ICS 35.240.50
CCS L 70

T/

团 体 标 准

T/CIE XXXX—2025

CAE通用隐式求解模块接口设计规范

CAE general design specifications for implicit solver module interface

XXXX－XX－XX 发布 XXXX－XX－XX 实施

发 布

目　　次

前言 II
1　范围 1
2　规范性引用文件 1
3　术语和定义 1
3.1　CAE 隐式求解器接口 1
3.2　数学库接口 1
3.3　特征值求解器接口 2
3.4　拓扑优化模块接口 3
3.5　非凸优化模块接口 3
4　缩略语 3
5　CAE 通用底座架构设计原则 3
6　数据类型定义 4
6.1　单元数据（data_t） 4
6.2　矩阵（matrix_t） 4
7　CAE 隐式求解器接口设计 4
7.1　单元计算模块 4
7.2　载荷计算模块 6
7.3　静力分析模块 8
7.4　模态分析 9
7.5　瞬态分析 11
8　通用数学库接口设计 13
8.1　核心概念 13
8.2　基本线性代数子程序（Basic Linear Algebra Subprograms） 13
8.3　稀疏矩阵 SM（Sparse Matrix） 18
8.4　稀疏求解器 SS（Sparse Solver） 24
9　特征值求解器接口设计 28
9.1　广义共轭梯度算法 GCGE 接口 28
9.2　并行扩展子空间算法 PASE 接口 29
10　拓扑优化模块接口设计 29
10.1　基于 MMC 进行拓扑优化设计 29
10.2　基于 SIMP 进行拓扑优化设计 30
10.3　基于水平集方法进行拓扑优化设计 30
11　非凸优化模块接口设计 31
11.1　标准优化问题模型类 31
11.2　对标准优化问题（Problem 类对象）求解 31
11.3　优化求解器核心函数 32

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国电子学会提出并归口。

本文件起草单位：大连理工大学、澎峰（北京）科技有限公司、中科院数学与系统科学研究院、大连星派科技有限公司、华南理工大学、武汉大学、重庆大学、河海大学、大连工业软件创新发展研究院。

本文件主要起草人：郭旭、梅跃、杜宗亮、刘畅、肖现涛、张先轶、谢和虎、魏鹏、张亮、刘辉、刘兆玮。

CAE 通用隐式求解模块接口设计规范

1 范围

本文件规定了通用CAE求解引擎模块接口规范的基本原则与接口详细设计标准。

本文件适用于CAE软件研发企业和CAE有限元仿真企业在设计、开发和使用CAE求解引擎模块接口时参考。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1 CAE 隐式求解器接口

3.1.1

刚度阵 stiffness matrix

描述结构抵抗变形的能力的矩阵，由材料属性和几何形状决定，通常表示为对称矩阵，用于有限元分析中计算结构在载荷作用下的位移响应。

3.1.2

质量阵 mass matrix

描述结构惯性特性的矩阵，反映质量分布对其动力学行为的影响，在振动分析中与刚度阵共同决定系统的固有频率和振型。

3.1.3

形函数 shape function

定义在单元节点上的插值函数，用于描述有限元法中近似单元内部的位移场，通过节点位移插值得到连续位移分布，是单元刚度阵和质量阵计算的基础。

3.1.4

载荷 load

作用在结构上的外力或激励，包括集中力、分布力、惯性力等，影响结构的静力学和动力学响应。

3.1.5

特征值 eigenvalue

在结构动力学中，用于求解系统的固有频率和振型，特征值的平方根对应系统的自由振动频率。结构动力学中的特征值分析揭示了系统的内在振动特性，是避免共振、优化设计的基础。

3.1.6

模态空间 modal space

由系统的固有振型（模态）张成的空间，用于将耦合的动力学方程解耦，简化振动分析和响应计算。

3.1.7

等效刚度阵 equivalent stiffness matrix

在动态响应计算中，由原刚度阵、质量阵和阻尼阵加权组合而成的矩阵，用于将时变载荷下的动力学方程转化为等效静力问题。

3.2 数学库接口

3.2.1

队列 queue

一种线性的数据结构，负责在各种计算设备上管理执行队列。它抽象了队列管理的复杂性，允许开发者针对特定设备（如CPU、GPU等）初始化、同步和管理执行队列。

3.2.2

内存缓冲区 buffer

用于管理和操作在特定队列上分配的内存缓冲区。该类提供了一系列构造函数、拷贝控制操作、以及数据传输方法，支持在主机和设备间进行数据的高效传输。

3.2.3

向量 vector

一维线性数据集合，支持标量和元素级运算。在数值计算中用于存储有序数值序列，是线性代数的基础结构，常用于表示空间中的点、方向或方程组右侧的常量项。

3.2.4

矩阵 matrix

二维数值数组，表示线性变换或线性方程组。支持矩阵乘法、转置等操作，是数值计算的核心结构，广泛应用于图像处理、机器学习和物理模拟等领域。

3.2.5

三角矩阵 triangular matrix

非零元素仅位于主对角线一侧的矩阵，分为上三角矩阵（非零元素仅位于主对角线及以上）和下三角矩阵（非零元素仅位于主对角线及以下）。这类矩阵在LU分解和高斯消元法中起关键作用，可简化线性方程组的求解过程。

3.2.6

带状矩阵 band matrix

非零元素集中在主对角线附近的带状区域（如三对角矩阵），适用于稀疏矩阵的高效存储和运算。在有限差分法或有限元分析中常见，可显著减少内存占用和计算复杂度。

3.2.7

正定矩阵 positive definite matrix

对称且所有特征值为正的矩阵，在优化问题、Cholesky分解和二次型分析中起核心作用，确保数值稳定性和唯一解的存在性。

3.3 特征值求解器接口

3.3.1

广义共轭梯度特征值求解器 generalized conjugate gradient eigensolver；GCGE

基于广义共轭梯度算法设计的大规模对称矩阵特征值问题并行解法器。对应的求解算法称为GCG(Generalized Conjugate Gradient)算法，该算法是一种子空间投影方法。

3.3.2

并行扩展子空间特征求解器 parallel augmented subspace eigensolver；PASE

基于扩展子空间算法和多重网格设计的大规模特征值问题解法器,结合提供的几何或者代数多重网格的信息，使得大规模特征值问题求解的计算量和边值问题求解的计算量相当。

3.3.3

单向量 single vector

并行数值计算的核心组件，专为高效处理分布式数据设计。支持分布式存储，允许在不同MPI进程间划分数据，适用于迭代求解器中的右端项和未知量存储。

3.3.4

多向量 multiple vectors

在单向量基础上的扩展，支持对多个向量的批处理操作，通过数据局部性优化减少通信开销。在Krylov子空间方法、特征值计算等场景中发挥关键作用，尤其适合大规模科学计算问题的高效求解。

3.3.5

稀疏矩阵 sparse matrix

专为大规模并行计算设计的高效数据结构，采用分布式存储策略，将矩阵非零元素划分到多个MPI进程上，显著降低内存需求和通信开销。

3.4 拓扑优化模块接口

3.4.1

移动可变形组件拓扑优化方法 moving morphable component topology optimization method

以可移动、变形的组件为设计基元进行拓扑优化的方法，每个组件的设计参数包括组件中心坐标，组件的半长、半宽、半高以及在空间中的三个旋转角度。

3.4.2

固体各向同性材料惩罚拓扑优化方法 solid isotropic material with penalization topology optimization method

以材料密度分布为设计变量进行拓扑优化的方法，通过有限元网格离散设计域，每个单元赋予相对密度值作为变量，每个单元的设计变量为连续密度值，采用固体各向同性材料惩罚模型建立宏观弹性模量插值关系。

3.4.3

水平集方法 level set topology optimization method

采用基于水平集方法的隐式表达，以水平集函数的描述参数为设计变量进行拓扑优化，在基于有限元方法的结构分析框架下，以有限单元的线性形函数为基函数，以节点水平集函数值为设计参数；优化目标函数为柔顺度，约束为体积约束。

