1. 一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，包括以下步骤：

根据有限单元集合体构建偏微分方程并使用有限元法进行求解，得到初步刚度矩阵和右矩阵；

对初步刚度矩阵进行分块处理，得到刚度矩阵；

对刚度矩阵和右矩阵进行分解处理，得到分解后的刚度矩阵和分解后的右矩阵；

通过递归分治方法对分解后的刚度矩阵和分解后的右矩阵进行化简处理，并构建简化线性方程组；

对简化线性方程组中的解矩阵进行求解，得到解值。

1. 根据权利要求1所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述根据有限单元集合体构建偏微分方程并使用有限元法进行求解，得到初步刚度矩阵和右矩阵这一步骤，其具体包括：

根据静力等效原则将作用于有限单元集合内的每个有限单元的外力简化至每个有限单元的结点上，得到等效结点力；

根据弹性力学导出在等效结点力作用下结点位移与等效结点力之间的关系，并构建偏微分方程；

基于偏微分方程的系数构建初步刚度矩阵和右矩阵。

1. 根据权利要求2所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述对初步刚度矩阵进行分块处理，得到刚度矩阵这一步骤，其具体包括：

对初步刚度矩阵内的元素进行编号处理，得到带编号的初步刚度矩阵；

对带编号的初步刚度矩阵的性质进行分析，判断带编号的刚度矩阵是否正定；

判断到带编号的刚度矩阵具有正定性质，选取对应的分块方案，输出刚度矩阵。

1. 根据权利要求3所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述对刚度矩阵和右矩阵进行分解处理，得到分解后的刚度矩阵和分解后的右矩阵这一步骤，其具体包括：

对刚度矩阵进行DS分解，得到分解后的刚度矩阵；

对右矩阵进行DY分解，得到分解后的右矩阵。

1. 根据权利要求4所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述DS分解公式具体如下所示：

上式中，表示刚度矩阵的对角分块矩阵，表示单位矩阵，和表示“长钉”矩阵。

1. 根据权利要求4所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述DY分解公式具体如下所示：

上式中，表示DY分解后的右矩阵，表示初始右矩阵。

1. 根据权利要求6所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述通过递归分治方法对分解后的刚度矩阵和分解后的右矩阵进行化简处理，并构建简化线性方程组这一步骤，其具体包括：

