



从PeriDyno到CAE软件 增量集成开发平台 MxSimLab

蔡勇

湖南大学 & 迈曦软件

2023.7.9

群聊: MxSimLab交流群



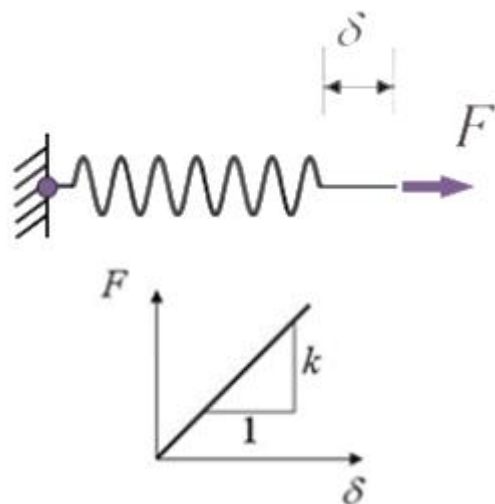
该二维码7天内(7月16日前)有效, 重新进入将更新

- 1 有 限 元 基 础 方 法
- 2 PeriDyno到MxSimLab
- 3 M x S i m L a b 教 学

群聊: MxSimLab交流群



该二维码7天内(7月16日前)有效, 重新进入将更新



胡克定律，曾译为虎克定律，是力学弹性理论中的一条基本定律，表述为：固体材料受力之后，材料中的应力与应变（单位变形量）之间成线性关系。满足胡克定律的材料称为线弹性或胡克型（英文Hookean）材料。

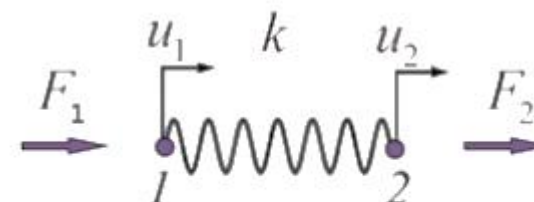
弹簧的力与
位移的关系

$$F = k\delta$$

k为弹簧的刚度系数

基于端点的描述

有限元中，为了规范化描述



基于节点的弹簧描述

节点位移 (u_1, u_2) 节点力 (F_1, F_2)

相当于描述弹簧变形行为的一个物理方程

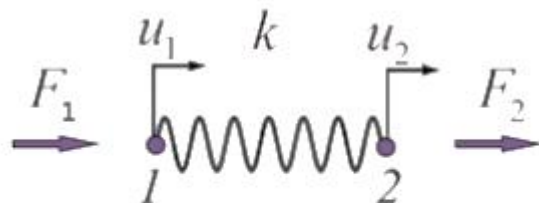


迈曦软件
MAIXI SOFTWARE

群聊: MxSimLab交流群



该二维码7天内(7月16日前)有效, 重新进入将更新



基于节点的弹簧描述

$$\left. \begin{aligned} k(u_2 - u_1) &= F_2 \\ F_1 + F_2 &= 0 \end{aligned} \right\}$$



$$\left. \begin{aligned} k(u_1 - u_2) &= F_1 \\ k(u_2 - u_1) &= F_2 \end{aligned} \right\}$$



矩阵形式

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

矩阵系数 节点位移 节点力

**称为弹簧的
平衡方程，
或刚度方程**

节点位移 (u_1, u_2) 节点力 (F_1, F_2)