3.5 非凸优化模块接口

3.5.1

标准优化问题模型 standard optimization problem model

在给定约束条件下，寻找使目标函数最小化（或最大化）的变量值的问题模型。通过数学形式描述现实决策问题，这一模型是运筹学、机器学习、工程优化等领域的基石。

3.5.2

盒式约束 box constraint

优化问题中对决策变量直接显式定义上下界的约束形式，仅需指定每个变量的取值范围，无需复杂约束函数，广泛应用于物体系统、超参数优化和资源分配等场景。

4 缩略语

下列缩略语适用于本文件。

BLAS：基础线性代数子程序（Basic Linear Algebra Subprograms）
CAE：计算机辅助工程（Computer Aided Engineering）
COO：坐标存储（Coordinate）
CSR：压缩行存储（Compressed Sparse Row）
CSC：压缩列存储（Compressed Sparse Column）
MMC：移动可变形组件（Moving Morphable Component）
GCGE：广义共轭梯度特征值求解器（Generalized Conjugate Gradient Eigensolver）
PASE：并行扩展子空间特征求解器（Parallel Augmented Subspace Eigensolver）
SIMP：固体各向同性材料惩罚(Solid Isotropic Material with Penalization)
SpTRSV：稀疏三角矩阵求解（Sparse Triangular Solve）
SpMM：稀疏矩阵-矩阵乘法（Sparse Matrix-Matrix Multiplication）
SS：稀疏求解器（Sparse Solver）

5 CAE 通用底座架构设计原则

CAE通用底座架构设计应遵循以下原则：

a) 开放性与可替代性：支持功能模块的横向扩展与动态替换，确保第三方或自研模块可通过标准化接口接入，兼容主流商业与开源工具链。

b) 标准化设计：统一接口协议、数据格式及异常处理规范，采用跨行业通用标准，降低跨平台协作成本。

6 数据类型定义

6.1 单元数据（data_t）

单元数据的数据模型定义如表1所示。

表 1 单元数据的数据模型定义说明

属性类别	中文名称	英文名称	定义	值类型	长度	是否必需	约束	备注
基本信息	单元唯一标识	element_id	单元的唯一标识符	int	4 字节	是	全局唯一	用于区分不同单元
拓扑信息	节点数量	num_nodes	单元包含的节点数量	int	4 字节	是	> 0	例如三角形单元为 3
空间信息	空间维度	dimension	单元所在的空间维度	int	4 字节	是	2（2D）或 3（3D）	决定节点坐标的维度
连接关系	节点ID数组	node_ids	构成单元的节点ID列表	int*	num_nodes* 4 字节	是	数组长度 = num_nodes	存储节点的全局编号
材料属性	材料ID	material_id	关联的材料属性ID	int	4 字节	是	对应材料库中的记录	用于指定单元的材料特性

6.2 矩阵（matrix_t）

矩阵的数据模型定义如表2所示。

表 2 数据模型定义说明

属性类别	中文名称	英文名称	定义	值类型	长度	是否必需	约束	备注
结构信息	行数	rows	矩阵的行数	int	4 字节	是	>= 1	
结构信息	列数	cols	矩阵的列数	int	4 字节	是	>= 1	
数据存储	数据指针	data	矩阵数据存储区域	double*	rows×cols×8 字节	是	列优先存储	按列优先排列

7 CAE 隐式求解器接口设计

7.1 单元计算模块

7.1.1 单元刚度矩阵计算接口

单元刚度矩阵计算接口根据所给单元的节点坐标、连接关系以及材料参数（如杨氏模量、泊松比），计算该单元的局部刚度矩阵。刚度矩阵反映了单元在单位位移下所产生的内力，是有限元法中求解全局刚度矩阵的基本组成部分。该接口不依赖任何全局变量，计算结果存储于由调用方分配的输出矩阵中，可用于结构静力分析、模态分析及动力响应等，具体接口定义如表3所示：

表 3 单元刚度矩阵计算接口定义

接口名	compute_stiffness_matrix			
描述	单元刚度矩阵计算			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	elem：单元数据	data_t	包括节点坐标、连接关系和材料属性等。调用方需设置num_nodes、node_ids和material_id等字段。	是
二、出参				

表3　单元刚度矩阵计算接口定义（续）

接口名	compute_stiffness_matrix			
序号	参数名	数据类型	描述	是否必填
1	K_out	matrix_t	输出刚度矩阵的地址，调用方需确保K_out->rows = dof, K_out->cols = dof，K_out->data已分配空间（dof = 单元总自由度）。函数计算所得刚度矩阵存储于data。	是
三、返回值				
数据类型			描述	
int			0表示成功，非零为错误码。	

7.1.2　单元质量矩阵计算

单元质量矩阵计算接口计算有限元单元的局部质量矩阵，通常用于模态分析或动力响应分析中。质量矩阵反映了单元的惯性特性，基于节点质量分布及单元密度生成。函数通过数值积分或解析形式构建，调用者需提供单元几何信息和材料密度，输出由调用方提供存储空间，具体接口定义如表4所示：

表 4　单元质量矩阵计算接口定义

接口名	compute_mass_matrix			
描述	单元质量矩阵计算			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	elem：单元数据	data_t	包括节点坐标、连接关系和材料属性等。调用方需设置num_nodes、node_ids和material_id等字段。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	M_out	matrix_t	输出质量矩阵的地址，调用方需确保M_out->rows = dof, M_out->cols = dof，M_out->data已分配空间（dof = 单元总自由度）。函数计算所得刚度矩阵存储于data。	是
三、返回值				
数据类型			描述	
int			0 表示成功，非零为错误码。	

7.1.3　形函数值及导数计算接口

此函数在给定局部坐标系下（例如高斯积分点）计算形函数值及其空间导数，用于构造插值函数、刚度矩阵和应力应变等计算。形函数定义了单元内场变量的插值关系，是有限元法的核心组成部分。该函数支持不同维度和阶次的单元，输出包括节点的插值系数（N）及其对局部坐标的偏导数（dN），可用于积分、应变-位移关系构建等任务，具体接口定义如表5所示：

表 5　形函数值及导数计算接口定义

接口名	compute_shape_functions			
描述	形函数值及导数计算			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	elem：单元数据	data_t	包括节点坐标、连接关系和材料属性等。调用方需设置num_nodes、node_ids和material_id等字段。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	N_out	matrix_t	输出形函数值数组，长度 = num_nodes。	是
2	dN_out	matrix_t	输出形函数导数数组，长度 = num_nodes * dimension，按节点依次存储每个方向导数。	是
三、返回值				
数据类型			描述	
int			0表示成功，非零为错误码。	

7.2 载荷计算模块

7.2.1 概述

该模块用于根据外部作用计算等效结点载荷或全局载荷向量。结构体与函数定义在load::*命名空间。

7.2.2 单个结点集中力的全局载荷向量计算接口

计算作用于指定节点和自由度的集中点载荷，将其转换为局部节点力向量，用于后续全局力向量的组装与分析处理，具体接口定义如表6所示：

表 6 单个结点集中力的全局载荷向量计算接口定义

接口名	compute_point_load			
描述	单个结点集中力的全局载荷向量计算			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	mesh	data_t	网格模型。	是
2	pl	data_t	点载荷数据，包括node_id和force。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	F_out	matrix_t	输出全局载荷向量,长度 = mesh->num_nodes * mesh->dimension，调用方需分配，力分量填充对应结点对应自由度的位置。	是
三、返回值				
数据类型			描述	
int			0 表示成功，非零为错误码。	

7.2.3 计算作用在单元表面上的均布压力等效结点载荷

根据施加在单元表面上的压力载荷进行数值积分，计算其等效节点力，对应于真实工况下的面压力分布效果，具体接口定义如表7所示：

表 7 单元表面上的均布压力等效结点载荷计算接口定义

接口名	compute_surface_pressure			
描述	计算作用在单元表面上的均布压力等效结点载荷			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	mesh	data_t	网格模型。	是
2	sl	data_t	面载荷数据，包括element_id、pressure和normal。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	F_out	matrix_t	输出全局载荷向量，调用方需分配，相应单元的结点力分配到对应位置。	是
三、返回值				
数据类型			描述	
int			0 表示成功，非零为错误码。	