基于递归分治方法，对分解后的刚度矩阵进行迭代化简处理，直至刚度矩阵的分区数小于预设阈值，得到最终的刚度矩阵；

基于递归分治方法，对分解后的右矩阵进行迭代化简处理，直至右矩阵的分区数小于预设阈值，得到最终的右矩阵；

整合最终的刚度矩阵和最终的右矩阵，构建简化线性方程组。

1. 根据权利要求7所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述简化线性方程组表示如下：

上式中，表示最终的刚度矩阵，表示最终的右矩阵，表示解矩阵。

1. 根据权利要求8所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述递归分治公式表示如下：

上式中，表示最终的刚度矩阵，表示最终的右矩阵，表示简化后的刚度矩阵，表示简化后的右矩阵，、和表示刚度矩阵从第一个至第个上、下分块矩阵。

1. 根据权利要求9所述一种并行求解有限元问题中大型带状线性方程组的方法，其特征在于，所述对简化线性方程组中的解矩阵进行求解，得到解值这一步骤，其具体包括：

对最终的线性方程组中的解矩阵的上、下分块矩阵值进行求解，得到上分块矩阵值和下分块矩阵值；

根据恢复公式求解解矩阵的中分块矩阵值，得到中分块的矩阵值；

整合上分块的矩阵值、中分块的矩阵值和下分块的矩阵值，得到解值。

**一种并行求解有限元问题中大型带状线性方程组的方法**

**技术领域**

本发明涉及仿真模拟和数学工业软件领域，尤其涉及一种并行求解有限元问题中大型带状线性方程组的方法。

**背景技术**

计算机的问世推进了新一轮的工业革命，利用这一有力的计算工具能够解决一些复杂的力学和物理问题，很长一段时间科学家都在研究一种近似求解微分方程的方法——数值解法，有限单元法就是当前数值解法中较为流行的方法，其基本思想就是将连续的求解区域离散为一组有限个、且按一定方式相互联结在一起的单元组合体，在工程应用方面，气体弹性力学是现代飞行器或飞行在空中的对象研制中必须考虑的一个重要问题.经典空气弹性力学只研究作用于飞行物体结构上的弹性力、惯性力和空气动力间的相互作用，以及由此引起的各种静态、动态稳定性及结构响应间的问题。例如，开启降落伞伞衣过程的研究是气动弹性力学最复杂的问题之一，使用并行计算方法，对流体采用稳定的时空有限元公式，对弹性结构采用源于虚拟工作的有限元公式，之后对有限元公式产生的带状线性方程组进行并行求解，这些单元能按不同的联结方式进行组合，且单元本身又可以有不同形状，所以可模型化几何形状复杂的求解域，但对于大多数这些问题中的方程，它们的某些特征具有非线性性质，而且求解区域的几何形状较为复杂，也就无法得到精确的解析解。

**发明内容**

为了解决上述技术问题，本发明的目的是提供一种并行求解有限元问题中大型带状线性方程组的方法，本发明利用递归分治算法对求解密集带状线性方程组进行求解，能够在不降低求解值准确度的情况下提高密集带状线性方程组的求解效率。

本发明所采用的第一技术方案是：一种并行求解有限元问题中大型带状线性方程组的方法，包括以下步骤：

根据有限单元集合体构建偏微分方程并使用有限元法进行求解，得到初步刚度矩阵和右矩阵；

对初步刚度矩阵进行分块处理，得到刚度矩阵；

对刚度矩阵和右矩阵进行分解处理，得到分解后的刚度矩阵和分解后的右矩阵；

通过递归分治方法对分解后的刚度矩阵和分解后的右矩阵进行化简处理，并构建简化线性方程组；

对简化线性方程组中的解矩阵进行求解，得到解值。

进一步，所述根据有限单元集合体构建偏微分方程并使用有限元法进行求解，得到初步刚度矩阵和右矩阵这一步骤，其具体包括：

根据静力等效原则将作用于有限单元集合内的每个有限单元的外力简化至每个有限单元的结点上，得到等效结点力；

根据弹性力学导出在等效结点力作用下结点位移与等效结点力之间的关系，构建偏微分方程；

基于偏微分方程的系数构建初步刚度矩阵和右矩阵。

进一步，所述对初步刚度矩阵进行分块处理，得到刚度矩阵这一步骤，其具体包括：

对初步刚度矩阵内的元素进行编号处理，得到带编号的初步刚度矩阵；

对带编号的初步刚度矩阵的性质进行分析，判断带编号的刚度矩阵是否正定；

判断到带编号的刚度矩阵具有正定性质，选取对应的分块方案，输出刚度矩阵。

进一步，所述对刚度矩阵和右矩阵进行分解处理，得到分解后的刚度矩阵和分解后的右矩阵这一步骤，其具体包括：

对刚度矩阵进行DS分解，得到分解后的刚度矩阵；

对右矩阵进行DY分解，得到分解后的右矩阵。

进一步，所述DS分解公式具体如下所示：

上式中，表示刚度矩阵的对角分块矩阵，表示单位矩阵，和表示“长钉”矩阵。

进一步，所述DY分解公式具体如下所示：

上式中，表示分块总数。

进一步，所述通过递归分治方法对分解后的刚度矩阵和分解后的右矩阵进行化简处理，并构建简化线性方程组这一步骤，其具体包括：

基于递归分治方法，对分解后的刚度矩阵进行迭代化简处理，直至刚度矩阵的分区数小于预设阈值，得到最终的刚度矩阵；

基于递归分治方法，对分解后的右矩阵进行迭代化简处理，直至右矩阵的分区数小于预设阈值，得到最终的右矩阵；

整合最终的刚度矩阵和最终的右矩阵，构建简化线性方程组。

进一步，所述简化线性方程组表示如下：

上式中，表示最终的刚度矩阵，表示最终的右矩阵，表示解矩阵。

进一步，所述所述递归分治公式表示如下：

上式中，表示最终的刚度矩阵，表示最终的右矩阵，表示简化后的刚度矩阵，表示简化后的右矩阵，、和表示刚度矩阵从第一个至第个上、下分块矩阵。

进一步，所述对简化线性方程组中的解矩阵进行求解，得到解值这一步骤，其具体包括：

对最终的线性方程组中的解矩阵的上、下分块矩阵值进行求解，得到上分块矩阵值和下分块矩阵值；

根据恢复公式求解解矩阵的中分块的矩阵值，得到中分块的矩阵值；

整合上分块的矩阵值、中分块的矩阵值和下分块的矩阵值，得到解值。

本发明方法及系统的有益效果是：本发明通过递归分治算法对密集带状线性方程组进行求解，对密集带状线性方程组进行编码、分块和分解处理，能够在确保解值的准确度的同时，提高密集带状线性方程组的求解效率。

