(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102362274 A
(43)申请公布日 2012.02.22

(21)申请号 201080013417.1

(22)申请日 2010.03.03

(30)优先权数据
09156017.7 2009.03.24 EP

(85)PCT申请进入国家阶段日
2011.09.23

(86)PCT申请的申请数据
PCT/IB2010/050912 2010.03.03

(87)PCT申请的公布数据
WO2010/109359 EN 2010.09.30

(71)申请人 国际商业机器公司
地址 美国纽约

(72)发明人 K·贝卡斯 A·库利奥尼

(74)专利代理机构 北京市中咨律师事务所 11247
代理人 于静 杨晓光

(51)Int.Cl.
G06F 17/12(2006.01)

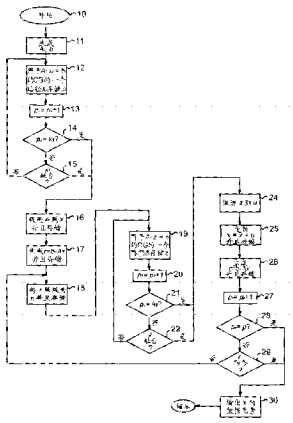
权利要求书 3 页 说明书 8 页 附图 3 页

(54)发明名称

方程的线性系统的处理

(57)摘要

提供用于生成与满足 Ax = b 的 n×1 向量 x 相对应的 n 个高精度数据元素的设备和计算机程序，其中 A 是与 n×n 个预定的高精度数据元素相对应的对称的、正定 n×n 矩阵并且 b 是与 n 个预定的高精度数据元素相对应的 n×1 向量。设备 (1) 包括：存储器 (3)，用于存储定义所述矩阵 A 和向量 b 的数据元素的输入数据，以及控制逻辑 (2)。在第一处理步骤 (a)，控制逻辑 (2) 实现用于通过所述输入数据生成与满足 $A_1x_1 = b_1$ 的 n×1 向量 x_1 相对应的 n 个低精度数据元素的第一迭代处理。此处，A_1 是与矩阵 A 的 n×n 数据元素相对应的低精度 n×n 矩阵并且 b_1 是与向量 b 的 n×1 个数据元素相对应的低精度 n×1 向量。控制逻辑 (2) 在出现第一收敛条件时控制逻辑 (2) 终止第一迭代处理。在步骤 (b) 控制逻辑将向量 x_1 的数据元素转换成高精度数据元素以获得当前的解向量 x。在步骤 (c)，控制逻辑 (2) 实现用于生成与基于向量 b 和向量积 Ax 之间的差异的 n×1 纠正向量相对应的 n 个低精度数据元素的第二迭代处理。控制逻辑 (2) 在出现第二收敛条件时控制逻辑终止第二迭代处理。在步骤 (d)，控制逻辑 (2) 从所述纠正向量的 n 个低精度数据元素产生 n×1 更新向量 u 的各个高精度数据元素，并且于是在步骤 (4)，更新所述当前解向量 x 的数据元素，使得 x = x+u。控制逻辑 (2) 执行步骤 (c) 到 (e)，直到出现第三收敛条件。



CN 102362274 A

1. 一种设备 (1)，用于生成与满足 Ax ＝ b 的 n×1 向量 x 相对应的 n 个高精度数据元素，其中 A 是与 n×n 个预定的高精度数据元素相对应的对称的、正定 n×n 矩阵并且 b 是与 n 个预定的高精度数据元素相对应的 n×1 向量，设备 (1) 包括 ：存储器 (3)，用于存储定义所述矩阵 A 和向量 b 的数据元素的输入数据，以及控制逻辑 (2)，适用于 ：

(a) 实现用于从所述输入数据生成与满足 $A_1x_1 = b_1$ 的 n×1 向量 x_1 相对应的 n 个低精度数据元素的第一迭代处理 ；其中 A_1 是与矩阵 A 的 n×n 个数据元素相对应的低精度 n×n 矩阵并且 b_1 是与向量 b 的 n×1 个数据元素相对应的低精度 n×1 向量，在出现第一收敛条件时控制逻辑 (2) 终止第一迭代处理 ；

(b) 将向量 x_1 的数据元素转换成高精度数据元素以获得当前的解向量 x ；

(c) 实现用于生成与基于向量 b 和向量积 Ax 之间的差异的 n×1 纠正向量相对应的 n 个低精度数据元素的第二迭代处理，在出现第二收敛条件时控制逻辑终止第二迭代处理 ；

(d) 从所述纠正向量的 n 个低精度数据元素产生 n×1 更新向量 u 的各个高精度数据元素 ；

(e) 更新所述当前解向量 x 的数据元素，使得 x ＝ x+u ；以及

(f) 执行步骤 (c) 到 (e)，直到出现第三收敛条件。

2. 根据权利要求 1 所述的设备 (1)，其中所述控制逻辑 (2) 适于 ：

在步骤 (b) 中生成当前解向量 x 之后，生成与指示向量 b 和向量积 Ax 之间的差异的当前 n×1 误差向量 r 相对应的 n 个数据元素 ；

执行步骤 (c)，使得具有所述纠正向量的矩阵 A_1 的向量积依赖于误差向量 r ；以及

在步骤 (d)，通过将纠正向量的数据元素转换成高精度数据元素来生成更新向量 u 的数据元素。

3. 根据权利要求 2 所述的设备 (1)，其中所述控制逻辑 (2) 适于 ：

生成高精度的误差向量 r 的数据元素，使得 r ＝ b-Ax ；

将误差向量 r 的数据元素转换成各个低精度数据元素以获得低精度误差向量 r_1 ；

执行步骤 (c)，使得具有所述纠正向量的矩阵 A_1 的向量积等于低精度误差向量 r_1 ；以及

在步骤 (e)，在更新当前解向量 x 的数据元素之后，更新当前误差向量 r 的数据元素使得 r ＝ b-Ax ；

其中第三收敛条件依赖于当前误差向量 r。

4. 根据前述权利要求中任意一项所述的设备 (1)，其中第一收敛条件依赖于以下内容的至少一个，或第一次出现 ：

完成预定数量的第一迭代处理的路径 ；

实现向量 x_1 对预定公差的解 ；以及

第一迭代处理的连续路径中未检测到用于向量 x_1 的解的改变。

5. 根据前述权利要求中任意一项所述的设备 (1)，其中第二收敛条件依赖于以下内容的至少一个，或第一次出现 ：

完成预定数量的第二迭代处理的路径 ；

实现纠正向量对预定公差的解 ；以及

第一迭代处理的连续路径中未检测到用于纠正向量的解的改变。

6. 根据前述权利要求中任意一项所述的设备 (1)，其中第三收敛条件依赖于以下内容

的至少一个，或第一次出现：

完成预定数量的步骤 (c) 到 (e) 的路径；

实现用于基于当前解向量 x 的向量对预定公差的解；以及

步骤 (c) 到 (e) 的连续路径中未检测到用于依赖于当前解向量 x 的所述向量的解的改变。

7. 根据权利要求 6 和 3 所述的设备 (1)，其中依赖于当前解向量 x 的所述向量包括当前误差向量 r。

8. 根据前述权利要求中任意一项所述的设备 (1)，其中定义矩阵 A 的数据元素的所述输入数据包括定义在任意 n×1 向量的元素上每个矩阵 A 元素 a(i，j) 的应用的函数 F，其中 1 ≤ i ≤ n 并且 1 ≤ j ≤ n 分别是矩阵 A 元素的行和列索引，控制单元 (2) 适于使用所述函数 F 来执行步骤 (a) 到 (e)。

