中国科学院大学

University of Chinese Academy of Sciences

硕士学位论文

面向申威众核处理器的多精度矩阵乘法代码自动生成技术研究

作者姓名：严愉程

指导教师：刘芳芳 正高级工程师 中国科学院软件研究所

学位类别：工学硕士

学科专业：计算机软件与理论

培养单位：中国科学院软件研究所

2024 年 6 月

LaTeX Thesis Template

of

The University of Chinese Academy of Sciences $^{\pi}\pi^{\pi}$

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Natural Science

in Fluid Mechanics

By

Mo Huangrui

Supervisor: Professor Liu Qingquan

Institute of Mechanics, Chinese Academy of Sciences

June, 2014

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

本文是中国科学院大学学位论文模板 ucasthesis 的使用说明文档。主要内容为介绍 LaTeX 文档类 ucasthesis 的用法，以及如何使用 LaTeX 快速高效地撰写学位论文。

关键词：中国科学院大学，学位论文，LaTeX 模板

Abstract

This paper is a help documentation for the LaTeX class ucasthesis, which is a thesis template for the University of Chinese Academy of Sciences. The main content is about how to use the ucasthesis, as well as how to write thesis efficiently by using LaTeX.

Keywords: University of Chinese Academy of Sciences (UCAS), Thesis, LaTeX Template

目 录

图形列表

表格列表

符号列表

字符

Symbol	Description	Unit
R	the gas constant	$m^2 \cdot s^{-2} \cdot K^{-1}$
C_v	specific heat capacity at constant volume	$m^2 \cdot s^{-2} \cdot K^{-1}$
C_p	specific heat capacity at constant pressure	$m^2 \cdot s^{-2} \cdot K^{-1}$
E	specific total energy	$m^2 \cdot s^{-2}$
e	specific internal energy	$m^2 \cdot s^{-2}$
h_T	specific total enthalpy	$m^2 \cdot s^{-2}$
h	specific enthalpy	$m^2 \cdot s^{-2}$
k	thermal conductivity	$kg \cdot m \cdot s^{-3} \cdot K^{-1}$
S_{ij}	deviatoric stress tensor	$kg \cdot m^{-1} \cdot s^{-2}$
τ_{ij}	viscous stress tensor	$kg \cdot m^{-1} \cdot s^{-2}$
δ_{ij}	Kronecker tensor	1
I_{ij}	identity tensor	1

算子

Symbol	Description
Δ	difference
∇	gradient operator
$\delta^{\pm}$	upwind-biased interpolation scheme

缩写

CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
EOS	Equation of State
JWL	Jones-Wilkins-Lee
WENO	Weighted Essentially Non-oscillatory
ZND	Zel'dovich-von Neumann-Doering

第 1 章　绪论

1.1　研究背景

高性能计算 (High-Performance Computing, HPC) 又称超级计算，是一种利用超级计算机和数值计算算法对复杂科学、工程和商业问题进行求解的计算机应用。高性能计算的应用领域包括人工智能、天气预报、气候模拟、地震模拟、生物医学、材料科学、工程设计、金融风险分析等。矩阵计算在这些计算应用中会被反复调用，并且在其中耗时较多。随着高性能计算机体系结构不断演进，矩阵计算需求的不断变化，与高性能计算相关的数学软件，特别是矩阵计算库，需要不断优化以适应新的硬件架构和新的矩阵计算需求，才能充分发挥硬件的计算能力。

基础线性代数子程序 (Basic Linear Algebra Subprograms)[1] 构成了数学库的基石，是高性能计算领域不可或缺的一个组成部分。其广泛应用于科学计算、机器学习，以及人工智能等多个研究前沿。BLAS 的高度优化函数是许多大规模计算密集型科学应用的核心依赖，包括但不限于天体物理学、量子化学、大气动力学以及材料科学等。开发和优化高性能 BLAS 库对于加速这些领域的研究进程，提升计算效率具有至关重要的影响。BLAS 库通常有硬件厂商提出针对特定硬件平台高度优化的版本，例如 cuBLAS[2], oneMKL-BLAS[3], AOCL-BLAS[4], rocBLAS[5], APL-BLAS[6] 等。另外也有第三方的开源实现例如 BLIS[7], MAGMA[8], OpenBLAS[9], GotoBLAS[10] 等