**附图说明**

图1是本发明一种并行求解有限元问题中大型带状线性方程组的方法的步骤流程图；

图2是本发明一种并行求解有限元问题中大型带状线性方程组的方法恢复公式示意图；

图3是本发明在实验测试递归化简阶段刚度矩阵数据之间传递的示意图；

图4是本发明在实验测试递归化简阶段右矩阵数据之间传递的示意图。

**具体实施方式**

下面结合附图和具体实施例对本发明做进一步的详细说明。对于以下实施例中的步骤编号，其仅为了便于阐述说明而设置，对步骤之间的顺序不做任何限定，实施例中的各步骤的执行顺序均可根据本领域技术人员的理解来进行适应性调整。

参照图1，本发明提供了一种并行求解有限元问题中大型带状线性方程组的方法，该方法包括以下步骤：

S1、根据有限单元集合体构建偏微分方程并使用有限元法进行求解，得到初步刚度矩阵和右矩阵；

S11、根据静力等效原则将作用于有限单元集合内的每个有限单元的外力简化至每个有限单元的结点上，得到等效结点力；

S12、根据弹性力学导出在等效结点力作用下结点位移与等效结点力之间的关系，构建偏微分方程；

S13、基于偏微分方程的系数构建初步刚度矩阵和右矩阵。

具体地，参照图1，在使用有限单元法处理弹性力学问题时，得到刚度矩阵的基本步骤是：将一个受力的连续弹性体“离散化”，即将它看作是由一定数量的有限小的单元集合体，认为这些单元之间只在结点上相互联系，即只有结点才能传递力。按静力学等效原则将作用于每个单元的外力简化到结点上去，成为等效结点力。根据弹性力学基本方程，设法导出结点位移和结点力之间的关系；建立每个结点的平衡方程式即为偏微分方程，得到一个以结点位移为基本未知数的偏微分方程组，这就是按位移求解的基本方程，再引入位移边界条件，即在那些可以不计位移的边界点。用线性代数解偏微分方程解得全部未知的结点位移，进而推求各单元的应变、应力。此偏微分方程组的系数矩阵即为刚度矩阵，应力使用刚度矩阵表示，应变使用右矩阵表示。

S2、对初步刚度矩阵进行分块处理，得到刚度矩阵；

S21、对初步刚度矩阵内的元素进行编号处理，得到带编号的初步刚度矩阵；

S22、对带编号的初步刚度矩阵的性质进行分析，判断带编号的刚度矩阵是否正定；

S23、判断到带编号的刚度矩阵具有正定性质，选取对应的分块方案，输出刚度矩阵。

具体地，分析刚度矩阵的性质，例如是否对称，是否正定，编号，对矩阵沿着对角线进行分块，分成块，刚度矩阵规模较小（阶数在10万以下时）使用分块（分区）数在8以下效率较高；刚度矩阵规模较大时，可适当增多分块（分区）数。根据矩阵性质可进行不同的矩阵分解方式，例如，若带编号的刚度矩阵不具有正定性质具有对角占优性质，则可使用不选主元的LU分解效率更高，若带编号的刚度矩阵具有正定性质则可使用cholesky分解，效率更高。