有限元基础方法（以弹簧单元为例）



为什么强调矩阵形式？

- ✓ 矩阵可被看成是数学的有序排列，可以使得求解过程以一种紧凑、美观的方式表达出来。
- ✓ 满足有限元分析标准化和规范化两大特点。这种特点使得大规模分析和计算成为可能，当采用了现代化的计算机以及所编制的软件作为实现平台时，则复杂工程问题的大规模分析就变为现实。
- ✓ 易于在计算机中实现。

```
>> A=[1 1 1 1;2 2 2 2;3 3 3 3;4 4 4 4]
A =
     1     1     1     1
     2     2     2     2
     3     3     3     3
     4     4     4     4

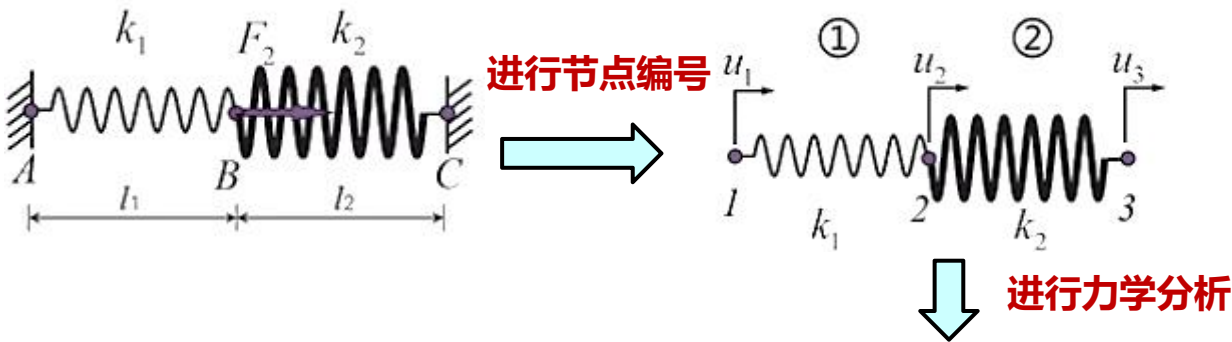
>> B=[1 5 9 2;6 3 5 7;2 5 8 9;4 5 6 3]
B =
     1     5     9     2
     6     3     5     7
     2     5     8     9
     4     5     6     3

>> C=A*B
C =
    13    18    28    21
    26    36    56    42
    39    54    84    63
    52    72   112    84
```



有限元基础方法（以弹簧单元为例）

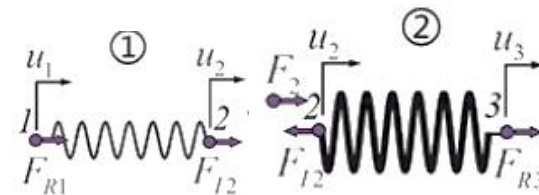
例：一个两弹簧系统的分析



构建1的平衡方程

$$\left. \begin{aligned} k_1(u_2 - u_1) &= F_{I2} \\ F_{R1} + F_{I2} &= 0 \end{aligned} \right\}$$

$$\left. \begin{aligned} k_1(u_1 - u_2) &= F_{R1} \\ k_1(u_2 - u_1) &= F_{I2} \end{aligned} \right\}$$



构建2的平衡方程

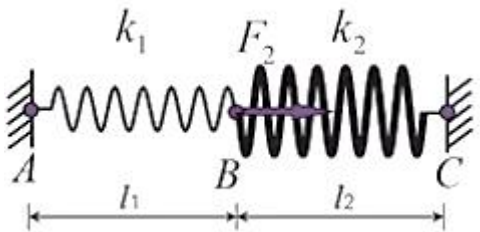
$$\left. \begin{aligned} k_2(u_3 - u_2) &= F_{R3} \\ F_2 - F_{I2} + F_{R3} &= 0 \end{aligned} \right\}$$

$$\left. \begin{aligned} k_2(u_2 - u_3) &= F_2 - F_{I2} \\ k_2(u_3 - u_2) &= F_{R3} \end{aligned} \right\}$$

为了方便计算或计算编程实现，如果构建整体的平衡方程？



例：一个两弹簧系统的分析



如何求解了？

$$\left. \begin{aligned} k_1(u_1 - u_2) &= F_{R1} \\ k_1(u_2 - u_1) &= F_{I2} \end{aligned} \right\}$$



$$\begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} F_{R1} \\ F_{I2} \end{bmatrix}$$



为了在一个系统里进行组装，
需要进行扩充

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_{R1} \\ F_{I2} \\ 0 \end{bmatrix}$$



$$\left. \begin{aligned} k_2(u_2 - u_3) &= F_2 - F_{I2} \\ k_2(u_3 - u_2) &= F_{R3} \end{aligned} \right\}$$



$$\begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_2 - F_{I2} \\ F_{R3} \end{bmatrix}$$



$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_{R1} \\ F_2 \\ F_{R3} \end{bmatrix}$$