9. 根据前述权利要求中任意一项所述的设备 (1)，其中第一迭代处理包括共轭梯度方法。

10. 根据前述权利要求中任意一项所述的设备 (1)，其中第二迭代处理包括共轭梯度方法。

11. 根据前述权利要求中任意一项所述的设备 (1)，其中控制逻辑 (2) 包括多个处理器，其被安排共同地并行操作以实现步骤 (a) 到 (e)。

12. 一种计算机程序，用于使计算机生成与满足 Ax ＝ b 的 n×1 向量 x 相对应的 n 个高精度数据元素，其中 A 是与 n×n 个预定的高精度数据元素相对应的对称的、正定 n×n 矩阵并且 b 是与 n 个预定的高精度数据元素相对应的 n×1 向量，计算机程序包括程序代码装置，用于促使计算机访问输入数据，其存储于计算机的存储器 (2) 中并且定义所述矩阵 A 和向量 b 的数据元素；以及：

(a) 实现用于从所述输入数据生成与满足 $A_1x_1 = b_1$ 的 n×1 向量 x_1 相对应的 n 个低精度数据元素的第一迭代处理；其中 A_1 是与矩阵 A 的 n×n 个数据元素相对应的低精度 n×n 矩阵并且 b_1 是与向量 b 的 n×1 个数据元素相对应的低精度 n×1 向量，在出现第一收敛条件时终止第一迭代处理；

(b) 将向量 x_1 的数据元素转换成高精度数据元素以获得当前的解向量 x；

(c) 实现用于生成与基于向量 b 和向量积 Ax 之间的差异的 n×1 纠正向量相对应的 n 个低精度数据元素的第二迭代处理，在出现第二收敛条件时终止第二迭代处理；

(d) 从所述纠正向量的 n 个低精度数据元素产生 n×1 更新向量 u 的各个高精度数据元素；

(e) 更新所述当前解向量 x 的数据元素，使得 x ＝ x+u；以及

(f) 执行步骤 (c) 到 (e)，直到出现第三收敛条件。

13. 根据权利要求 12 所述的计算机程序，包括程序代码装置，用于促使计算机：

在步骤 (b) 中生成当前解向量 x 之后，生成与指示向量 b 和向量积 Ax 之间的差异的当前 n×1 误差向量 r 相对应的 n 个数据元素；

执行步骤 (c)，使得具有所述纠正向量的矩阵 A1 的向量积依赖于误差向量 r；以及

在步骤 (d)，通过将纠正向量的数据元素转换成高精度数据元素来生成更新向量 u 的数据元素。

14. 根据权利要求 13 所述的计算机程序，包括程序代码装置，用于促使计算机：

生成高精度的误差向量 r 的数据元素，使得 r = b-Ax；

将误差向量 r 的数据元素转换成各个低精度数据元素以获得低精度误差向量 r_1；

执行步骤 (c)，使得具有所述纠正向量的矩阵 A_1 的向量积等于低精度误差向量 r_1；以及

在步骤 (e)，在更新当前解向量 x 的数据元素之后，更新当前误差向量 r 的数据元素使得 r = b-Ax；

其中第三收敛条件依赖于当前误差向量 r。

15. 根据权利要求 12 到 14 中任意一项所述的计算机程序，其中

第一收敛条件依赖于以下内容的至少一个，或第一次出现：完成预定数量的第一迭代处理的路径；实现向量 x_1 对预定公差的解；以及第一迭代处理的连续路径中未检测到用于向量 x_1 的解的改变。

第二收敛条件依赖于以下内容的至少一个，或第一次出现：完成预定数量的第二迭代处理的路径；实现纠正向量对预定公差的解；以及第一迭代处理的连续路径中未检测到用于纠正向量的解的改变。

第三收敛条件依赖于以下内容的至少一个，或第一次出现：完成预定数量的步骤 (c) 到 (e) 的路径；实现用于基于当前解向量 x 的向量对预定公差的解；以及步骤 (c) 到 (e) 的连续路径中未检测到用于依赖于当前解向量 x 的所述向量的解的改变。

方程的线性系统的处理

技术领域

[0001] 本发明一般涉及方程的线性系统的处理。提供用于方程的线性系统的混合精度处理以生成高精度解的设备和计算机程序。

背景技术

[0002] 现代的处理器典型地能够以高精度和低精度执行处理操作。精度确定可以用于表示浮点数的小数部分的位数。此处的术语“高”和“低”简单地用于区分两种不同的精度等级（以及由此确定“小数位”的数量），一个等级比另一等级要高，但是并不暗示在各个精度等级上进行任意特定的限制。因此，在低和高精度中使用的小数位的实际数量可能从一个系统到另一系统进行实质上的变化。例如，当前的 IEEE 标准为低精度（也称为“单精度”）处理指定 32 位，等于在小数点之后具有 8 个小数位，以及为高精度（或“双精度”）处理指定 64 位，等于在小数点之后具有 16 个小数位。然而，许多嵌入式系统使用不同的规范，例如 8 位和 16 位，或 10 和 20 位。

[0003] 根据处理器的类型，由专用处理逻辑或由相同处理器硬件的适当软件控制来实现低精度和高精度操作。任意一种方式，低精度操作不太复杂并且显著地快于高精度操作。在复杂任务需要高精度结果时，因此可以使用“混合 - 精度”方法。通过混合精度处理，可以按低精度执行任务的一些部分，并且按高精度执行其他部分，而获得高精度的整体结果。作为多种科学和工程应用的基础的这种处理任务的实例是方程的线性系统的解。这个任务需要处理器生成与维度 n×1 的向量 x 的元素相对应的 n 个高精度数据元素，使得

[0004] $Ax = b$

[0005] 其中 A 是维度 n×n 的对称正定矩阵并且 b 是维度 n×1 的向量。通过必须被存储在存储器中并且在需要处理操作时被存取的 n×n 高精度数据元素来定义矩阵 A。右侧向量 b 类似地由存储在系统存储器中的 n 个高精度数据元素来定义。当矩阵 A 是稠密的，生成与解向量 x 相对应的高精度数据元素的任务是高度处理器密集的。特别地，当系数矩阵 A 的所有元素非零，任务需要处理器中的多个算术操作，其随着矩阵 A 的维度 n 立方地增长。