7.2.4 计算体力的单元等效结点载荷

通过对单元内部体力（如重力、惯性力）进行体积分，将体力转换为等效节点载荷，用于动力学或静力学分析，具体接口定义如表8所示：

表 8 体力转换为等效节点载荷接口定义

接口名	compute_body_force

表8　体力转换为等效节点载荷接口定义（续）

接口名	compute_body_force			
描述	计算体力的单元等效结点载荷			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	elem	data_t	单元数据。	是
2	acceleration	matrix_t	作用于单元的加速度向量（长度 = dimension）。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	F_out	matrix_t	输出单元等效结点载荷，长度 = dof（需分配）。	是
三、返回值				
数据类型			描述	
int			0 表示成功，非零为错误码。	

7.2.5　计算作用在一维单元上的线性分布载荷等效结点力

将边界或梁构件上的线性分布载荷进行积分转换为节点力，可考虑变截面、角度或分布函数影响，具体接口定义如表9所示：

表 9　计算一维单元上的线性分布载荷等效结点力接口定

接口名	compute_line_load			
描述	计算一维单元上的线性分布载荷等效结点力			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	mesh	data_t	网格模型。	是
2	ll	data_t	线载荷数据，包含element_id和intensity。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	F_out	matrix_t	输出单元等效结点载荷，长度 = dof（需分配）。	是
三、返回值				
数据类型			描述	
int			0 表示成功，非零为错误码。	

7.2.6　组合多个点载荷和面载荷生成全局载荷向量

将各类载荷（点、面、线、体力）转换后的局部力矢量进行叠加，组装为系统求解所需的全局力向量，具体接口定义如表10所示：

表 10　计算一维单元上的线性分布载荷等效结点力接口定义

接口名	compute_line_load			
描述	计算一维单元上的线性分布载荷等效结点力			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	mesh	data_t	网格模型。	是
2	point_loads	matrix_t	点载荷数组，长度num_point。	是
3	surface_loads	matrix_t	面载荷数组，长度num_surface。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	F_out	matrix_t	输出全局载荷向量，调用方需分配。函数会累加各载荷贡献。	是

表10　计算一维单元上的线性分布载荷等效结点力接口定义（续）

三、返回值	
数据类型	描述
int	0 表示成功，非零为错误码。

7.3　静力分析模块

7.3.1　概述

静力分析模块用于求解结构在静载荷作用下的响应，通过构建和求解线性方程组获得节点位移、单元应力等关键结果。下列接口均在static::*命名空间中实现。

7.3.2　求解有限元静力平衡方程

该函数用于求解有限元静力平衡方程 KU = F，通过直接解法计算节点的全局位移结果，是后续应力和反力分析的基础，具体接口定义如表11所示：

表 11　求解有限元静力平衡方程接口定义

接口名	solve			
描述	求解有限元静力平衡方程			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	global_stiffness	matrix_t	系统整体刚度矩阵，维度为n×n，必须为对称正定。	是
2	global_force	matrix_t	节点外力向量，维度为 n×1。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	global_displacement	matrix_t	输出位移向量，维度为n×1，调用前需分配内存。	是
三、返回值				
数据类型			描述	
int			0 表示求解成功；非 0 表示输入非法或解算失败。	

7.3.3　计算指定节点的各自由度位移

从全局位移向量中提取指定节点的各自由度位移信息，支持多自由度节点，便于后处理和结果可视化等模块使用，具体接口定义如表12所示：

表 12　计算指定节点的各自由度位移接口定义

接口名	get_nodal_displacement			
描述	计算指定节点的各自由度位移			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	global_displacement	matrix_t	整体自由度位移向量，维度为 ndof×1。	是
2	node_index	matrix_t	目标节点编号，从 0 开始。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	nodal_displacement	matrix_t	目标节点的局部位移。	是
三、返回值				
数据类型			描述	
int			0 表示成功；非 0 表示索引越界或参数错误。	

7.3.4　计算单元在当前变形状态下的应力场

根据单元自由度位移向量、应变-位移矩阵B以及材料刚度矩阵D，计算单元在当前变形状态下的应力场，是有限元后处理中的关键步骤，具体接口定义如表13所示：

表 13　计算单元在当前变形状态下的应力场接口定义

接口名	compute_element_stress			
描述	计算单元在当前变形状态下的应力场			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	element_displacement	matrix_t	单元自由度位移，维度为 ndof×1。	是
2	B_matrix	matrix_t	应变-位移矩阵，维度为 strain_size×ndof。	是
3	D_matrix	matrix_t	材料本构矩阵，维度为 strain_size×strain_size。	
二、出参				
序号	参数名	数据类型	描述	是否必填
1	stress_output	matrix_t	输出应力向量，维度为 strain_size×1，如二维问题中为 3 分量。	是
三、返回值				
数据类型			描述	
int			0 表示应力计算成功；非 0 表示维度不一致或输入无效。	

7.4　模态分析

7.4.1　概述

模态分析用于计算结构在自由振动下的固有频率和模态形状。该模块包含解算广义特征值问题、频率转换和模态转换等函数，位于modal::*命名空间。

7.4.2　模态特征值求解

求解结构广义特征值问题 $K\phi = \lambda M\phi$，该函数是模态分析的核心步骤，返回最低阶若干模态频率及其模态向量，常用稀疏迭代法或LAPACK库进行高效求解，具体接口定义如表14所示：

表 14　模态特征值求解接口定义

接口名	solve_eigenproblem			
描述	模态特征值求解			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	K, M	matrix_t	全局刚度矩阵和质量矩阵，均为对称正定矩阵。	是
2	num_modes	matrix_t	求解的模态数量（≤矩阵阶数）。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	eigenvalues	matrix_t	长度为num_modes的数组，返回每个模态的特征值（λ_i）。	是
2	eigenvectors	matrix_t	输出特征向量矩阵，调用方需设置rows = dof, cols = num_modes并分配数据。第i列对应第i模态的振型向量。	是
三、返回值				
数据类型			描述	
int			0 表示成功；非 0 表示输入无效。	

7.4.3　物理坐标与模态坐标转换

将物理坐标下的位移向量转换为模态坐标系中的模态响应。函数通过计算 $q = \Phi^T M u$ 实现模态投影，是模态叠加法、降阶模拟等方法的基础步骤，具体接口定义如表15所示：

表 15　物理坐标与模态坐标转换接口定义

表 15　物理坐标与模态坐标转换接口定义（续）

接口名	transform_to_modal			
描述	物理坐标与模态坐标转换			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	eigenvectors	matrix_t	特征向量矩阵，大小为dof × num_modes。	是
2	M	matrix_t	全局质量矩阵。	是
3	u_phys	matrix_t	物理坐标下的位移向量，长度 = dof。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	u_modal	matrix_t	模态坐标向量，长度 = num_modes。	是
三、返回值				
数据类型			描述	
int			0 表示成功；非 0 表示输入无效。	

7.4.4　模态坐标与物理坐标转换

将模态坐标系下的模态位移组合转换回实际物理空间中的全局位移。通过 $u = \Phi q$ 公式完成变换，用于可视化模态响应或进行频域、时域响应合成，具体接口定义如表16所示：

表 16　模态坐标与物理坐标转换接口定义

接口名	transform_to_physical			
描述	模态坐标与物理坐标转换			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	eigenvectors	matrix_t	特征向量矩阵，大小为dof × num_modes。	是
2	u_modal	matrix_t	模态位移向量，长度 = num_modes。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	u_phys	matrix_t	输出的物理位移向量，长度 = dof。	是
三、返回值				
数据类型			描述	
int			0 表示成功；非 0 表示输入无效。	

7.4.5　计算模态有效质量等

计算每个模态所对应的有效质量，用于评估模态对整体结构响应的贡献程度。函数通过 $m_{\text{eff}} = \phi^T M \phi$ 形式获取，用于模态参与比计算或振动能量分布分析，具体接口定义如表17所示：

表 17　计算模态有效质量等接口定义

接口名	compute_effective_mass			
描述	计算模态有效质量等			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	M	matrix_t	质量矩阵。	是
2	eigenvectors	matrix_t	特征向量矩阵（dof × num_modes）。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	u_phys	matrix_t	输出的物理位移向量，长度 = dof。	是