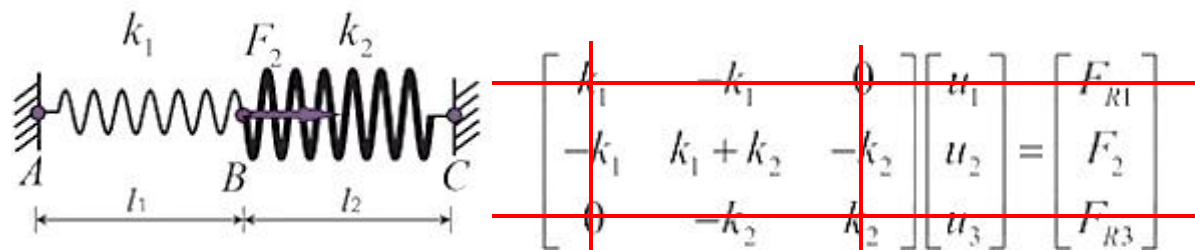
系统总体的平衡方程

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_{R1} \\ F_2 - \boxed{F_{I2} + F_{I2}} \\ F_{R3} \end{bmatrix}$$

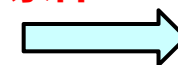
一对作用力和反作用力
和为“0”



例：一个两弹簧系统的分析



考虑边界
条件



$$\begin{aligned} u_1 &= 0 \\ u_3 &= 0 \end{aligned}$$

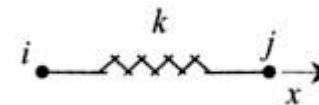


$$\left. \begin{aligned} F_{R1} &= -k_1 u_2 = -\frac{k_1 F_2}{k_1 + k_2} \\ F_{R3} &= -k_2 u_2 = -\frac{k_2 F_2}{k_1 + k_2} \end{aligned} \right\} \leftarrow u_2 = \frac{F_2}{k_1 + k_2} \leftarrow (k_1 + k_2) u_2 = F_2$$



弹簧元是总体和局部坐标一致的一维有限单元

每个弹簧元有两个节点 (node)



单刚矩阵为: $k = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \quad 2 \times 2$

总刚矩阵: $n \times n$

结构方程: $[K]\{U\} = \{F\}$

单元节点力: $\{f\} = [k]\{u\}$



3、MATLAB函数编写

3.1 单元刚度矩阵的形成

function y = SpringElementStiffness(k)

**%SpringElementStiffness This function returns the element stiffness
%matrix for a spring with stiffness k.
%The size of the element stiffness matrix is 2 x 2.**

y = [k -k ; -k k];

$$k = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$$

群聊: MxSimLab交流群



该二维码7天内(7月16日前)有效, 重新进入将更新

3、MATLAB函数编写

3.2 整体刚度矩阵的形成

function y = SpringAssemble(K,k,i,j)

```
%SpringAssemble This function assembles the element stiffness  
%               matrix k of the spring with nodes i and j into the  
%               global stiffness matrix K.  
%               This function returns the global stiffness matrix K  
%               after the element stiffness matrix k is assembled.
```

```
K(i,i) = K(i,i) + k(1,1);  
K(i,j) = K(i,j) + k(1,2);  
K(j,i) = K(j,i) + k(2,1);  
K(j,j) = K(j,j) + k(2,2);  
y = K;
```

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_{R1} \\ F_2 \\ F_{R3} \end{bmatrix}$$



3、MATLAB函数编写

3.3 节点载荷计算

function y = SpringElementForces(k,u)

**%SpringElementForces This function returns the element nodal force
% vector given the element stiffness matrix k
% and the element nodal displacement vector u.**

y = k * u;

$$[K]\{U\} = \{F\}$$



4、实例计算分析应用

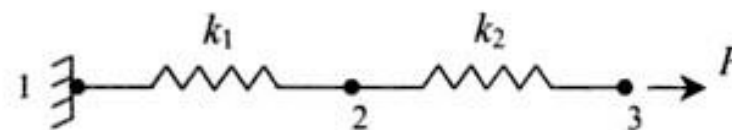
如图所示二弹簧元结构，假定 $k_1=100\text{kN/m}$ ， $k_2=200\text{kN/m}$ ， $P=15\text{kN}$ 。