S3、对刚度矩阵和右矩阵进行分解处理，得到分解后的刚度矩阵和分解后的右矩阵；

S31、对刚度矩阵进行DS分解，得到分解后的刚度矩阵；

具体地，根据刚度矩阵的大小和性质，选择合适的分块方案。

首先，沿着的对角线进行分块，如下：

上式中，表示刚度矩阵内的分块矩阵大小为，表示刚度矩阵内的分块矩阵大小为，表示刚度矩阵带宽为；

其结构如下：

上式中，，表示密集方阵，大小为。

对进行分解：

上式中，表示刚度矩阵的对角分块矩阵，表示单位矩阵，和表示“长钉”矩阵。

S32、对右矩阵进行DY分解，得到分解后的右矩阵。

具体地，对右矩阵进行分解：

上式中，表示分块总数。

如此，求解就转换为求解如下的

因为是解耦的，所以每个分区可以独立地进行处理如下公式：

上式中，表示DY分解后的右矩阵，表示对角分块矩阵，表示初始右矩阵。

这里比较巧妙的是，对每个都可以进行如下分块：

上式中，表示“钉子”矩阵，表示解矩阵，表示右矩阵， 表示上分块（t表示top），表示下分块（b表示bottom），上下分块大小均为，，为中间分块，大小均为。

这样，可由如下表达式计算：

上式中， 表示上分块（t表示top），表示下分块（b表示bottom），上下分块大小均为，，为中间分块，大小均为。

在下一递归化简阶段，为减少各个分区节点的计算量，首先可以忽略占比较大的中间分块、，就转换为如下的简化表达式：

S4、通过递归分治方法对分解后的刚度矩阵和分解后的右矩阵进行化简处理，并构建简化线性方程组；

S41、基于递归分治方法，对分解后的刚度矩阵进行迭代化简处理，直至刚度矩阵的分区数小于预设阈值，得到最终的刚度矩阵；

具体地，利用递归分治方法对分解后的刚度矩阵再次进行化简，直到分区数简化到为2为止。

如下为去除中间分块后，初始第一递归层的简化方程组，这里举一个分区数为4的例子：

上式中，第一层递归的右矩阵，，，，，表示第一层递归的上下分块。

如下对进行化简，得到分区数为一半的：

对于当前递归层的每一个分区都有：

每个分区的交界处，称之为简化系统；计算当前递归层的每一个分区可得到的“长钉”矩阵和，其中每个分区、和的大小都相同，至此得到下一递归层的简化矩阵，如果的分区数较多时，递归层数也就更多，当化简到，分区数为2时，递归终止，得到最终的刚度矩阵，如下表所示为递归化简分解后的刚度矩阵的进程处理分配表：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **进程**  **递归 分块** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **第一层** |  | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** |
|  | **×** | **×** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** | **×** | **√** |
| **第二层** |  | **×** | **√** | **×** | **×** | **×** | **√** | **×** | **×** | **×** | **√** | **×** | **×** | **×** | **×** | **×** | **×** |
|  | **×** | **×** | **×** | **×** | **×** | **×** | **√** | **×** | **×** | **×** | **√** | **×** | **×** | **×** | **√** | **×** |
| **第三层** |  | **×** | **×** | **×** | **√** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** |
|  | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **×** | **√** | **×** | **×** | **×** | **×** |

S42、基于递归分治方法，对分解后的右矩阵进行迭代化简处理，直至右矩阵的分区数小于预设阈值，得到最终的右矩阵；

具体地，同理，对于分解后的右矩阵*Y*也使用相同的递归分治方法：

对于当前递归层的每一个分区都有：

如下表所示为递归化简分解后的右矩阵*Y*的进程处理分配表：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 进程  递归 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 第一层 | √ | × | √ | × | √ | × | √ | × | √ | × | √ | × | √ | × | √ | × |
| 第二层 | × | √ | × | × | × | √ | × | × | × | √ | × | × | × | √ | × | × |
| 第三层 | × | × | × | ✓ | × | × | × | × | × | × | × | ✓ | × | × | × | × |
| 第四层 | × | × | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × |

可以发现递归化简和的操作是相似的，并且每一层的分区处理进程并不都是“忙碌”的，有很大一部分是“闲置”的，一个进程只会在某个递归层处理一个分区，同一层大半部分进程是处于闲置状态的，由于递归操作相似，为了进一步提升进程资源的利用率，所以递归化简矩阵和的操作可以合并，特别的，当时，化简矩阵和的计算量是接近的，那么递归化简阶段的算法效率将达到最高。例如，当进入第一层时，进程1，3，5，7，9，11，13，15为“忙碌”状态计算，其他进程闲置，此时若将两个递归阶段合并，那么进程2，4，6，8，10，12，14，16也会进入“忙碌”状态计算，如此就会大大提升各进程的利用率。