申威系列处理器的硬件架构、指令级、存储层次等一般与通用 CPU、GPU 存在较大差异。为一般商用多核处理器定制的 BLAS 库无法在 SW26010P 上运行，现有开源数学库在设计时也没有考虑异构众核处理器的特点，其优化方案也不能充分发挥申威众核处理器的硬件特性。因此，针对申威众核处理器特有的体系结构设计高性能 BLAS 库十分必要。BLAS 包括向量和矩阵的基本操作，它分为 3 个等级。BLAS 1、2 级函数是典型的访存密集型任务，其性能受限于系统访存带宽，BLAS 3 级函数在大规模条件下是计算密集型任务。

研究表明,BLAS 库中多数 Level-3 计算函数都可以通过调用 GEMM 函数完成, 即,GEMM 函数可以作为整个 Level-3 BLAS 的构建基础。因此, 优化 GEMM 函数便成为高性能 BLAS 库开发工作的重中之重。GEMM 各种计算规模，转置与非转置，低精度、复数等输入类型等变化，会产生组合爆炸的现象，往往需要大量的工作量来应对这些变化。本论文将从自动代码生成与自动调优这一方面展开工作，研究面向申威众核处理器的多规模多精度的稠密矩阵计算代码自动生成系统。该技术在面向新的架构或者其他体系结构时，提供一定的自适应能力，期望能够以极少的修改工作量就能完成适配。在上述架构的基础上，因为存在减少冗余访存带宽的需求，本论文还将研究算子融合，多精度异形矩阵计算等优化策略。具备较高的应用价值，期望推动国产平台上数学库以及相关应用的发

展。

本研究面向 SW26010-pro 众核处理器进行 BLAS(Basic Linear Algebra Subprograms) 高性能基础线性代数库自动代码生成技术的研究。对 BLAS 进行深度优化，结合代码自动生成技术，实现不同规模下的自动调优框架。同时结合算子融合等技术，推动国产平台上数学库以及相关应用的发展。

1.2 研究目标和主要工作

1.2.1 面向矩阵乘法的代码自动生成工具

1.2.2 自适应性能调优

1.2.3 基于 DSL 的矩阵乘法算子融合拓展

1.3 论文组织结构

第 2 章　问题背景与国内外研究进展

2.1　背景知识概述

2.1.1　矩阵乘法计算及其优化原理

BLAS 三级函数中的通用矩阵乘法 (General Purpose Matrix Multiply, GEMM)，是矩阵相乘并累加的运算 [1]，其基本形式如下所示：

$$C = \alpha AB + \beta C \tag{2.1}$$

其中输入矩阵 A 的大小为 $M \times K$，输入矩阵 B 的大小为 $K \times N$，有转置和非转置两种情况，输出矩阵 C 的大小为 $M \times N$。α 和 β 为标量，是矩阵乘法的运算系数。矩阵乘法的数据类型可以为，单精度浮点实数，双精度浮点实数，单精度浮点复数，双精度浮点复数等，针对人工智能深度学习领域的计算特点，半精度 (half) 数据类型，也开始被广泛利用。矩阵乘法的计算复杂度为 $O\left(N^3\right)$，访存量为 $O\left(N^2\right)$，计算访存比为 $O(N)$，当矩阵的规模较大时，是典型的计算密集型应用。代码1是矩阵乘法算法的一个简单的实现，可以看到矩阵乘法其实是在 M N K 三个维度上的三层嵌套循环，在维度 K 上进行归约和累加的操作。

```
for (int i = 0; i < M; i++) {
    for (int j = 0; j < N; j++) {
        C[i][j] *= beta;
        for (int k = 0; k < K; k++) {
            C[i][j] += alpha * A[i][k] * B[k][j];
        }
    }
}
```

Listing 1 矩阵乘法的简单实现

在多核或者众核 CPU 上，矩阵乘法优化主要有分块算法、多核并行任务划分、SIMD 向量汇编内核，这三个重要的优化手段。

TODO

TODO 可以提到 winograd 优化方式?