[0006] 此外，上述任务的混合精度方法已经基于矩阵的分解（变换）。处理操作的基本步骤如下。首先，必须通过 n×n 高精度数据元素来构建矩阵 A 并且矩阵 A 必须存储在系统存储器中。于是降级（四舍五入）矩阵 A 在系统存储器中产生低精度拷贝 A_1。这是通过适当的四舍五入处理将矩阵 A 的高精度数据元素转换成各个低精度数据元素来完成的。接下来，处理器实现分解过程，其中矩阵 A_1 被分解为：

[0007] $A_1 = LL^T$

[0008] 其中 L 是下三角矩阵并且 L^T 表示其移相。通过乔列斯基 (Cholesky) 分解来执行变换。这种技术是现有技术中公知的并且不需要在此处详细地讨论。重要的是要注意，当可以通过低精度硬件来实现分解时，这种分解的成本随着维度 n 的立方来增长。为生成具有系数矩阵 L 和 L^T 的线性系统的向量解的数据元素的下面的处理于是需要随着矩阵维度 n 二次方增长的成本。这是通过使用如下的迭代细化方法来完成的。

[0009] 最初，通过求解 $L(L^Tx_1) = b_1$ 来获得用于 Ax = b 的低精度合适解向量 x_1，其中 b_1 是具有与低精度向量 b 的 nx1 数据元素相对应的元素的 nx1 向量。通过适当的转换过程向各个高精度数据元素促进向量 x1 的低精度数据元素，以获得当前解向量 x。这可以通过多种方式来完成，例如通过在每个例子中选择最接近的高精度值。于是，对应于当前 nx1 的剩余（误差）向量 r，生成 n 个高精度数据元素，使得

[0010] r = b-Ax

[0011] 处理器于是执行迭代处理直到收敛：

[0012] (1) 生成与线性系统 $L(L^Tz) = r_1$ 中的 nx1 向量 z 相对应的低精度数据元素，其中 r_1 对应于转换为低精度的当前误差向量 r；

[0013] (2) 将向量 z 的数据元素转换成高精度元素以获得高精度向量 z；

[0014] (3) 更新当前高精度解向量 x 的数据元素使得 $x = x+z_h$；

[0015] (4) 更新当前高精度误差向量 r 的数据元素使得 r = b-Ax；

[0016] (5) 重复步骤 1 到 4 直到检测到收敛（典型地，当 r 足够或不进行处理）。

[0017] 在上面的过程中，乔列斯基分解需要在外部的存储器中形成的矩阵 A。然后，在接下来的操作中每次计算高精度误差向量 r 时由处理器从存储器提取这个矩阵。在典型的应用中，矩阵 A 可以非常大，例如维度 n = 10,000 或甚至更大，导致在处理器和存储器子系统之间的相当大的业务量。由于所要求的处理器协作等级以及现有技术中没有很好的等级，矩阵变换过程难以在并行处理环境中实现。如上面注意的之外，整体复杂度仍然与矩阵 A 的维度 n 成立方关系。这些和其它问题限制整体处理效率并且会限制能够处理许多应用的处理要求的硬件类型。实际上，仅立方复杂度就限制了通过基于当前单个和并行处理器的计算系统可以处理的问题的大小。

发明内容

[0018] 本发明的一个方面提供了一种设备，用于生成与满足 Ax = b 的 n×1 向量 x 相对应的 n 个高精度数据元素，其中 A 是与 n×n 个预定的高精度数据元素相对应的对称的、正定 n×n 矩阵并且 b 是与 n 个预定的高精度数据元素相对应的 n×1 向量。设备包括：存储器，用于存储定义所述矩阵 A 和向量 b 的数据元素的输入数据，以及控制逻辑，适用于：

[0019] (a) 实现用于从所述输入数据生成与满足 $A_1x_1 = b_1$ 的 n×1 向量 x_1 相对应的 n 个低精度数据元素的第一迭代处理；其中 A_1 是与矩阵 A 的 n×n 个数据元素相对应的低精度 n×n 矩阵并且 b_1 是与向量 b 的 n×1 个数据元素相对应的低精度 n×1 向量，在出现第一收敛条件时控制逻辑终止第一迭代处理；

[0020] (b) 将向量 x_1 的数据元素转换成高精度数据元素以获得当前的解向量 x；

[0021] (c) 实现用于生成与基于向量 b 和向量积 Ax 之间的差异的 n×1 纠正向量相对应的 n 个低精度数据元素的第二迭代处理，在出现第二收敛条件时控制逻辑终止第二迭代处理；

[0022] (d) 从所述纠正向量的 n 个低精度数据元素产生 n×1 更新向量 u 的各个高精度数据元素；

[0023] (e) 更新所述当前解向量 x 的数据元素，使得 x = x+u；以及

[0024] (f) 执行步骤 (c) 到 (e)，直到出现第三收敛条件。

[0025] 替代现有方法的基于乔列斯基分解的技术，本发明的实施方式实现用于在步骤(a)中生成解向量 x_1 的低精度数据元素的迭代处理，并且还实现用于生成纠正向量（步骤(c)）的低精度数据元素的迭代处理，其用于更新当前高精度解向量 x。在这个过程中不需要矩阵变换，但是优势是仍然可以获得低精度处理的加速性能。每个迭代细化步骤引起对于矩阵大小 n 的平方成本，仅在与现有系统的立方成本相比较的具有矩阵大小 n 的平方的最坏的方式时，处理操作的整体成本增加。此外，可以仅使用矩阵 A 与另一向量（步骤(a)中的 x_1 和步骤(c)中的纠正矢量）的矩阵矢量积来实现整个处理操作。这样避免了与上面讨论的基于变换的处理相关联的并行处理问题。实际上，基于矩阵向量积的操作特别愿意并行实现，其中本发明的实施方式可以按大规模并行的实现方式来实现。除了这样之外，矩阵 A 的形成并不是处理设备的操作的前提条件。如上所解释的，在现有技术中，矩阵 A 必须首先被构建在存储器中并且然后在每次迭代中为生成误差向量 r 而从存储器中获取矩阵 A。相对的，甚至在不形成矩阵 A 的情况，也可以生成下本方面实施方式中所需要的向量积。特别地，常见的做法是将矩阵 A 定义为某个函数，其中具有 A 的矩阵向量积的计算是简单的且计算便宜的。这将在下面进行进一步介绍，但是由于可以避免在每次迭代中从存储器装载所执行的矩阵，效果是实质上简化处理并且显著地降低存储子系统的业务量。

