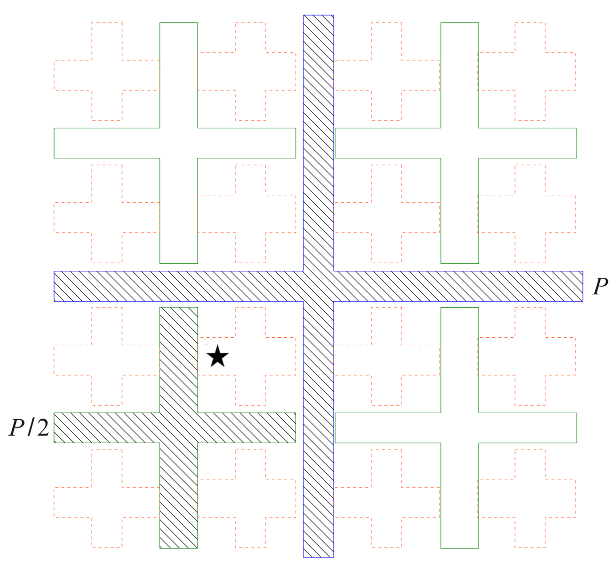

Figure 6: In order to count the number of receivers for the column ★ in a third-level separator, we need to look at all the neighboring separators of its submesh (these separators are hatched) and choose the one with the largest subset of processes. In this case, the largest subset belongs to S^1_1. Thus, the number of receivers for ★ is $|P| = p = 4^\sigma$.

Couting the number of receivers this way, we encounter two more problems. The first problem is that the process of determining neighboring separators for those separators who touch the edges of the original mesh is different from the same process for the separators burried deep in the original mesh and not touching its edges. For example, on Figure 6, the submesh containing the ★ column is bordered from all sides, while another third-level separator, say, at the lower-left corner of the original mesh, is bordered only from two sides, and determining the number of receivers heavily depends on the borders of submeshes.

Thus, to make the computation of the number of receivers tractable, we will use Assumption 1 that the original mesh is bordered. This assumption makes all the separators at all levels of $\mathrm{T}(\boldsymbol{A})$ similar in terms of determining their neighbors. In the context of counting the number of receivers, Assumption 1 corresponds to an underapproximation – those separators that touch the edges of the original mesh are treated as touching some high-level separators surrounding the original mesh. Thus, at this point, the contribution of these separators to the overall parallel communication volume is not counted. However, in future, this assumption can be removed and the communication volume can be refined correspondingly.

The third problem with counting the number of receivers is that some separators are so small that their submeshes touch a very small parts of the neighboring separators. As a result, the touched parts of these neighboring separators may contain columns corresponding only to a part of the whole set of processes assigned to these separators.

For example, in Figure 4, all the separators at the second level touch the first-level separator, but neither of them touches its part big enough to send columns to all the $4^\sigma = 4^2 = 16$ processes assigned to S^1_1. (In fact, in this example the ratio of the number of processes to the size of S^1_1 is so large that the columns of S^1_{V1} use only 15 of 16 processes that should be assigned to S^1_{V1} according to our assignment strategy.)

Thus, we make another assumption:

Assumption 3. *If a separator's submesh touches a higher-level separator, let us think that the columns of the separator in the submesh are to be sent to* all *the processes assigned to this higher-level separator even if only a small part of this higher-level separator is touched by the submesh.*

This third assumption is unlikely to be removed from our analysis in future. However, this is not critical, since this last assumption does not affect the result much.

Let us compute the volume of parallel communication under the conditions of the three made assumptions.

$$
\begin{aligned}
comm &= \sum_{i=1}^{\sigma} \sum_{q=1}^{\#_i} \sum_{l \in S^i_q} \Big(\eta(L^{\llcorner}_{*l}) \times \text{“number of receivers”} \Big) = \Big(\text{by Assumption 1} \Big) = \\
&= \sum_{i=1}^{\sigma} \sum_{q=1}^{\#_i} \left(\text{“number of receivers”} \times \sum_{l \in S^i_q} \eta(L^{\llcorner}_{*l}) \right) = \Big(\text{since } \eta(L^{\llcorner}_{*l}) \text{ does not depend on } q \Big) = \\
&= \sum_{i=1}^{\sigma} \left(\sum_{l \in S^i_*} \eta(L^{\llcorner}_{*l}) \times \sum_{q=1}^{\#_i} \text{“number of receivers”} \right).
\end{aligned}
$$