表17　计算模态有效质量等接口定义（续）

三、返回值	
数据类型	描述
int	0 表示成功；非 0 表示输入无效。

7.5　瞬态分析

7.5.1　概述

时程分析用于计算结构随时间的动态响应，输出随时间变化的位移、应力等结果。所解算方程为$M\ddot{u}+C\dot{u}+Ku=F(t)$，通常采用Newmark等时间积分方法。

7.5.2　初始化瞬态分析状态

初始化瞬态分析状态，包括时间步长、初始响应量（位移、速度、加速度），以及积分参数设定，具体接口定义如表18所示：

表 18　初始化瞬态分析状态接口定义

接口名	initialize			
描述	初始化瞬态分析状态			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	time_step	float	时间步长，单位为秒。	是
2	total_steps	int	总时间步数量。	是
3	initial_displacement	matrix_t	初始位移向量。	是
4	initial_velocity	matrix_t	初始速度向量。	是
5	initial_acceleration	matrix_t	初始加速度向量。	是
6	params	matrix_t	时间积分参数，如Newmark-beta的beta和gamma值。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无			否

7.5.3　计算等效载荷

基于当前时间点响应与积分参数，计算等效外载荷，用于动态平衡方程的右端项，具体接口定义如表19所示：

表 19　计算等效外载荷接口定义

接口名	compute_effective_force			
描述	计算等效外载荷			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	external_force	matrix_t	当前时刻的外部载荷。	是
2	displacement, velocity, acceleration	matrix_t	当前状态向量。	是
3	mass_matrix	matrix_t	结构质量矩阵。	是
4	damping_matrix	matrix_t	阻尼矩阵。	是
6	params	matrix_t	时间积分参数。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		计算得到的等效载荷向量。	是

7.5.4 计算等效刚度阵

构造当前时间步下的等效刚度矩阵，用于瞬态响应求解的左端项，具体接口定义如表20所示：

表 20 计算等效刚度阵接口定义

接口名	compute_effective_stiffness			
描述	计算等效刚度阵			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	stiffness_matrix	matrix_t	原始刚度矩阵。	是
2	mass_matrix	matrix_t	质量矩阵。	是
3	damping_matrix	matrix_t	阻尼矩阵。	是
4	params	matrix_t	积分方法参数。	是
6	time_step	matrix_t	当前时间步长。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		等效刚度矩阵	是

7.5.5 瞬态积分状态更新

执行隐式积分，求解当前时刻的响应变量（位移、速度、加速度），具体接口定义如表21所示：

表 21 瞬态积分状态更新接口定义

接口名	update_state			
接口名	update_state			
描述	瞬态积分状态更新			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	effective_stiffness	matrix_t	上一步计算的等效刚度矩阵。	是
2	effective_force	matrix_t	当前时间点的等效载荷。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	displacement, velocity, acceleration	matrix_t	更新后的响应变量。	是

7.5.6 瞬态分析主控函数

瞬态分析主控函数协调各子模块执行全过程求解与输出，具体接口定义如表22所示：

表 22 瞬态分析主控函数接口定义

接口名	solve			
描述	瞬态分析主控函数			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	model	data_t	包含质量、刚度、阻尼矩阵的有限元模型。	是
2	time_settings	matrix_t	时间步长、总时间等控制信息。	是
3	load_manager	matrix_t	管理各时间步外载荷。	是
4	params	matrix_t	数值积分参数。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		过程输出由子函数处理	否

8 通用数学库接口设计

8.1 核心概念

以下是数学库中各类数据结构的简要描述，它们在数值计算、优化问题以及其他相关领域中发挥着重要作用，具体内容如表23所示：

表 23 数学库

名称	描述
queue	负责在各种计算设备上管理执行队列。它抽象了队列管理的复杂性，允许开发者针对特定设备（如CPU、GPU 等）初始化、同步和管理执行队列。
buffer	用于管理和操作在特定队列上分配的内存缓冲区。该类提供了一系列构造函数、拷贝控制操作、以及数据传输方法，支持在主机和设备间进行数据的高效传输。
向量	表示一维线性数据集合，支持标量和元素级运算。在数值计算中用于存储有序数值序列，是线性代数的基础结构，常用于表示空间中的点、方向或方程组右侧的常量项。
矩阵	二维数值数组，表示线性变换或线性方程组。支持矩阵乘法、转置等操作，是数值计算的核心结构，广泛应用于图像处理、机器学习和物理模拟等领域。
三角矩阵	分为上三角矩阵（非零元素仅位于主对角线及以上）和下三角矩阵（非零元素仅位于主对角线及以下）。这类矩阵在 LU 分解和高斯消元法中起关键作用，可简化线性方程组的求解过程。
带状矩阵	非零元素集中在主对角线附近的带状区域（如三对角矩阵），适用于稀疏矩阵的高效存储和运算。在有限差分法或有限元分析中常见，可显著减少内存占用和计算复杂度。
正定矩阵	对称矩阵且所有特征值为正，在优化问题、Cholesky分解和二次型分析中起核心作用，确保数值稳定性和唯一解的存在性。

8.2 基本线性代数子程序（Basic Linear Algebra Subprograms）

8.2.1 计算两个实向量的点积。

dot函数执行两个向量的点积计算，具体接口定义如表24所示：

表 24 瞬态分析主控函数接口定义

接口名	dot			
描述	瞬态分析主控函数			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	n	int	指向向量 x 和 y 中的元素个数。	是
3	x	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
4	incx	int	向量X的步长。	是
5	y	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是
6	incy	int	向量y的步长。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	result	matrix_t	结果存放的地址。	是

8.2.2 将一个向量复制到另一个向量。

copy函数将一个向量复制到另一个向量，具体接口定义如表25所示：

表 25 将一个向量复制到另一个向量接口定义

接口名	copy

表 25　将一个向量复制到另一个向量接口定义（续）

接口名	copy			
描述	将一个向量复制到另一个向量			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	n	int	指向向量 x 和 y 中的元素个数。	是
3	x	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
4	incx	int	向量X的步长。	是
5	y	int	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是
6	incy	int	向量y的步长。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	y	matrix_t	存储输入/输出向量y的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是

8.2.3　计算向量标量乘积并将结果添加到向量中。

axpy 函数计算标量-向量乘积并将结果添加到一个向量中，具体接口定义如表26所示：

表 26　计算向量标量乘积并将结果添加到向量中接口定义

接口名	axpy			
接口名	axpy			
描述	计算向量标量乘积并将结果添加到向量中			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	n	int	指向向量 x 和 y 中的元素个数。	是
3	x	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
4	incx	int	向量X的步长。	是
5	y	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是
6	incy	int	向量y的步长。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	y	matrix_t	存储输入/输出向量y的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是

8.2.4　计算向量与标量的乘积

scal 函数计算标量-向量乘积，具体接口定义如表27所示：

表 27　计算向量与标量的乘积接口定义

接口名	scal			
描述	计算向量与标量的乘积			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是

表 27　计算向量与标量的乘积接口定义（续）

序号	参数名	数据类型	描述	是否必填
2	n	int	指向向量 x 和 y 中的元素个数。	是
3	x	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
4	incx	int	向量X的步长。	是
5	y	int	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是
6	incy	int	向量y的步长。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	x	matrix_t	存储输入/输出向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是

8.2.5　计算向量与标量的乘积

swap 函数给定两个具有n个元素的向量x和y，swap例程返回交换后的向量y和x，每个向量替换另一个，具体接口定义如表28所示：

表 28　计算向量与标量的乘积接口定义

接口名	swap			
描述	计算向量与标量的乘积			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	n	int	指向向量x和y中的元素个数。	是
3	x	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
4	incx	int	向量X的步长。	是
5	y	int	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是
6	incy	int	向量y的步长。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	x	matrix_t	存储输入/输出向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
2	y	matrix_t	存储输入/输出向量y的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是

8.2.6　使用一般矩阵计算矩阵-向量乘积

gemv 函数计算标量-矩阵-向量乘积并将结果加到标量-向量乘积上，使用的是一般矩阵，具体接口定义如表29所示：

表 29　使用一般矩阵计算矩阵-向量乘积接口定义

接口名	gemv			
描述	使用一般矩阵计算矩阵-向量乘积			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是