求：系统的整体刚度矩阵；

节点2、3的位移；

节点1的支反力；

每个弹簧的内力



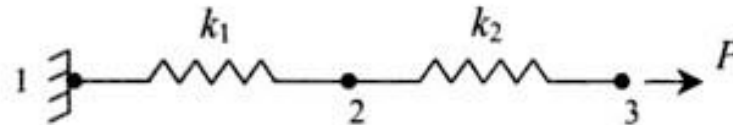
解：

步骤1：离散化域

Element Number	Node i	Node j
1	1	2
2	2	3



4、实例计算分析应用



步骤2：形成单元刚度矩阵

调用 `function y = SpringElementStiffness(k)`函数

```
k1=SpringElementStiffness(100);
```

k1 =

```
100 -100
-100 100
```

```
k2=SpringElementStiffness(200);
```

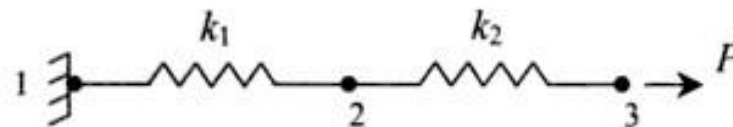
k2 =

```
200 -200
-200 200
```



4、实例计算分析应用

步骤3：集成整体刚度矩阵



调用 function $y = \text{SpringAssemble}(K,k,i,j)$ 函数

$n=3; K = \text{zeros}(n,n);$

$K =$

0	0	0
0	0	0
0	0	0

$K = \text{SpringAssemble}(K,k1,1,2)$

$K =$

100	-100	0
-100	100	0
0	0	0

$K = \text{SpringAssemble}(K,k2,2,3)$

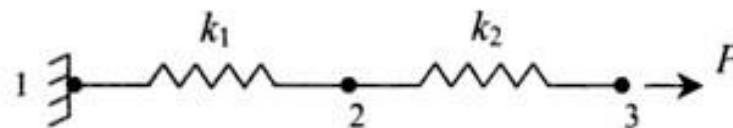
$K =$

100	-100	0
-100	300	-200
0	-200	200



4、实例计算分析应用

步骤4：引入边界条件



$$\begin{bmatrix} 100 & -100 & 0 \\ -100 & 300 & -200 \\ 0 & -200 & 200 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}$$

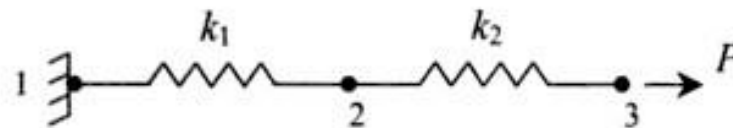
已知边界条件: $U_1 = 0, F_2 = 0, F_3 = 15$

$$\begin{bmatrix} 100 & -100 & 0 \\ -100 & 300 & -200 \\ 0 & -200 & 200 \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ 0 \\ 15 \end{Bmatrix}$$



5、实例计算分析应用

步骤5：解方程



```
U=zeros(2,1);
```

```
F=[0;15];
```

```
K = K(2:3,2:3);
```

```
K(1,:)=[];  
K(:,1)=[];
```

$$\begin{bmatrix} 300 & -200 \\ -200 & 200 \end{bmatrix} \begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 15 \end{Bmatrix}$$

```
U=K\F
```

```
U=inv(K)*F
```

```
U =  
    0.1500  
    0.2250
```



5、实例计算分析应用

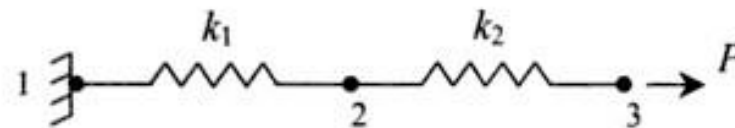
步骤6：后处理

$$U=[0;U]$$

$$U = \begin{bmatrix} 0 \\ 0.1500 \\ 0.2250 \end{bmatrix}$$

$$F=K*U$$

$$F = \begin{bmatrix} -15.0000 \\ 0.0000 \\ 15.0000 \end{bmatrix}$$



$$u1=U(1:2);$$
$$f1=SpringElementForces(k1,u1);$$

$$f1 = \begin{bmatrix} -15.0000 \\ 15.0000 \end{bmatrix}$$

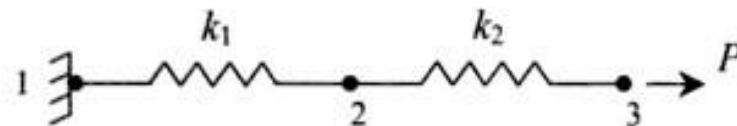
$$u2=U(2:3);$$
$$f2=SpringElementForces(k2,u2);$$

$$f2 = \begin{bmatrix} -15.0000 \\ 15.0000 \end{bmatrix}$$



5、实例计算分析应用

```
k1=SpringElementStiffness(100);  
k2=SpringElementStiffness(200);  
n=3;  
K=zeros(n,n);  
K=SpringAssemble(K,k1,1,2);  
K=SpringAssemble(K,k2,2,3);  
U=zeros(2,1);  
F=[0;15];  
K = K(2:3,2:3);  
KK=K;  
U=K\F  
U=[0;U];  
F=K*U;  
u1=U(1:2);  
f1=SpringElementForces(k1,u1)  
u2=U(2:3);  
f2=SpringElementForces(k2,u2)
```