2.1.2　申威 SW26010P 众核处理器

2.1.3　Roofline 模型

2.2　国内外研究现状

申威处理器上存在不少针对 BLAS 算子库的优化工作，比如 [11,12] 面向 SW26010-pro 众核处理器探索了 BLAS 基础线性代数程序集的高性能实现，充分

利用了 SW26010-pro 众核处理器的硬件特性。针对异形矩阵乘法，则是需要针对 MNK 三个规模的大小，设计不同的分块，规约方式。

在高性能代码自动生成与自动调优方面,Tensile[13] 实现了一套 GPU 汇编 kernel 生成器，定义了一系列和性能相关的 kernel 参数配置，针对固定矩阵规模或一个范围内的矩阵规模，搜索并选择对应性能最好的 kernel。Halide[14] 引入了一种可以描述循环优化原语的调度语言。将计算和调度分离，适用于手动优化和自动搜索。Halide 有三个基于不同技术的自动调度器版本 [15][16][17] 。其中，使用波束搜索和学习成本模型的最新版本表现最好。TVM[18] 使用了类似的调度语言，并包括了一个模板引导搜索框架。而 Ansor[19] 基于 TVM 的框架，针对深度学习工作负载，引入了新的搜索空间，摆脱了对预先定义模版框架的依赖，同时用进化搜索，XGBOOST 筛选器的方式，优化了搜索和自动调优的效率。Ansor 依赖于 LLVM 编译器后端，目前还没有办法直接搜索和生成高性能的汇编 kernel，这是导致他在一些处理器架构上生成 kernel 效率不高的原因之一。而 ukernel[20] 则是实现了一个针对 GEMM 汇编 kenrel 的编译器后端。swATOP[21] 在 SW26010-pro 处理器上自动生成 DNN 算子，同样是采用调度器和 IR 优化器的两层架构，并提供自动调优的能力，在 IR 上需要插入调用 DMA 原语，以及 GEMM 原语。

针对矩阵乘法分块参数选择的问题，Le Xu[22] 等人，在 SW26010-pro 处理器上，首先根据不同循环次序，用公式定义了 DMA 访存量和分块大小 BM、BN、BK 之间的关系，以及计算密度和分块大小之间的关系。同时加入最多使用的 LDM 大小，和每次 DMA 或 RMA 的最小大小作为约束条件。在满足约束条件的情况下，作者训练了一个神经网络，目标函数是最大化计算密度，这样针对某一给定的矩阵计算规模 M-N-K，神经网络只需要一次推理的时间就能给出具体的分块参数 BM、BN、BK。

TensorIR[23]、GrapheneIR[24] 都是针对 Tensor 计算提出了一套中间描述语言 IR，然后交给深度学习编译器优化生成高性能的 CUDA 程序。TensorIR 使用 block 作为编译器调度的基本单位，block 之间存在消费者和生产者的关系。基于 block 编译器可以调度出软件流水线并且调用 TensorCore 来完成一个 block 的计算任务。GrapheneIR 提供了基于 IR 层面的算子融合能力。

多面体编译模型是一种先进的编程优化技术，它能够把程序优化的问题转化为整数线性规划问题。该模型通过对程序中的循环结构进行仿射变换，优化数据的重用策略，从而减少数据处理过程中的时间和空间开销。在申威处理器上，研究者陶小涵 [25] 等人使用多面体模型 [26]，插入各种节点，实现了 GEMM 通用矩阵乘法的自动编译生成，但是缺乏框架的灵活性即自动搜索和调优的能力。