表 29　使用一般矩阵计算矩阵-向量乘积接口定义（续）

2	n	int	指向向量 x 和 y 中的元素个数。	是
3	x	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incx))。	是
4	incx	int	向量X的步长。	是
5	y	matrix_t	存储输入向量x的缓冲区。该缓冲区的大小必须至少为(1 + (n - 1)*abs(incy))。	是
6	incy	int	向量y的步长。	是
7	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
8	trans	int	指定应用于矩阵A　的转置操作。	是
9	m	int	指定矩阵op(A)的行数。m的值必须至少为零。	是
10	n	int	指定矩阵op(B)的列数。n的值必须至少为零。	是
11	alpha	float	矩阵-向量积的缩放因子	是
12	a	matrix_t	存储输入矩阵A的缓冲区。	是
13	lda	int	矩阵A的主导维度。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	y	matrix_t	存储输入/输出向量y的缓冲区。	是

8.2.7　计算三角矩阵-向量乘积

trmv函数计算一个三角矩阵的矩阵-向量乘积，具体接口定义如表30所示：

表 30　计算一个三角矩阵的矩阵-向量乘积接口定义

接口名	trmv			
描述	计算一个三角矩阵的矩阵-向量乘积			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	n	int	指向向量x和y中的元素个数。	是
3	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
4	trans	int	指定应用于矩阵A　的转置操作。	是
5	n	int	指定矩阵op(B)的列数。n的值必须至少为零。	是
6	a	matrix_t	存储输入矩阵A的缓冲区。	是
7	lda	int	矩阵A的主导维度。	是
8	x	matrix_t	存储输入/输出向量x的缓冲区。	是
9	incx	int	向量x的步长。	是
10	upper_lower	char	指定矩阵 A 是上三角还是下三角。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	x	matrix_t	存储输入/输出向量y的缓冲区。	是

8.2.8　求解系数为三角矩阵的线性方程组

trsv 函数求解系数为三角矩阵的线性方程组，具体接口定义如表31所示：

表 31　求解系数为三角矩阵的线性方程组接口定义

接口名	trsv

表31　求解系数为三角矩阵的线性方程组接口定义（续）

接口名	trsv			
描述	求解系数为三角矩阵的线性方程组			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	n	int	指向向量 x 和 y 中的元素个数。	是
3	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
4	trans	int	指定应用于矩阵A　的转置操作。	是
5	n	int	指定矩阵op(B)的列数。n的值必须至少为零。	是
6	a	matrix_t	存储输入矩阵A的缓冲区。	是
7	lda	int	矩阵A的主导维度。	是
8	x	matrix_t	存储输入/输出向量x的缓冲区。	是
9	incx	int	向量x的步长。	是
10	upper_lower	char	指定矩阵 A 是上三角还是下三角。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	x	matrix_t	存储输入/输出向量y的缓冲区。	是

8.2.9　计算通用矩阵-矩阵乘积

GEMM函数计算两个矩阵的乘积（可能对输入矩阵进行转置或共轭转置操作），并将结果缩放后与另一标量缩放的矩阵相加，具体接口定义如表32所示：

表 32　计算通用矩阵-矩阵乘积接口定义

接口名	gemm			
描述	计算通用矩阵-矩阵乘积			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	transa	int	指定应用于矩阵A的转置操作。	是
4	transb	int	指定应用于矩阵B的转置操作。	是
5	m	int	指定矩阵op(A)和矩阵C的行数m的值必须至少为零。	是
6	n	int	指定矩阵op(B)和矩阵C的列数。n的值必须至少为零。	是
7	k	int	指定矩阵op(A)的列数和矩阵op(B)的行数。k的值必须至少为零。	是
8	alpha	float	矩阵-矩阵乘积的缩放因子。	是
9	a	matrix_t	存储输入矩阵A的缓冲区。	是
10	lda	int	矩阵A的主导维度。	是
	b	matrix_t	存储输入矩阵B的缓冲区。	是
	ldb	int	矩阵B的主导维度。	是
	beta	float	矩阵C的缩放因子。	是
	c	matrix_t	存放输入/输出矩阵C的缓冲区。	是
	ldc	int	矩阵C的主导维度。	是
二、出参				

表 32　计算通用矩阵-矩阵乘积接口定义（续）

序号	参数名	数据类型	描述	是否必填
1	c	matrix_t	存放输入/输出矩阵C的缓冲区。	是

8.2.10　计算矩阵-矩阵乘积

trmm函数计算两个矩阵的乘积（可能对输入矩阵进行转置或共轭转置操作），其中一个矩阵是三角矩阵，另一个为通用矩阵，具体接口定义如表33所示：

表 33　计算矩阵-矩阵乘积接口定义

接口名	trmm			
描述	计算矩阵-矩阵乘积			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	执行操作的队列。	是
2	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	left_right	char	指定 A 是在乘法的左边(side::left)还是右边(side::right)。	是
4	upper_lower	char	指定是使用矩阵 A 的上三角形部分还是下三角形部分。	是
5	transa	int	指定应用于矩阵A的转置操作。	是
6	unit_diag	char	指定矩阵A是否为单位三角矩阵。	是
7	m	int	指定矩阵op(A)和矩阵C的行数m的值必须至少为零。	是
8	n	int	指定矩阵op(B)和矩阵C的列数。n的值必须至少为零。	是
9	alpha	matrix_t	矩阵-矩阵乘积的缩放因子。	是
10	a	matrix_t	存储输入矩阵A的缓冲区。	是
11	lda	matrix_t	矩阵A的主导维度。	是
12	b	matrix_t	存储输入矩阵B的缓冲区。	是
13	ldb	matrix_t	矩阵B的主导维度。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	b	matrix_t	存储输入/输出矩阵B的缓冲区。	是

8.3　稀疏矩阵 SM（Sparse Matrix）

8.3.1　创建指定格式的稀疏矩阵

创建特定稀疏格式的稀疏矩阵，为后续数据处理提供准备。例如：创建CSR格式的系数矩阵空间，其他格式转换接口一致，具体接口定义如表34所示：

表 34　创建特定稀疏格式的稀疏矩阵接口定义

接口名	create			
描述	创建特定稀疏格式的稀疏矩阵			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	m/n/nnz	int	稀疏矩阵的行数、列数、非零元个数。	是
3	row_offsets	matrix_t	CSR格式矩阵的行偏移数组。	是
4	column_indices	matrix_t	CSR格式矩阵的列坐标数组。	是
5	values	matrix_t	CSR格式矩阵的非零元值数组。	是

表34　创建特定稀疏格式的稀疏矩阵接口定义（续）

二、出参				
序号	参数名	数据类型	描述	是否必填
1	matrix	matrix_t	创建的稀疏矩阵结构体。	是

8.3.2　稀疏矩阵格式转换

例如：将COO、CSR或CSC等非CSR格式稀疏矩阵存储格式转换为CSR格式，其他格式转换接口一致，具体接口定义如表35所示：

表 35　稀疏矩阵格式转换接口定义

接口名	convert			
描述	稀疏矩阵格式转换			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	old_mat	matrix_t	稀疏矩阵结构体，可以是COO、CSR、CSC格式矩阵。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	new_mat	matrix_t	转换后的CSR格式矩阵。	是

8.3.3　稀疏矩阵重排序

对稀疏矩阵行进行重排。主要用于三角分解与求解中，其排列会被保存起来，在求解时对应向量也会重排。排列向量有两种定义：

1. perm：若将 a 重排成 b，则 b[i] = a[p[i]]；

2. iperm：若将 a 重排成 b，则 b[p[i]] = a[i]。

矩阵重排支持CSR、CSC和COO。其中 new_mat 可以取 LibNameSPARSE_IN_PLACE，表示操作后的结果写到 old_mat 里，具体接口定义如表36所示：

表 36　稀疏矩阵重排序接口定义

接口名	reorder			
描述	稀疏矩阵重排序			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	old_mat	matrix_t	稀疏矩阵结构体，可以是COO、CSR、CSC格式矩阵。	是
3	ciperm	matrix_t	列重排向量。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	new_mat	matrix_t	重排行后的稀疏矩阵结构体。	是