S43、整合最终的刚度矩阵和最终的右矩阵，构建简化线性方程组。

具体地，整合最终的刚度矩阵和最终的右矩阵，得到解矩阵并构建简化线性方程组，所述简化线性方程组如下表示：

同样也可以表示为：

S5、对简化线性方程组中的解矩阵进行求解，得到解值。

S51、对最终的线性方程组中的解矩阵的上、下分块矩阵值进行求解，得到上分块矩阵值和下分块矩阵值；

具体地，求解简化解矩阵每个分区的上、下分块：

S52、根据恢复公式求解解矩阵的中分块的矩阵值，得到中分块的矩阵值；

具体地，参照图2，求解解矩阵每个分区的中间分块，在得到各个矩阵每个分区的上、下分块之后，就是利用如下公式求解解矩阵每个分区的中间分块，继而得到全局解矩阵*X*：

上式中，、、表示解矩阵分块，、、表示右矩阵分块，大小为；表示对角分块，大小为；，分别表示解矩阵第i+1个分块的上分块、第i-1个分块的下分块，表示第个对角分块的逆。

对以上公式进一步化简得到如下公式表示：

上式中， 表示上分块（t表示top）；表示下分块（b表示bottom），上下分块大小均为；，为中间分块，大小均为；，，，，，，，，分别表示右矩阵第*i*（*i=*1,2,3*,…,p*）个分块的上、中、下分块，表示分块的非零区域，表示分块的非零区域，、、、代表，分解后的上、下三角矩阵的逆。

以上公式为恢复阶段的简化形式，对于前个分区，对使用分解，则求解钉矩阵有：

上式中，，代表分解后的上、下三角矩阵的逆，表示钉子矩阵的第个分块；表示分块的非零区域。

而在最后一个分区，对使用分解，则求解钉矩阵有：

上式中，，代表分解后的上、下三角矩阵的逆；表示钉子矩阵的第个分块；表示分块的非零区域。

以上在不同分区对进行不同的分解方式的目的是：在求解和，不需要将、全部加载计算。

S53、整合上分块的矩阵值、中分块的矩阵值和下分块的矩阵值，得到解值。

具体地，利用如上公式，每个节点进程并行处理分配给自己的分区，得到每个分区的局部解向量，之后使用消息传递接口（MPI）进行节点间通信，在主进程中输出全局解向量，即解值。

基于以上本发明内容进行以下实验进行测试：

本实施例在天河二号超级计算机上运行测试，对比本发明方法和ScaLAPACK中的方法求解带状线性方程组的计算效率。其中在求解对角占优方程组时，将本发明不选主元的算法和ScaLAPACK中的pddbsv例程进行对比；在求解非对角占有方程组时，将本发明不选主元的算法和ScaLAPACK中的pdgbsv例程进行对比。

另一个对比就是：合并递归化简矩阵和的操作的算法和分离递归化简矩阵和的操作的算法进行比较。

本实例运行测试在天河二号WORK集群中的常规计算节点，，每个计算结点包含 2 个Intel Xeon E5-2692 v2 12核心的多核中央处理器（CPU），主频2.2GHz。每个计算结点拥有64GB内存（2个CPU 共用）。

运行测试的基准案例为n=1920000（矩阵阶数），b=401（矩阵带宽），nrhs=200的对角占优的带状线性方程组，分别使用p=4，8，16，32，64（进程/分区数）对算法进行测试。

在每个分区，本发明方法在进行DS、DY分解阶段计算“长钉”矩阵时，使用dgbalu例程对进行分解（若使用选主元的算法，则使用LAPACK的dgbtrf进行分解），使用dtbsm例程对三角线性方程组进行求解，若无需将全部三角线性方程组加载计算，也可使用该例程进行简化计算。

测试结果表明，在进程/分区数为16（8），32（16），64（32），128（64），256（128）的情况下（其中，括号内为分配的WORK集群节点数），并行运算的时间如下表：

|  |  |  |
| --- | --- | --- |
| 时间  进程数（节点数） | 递归SPIKE | ScaLAPACK  （pddbsv） |
| 4（2） | 124.9283s | 202.6931s |
| 8（4） | 77.3557s | 138.0912s |
| 16（8） | 44.6096s | 69.7490s |
| 32（16） | 27.4514s | 38.4871s |
| 64（32） | 20.3144s | 22.2468s |

对比ScaLAPACK，递归SPIKE算法的计算效率更高。理论上，递归SPIKE算法的扩展性较ScaLAPACK更高，但由于天河二号WORK集群和算法代码实现的问题，其节点间的通信时间稍大于计算时间，但在计算进程较少的情况下，递归SPIKE的算法效率更高。