The first factor in the outer sum is nothing else but the number of non-zeros in the columns of a separator – the number provided in Lemma 4. As to the second factor – the aggregate number of receivers at level i – its computation is still challenging. But there is one important difference from the situation we had before: now we do not need to count the number of receivers individually for every separator in a level. This is fortunate, since finding a sum is easier due to our ability to group summands. In particular, we are going to group the summands of the sum for the number of receivers by the number of receivers.

$$comm = \sum_{i=1}^{\sigma} \left(\sum_{l \in S_*^i} \eta(L_{*l}^{\llcorner}) \times \left[\sum_h |P_h| \times \text{“number of separators } S^i \text{ having } |P_h| \text{ receivers”} \right] \right),$$

where $|P_h|$ is the number of processes assigned to a separator S^h.

Lemma 5. *The number of processes $|P_h|$ assigned to a separator S^h is*

$$|P_h| = 4^{\sigma - h + 1}.$$

Proof. It is enough to notice that (i) the columns of the first-level separator S^1 are assigned to all 4^σ processes, and (ii) the number of processes assigned to every subsequent level decreases four-fold. Thus, we have

$$|P_1| = 4^\sigma,\ |P_2| = \frac{4^\sigma}{4},\ |P_3| = \frac{4^\sigma}{4^2},\ \ldots\ ,\ |P_h| = \frac{4^\sigma}{4^{h-1}} = 4^{\sigma - h + 1}.$$

□

For convenience, let us denote with $\tilde{H}_i^j$ the number of separators S^j at level j whose number of receivers is determined by a separator S^i at level i, that is whose number of receivers equals $|P_i|$. The expression for the parallel communication volume using this new notation will look as follows:

$$comm = \sum_{i=1}^{\sigma} \left(\sum_{l \in S_*^i} \eta(L_{*l}^{\llcorner}) \times \left[\sum_h |P_h| \times \tilde{H}_h^i \right] \right).$$

Notice that, aside from $\tilde{H}_h^i$ having not being computed, the expression for *comm* is still incomplete – the range for h is not specified. However, it is convenient to specify it after an expression for $\tilde{H}_h^i$ has been obtained, which follows.

Lemma 6 (about the number of separators having a particular number of receivers)**.** *The number $\tilde{H}_i^j$ of separators at level j whose number of receivers is determined by a separator at level i (i.e., whose number of receivers is $|P_i|$) is:*

$$\tilde{H}_i^j = \begin{cases} 4^i(2^{j-i} - 3), & j - i > 1, \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. For the proof, let us introduce a utility function h_i^j, that measures the number of vertical separators S_V^j residing along one (say, left) side of a vertical separator S_V^i. In other words, h_i^j

measures the length of S_V^i using the length of S_V^j as a unit. For example, the length of S_V^1 using S_V^2 as a unit is $h_1^2 = 2$.

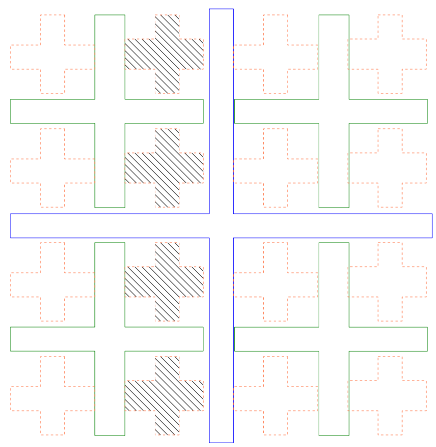

Figure 7: The number of separators of level 3 residing along the left side of a vertical separator of level 1 is $h_1^3 = 4$. The corresponding level-3 separators are hatched.

It is easy to see that

$$h_i^j = \frac{|S_V^i|}{|S_V^j|} = \text{(from Lemma 2)} = 2^{j-i}.$$

Another one utility function H_i^j measures the (*total*) number of separators S^j at level j that touch (*any*) separators S^i at level i. For example, the number of second-level separators touching first-level separator(s) is 4; the number of third-level separators touching second-level separators is 16; the number of fourth-level separators touching second-level separators is 48.