计算图优化是一种将各种计算操作（算子）作为基本处理单元的技术，目的是在保持算子内部结构不变的情况下，对程序的整体计算流程（图）进行优化。这种优化可以通过多种方式实现，包括优化数据布局、合并多个算子以减少操作、折叠固定的常量值、自动化处理数据批次，以及采用不同精度的计算来提高效

率。例如，针对海洋模式中求解正压模态的海表高度方，Mindspore[27] 将计算科学代码转换为对应的计算图，自动生成对应的融合算子，例如多个 element-wise 的算子，减少内存数据的重复搬运。

CUTLASS 是一个由 NVIDIA 推出的，基于 C++ 模板特性的矩阵计算库。该数学库利用 C++ 模板来支持灵活的矩阵运算分块策略，和不同矩阵数据精度，例如 int8，int4，FP8 等。基于模块化的理念，CUTLASS 提供了 device level，block level，warp level 等多个层级 api，方便用户使用自定义的算子融合策略。但是，该数学库在使用的时候，往往需要用户来指定分块参数，使得在某些特定的矩阵规模下取得比较好的性能。例如在矩阵 M、N 维度较小的情况下，CUTLASS 需要提供用户手动指定开启 split-k 的优化来提升 SM 的利用率。

综上所述，目前在国产申威 SW26010-pro 处理器上存在不少针对 BLAS 算子库的优化工作，但是仍然有很多的改进的空间。1. 胡怡 [12] 针对了双精度浮点类型较大规模矩阵和异形矩阵的优化，但是针对某个具体规模大小的矩阵乘法问题，在分块参数的选择以及具体算法的选择上，缺乏一定的自适应性，人工手动调优较为耗时。2. 该工作也没有针对单精度，半精度浮点类型的矩阵乘法进行优化。3. 可编程 Cache 天然适合利用算子融合的优化来进行加速，但是目前国产申威处理器上的 BLAS 库无法很好地利用算子融合来进行加速，需要开发人员手工编写。

因此，本论文研究目标，期望提出一种新的矩阵乘法计算以及代码自动生成框架，能够同时兼容不同的数据类型；针对较大规模矩阵和异形矩阵，具备自动调优，自动搜索的功能；同时基于 DSL 提供算子融合的能力。该计算框架对国产申威众核平台上计算生态的发展有重要意义。

2.3　小结

第 3 章　面向矩阵乘法的代码自动生成工具

本章节详细介绍了面向矩阵乘法的代码自动生成工具框架，主要包含以下几个部分：

- 领域特定语言 (Domain Specific Language) 及其语法分析器 (Parser)
- 调度树 (Schedule Tree) 对矩阵计算任务的建模
- 以及中间表示 (Intermediate Representation)
- 基于 AST 变换的编译优化算法
- 代码生成工具 (Code Generator)

3.1　针对矩阵计算问题的建模

芯片的理论峰值计算性能往往由计算核心的数量、峰值频率、指令发射宽度、每条指令完成多少浮点计算，这几个参数相乘来得到。

矩阵乘法算法作为一个计算密集型的算法，往往期望跑到 90% 以上的峰值性能即充分利用芯片的计算能力。但是，较高的目标峰值性能，对访存系统的延迟和带宽，软件算法的设计也提出了较高的要求。

近年来 CPU 还是 GPU 都在不断优化主存带宽，例如使用 DDR5、HBM 等更先进的存储硬件。同时也都使用了芯片内高速缓存 Cache，来降低数据的延迟。高速缓存 Cache 离计算单元物理距离更近，并且 SRAM 的硬件特性，能够有效降低数据延迟。同时配合配合矩阵乘法算法中的分块优化思路，将一部分会重复读取的数据放在高速缓存中，能够减少对主存的访问。

全局的高速缓存，还需要解决缓存一致性的问题，这在某种程度上，限制和核心数量的扩张，同时过多的缓存一致性冲突也会导致算法运行效率的降低。分布式可编程高速缓存则是很好得解决了这一问题，该技术得到了广泛的应用，例如 SW26010P 上的 LDM 和 NVIDIA GPU 中的共享内存 (Shared Memory)