参照图3和图4，在递归化简阶段，细化递归层间节点数据通信按相应的进程编号编号进行数据传递，进程间通信的传送与接收使用MPI的MPI\_Send和MPI\_Recv。

上述方法实施例中的内容均适用于本系统实施例中，本系统实施例所具体实现的功能与上述方法实施例相同，并且达到的有益效果与上述方法实施例所达到的有益效果也相同。

以上是对本发明的较佳实施进行了具体说明，但本发明创造并不限于所述实施例，熟悉本领域的技术人员在不违背本发明精神的前提下还可做作出种种的等同变形或替换，这些等同的变形或替换均包含在本申请权利要求所限定的范围内。

本发明公开了一种并行求解有限元问题中大型带状线性方程组的方法及系统，该方法包括：根据有限单元集合体构建偏微分方程并使用有限元法进行求解；对初步刚度矩阵进行分块处理，得到刚度矩阵；对刚度矩阵和右矩阵进行分解处理，得到分解后的刚度矩阵和分解后的右矩阵；通过递归分治方法对分解后的刚度矩阵和分解后的右矩阵进行化简处理，并构建简化线性方程组；对简化线性方程组中的解矩阵进行求解，得到解值。通过使用本发明，在确保解值的准确度下，进一步提高密集带状线性方程组的求解效率。本发明作为一种并行求解有限元问题中大型带状线性方程组的方法及系统，可广泛应用于仿真模拟和数学工业软件领域。

指定说明书附图中的图1为摘要附图

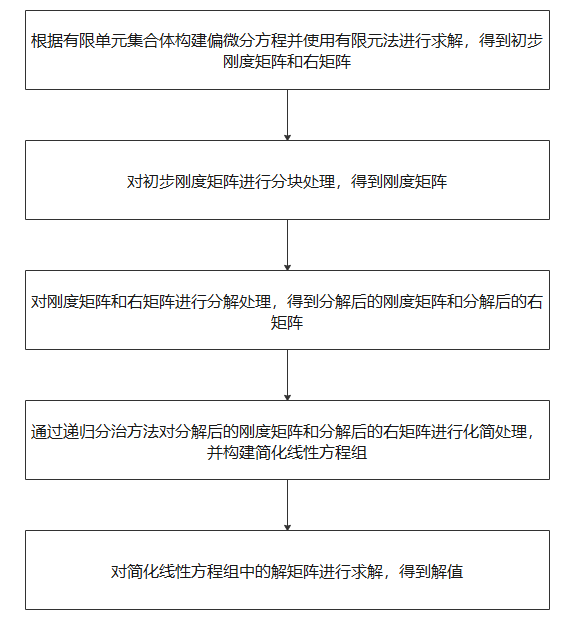


图1

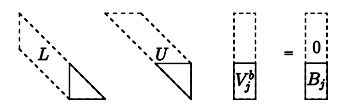


图2

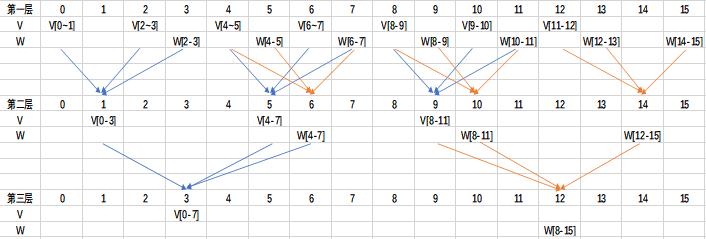


图3

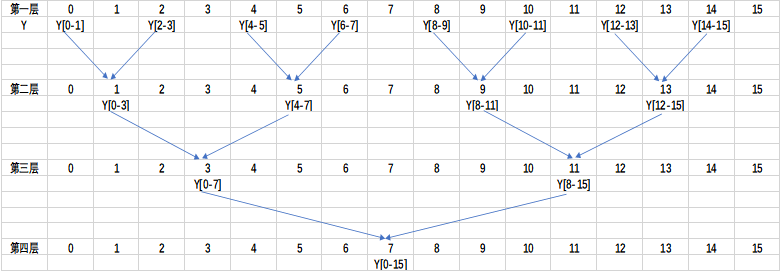


图4