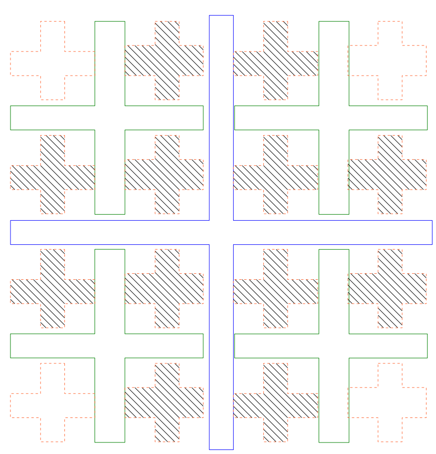

Figure 8: The number of level-3 separators residing along all the level-1 separators (of which there happens to be only one) is $H_1^3 = 12$. The corresponding 3-level separators are hatched.

It is easy to see that

$$H_i^j = \underbrace{|S^i|\times}_{\text{for every } S^i} \left[\underbrace{2h_i^j}_{\substack{\text{number of } S^j \\ \text{along } S_V^i}} + \underbrace{2(h_i^j - 2)}_{\substack{\text{number of } S^j \\ \text{along } S_L^i \text{ and } S_R^i \\ \text{but not along } S_V^i}} \right] = 4^i(2^{j-i} - 1).$$

In order to compute $\tilde{H}_i^j$, let us notice that those separators that should be counted in $\tilde{H}_i^j$ are exactly those counted in H_i^j excluding the separators that touch the edges of the sumbeshes that accomodate S^i.

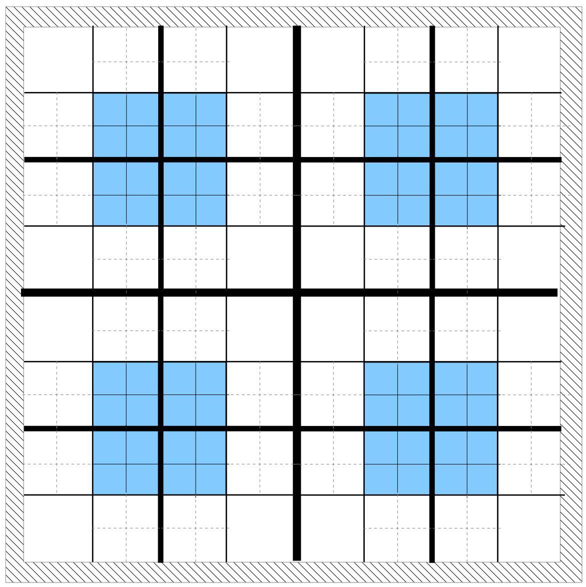

Figure 9: The number of separators of level 4 residing along the separators of level 2 and not touching the edges of the corresponding submeshes is $\tilde{H}_2^4 = 16$. The separators counted in $\tilde{H}_2^4$ are shaded, while the separators counted in H_2^4 and not counted in $\tilde{H}_2^4$ are displayed with dotted lines. Some of these not counted separators have not been counted because they touch the first-level separator; others – because they touch the border of the original mesh.

There are exactly 8 such separators to exclude per every separator S^i (here, we use the assumption that the original mesh is bordered). Thus,

$$\tilde{H}_i^j = H_i^j - 8|S^i| = 4^i(2^{j-i} - 3).$$

Finally, let us take a look at the obtained expression for $\tilde{H}_i^j$. It has positive values only if $j - i \geq 2$, which is reasonable. If $j = i$, then, obviously, there are no separators S^i along the separators S^i; hence, $\tilde{H}_i^i = 0$. If $j = i + 1$, then there are some separators $S^j \equiv S^{i+1}$ along the separators S^i, but all of these S^{i+1} touch the edges of the submeshes that accomodate S^i. Thus, the number of receivers for these S^{i+1} will be determined not by S^i, but by the higher-level neighbors of S^i. For example, in our model with the bordered original mesh, no separators of the second level will use the number of receivers determined by the first-level separator. As a result, we have $\tilde{H}_i^{i+1} = 0$. $\square$

Using Lemmas 4, 5, and 6, we can rewrite the expression for the volume of parallel communication:

$$
\begin{aligned}
comm &= \sum_{i=1}^{\sigma} \left[\sum_{l \in S_*^i} \eta(L_{*l}^{\llcorner}) \times \left(\sum_{h=1}^{i-2} |P_h| \times \tilde{H}_h^i \right) \right] = \\
&= \sum_{i=3}^{\sigma} \left[\left(31 \left(\frac{k}{2^i} \right)^2 - 11 \left(\frac{k}{2^i} \right) + \frac{7}{4} \right) \times \sum_{h=1}^{i-2} \left(4^{\sigma-h+1} \times 4^h (2^{i-h} - 3) \right) \right] = \\
&= 4^{\sigma} \sum_{i=3}^{\sigma} \left[\left(124 \left(\frac{k}{2^i} \right)^2 - 44 \left(\frac{k}{2^i} \right) + 7 \right) \times (2^i - 3i + 2) \right].
\end{aligned}
$$