[0026] 通过使用步骤(a)和(c)中的迭代处理进一步增强操作效率。在之前的系统中，获得对于迭代处理中每个路径中的特定精度的基于分解 LL_T 的线性系统的解。例如，用于 z 的 $L(L^Tz) = r_1$ 的解涉及固定数量的步骤并且产生定义的精度的向量 z，即具有与可用于低精度处理的比特的数量而计算的值相应的数据元素。这样增加了比迭代细化要求的理论属性更多的限制精度要求。在出现预定的收敛条件时，控制逻辑终止上述步骤(a)和(c)的每个迭代处理，并且这样允许本发明的实施方式自动地适应所需要的精度。收敛条件典型地定义为预定的最大数量的解的收敛或迭代的完成（实现了根据预定公差的解，或检测迭代之间没有进展）。因此，不仅是迭代处理能够使用快速、低精度的逻辑，还实现这些过程立即终止所需要的精度的解。这样是无法利用标准的基于矩阵因素分解的系统来实现的。

[0027] 虽然控制逻辑可能适于以各种方式执行步骤(c)和(d)，这些步骤优选地基于误差向量 r，其依赖于向量 b 和向量积 Ax 之间的差异，并且特别地基于具有 A 的矩阵向量积依赖于误差向量 r 的纠正向量的迭代生成。因此，在优选的实施方式中，控制逻辑适于：

[0028] 在步骤(b)中生成当前解向量 x 之后，生成与指示向量 b 和向量积 Ax 之间的差异的当前 n×1 误差向量 r 相对应的 n 个数据元素；

[0029] 执行步骤(c)，使得具有所述纠正向量的矩阵 A_1 的向量积依赖于误差向量 r；以及

[0030] 在步骤(d)，通过将纠正向量的数据元素转换成高精度数据元素来生成更新向量 u 的数据元素。

[0031] 优选地，通过控制逻辑以高精度生成误差向量 r 的数据元素使得 $r = b-Ax$，以及于是将误差向量 r 的数据元素转换成各个低精度数据元素以获得低精度误差向量 r_1。这样于是可以用于步骤(c)的第二迭代处理，使得具有纠正向量的矩阵 A_1 的向量积等于低精度误差向量 r_1。于是，在步骤(e)，在更新了当前解向量 x 的数据元素之后，控制逻辑可以更新当前误差向量 r 的数据元素，使得 $r = b-Ax$。在这种实施方式中，第三收敛条件方便地依赖于当前误差向量 r。

[0032] 还可以设想替代方式，在优选的实施方式中，第一和第二迭代处理中的每一个包

括已知的共轭梯度方法。一般地，各种收敛条件可以被设置为给定应用所需要的，但是这些条件优选地依赖于在考虑中的预定数量的迭代处理的路径的完成和 / 或检测的结果解的收敛（实现向量对预定公差的解，或者在过程的连续路径中没有检测进展）。

[0033] 可以在系统存储器中各自预定矩阵 A 和向量 b 的预定数据元素，或者可以由输入数据按任意传统方式来定义。例如，如之前所提及的，定义了矩阵 A 的数据元素的输入数据可以包括定义在任意 n×1 向量上的矩阵 A 的应用函数 F，其中 1 ≤ i ≤ n 并且 1 ≤ j ≤ n 分别是矩阵 A 元素的行和列索引。在这种情况下，控制单元适于使用所述函数 F 来执行步骤 (a) 到 (e)，用于特别快速地和计算廉价地生成与 A 的矩阵向量积。特别地，涉及矩阵元素 a(i，j) 的处理操作可以典型地完全在处理器高速缓存器中完成，并且通过生成利用 A 的矩阵向量积获得的存储器业务可以比之前的系统显著地减少。这将会在下面进一步地介绍。

[0034] 使用共享的和 / 或各自分配的存储器资源，体现本发明的设备可以通过一个或多个处理器来实现。为了更有效率的操作，以及特别是在复杂的操作中，控制逻辑理想地包括多个处理器，其被并行地安排以全部并行地实现步骤 (a) 到 (e)。一般来说，处理器可以使用专用的低和高精度硬件或使用可被软件配置以用于低和高精度操作的通用硬件。在使用多个处理器的情况下，它们可以被集成到单个芯片或分布在基于单一处理器和 / 或多处理器的计算系统的不同芯片上，例如，在分布式计算系统的多个计算机上。类似地，在体现本发明的设备的操作中使用的存储器可以包括一个或多个类型的存储器的一个或多个组件，从本地处理器高速缓存器到主存储器（诸如硬盘或备份存储媒介），以及所述存储器或组件可以全部或部分地通过控制逻辑的不同处理器来共享。

[0035] 本发明的第二方面提供了一种计算机程序，用于使计算机生成与满足 Ax ＝ b 的 n×1 向量 x 相对应的 n 个高精度数据元素，其中 A 是与 n×n 个预定的高精度数据元素相对应的对称的、正定 n×n 矩阵并且 b 是与 n 个预定的高精度数据元素相对应的 n×1 向量。计算机程序包括程度代码装置，用于促使计算机访问输入数据，其存储于计算机的存储器中并且定义所述矩阵 A 和向量 b 的数据元素，以及：

[0036] (a) 实现用于从所述输入数据生成与满足 $A_1x_1 = b_1$ 的 n×1 向量 x_1 相对应的 n 个低精度数据元素的第一迭代处理；其中 A_1 是与矩阵 A 的 n×n 个数据元素相对应的低精度 n×n 矩阵并且 b_1 是与向量 b 的 n×1 个数据元素相对应的低精度 n×1 向量，在出现第一收敛条件时终止第一迭代处理；

[0037] (b) 将向量 x_1 的数据元素转换成高精度数据元素以获得当前的解向量 x；

[0038] (c) 实现用于生成与基于向量 b 和向量积 Ax 之间的差异的 n×1 纠正向量相对应的 n 个低精度数据元素的第二迭代处理，在出现第二收敛条件时终止第二迭代处理；

[0039] (d) 从所述纠正向量的 n 个低精度数据元素产生 n×1 更新向量 u 的各个高精度数据元素；

[0040] (e) 更新所述当前解向量 x 的数据元素，使得 x ＝ x+u；以及

[0041] (f) 执行步骤 (c) 到 (e)，直到出现第三收敛条件。

[0042] 应当理解的是，术语“计算机”用于大部分通用场景并且包括具有用于实现计算机程序的数据处理能力的任意设备、组件或系统，并且因此可以包括如上所述的单个设备一个或多个处理器或设备的分布式系统。此外，体现本发明的计算机程序可以构成独立的程序或程序集、或者可以是大规模程序或程序集的一部分，以及可以提供例如，体现为计算

机刻度媒介，例如用于载入计算机的磁盘或电子传输。计算机程序的程序代码装置可以包括按照任意语言、代码或符号的旨在促使计算机执行正在考虑的方法的指令集的任意表达式，直接地或在 (a) 与另一语言、代码或符号的会话之后，以及 (b) 以不同的材料形式进行复制。