在矩阵乘法中，不论是在 GPU 还是 CPU 的平台，优先在 M N 这两个维度上进行任务划分。(TODO 这里插入 thread block swizzle 的图片，下图中的 Block(0,0) 和 Block(1,0) 都需要读取 A 矩阵的第一个行块。也就是说在相同行的 Block 都需要同一行 A 的数据，相同列的 Block 都需要同一列 B 矩阵的数据。这里也存在着重复数据读取的问题，在 GPU 和 SW26010P 处理器中，这部分数据可以被 L2 Cache 缓存，例如 GPU 中的矩阵乘法算法往往会用 ThreadBlock Swizzle 这一优化策略来提高 L2 Cache 的命中率。另外片上网络设计的引入，也可以更优雅的解决这一问题。

片上网络的引入和优势。正如前文 TODO 所提到的，SW26010P 处理器，在一个核组，即 64 个从核之间共享一个片上交换网络，并且该网络构成二维网格的组成形式。同样 NVIDIA GPU 在 hopper 架构中也引入了 Shared Memory 之间

的通信机制 (TODO 插入片上网络通信的图片)。在 Groq AI 推理芯片中，在矩阵乘法的计算中也是用片上网络的通信机制。

随着体系结构的发展，市面上出现了一些具备片上网络和核内分布式高速缓存的处理器，例如特斯拉的 DOJO，NVIDIA 的 HOPPER 还有 SW26010P 处理器等。片上网络的特点和使用基于消息通信机制

本研究针对分布式可编程高速缓存和核间片上网络的特点，结合矩阵乘法的特点，设计了一套用于矩阵乘法的代码自动生成框架。

3.2 领域特定语言 (Domain Specific Language)

3.3 面向矩阵乘法 IR 设计

3.3.1 DMA

3.3.2 RMA

3.3.3 MMA

3.3.4 其他 IR

3.4 用于优化的各种 AST 变换

3.4.1 循环展开变换

3.4.2 循环消除变换

在实际的应用当中，用户可能会需要一个特定规模的矩阵乘法，利用这个矩阵的规模信息我们能对矩阵乘法进行进一步的性能提升。结合后续章节提到的动态调优技术，我们能生成一套覆盖所有范围，并且针对某个区间调优的矩阵乘法库。

例如，在上面的例子中，我们可以看到，矩阵乘法的规模是 $M \times N \times K$，针对特定的规模信息，我们可以对前面生成的 AST 进行优化。

3.4.3 循环不变式外提变换

本研究使用了编译器技术中，循环不变式外提的中端优化技术，

循环不变式外提技术的定义。循环不变式外提（Loop Invariant Code Motion，简称 LICM）是一种编译器优化技术，旨在提高程序运行效率。它识别循环中的那些计算，这些计算在每次迭代中都产生相同的结果，也就是说，这些计算与循环变量的值无关，因而被称为“循环不变式”。一旦识别出循环不变式，编译器就会将这些计算从循环体中移动到循环之前，从而减少每次循环迭代时的计算量。

TODO 循环不变式外提的实现算法

TODO 循环不变式起作用的例子，以及对访存量的影响

3.4.4　软件流水线变换

软件流水线的定义。软件流水线（Software Pipelining）是一种高级循环优化技术，用于提高处理器执行并行性的一种编译时技术。它通过重新组织循环体内指令的执行顺序，使得在每个循环迭代中，可以同时对多个迭代的不同部分执行操作，类似于硬件流水线中不同阶段同时处理多个指令的方式。这样做的目的是在保持循环逻辑不变的前提下，最大化指令级并行（ILP），减少循环迭代之间的依赖性，提高 CPU 利用率和程序执行效率。