Notice that on the second line above, the lower bound for i in the outer sum changes from 1 to 3. The reason is that the first two summands contain a factor $\tilde{H}_h^i$, which is 0 for $i = 1$ and $i = 2$, due to the definition of $\tilde{H}_h^i$. The latter fact is a consequence of our treating the original mesh as bordered. Thus, as it manifests here, bordering the original mesh results in not counting parallel communication for the separators of level 1 and 2.

Another side effect of this change in the range of i is that the expression for the volume of parallel communication will work only for $\sigma > 2$, that is, for the number of processes $p > 16$. Thus, using this approximation we will need to extrapolate in order to see how parallel communication behaves when $p \leq 16$.

Finally, the obtained expression for *comm* is computable, and the result is stated as a theorem below.

Theorem. *The volume of parallel communication for the right-looking sparse Cholesky factorization algorithm using coarse analysis (i.e., using Assumptions 1, 2, and 3), is defined as follows:*

$$
\begin{aligned}
comm = 31 \Big[\frac{1}{3}(4^{\sigma} + 8) + 4(\sigma - 2^{\sigma}) \Big] k^2 + \\
+ 22 \Big[(9 - 2\sigma)4^{\sigma} - 2(4 + 3\sigma)2^{\sigma} \Big] k + \\
+ 7 \Big[\frac{\sigma(1 - 3\sigma)4^{\sigma}}{2} + 4^{\sigma}(2^{\sigma+1} - 3) \Big].
\end{aligned}
$$

The plot of *comm* for some selected values of σ is presented below.

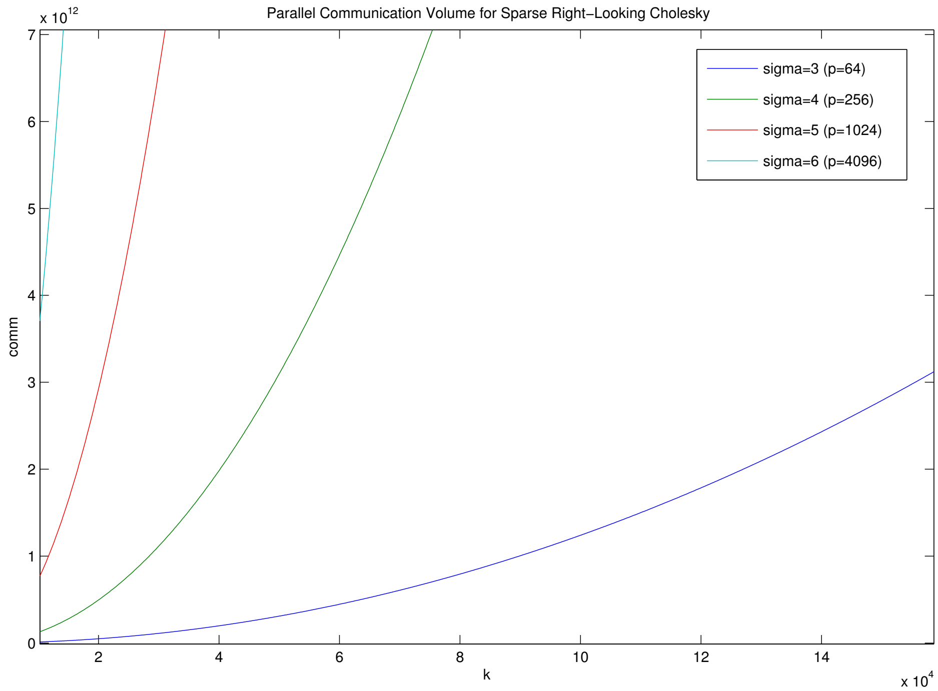

Figure 10: Dependence of the parallel communication volume for sparse right-looking Cholesky factorization algorithm on the mesh size k.

The obtained result for *comm* corresponds to the result obtained in [2]: the highest-order component in our expression is $\frac{31}{3}pk^2$, and the lower bound for *comm* computed in [2] is $O(pk^2)$.