8.3.4　trf 算法

一阶段LU分解，之后分别用于SpTRSV。要求A为CSC格式。输出参数中，lu表示因子矩阵L和U，其中L矩阵的对角线为1。实际上输出参数lu中存储的是L-I+U，且U的对角线是以倒数形式存储。riperm和ciperm 是在分解过程中分别对进行的重排，也就是排列阵Pr和Pc；rscale和cscale是分别对行列进行的放缩，也就是对角阵Dr和Dc，注意对矩阵的放缩在重排之前，具体接口定义如表37所示：

表 37　trf 算法接口定义

接口名	trf

表37 trf算法接口定义（续）

接口名	trf			
描述	trf算法			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	a	matrix_t	稀疏矩阵结构体，可以是COO、CSR、CSC格式矩阵。	是
3	riperm/ciperm	matrix_t	行列排列向量。	是
4	rscale/cscale	matrix_t	行列缩放向量。	是
5	options	matrix_t	LibNameSparseTRFOptions选项定义了矩阵分解类型（LU分解）、矩阵对称性、是否转置、重用策略、行列重排序算法（最小近似度AMD）、行列缩放策略（最大权匹配MC64）等。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	lu	matrix_t	稀疏矩阵LU分解结构体。	是

8.3.5 SpADD 算法

计算稀疏加法，具体接口定义如表38所示：

表 38 SpADD 算法接口定义

接口名	add			
描述	SpADD算法			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	a	struct	稀疏矩阵A结构体。	是
3	b	struct	稀疏矩阵B结构体。	是
4	alpha/beta	float	标量系数。	是
5	options	int	LibNameSparseADDOptions设置A、B矩阵是否转置。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	c	struct	结果稀疏矩阵C结构体。	是

8.3.6 mv 算法的稀疏矩阵预处理

SpMV的计算分为两阶段，LibName_sparse_mv_inspect为第一阶段：稀疏矩阵预处理，根据稀疏矩阵的非零元个数判断是否进行矩阵格式转换，具体接口定义如表39所示：

表 39 mv 算法的稀疏矩阵预处理接口定义

接口名	mv			
描述	mv算法的稀疏矩阵预处理			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	old_format	struct	稀疏矩阵结构体。包含了稀疏矩阵的行数、列数、非零元个数以及矩阵数据等信息。	是
3	options	struct	LibNameSparseMVOptions包含预期计算次数，矩阵是否转置以及矩阵的对称性。	是

表 39　mv 算法的稀疏矩阵预处理接口定义（续）

二、出参				
序号	参数名	数据类型	描述	是否必填
1	new_format	struct	预处理后的稀疏矩阵结构体，包含了稀疏矩阵的行数、列数、非零元个数以及矩阵数据等信息。	是

8.3.7　用于 mv 算法的稀疏矩阵计算部分

SpMV的计算分为两阶段，LibName_sparse_?_mv_execute为第二阶段：稀疏矩阵计算，根据预处理后的矩阵进行SpMV计算，具体接口定义如表40所示：

表 40　用于 mv 算法的稀疏矩阵计算部分接口定义

接口名	mv_execute			
描述	用于mv算法的稀疏矩阵计算部分			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	alpha/beta	float	标量系数，参与乘法和累加运算。	是
3	matrix	matrix_t	预处理后的稀疏矩阵结构体，包含了稀疏矩阵的行数、列数、非零元个数以及矩阵数据等信息。	是
5	x	matrix_t	稠密向量。	是
6	options	struct	LibNameSparseMVOptions包含预期计算次数，矩阵是否转置以及矩阵的对称性。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	y	matrix_t	结果向量。	是

8.3.8　SpTRSV 算法第一阶段

SpTRSV即对上三角或下三角矩阵T求解线性方程组。

SpTRSV中的第一阶段要求A为CSR或CSC格式。options必须是LibNameSPARSE_LOWER_TRI或LibNameSPARSE_UPPER_TRI表示矩阵是下三角还是上三角。另外options可取 LibNameSPARSE_UNIT_DIAG或LibNameSPARSE_RECP_DIAG，通过按位或操作符“|”与上下三角的标识同时传入。LibNameSPARSE_UNIT_DIAG 表示对角线全为1，且在矩阵中没有被表示出来，也就是参数a只表示了矩阵的严格上/下三角部分；LibNameSPARSE_RECP_DIAG表示对角线是否已经被取了倒数，这样会将除法替换为乘法。具体接口定义如表41所示：

表 41　SpTRSV 算法第一阶段接口定义

接口名	trsv_inspect			
描述	SpTRSV算法第一阶段			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	old_format	struct	稀疏矩阵结构体。	是
3	perm/iperm	matrix_t	排列向量。	是

表41 SpTRSV算法第一阶段接口定义（续）

序号	参数名	数据类型	描述	是否必填
6	options	struct	LibNameSparseTRSVOptions选项定义了TRSV的计算次数、矩阵是否转置、矩阵是下三角（LibNameSPARSE_LOWER_TRI）还是上三角（LibNameSPARSE_UPPER_TRI）、矩阵对角信息（LibNameSPARSE_GENERAL_DIAG、LibNameSPARSE_UNIT_DIAG、LibNameSPARSE_RECP_DIAG）。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	new_format	struct	稀疏三角求解结构体。	是

8.3.9 SpTRSV 算法第二阶段

完成TSR算法第二阶段的计算，具体接口定义如表42所示：

表 42 TSR 算法第二阶段接口定义

接口名	trsv_execute			
描述	TSR算法第二阶段			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	alpha	float	标量系数。	是
3	matrix	matrix_t	稀疏三角求解结构体。	是
4	b	matrix_t	右端向量。	是
5	options	struct	LibNameSparseTRSVOptions选项定义了TRSV的计算次数、矩阵是否转置、矩阵是下三角（LibNameSPARSE_LOWER_TRI）还是上三角（LibNameSPARSE_UPPER_TRI）、矩阵对角信息（LibNameSPARSE_GENERAL_DIAG、LibNameSPARSE_UNIT_DIAG、LibNameSPARSE_RECP_DIAG）。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	x	matrix_t	结果向量。	是

8.3.10 SpMM 算法第一阶段

SpMM的计算分为两阶段，LibName_sparse_mm_inspect为第一阶段：稀疏矩阵预处理，根据稀疏矩阵的非零元个数判断是否进行矩阵格式转换，具体接口定义如表43所示：

表 43 SpMM 算法第一阶段接口定义

接口名	mm_inspect			
描述	SpMM算法第一阶段			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	old_format	struct	稀疏矩阵结构体。包含了稀疏矩阵的行数、列数、非零元个数以及矩阵数据等信息。	是
3	options	struct	LibNameSparseMVOptions包含预期计算次数，矩阵是否转置以及矩阵	是

			的对称性。	

表43 SpMM算法第一阶段接口定义（续）

二、出参				
序号	参数名	数据类型	描述	是否必填
1	new_format	struct	预处理后的稀疏矩阵结构体，包含了稀疏矩阵的行数、列数、非零元个数以及矩阵数据等信息。	是

8.3.11 SpMM 算法第二阶段

SpMM的计算分为两阶段，LibName_sparse_?_mm_execute为第二阶段：稀疏矩阵计算，根据预处理后的矩阵进行SpMM计算，具体接口定义如表44所示：

表 44 spmm 算法第二阶段接口定义

接口名	mm_execute			
描述	spmm算法第二阶段			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	Handle	matrix_t	进行SPARSE BLAS计算的上下文。	是
2	alpha/beta	float	标量系数，参与乘法和累加运算。	是
3	matrix	matrix_t	预处理后的稀疏矩阵结构体，包含了稀疏矩阵的行数、列数、非零元个数以及矩阵数据等信息。	是
4	x	matrix_t	稠密矩阵。	是
5	nvec	int	稠密矩阵的列数。	是
6	ldx/ldy	int	矩阵的行数、列数。	是
7	options	struct	LibNameSparseMMOptions包含预期计算次数，矩阵是否转置以及矩阵的对称性。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	y	matrix_t	结果矩阵。	是