在 SW26010P 处理器中，我们主要使用两层软件流水线设计。第一层是在分块矩阵乘法，也就是 MMA 中。正如上文所提到的，这部分主要通过手写的汇编指令，实现 LDM 到寄存器的读写指令，和 SIMD 计算指令之间的异步并行。第二层则是在中间表示 IR 的粒度上，来构建流水线。SW26010P 支持异步 DMA 读写，以及异步 RMA 数据通信，计算则是需要同步完成。因此软件流水线主要发生于 DMA、RMA 、MMA 个阶段。

第 4 章　自适应调优框架

为方便使用及更好地展示 LaTeX 排版的优秀特性，ucasthesis 的框架和文件体系进行了细致地处理，尽可能地对各个功能和板块进行了模块化和封装，对于初学者来说，众多的文件目录也许一开始让人觉得有些无所适从，但阅读完下面的使用说明后，会发现原来使用思路是简单而清晰的，而且，当对 LaTeX 有一定的认识和了解后，会发现其相对 Word 类排版系统极具吸引力的优秀特性。所以，如果是初学者，请不要退缩，请稍加尝试和坚持，以领略到 LaTeX 的非凡魅力，并可以通过阅读相关资料如 LaTeX Wikibook [**?**] 来完善自己的使用知识。

4.1　先试试效果

第 5 章　算子融合

为方便使用及更好地展示 LaTeX 排版的优秀特性，ucasthesis 的框架和文件体系进行了细致地处理，尽可能地对各个功能和板块进行了模块化和封装，对于初学者来说，众多的文件目录也许一开始让人觉得有些无所适从，但阅读完下面的使用说明后，会发现原来使用思路是简单而清晰的，而且，当对 LaTeX 有一定的认识和了解后，会发现其相对 Word 类排版系统极具吸引力的优秀特性。所以，如果是初学者，请不要退缩，请稍加尝试和坚持，以领略到 LaTeX 的非凡魅力，并可以通过阅读相关资料如 LaTeX Wikibook [**?**] 来完善自己的使用知识。

5.1　先试试效果

第 6 章　实验分析

先试试效果

第 7 章　总结与展望

附录 A 中国科学院大学学位论文撰写要求

学位论文是研究生科研工作成果的集中体现，是评判学位申请者学术水平、授予其学位的主要依据，是科研领域重要的文献资料。根据《科学技术报告、学位论文和学术论文的编写格式》（GB/T 7713-1987）、《学位论文编写规则》（GB/T 7713.1-2006）和《文后参考文献著录规则》（GB7714—87）等国家有关标准，结合中国科学院大学（以下简称“国科大”）的实际情况，特制订本规定。

A.1 论文无附录者无需附录部分

A.2 测试公式编号 $\Lambda, \lambda, \theta, \bar{\Lambda}, \sqrt{S_{NN}}$

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \\ \frac{\partial (\rho \boldsymbol{V})}{\partial t} + \nabla \cdot (\rho \boldsymbol{V} \boldsymbol{V}) = \nabla \cdot \boldsymbol{\sigma} \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \boldsymbol{V}) = \nabla \cdot (k \nabla T) + \nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{V}) \end{cases} \quad \ldots \text{(A.1)}$$

$$\frac{\partial}{\partial t} \int_{\Omega} u \, \mathrm{d}\Omega + \int_{S} \mathbf{n} \cdot (u \boldsymbol{V}) \, \mathrm{d}S = \dot{\phi} \quad \ldots \text{(A.2)}$$

$$\mathcal{L}\{f\}(s) = \int_{0^-}^{\infty} f(t) e^{-st} \, \mathrm{d}t, \; \mathscr{L}\{f\}(s) = \int_{0^-}^{\infty} f(t) e^{-st} \, \mathrm{d}t$$

$$\mathcal{F}\big(f(x + x_0)\big) = \mathcal{F}\big(f(x)\big) e^{2\pi i \xi x_0}, \; \mathscr{F}\big(f(x + x_0)\big) = \mathscr{F}\big(f(x)\big) e^{2\pi i \xi x_0}$$

mathtext: $A, F, L, 2, 3, 5, \sigma$, mathnormal: $A, F, L, 2, 3, 5, \sigma$, mathrm: $\mathrm{A}, \mathrm{F}, \mathrm{L}, 2, 3, 5, \sigma$.