Computing the number of messages sent during parallel communication is very much similar to computing the volume of data to be communicated. Let us recall one of the early expressions for the parallel communication volume:

$$comm = \sum_{i=1}^{\sigma} \left(\sum_{l \in S_*^i} \eta(L_{*l}^{\llcorner}) \times \left[\sum_h |P_h| \times \tilde{H}_h^i \right] \right).$$

Taking into account that in the right-looking Cholesky algorithm, every column to be sent constitute an individual update, that is an individual message, in order to count the number of messages we simply need to count the number of columns to be sent. The latter corresponds to replacing $\eta(L_{*l}^{\llcorner})$ in the expression above with 1. Thus,

$$msg = \sum_{i=1}^{\sigma} \left(\sum_{l \in S_*^i} (1) \times \left[\sum_h |P_h| \times \tilde{H}_h^i \right] \right) = \sum_{i=1}^{\sigma} \left(|S_*^i| \times \left[\sum_h |P_h| \times \tilde{H}_h^i \right] \right).$$

All the quantities in the expression above are known. Thus, *msg* can be computed, and the result follows as a theorem.

Theorem. *The number of messages sent during parallel communication in the sparse right-looking Cholesky factorization algorithm using coarse analysis (e.g., Assumptions 1, 2, and 3), is*

$$msg = 2\Big[(2\sigma - 9)4^{\sigma} + (6\sigma + 8)2^{\sigma}\Big]k +$$

$$+ \Big[4^{\sigma}\left(\frac{9}{2}\sigma^2 + \frac{5}{2}\sigma - 9 - 6\cdot 2^{\sigma}\right) + 2^{\sigma}(12\sigma + 16)\Big].$$

The plot of msg for some selected values of σ is presented below.

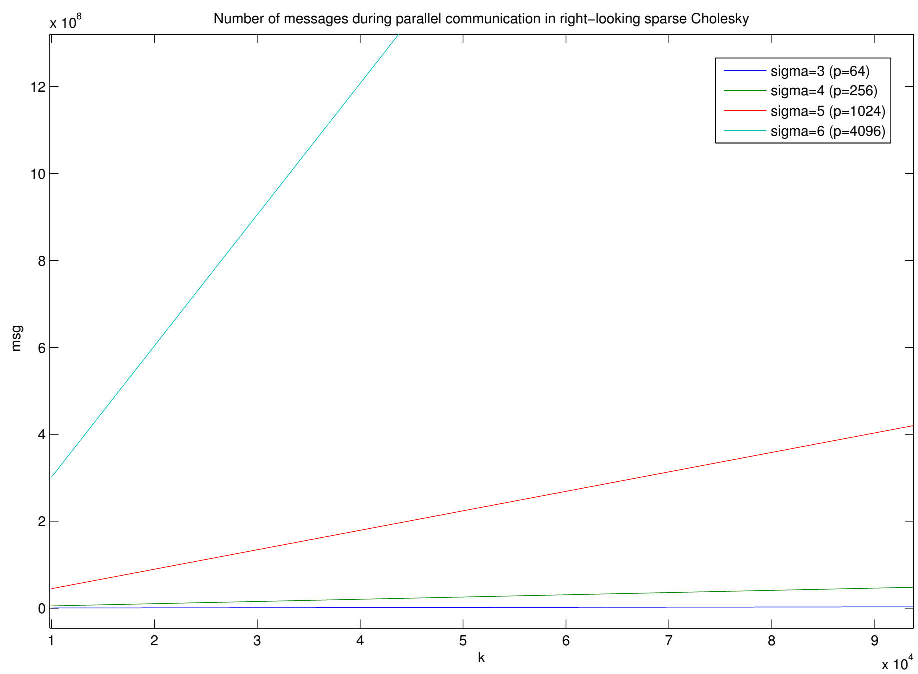

Figure 11: Dependence of the number of messages sent during parallel communication in sparse right-looking Cholesky factorization algorithm on the mesh size k.

We can see that the number of messages is linear relatively to the size k of the original mesh. The highest-order term in the obtained expression for msg is $4\sigma 4^{\sigma}k$, and, thus, $msg = O(p\log_2(p)k)$.

References

[1] Alan George, Joseph Liu, and Esmond Ng. *Computer Solution of Sparse Positive Definite Systems.* A draft from July 6, 2012.

[2] Alan George, Joseph Liu, and Esmond Ng. Communication results for parallel sparse cholesky factorization on a hypercube. *Parallel Computing*, 10(3):287–298, 1989.