[0043] 通常来说，此处介绍的通过参考本发明的一个方面的实施方式的特征的情况下，相应特征可以提供在本发明的另一方面的实施方式中。

附图说明

[0044] 现在将通过实例的方式，参照附图来介绍本发明的优选实施方式，其中：

[0045] 图 1 是体现本发明的处理设备的示例性框图；

[0046] 图 2 是示出了图 1 的设备的操作的流程图；

[0047] 图 3 示出了体现本发明的设备的示例性实现方式；

[0048] 图 4 是将本发明实施方式的运行时间与现有技术系统进行比较的示意图；

[0049] 图 5 是示出了本方面实施方式和现有技术系统中使用的存储器的表格。

具体实施方式

[0050] 图 1 是体现本发明的处理设备的简化示意图，其指示要被介绍的操作中涉及的主要组件。设备 1 包括在图中表示为控制器 2 的控制逻辑，以及此处以简化形式由高速缓存存储器 4 和主存储器 5 表示的存储器 3。控制器 2 的控制逻辑包括高和低精确逻辑，其中控制器 2 可以按高和低精度来执行处理操作。一般来说，可以在硬件或软件或它们的结合中实现控制器的控制逻辑。在本实施方式中，然而，由执行所介绍的功能的软件配置的一个或多个处理器核心来执行逻辑。通过此处的说明所属领域的技术人员容易了解合适的软件。虽然一般来说，要介绍的高和低精度操作可以由设备 1 的不同处理器来执行，在本实例中，我们假设控制器 1 的处理器在软件的控制下单独地操作以按高或低精度执行操作。此处的高速缓存存储器 4 代表控制器 2 的主要工作存储器，例如级别 1 高速缓存存储器。主存储器 5 代表控制器 2 可存取的存储器子系统的剩余者并且可以包括各种类型的存储器，例如，附加的高速缓存级别、硬盘和备份存储媒体。

[0051] 设备 1 适于实现与 nx1 向量 x 相应的用于生成 n 个高精度数据元素的过程，由下面的定义来表示方程的线性系统：

[0052] $Ax = b$

[0053] 此处，A 是维度为 n×n 的对称正定稠密矩阵并且 b 是 n×1 的向量。由存储器 5 中存储的输入数据来定义与矩阵 A 的元素相对应的 n×n 高精度数据元素以及与向量 b 的元素相对应的 n 个高精度数据元素。更为具体地，此处直接通过标量函数 $F() = a(i, j)$ 来定义矩阵 A 的高精度数据元素，其中 $a(i, j)$ 表示具有行索引 $i(1 \leqslant i \leqslant n)$ 和列索引 $j(1 \leqslant j \leqslant n)$ 的元素。在本实例中，我们假设在存储器 5 中直接地定义向量 b 的数据元素。存储器 5 还保持指定定义在处理操作中使用的三个收敛条件的参数的数据 $C_1(k_1, d_1)$、$C_2(k_2, d_2)$ 和 $C_3(p, c)$。这些参数将在下面进行解释。

[0054] 在图 2 的流程图中示出了由设备 1 执行的生成解向量 x 的高精度数据元素的关键步骤。操作在步骤 10 处开始，典型地以响应于通过设备的 I/O（输入 / 输出）接口（未示

出）的操作提示符或来自另一应用的请求，是运行在控制器2的处理器上还是在与控制器2进行通信的远程处理器上。在步骤11，控制器2访问主存储器1以获取高精度函数F()并且在高速缓存存储器4中创建低精度拷贝$F_1()$。这可以通过降级、或四舍五入等已知方式来完成高精度函数到低精度的表示。类似地，控制器1获取向量b的高精度数据元素并且将这些降级为各个低精度元素以生成低精度向量b_1。基于存储容量和维度n，可以在高速缓存器4中保留或在主存储器5中预存储向量b_1的数据元素。

[0055] 接下来，在步骤12中，控制器1执行用于生成与最初的、近似的n×1解向量x_1相对应的n个低精度数据元素的第一迭代处理的一个路径，满足

[0056] $A_1x_1 = b_1$

[0057] 此处，A_1是函数$F_1()$所定义的n×n低精度矩阵。在本实施方式中，为第一迭代处理使用的技术是共轭梯度(CG)处理。CG处理是用于方程的线性系统的解的公知技术，并且不需要在此处进一步介绍。控制器2仅使用函数$F_1()$和向量b_1的元素来执行必要的计算，可以根据需要从主存储器5向高速缓存器4回调向量b_1的元素。在CG处理的一个路径之后获得的向量x_1的低精度元素存储在存储器3中，并且操作前进到步骤13，其中控制器2对路径计数器p_x进行加1。在判断步骤14，控制器2检查当前路径计数p_x是否等于预设的参数k_1，其指示第一迭代处理的允许路径的最大数量。假设不是（在步骤14决定“否”(N)），控制器2于是在判断步骤15判断解x_1的收敛是否已经出现。如果两个事件中的任一个出现，则此处检测为收敛。第一事件是已经到达为x_1预设的下降公差d_1。即，控制器2通过数量≥d_1检查用于x_1的当前解是否与第一路径中获得的解不同，其中，下降公差d_1典型地指百分数改变。第二事件是在过程的这个路径中没有实现进展，即，x_1的解与之前路径中获得的解没有改变。假设在步骤15没有检测到收敛，操作返回到步骤12用于解处理的另一路径。所述处理因此迭代直到出现定义为如在步骤14中完成的k_1路径的第一次出现的第一收敛条件C_1或在步骤15处识别的收敛为止。在检测收敛条件C_1（在步骤14或步骤15处判断“是”(Y)），控制器2终止第一迭代处理。

[0058] 在完成第一迭代处理时，操作前进到步骤16，其中控制器2创建第一迭代处理所输出的解向量x_1的高精度拷贝。这可以通过将向量x_1的低精度数据元素转换成各个高精度数据元素以获得存储器3中存储的当前的、高精度解向量x的已知方式来完成。接下来，在步骤17，控制器2生成满足如下条件的与当前nx1误差向量r相对应的n个高精度数据元素：

[0059] $r = b-Ax$

[0060] 根据需要可以使用函数F()和可以向高速缓存器4回调的向量x和b的元素来执行上述操作。误差向量r的结果高精度元素与这个误差向量的低精度拷贝r_1一起存储在存储器3中，其在步骤18通过将向量r的元素转换为各个低精度数据元素来生成。

[0061] 在步骤19，控制器2于是执行用于生成满足如下公式的与nx1纠正向量z相对应的n个低精度数据元素的第二迭代处理的第一路径：

[0062] $A_1z = r_1$

[0063] 再次，在本实例中共轭梯度技术用于第二迭代处理，并且控制器2可以使用函数$F_1()$来执行必要的计算。将过程的一个路径之后获得的向量z的低精度元素存储在存储器3中，并且操作前进到步骤20，其中控制器2对路径计数器p_z进行加1。在判断步骤21，控