mathbf: $\mathbf{A}, \mathbf{F}, \mathbf{L}, \mathbf{2}, \mathbf{3}, \mathbf{5}, \boldsymbol{\sigma}$, mathit: $\mathit{A}, \mathit{F}, \mathit{L}, 2, 3, 5, \sigma$, mathsf: $\mathsf{A}, \mathsf{F}, \mathsf{L}, 2, 3, 5, \sigma$.

mathtt: $\mathtt{A}, \mathtt{F}, \mathtt{L}, 2, 3, 5, \sigma$, mathfrak: $\mathfrak{A}, \mathfrak{F}, \mathfrak{L}, 2, 3, 5, \sigma$, mathbb: $\mathbb{A}, \mathbb{F}, \mathbb{L}, \mathbb{2}, \mathbb{3}, \mathbb{5}, \sigma$.

mathcal: $\mathcal{A}, \mathcal{F}, \mathcal{L}, 2, 3, 5, \sigma$, mathscr: $\mathscr{A}, \mathscr{F}, \mathscr{L}, 2, 3, 5, \sigma$, boldsymbol: $\boldsymbol{A}, \boldsymbol{F}, \boldsymbol{L}, 2, 3, 5, \boldsymbol{\sigma}$.

vector: $\boldsymbol{\sigma}, \boldsymbol{T}, \boldsymbol{a}, \boldsymbol{F}, \boldsymbol{n}$, unitvector: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

matrix: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$, unitmatrix: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

tensor: $\boldsymbol{\sigma}, \mathsf{\mathbf{T}}, \mathsf{\mathbf{a}}, \mathsf{\mathbf{F}}, \mathsf{\mathbf{n}}$, unittensor: $\boldsymbol{\sigma}, \mathsf{\mathbf{T}}, \mathsf{\mathbf{a}}, \mathsf{\mathbf{F}}, \mathsf{\mathbf{n}}$

A.3 测试生僻字

霜蟾盥薇曜灵霜颸妙鬘虚霩淩澌菀枯菡萏泬寥窅冥磑磑濩落雩雩便嬛岩峣瀺灂姽婳愔嫕飒纚棽俪緸冤莩甲摛藻卮言倥侗椒觞期颐夜阑彬蔚倥偬澄廓簪缨陟遐迤逦缥缃鹣鲽憯懔闺闼璀错媕婀噌吰澒洞阛阓覼缕玓瓑逡巡諓諓琭琭瀌瀌踽踽叆叇氤氲瓠犀流眄蹀躞赟嬛茕頔璎珞螓首蘅皋惏悷缱绻昶皴皱颟顸愀然菡萏卑陬纯懿犇麤掱暒墌墍墎墏墐墒墒墓墔墕墖墘墖墚墛坠墝增墠墡墢墣墤墥墦