8.4 稀疏求解器 SS（Sparse Solver）

8.4.1 配置稀疏求解器参数

设置求解器的配置信息,初始化相应求解器内部结构，具体接口定义如表45所示：

表 45 配置稀疏求解器参数接口定义

接口名	setconf			
描述	配置稀疏求解器参数			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	ipr	matrix_t	长度为P_IMAX的数组。	是
2	dpr	matrix_t	长度为P_DMAX的数组。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		返回一个ID，用于指定求解器实例。	是

8.4.2 从内存读取矩阵

从内存地址ptr读取特定格式的稀疏矩阵，写入到求解器内部的求解器数据结构中，具体接口定义如表46所示：

表 46　从内存读取矩阵接口定义

接口名	memload			
描述	从内存读取矩阵			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	id	char	setconf生成的求解器实例号。	是
2	ptr	matrix_t	内存地址。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		更新内部求解器的数据结构。	否

8.4.3　执行指定求解步骤

根据参数执行指定求解步骤，具体接口定义如表47所示：

表 47　执行指定求解步骤接口定义

接口名	execute			
描述	执行指定求解步骤			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	id	char	setconf生成的求解器实例号。	是
2	op	int	步骤代号。op=1 代表结构分析，op=2 代表数值分解。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		返回值为0表示正常完成，其他表示执行异常。	否

8.4.4　执行方程求解

计算线性方程Ax=b的解。

本接口执行前需要先执行execute(id,1)和execute(id,2)确保已完成符号分析和数值分解，具体接口定义如表48所示：

表 48　执行方程求解接口定义

接口名	solve			
描述	执行方程求解			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	id	char	setconf生成的求解器实例号。	是
2	b	matrix_t	方程右侧的常数项系数。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	无		方程的解更新在b中（x覆盖b）。 如果执行正常结束返回0，其他值代表执行异常。	否

8.4.5 执行一般矩阵的 QR 分解

geqrf 函数执行一般矩阵A的QR分解。不进行枢轴化。函数不显式形成矩阵Q;相反,Q被表示为min(m, n)个基本反射器的乘积。提供了与此表示形式的Q一起工作的例程，具体接口定义如表49所示：

表 49 执行一般矩阵的 QR 分解接口定义

接口名	geqrf			
描述	执行一般矩阵的QR分解			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	setconf生成的求解器实例号。	是
2	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	m	int	矩阵A的行数，必须至少为零。	是
4	n	int	矩阵A的列数，必须至少为零。	是
5	a	matrix_t	存储输入矩阵A的缓冲区。	是
6	lda	int	矩阵A的主导维度。如果使用列主序布局，则至少为m，如果使用行主序布局，则至少为n。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	a	matrix_t	输出缓冲区，被分解数据覆盖。	是
2	tau	matrix_t	输出缓冲区，其大小至少为max(1, min(m, n)) 。包含矩阵Q分解成基本反射器乘积的标量。	是

8.4.6 计算一般 m×n 矩阵的 LU 分解

该例程计算一般m × n矩阵A的LU分解为A = PLU，其中P是一个置换矩阵，L是单位对角元素的下三角矩阵（如果m > n则为下梯形矩阵），U 是上三角矩阵（如果m < n则为上梯形矩阵）。该例程使用部分主元法进行行交换，具体接口定义如表50所示：

表 50 计算一般 m × n 矩阵的 LU 分解接口定义

接口名	getrf			
描述	计算一般m × n矩阵的LU分解			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	setconf生成的求解器实例号。	是
2	layout	char	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	m	int	矩阵A的行数，必须至少为零。	是
4	n	int	矩阵A的列数，必须至少为零。	是
5	a	matrix_t	存储输入矩阵A的缓冲区。	是
6	lda	int	矩阵A的主导维度。如果使用列主序布局，则至少为m，如果使用行主序布局，则至少为n。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	a	matrix_t	输出缓冲区，被L和U覆盖。L的单位对角元素不被存储。	是
2	ipiv	matrix_t	数组，大小至少为max(1,min(m,n))。包含枢轴索引；对于1≤i≤ min(m,n)，行i被与行ipiv(i)交换。	是

8.4.7 计算对称（埃尔米特）正定矩阵的乔列斯基分解

该函数是计算对称正定矩阵或复数数据的埃尔米特正定矩阵A的乔列斯基分解，具体接口定义如表51所示：

表51 计算对称（埃尔米特）正定矩阵的乔列斯基分解接口定义

接口名	potrf			
描述	计算对称（埃尔米特）正定矩阵的乔列斯基分解			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	setconf生成的求解器实例号。	是
2	layout	struct	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	m	int	矩阵A的行数，必须至少为零。	是
4	n	int	矩阵A的列数，必须至少为零。	是
5	a	matrix_t	存储输入矩阵A的缓冲区。	是
6	lda	int	矩阵A的主导维度。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	a	matrix_t	缓冲区a被乔列斯基因子U或L覆盖，具体取决于upper_lower。	是

8.4.8 计算由 getrf 确定的一个 LU 分解的一般矩阵的逆矩阵

该函数计算一般矩阵A的逆矩阵A^{-1}。在调用此函数之前，先调用getrf对A进行因式分解，具体接口定义如表52所示：

表52 计算由 getrf 确定的一个 LU 分解的一般矩阵的逆矩阵接口定义

接口名	getri			
描述	计算由 getrf 确定的一个LU分解的一般矩阵的逆矩阵			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	setconf生成的求解器实例号。	是
2	layout	struct	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	n	int	矩阵A的列数，必须至少为零。	是
4	a	matrix_t	存储输入矩阵A的缓冲区。	是
5	lda	int	矩阵A的主导维度。	是
6	ipiv	matrix_t	缓冲区ipiv，由getrf返回。ipiv的维度至少为max(1, n)。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	a	matrix_t	被覆写为n * n的矩阵A。	是

8.4.9 使用乔列斯基分解解对称（埃尔米特）正定线性方程组

该函数求解实对称正定或复埃尔米特正定矩阵 A 的线性方程组，具体接口定义如表53所示：

表53 使用乔列斯基分解解对称（埃尔米特）正定线性方程组接口定义

表53　使用乔列斯基分解解对称（埃尔米特）正定线性方程组接口定义（续）

接口名	potrs			
接口名	potrs			
描述	使用乔列斯基分解解对称（埃尔米特）正定线性方程组			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	queue	matrix_t	setconf生成的求解器实例号。	是
2	layout	struct	指定二维数组存储是行主存储（RowMajor）还是列主存储（ColMajor）。	是
3	upper_lower	char	指定是使用矩阵A的上三角形部分还是下三角形部分。	是
4	n	int	矩阵A的列数，必须至少为零。	是
5	nrhs	int	右侧矩阵B的列数。	是
6	a	matrix_t	存储输入矩阵A的缓冲区。	是
7	lda	int	矩阵A的主导维度。	是
8	b	matrix_t	存储矩阵B的缓冲区。	是
9	ldb	int	矩阵B的主导维度。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	b	matrix_t	解矩阵结果覆盖缓冲区b。	是

9　特征值求解器接口设计

9.1　广义共轭梯度算法 GCGE 接口

使用GCGE算法，实现求解如下广义特征值问题$AX = BX\Lambda$的前nev个特征对。其中$A \in \mathbb{R}^{n\times n}$是稀疏对称矩阵，$B \in \mathbb{R}^{n\times n}$是稀疏对称正定矩阵，$X \in \mathbb{R}^{n\times nev}$为求解的特征向量，满足正交性$X^T BX = I$，$\Lambda$是nev阶对角阵，对角元存储对应的特征值，具体接口定义如表54所示：

表 54　广义共轭梯度算法 GCGE 接口定义

接口名	EigenSolver			
描述	广义共轭梯度算法GCGE			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	A	matrix_t	表示矩阵A。	是
2	B	matrix_t	表示矩阵B。	是
3	evec	matrix_t	若nevGiven不为0，则作为输入参数，其指向的多向量的前nevGiven列表示已知的特征向量近似，可代入算法参与运算。	是
4	nevGiven	int	已知特征向量近似的数量。一般输入0。	是
5	nevConv	matrix_t	指向需要计算的特征对数量。	是
6	ops	Struct*	操作对象结构体指针。包含矩阵操作、线性求解器、GCG参数等信息。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	eval	double*	指向特征值的指针。调用前需分配内存空间。	是