制器 2 检查当前路径技术 p_z 是否等于预设参数 k_2，其指示第二迭代处理的允许路径的最大数量。如果否，控制器 2 在判断步骤 22 确定解 z 的收敛是否已经出现。如果 (1) 已经到达用于 z 的预设下降公差 d_2，或者 (2) 在处理的这个路径中没有实现进展，即由于之前的路径未改变用于 z 的解，则这里再次检测到收敛。假设在步骤 22 没有检测到收敛，操作返回到步骤 19 用于解处理的另一路径。处理因此迭代，直到定义为在步骤 21 处完成的 k_2 路径的第一次出现的第二收敛条件的出现或者在步骤 22 处识别到收敛。在检测到收敛条件 C_2（在步骤 21 或步骤 22 的 Y）时，控制器 2 终止第二迭代处理。

[0064] 在完成第二迭代处理时，操作前进到步骤 24，其中控制器 2 将纠正向量 z 的数据元素转换成各个高精度数据元素，以产生存储在存储器 3 中的高精度更新向量 u。接下来，在步骤 25，控制器 2 更新当前高精度解向量 x 的数据元素，使得

[0065] $x = x+u$

[0066] 并且在存储器 3 中存储更新的解 x。在更新了解向量 x 之后，控制器 2 在步骤 26 中更新当前高精度误差向量 r，使得：

[0067] $r = b-Ax$，

[0068] 其中，使用定义矩阵 A 的函数 F() 再次执行计算。在存储器 3 中存储新向量 r 的高精度元素，并且操作前进到步骤 27，其中控制器 2 对用于第三迭代处理的路径计数器 p_r 增加 1。在判断步骤 28，控制器 2 检查当前路径计数 p_r 是否等于预设参数 p，其指示第三迭代处理的允许路径的最大数量。如果否，控制器 2 在判断步骤 29 判断解 r 的收敛是否已经出现。如前，如果 (1) 已经达到用于 r 的预设公差，或者 (2) 从处理的前述路径以来用于 r 的解没有实现进展，则检测到收敛。在这种情况下，公差 c 指定误差阈值，其中欧几里得 (Euclidean) 规范小于 c 的误差向量 r 被视为其所要求的公差内。假设在步骤 29 没有检测到收敛，操作返回到步骤 18 用于第三迭代处理的另一路径，再次要求用于新误差向量 r 的第二迭代处理的执行。步骤 18 至 29 的处理因此迭代直到出现被定义为在步骤 28 处完成的 p 路径的第一次出现的第三收敛条件 C_3 或者在步骤 29 处被识别的误差向量 r 的解的收敛。在检测到收敛条件 C_3（在步骤 28 或 29 的是）时，控制器 2 终止第三迭代处理。在步骤 30，将对应于最终解向量 x 的高精度数据元素输出到发起所述处理的运算器或应用，并且处理操作完成。

[0069] 前述设备为方程的线性系统的解提供优良的操作效率，甚至在矩阵 A 是稠密的并且维度 n 非常大的情况下。设备利用快速、低精度处理并且处理操作的复杂度仅是 $O(kn^2)$，对于 k1 = k2 = k。这与之前介绍的引发成本 $o(n^3)$ 现有系统有明显的对照。

[0070] 注意到，在上述处理中所有的涉及矩阵 A 或 A_1 的计算仅要求具有另一向量的 A 或 A_1 的矩阵向量积。这些计算不要求矩阵 A 的先有信息，但是可以由控制器 2 仅适当地使用函数 F() 或 F_1() 和根据需要可以从主处理器 5 向高速缓存器 5 回调的正在考虑的向量元素来执行。那就是，涉及矩阵 A（或 A_1）必要的处理操作可以全部在处理器高速缓存器中执行。与之前介绍的先有方法相比，这样显著地降低了控制器 2 的处理器和存储子系统之间的业务量。特别地，在上面的实施方式中，具有 A（或 A_1）的矩阵向量积的计算仅需要 O(kn) 次存储移动，其中通过相比较 k 典型地小于维度 n。相对的，现有方案要求在分解之前在系统存储器中形成矩阵 A。这个矩阵被分解为 $A = LL_T$，其中 L 是具有 O(n/2) 个元素的下三角矩阵。以低精度存储的矩阵 L 用于所有的解实例。高精度的原始矩阵 A 适用于误差的计

算。因此，在现有系统中，$O(n^2)$ 个数据需要在所述过程的每个迭代细化步骤中从主存储器移动到处理器。

[0071] 定义三个收敛条件 $C_1(k_1, d_1)$、$C_2(k_2, d_2)$ 和 $C_3(p, c)$ 的参数可以被设置为根据在给定应用中所需要的准确度的要求。在优选的实施方式中，为了实现的便利选择 $k_1 = k_2 = k$，其中在第一和第二迭代处理中路径的最大数量是相同的。注意到，与现有系统中计算误差的过程不同，这些过程不需要完全地朝向误差向量 r 来解决，但是可以自动地适应所要求的准确度。

[0072] 如之前所介绍的，可以有并行操作的多个处理器来实现设备 1，这些处理器一起实现上面介绍的处理操作。图 3 示出了这种实施方式的简单实例。此处，通过多个处理器来实现控制器 2 的功能，在具有各个级别一个 (L1) 高速缓存器的情况下，通过总线接口 (I/F) 与共享存储器子系统继续通信。由于处理操作是基于矩阵向量积的，所使用的处理器数量可以使较大的。这种大量的并行实现方式提供优良的操作效率。图 4 用两个不同的矩阵维度 n 的值和各种数量的并行处理器来比较上面介绍的现有系统和本发明实施方式的运行时间。垂直轴对应于运行时间并且水平轴对应于使用的处理器的数量。矩阵大小 n 的两个值为 32768 和 49152。以实线示出的上面的轨迹对与现有方案对应，并且以虚线示出的下面的轨迹对与本发明的实施方式相对应。注意到，垂直轴的量度是对数的。直接清楚地是，本发明的实施方式提供实质上的改进，减少运行时间至少一个量级。

[0073] 本发明的实施方式还提供明显的存储器使用和带宽改进。图 5 的表格指示了使用如上所述的现有方案和本发明的实施方式的、用于矩阵 A 的各种大小的、误差向量 r 的迭代细化的每个步骤的按 G 字节的存储器使用（所要求的带宽）。所述表格中顶部行给出利用不同值 n 的现有系统的结果。下面的四行给出所介绍的实施方式中 $k_1 = k_2 = k$ 的不同值的结果。通过本发明的实施方式所实现的显著改进是非常明显的。