墧墨墩墪樽墜墭堕墯墰墱墲坟墴墵垯墷墸墹墺墙墼墽垦墿壀壁壂壃壄壅壆坛壈
壉壊垱壌壍埙壏壐壑壒压壔壕壖壗垒圹垆壛壜壝垄壠壡坜壣壤壥壦壧壨坝壪圭
嫶嫷嫸嫹嫺娴嫼嫽嫾婳妫嬁嬂嬃嬄嬅嬆嬇娆嬉嬊娇嬍嬎嬏嬐嬑嬒嬓嬔嬕嬖嬗嬘
嬙嬚嬛嬜嬞嬟嬠嫒嬢嬣嬥嬦嬧嬨嬩嫔嬫嬬奶嬬嬮嬯婴嬱嬲嬳嬴嬵嬶嬷婶嬹嬺嬻
嬼嬽嬾嬿孀孁孂娘孄孅孆孇孆孈孉孊娈孋孊孍孎孏嫫婿媚嵭嵮嵯嵰嵱嵲嵳嵴
嵵嵶嵷嵸嵹嵺嵻嵼嵽嵾嵿嶀嵝嶂嶃崭嶅嶆岖嶈嶉嶊嶋嶌嶍嶎嶏嶐嶑嶒嶓岙嶕嶖
嶗嶘嶙嶚嶛嶜岛峤嶟峣嶡嶢嶣嶤峄峃嶧嶨嶩嶪嶫嶬嶭崄嶯嶰嶱嶲嶳岙嶵嶶嶷嶸嶹岭
巀屿岳帋巂巃巄巅巆巇巈巉巊巋巌巍巎巏巐巑巒峦巔巕巖岩巘巙巚帠帡帢
帣帤帨帩帪帬帯帰帱帲帴帵帷帹帺帻帼帽帾帿幁幂帏幄幅幆幇幈幉幊幋幌幍
幎幐幑幒幓幖幙幚幛幜幝幞帜幠幡幢币幤幥幦幧幨幩幪幭幮幯幰庁庂広庅庈庉
庌庍庎庒庘庛庝庡庢庣庤庥庨庩庪库庬庮庯庰庱庲庳庴庵庹庺庻庼庽庿廀厕廃厩廅廆廇廋廌
廍庼廏廐廑廒廔廕廖廗廘廙廛廜廞庑廤廥廦廧廨廭廮廯廰痈廲巡廸廹廻廼廽廿
弁弅弆弇弉弖弙弚弜弝弞弡弢弣弤弨弩弪弫弬弭弮弰弲弴弶弸弻弽弿彀彁彂
彃彄彅彆彇彉彋彍彎彏彐彑彔彚彛彜彝彟彴彵彶彷彸役彺彻彽彾佛徂徃徆徇徉后徍徎徏径徒従徔徕徖徙
徚徛徜徝从徟徕御徢徣徤徥徦徧徨复循徫旁徭微徯徰徱徲徳徴徵徶德徸彻徺徻
徼忀忁忂忇忈忊忋忎忓忔忕忚忛応忞忟忢忣忥忦忨忩忬忯忰忲忳忴忶忷忹忺忼怇
怈怉怋怌怐怑怓怗怘怚怞怟怢怣怤怬怭怮怰怱怲怳怴怶怷怸怹怺怽怾恀恄
恅恆恇恈恉恊恌恍恎恏恑恓恔恖恗恘恛恜恞恟恠恡恥恦恮恱恲恴恵恷恾悀
悁悂悅悆悇悈悊悋悎悏悐悑悓悕悗悘悙悜悞悡悢悤悥悧悩悪悮悰悳悵悶悷悹悺悽
悾惀惁惂惃惄惇惈惉惌惍惎惏惐惒惓惔惖惗惙惛惞惡惢惣惤惥惪惱惲惵惷惸惻
惼惽惾惿愂愃愄愅愇愊愋愌愐愑愒愓愔愖愗愘愙愛愜愝愞愡愢愥愨愩愪愬愭
愮愯愰愱愲愳愴愵愶愷愸愹愺愻愼愽愾慀慁慂慃慄

参考文献

[1] An updated set of basic linear algebra subprograms (BLAS) [J/OL]. ACM Transactions on Mathematical Software, 2002, 28(2): 135-151. DOI: 10.1145/567806.567807.

致 谢

感谢党和国家

作者简历及攻读学位期间发表的学术论文与研究成果

本科生无需此部分。

作者简历:

ucasthesis 作者

严愉程，浙江省宁波市人，中国科学院软件研究所，硕士研究生。

已发表（或正式接受）的学术论文:

1. ucasthesis: A LaTeX Thesis Template for the University of Chinese Academy of Sciences, 2014.

申请或已获得的专利:

（无专利时此项不必列出）

参加的研究项目及获奖情况:

可以随意添加新的条目或是结构。