表54　广义共轭梯度算法GCGE接口定义（续）

序号	参数名	数据类型	描述	是否必填
2	evec	matrix_t	特征向量。与eval的特征信息对应。	是
3	nevConv	Int*	指向已经收敛的特征对数量。	是

9.2　并行扩展子空间算法 PASE 接口

利用并行扩展子空间算法，求解如下广义特征值问题$AX = BX\Lambda$的前nev个特征对，其中$A, B \in \mathbb{R}^{n\times n}$是稀疏对称矩阵，通常由有限元方程得到。$X \in \mathbb{R}^{n\times nev}$为求解的特征向量，满足正交性$X^T BX = I$，$\Lambda$是nev阶对角阵，对角元存储相应特征值，具体接口定义如表55所示：

表 55　并行扩展子空间算法 PASE 接口定义

接口名	PASE_EigenSolver			
接口名	PASE_EigenSolver			
描述	并行扩展子空间算法PASE			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	param	struct*	参数结构体指针。调用接口函数前需要构造，特别是要指定容许误差、空间层数以及多层网格上的刚度矩阵和质量矩阵以及相邻网格层之间的插值矩阵。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	eval	double*	指向特征值的双重指针。调用前为空指针。	是
2	evec	matrix_t*	指向特征向量的多向量指针，与eval的特征值信息对应。调用前为空指针。	是
三、返回值				
数据类型			描述	
int			输出 0 意味着函数被成功调用。	

10　拓扑优化模块接口设计

10.1　基于 MMC 进行拓扑优化设计

以可移动变形组件作为设计变量进行拓扑优化。优化目标为结构柔顺度最小，约束为体积约束，具体接口定义如表56所示：

表 56　基于 MMC 进行拓扑优化设计接口定义

接口名	MovingMorphableComponents3D			
描述	基于MMC进行拓扑优化设计			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	int numEx	int	设计域在X方向的网格数量。	是
2	int numEy	int	设计域在Y方向的网格数量。	是
3	int numEz	int	设计域在 Z 方向的网格数量。	是
4	double DL	float	设计域在X方向的尺寸。	是
5	double DW	float	设计域在Y方向的尺寸。	是

6	double DH		设计域在Z方向的尺寸。	是
7	int CellX	int	以单胞为单位决定组件的初始布局，在 X 方向的单胞数量。	是
8	int CellY	int	以单胞为单位决定组件的初始布局，在Y方向的单胞数量。	是
9	int CellZ	int	以单胞为单位决定组件的初始布局，在 Z 方向的单胞数量	是
10	int numVal	int	设计变量个数（默认为9）。	是
11	double volFraction	float	限制的最大体积分数。	是
12	int case_num	int	边界条件算例，不同序号对应不同边界条件。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	std::string result_flie	char	优化结果的单元密度.vtk文件路径。	是
2	std::string txt_flie	char	优化结果未用Heavisde函数处理的密度文件路径。	是

10.2 基于 SIMP 进行拓扑优化设计

以单元材料密度作为设计变量进行拓扑优化。优化目标为结构柔顺度最小，约束为体积约束，具体接口定义如表57所示：

表 57 基于 SIMP 进行拓扑优化设计接口定义

接口名	SIMPTopologyOptimization3D			
描述	基于SIMP进行拓扑优化设计			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	int numEx	int	设计域在X方向的网格数量。	是
2	int numEy	int	设计域在Y方向的网格数量。	是
3	int numEz	int	设计域在Z方向的网格数量。	是
4	double DL	float	设计域在X方向的尺寸。	是
5	double DW	float	设计域在Y方向的尺寸。	是
6	double DH	float	设计域在Z方向的尺寸。	是
7	double volFraction	float	限制的最大体积分数。	是
8	int case_num	int	边界条件算例，不同序号对应不同边界条件。	是
9	double penalFactor	float	SIMP材料惩罚因子	是
10	double filterRadius	float	密度过滤的半径	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	std::string result_flie	char	优化结果的单元密度.vtk文件路径。	是

10.3 基于水平集方法进行拓扑优化设计

以节点水平集函数值作为设计变量进行拓扑优化，优化目标为结构柔顺度最小，约束为体积约束，具体接口定义如表58所示：

表 58 基于水平集方法进行拓扑优化设计接口定义

接口名	LevelSet3D			
描述	基于水平集方法进行拓扑优化设计			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	int numEx	int	设计域在X方向的网格数量。	是
2	int numEy	int	设计域在Y方向的网格数量。	是
3	int numEz	int	设计域在Z方向的网格数量。	是
4	double DL	float	设计域在X方向的尺寸。	是
5	double DW	float	设计域在Y方向的尺寸。	是
6	double DH	float	设计域在Z方向的尺寸。	是
7	int PhiInit	matrix_t	初始布局中，水平集函数初始值。	是
8	int PhiUp	matrix_t	水平集函数值上限。	是
9	int PhiLo	matrix_t	水平集函数值下限。	是
10	int numVal	int	设计变量个数（默认为有限元节点数）。	是
11	double volFraction	matrix_t	限制的最大体积分数。	是
12	int case_num	int	边界条件算例，不同序号对应不同边界条件。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	std::string result_flie	char	优化结果的单元密度.vtk文件路径。	是

11 非凸优化模块接口设计

11.1 标准优化问题模型类

Problem类用于求解带边界约束的非线性优化问题，支持多种优化算法，具体接口定义如表59所示：

表 59 Problem:标准优化问题模型类计接口定义

接口名	Problem			
描述	Problem:标准优化问题模型类计			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	nvar	int	优化变量维度，必须满足 nvar > 0。	是
2	x_L	matrix_t	变量下界缓冲区，该缓冲区大小为nvar。	是
3	x_U	matrix_t	变量上界缓冲区，该缓冲区大小为nvar。	是
4	f	float*	目标函数指针，输入x返回标量值 $f(x)$。	是
5	grad_f	matrix_t*	梯度函数指针，输入x，输出梯度到grad缓冲区。	是

11.2 对标准优化问题（Problem 类对象）求解

solve函数对Problem类对象定义的优化问题进行求解，支持多种迭代算法，输出问题的近似最优解，具体接口定义如表60所示：

表 60 对标准优化问题（Problem 类对象）求解接口定义

接口名	solve
描述	对标准优化问题（Problem类对象）求解

一、入参				
序号	参数名	数据类型	描述	是否必填
1	n	int	优化问题的维度（变量个数），必须满足 n > 0。	是
2	x	matrix_t	初始点。	是
3	x_L	matrix_t	变量下界缓冲区 。	是
4	x_U	matrix_t	变量上界缓冲区。	是
5	f	matrix_t*	目标函数指针，输入x返回标量值 $f(x)$。	是
6	grad_f	matrix_t*	梯度函数指针，输入x，输出梯度到grad缓冲区。	是
7	method	int	优化算法选择。	是
8	options	char	算法参数字典。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	result	matrix_t	近似最优解，要求预分配大小为n。	是
2	optimal_value	float	近似最优目标函数值。	是
3	optimal_grad	matrix_t	近似最优解处的梯度。	是
4	iterations	int	实际迭代次数。	是

11.3 优化求解器核心函数

umatr函数实现信赖域反射法的优化算法，适用于盒式约束优化问题，具体接口定义如表61所示：

表 61 优化求解器核心函数接口定义

接口名	umatr			
描述	优化求解器核心函数			
一、入参				
序号	参数名	数据类型	描述	是否必填
1	n	int	优化问题的维度（变量个数），必须满足 n > 0。	是
2	x_0	matrix_t	初始点。	是
3	x_L	matrix_t	变量下界缓冲区 。	是
4	x_U	matrix_t	变量上界缓冲区。	是
5	f	float*	目标函数指针，输入x返回标量值 $f(x)$。	是
6	grad_f	matrix_t*	梯度函数指针，输入x，输出梯度到grad缓冲区。	是
二、出参				
序号	参数名	数据类型	描述	是否必填
1	x_opt	matrix_t	近似最优解，要求预分配大小为n。	是
2	f_opt	float	近似最优目标函数值。	是
3	iter	int	算法迭代次数。	是