[0074] 应当明白的是，可以对上面介绍的示例性实施方式做出许多改变和改进。通过实例的方式，用于迭代处理的收敛条件会依赖于那些指定的不同集合的事件并且可以按各种方式依赖于这些事件的组合。然而，优选的实施方式总是指定用于每个过程的最大迭代数量。其中为收敛条件指定公差，这将会以给定应用所需要的各种方式来定义。同样，当在上面的第一和第二迭代处理中使用共轭梯度方法时，如果需要此处可以使用用于方程的线性系统的其它迭代解决技术。在不脱离本发明范围的情况下，可以对所介绍实施方式做出许多其它的改变和改进。

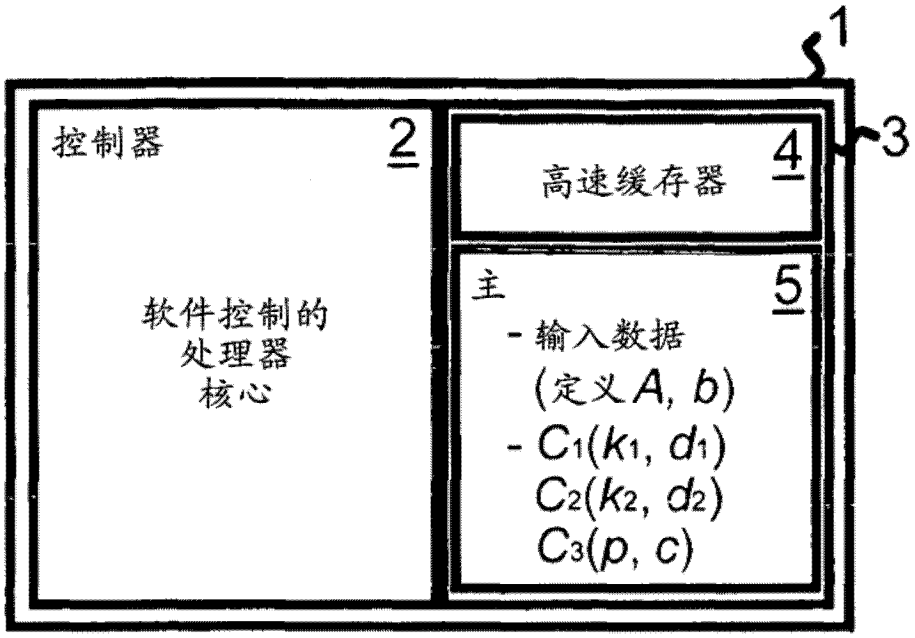

1
控制器
2
软件控制的
处理器
核心
高速缓存器
4
3
主
5
- 输入数据
(定义A, b)
- C1(k1, d1)
C2(k2, d2)
C3(p, c)

图 1

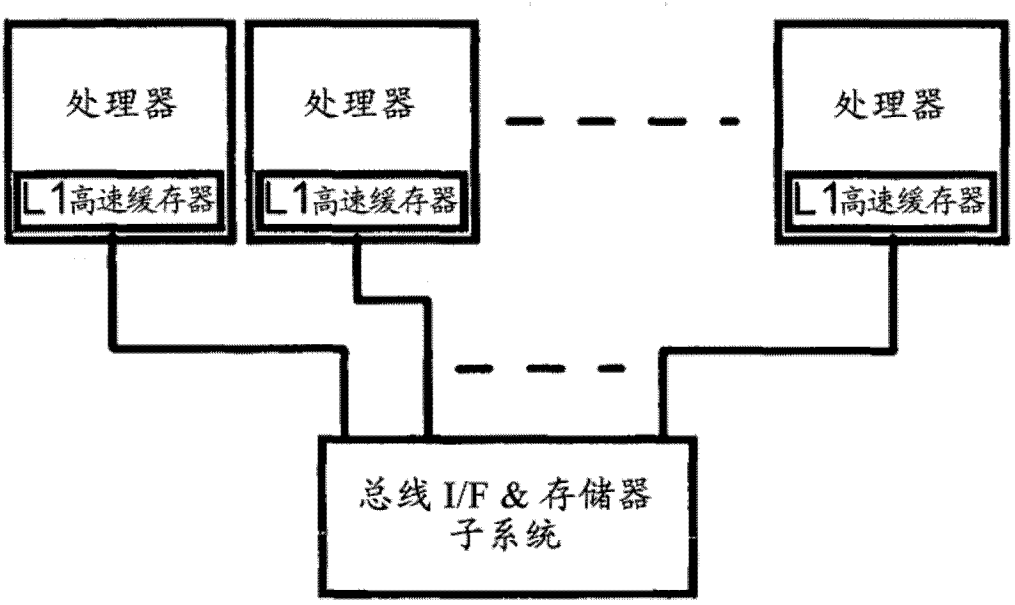

处理器
L1高速缓存器
处理器
L1高速缓存器
处理器
L1高速缓存器
总线 I/F & 存储器
子系统

图 3

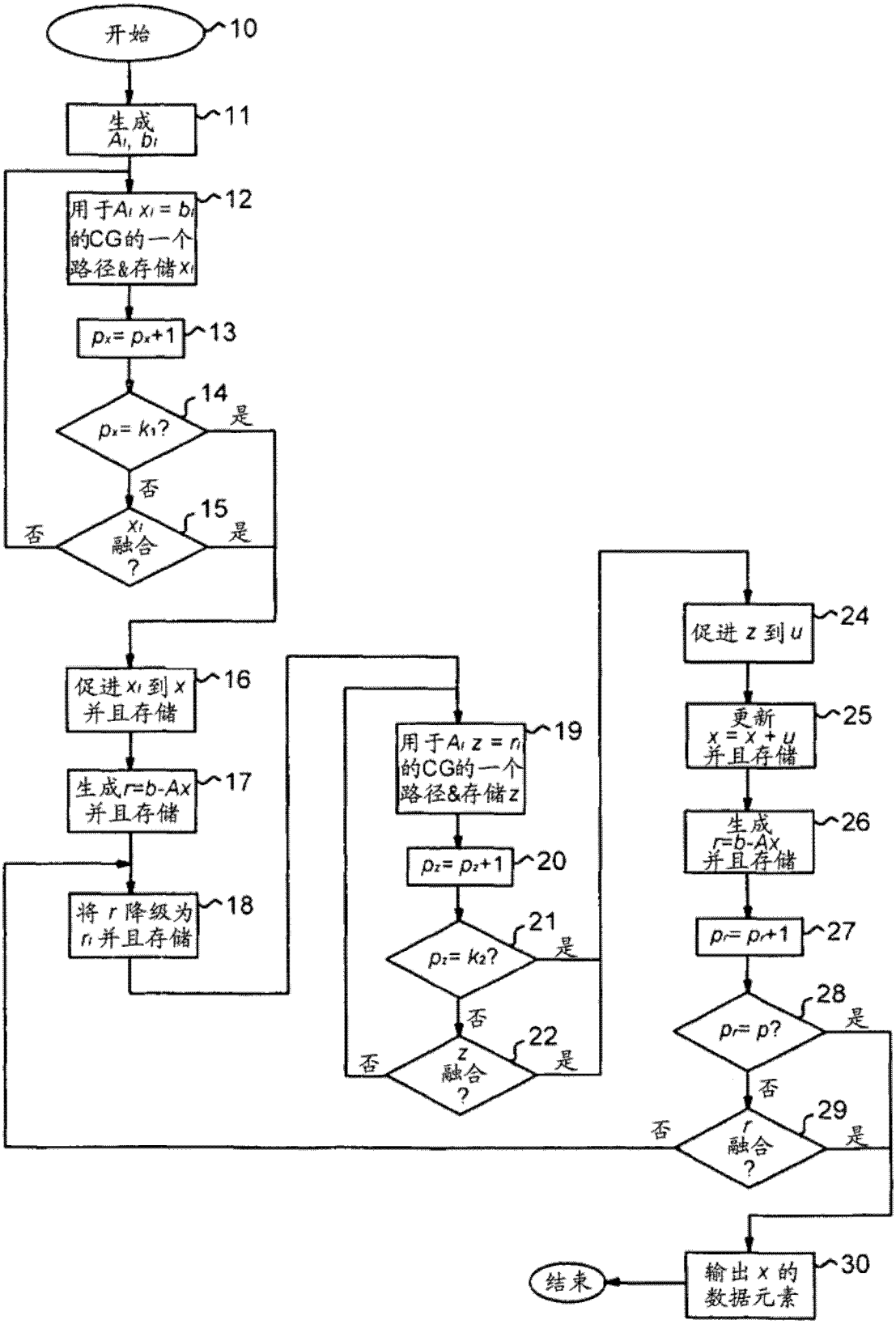

开始
10
生成 A_l, b_l
11
用于 A_l x_l = b_l 的CG的一个路径&存储 x_l
12
p_x = p_x+1
13
p_x = k_1?
14
是
否
x_l 融合 ?
15
是
否
促进 x_l 到 x 并且存储
16
生成 r=b-Ax 并且存储
17
将 r 降级为 r_l 并且存储
18
用于 A_l z = r_l 的CG的一个路径&存储 z
19
p_z = p_z+1
20
p_z = k_2?
21
是
否
z 融合 ?
22
是
否
促进 z 到 u
24
更新 x = x + u 并且存储
25
生成 r=b-Ax 并且存储
26
p_r = p_r+1
27
p_r = p?
28
是
否
r 融合 ?
29
是
否
输出 x 的数据元素
30
结束

图 2

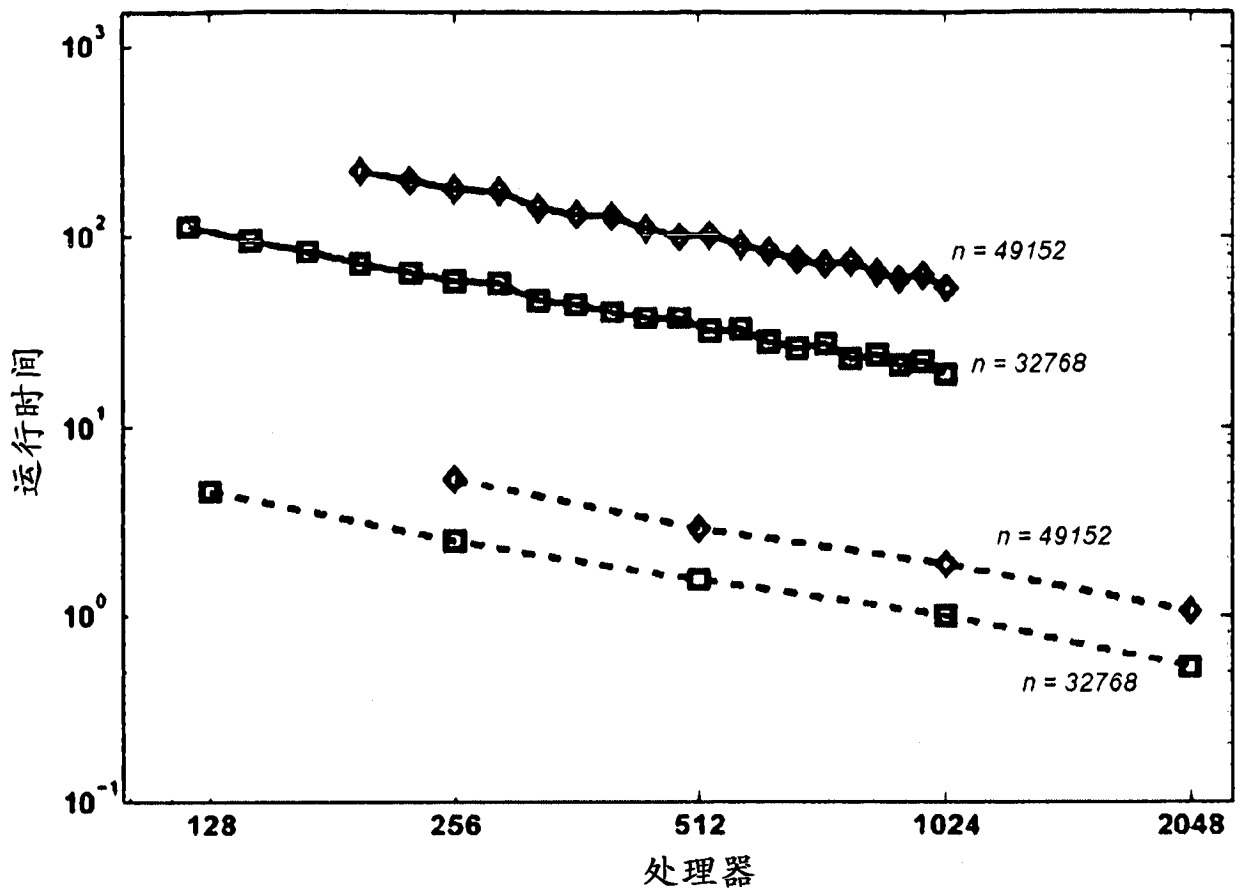


图 4

	n = 10000	n = 20000	n = 30000	n = 40000	n = 50000
存储器,现有技术	0.381GB	1.54GB	3.43GB	6.1GB	9.54GB
存储器, k=100	0.003GB	0.007GB	0.011GB	0.015GB	0.019GB
存储器, k=200	0.006GB	0.0014GB	0.022GB	0.03GB	0.038GB
存储器, k=500	0.015GB	0.035GB	0.055GB	0.075GB	0.095GB
存储器, k=1000	0.03GB	0.07GB	0.11GB	0.15GB	0.19GB

图 5