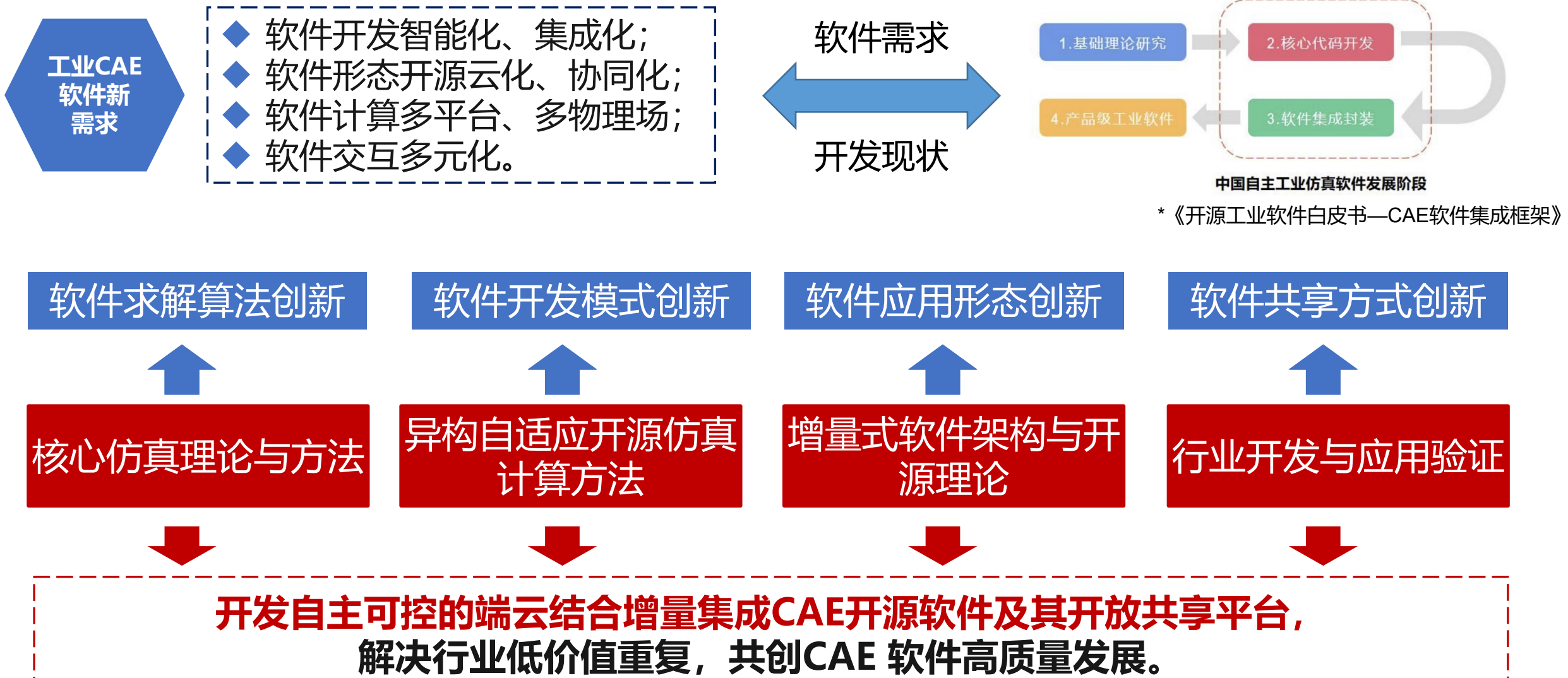
- 1 有 限 元 法 基 础
- 2 PeriDyno到MxSimLab
- 3 M x S i m L a b 教 学

群聊: MxSimLab交流群



该二维码7天内(7月16日前)有效, 重新进入将更新

CAE软件行业发展趋势及需求



CAE软件行业发展趋势及需求

不同任务驱动下对CAE软件学习、开发和应用的多层需求



社会大众、学生、科研人员、工程应用人员

降低行业低价值重复，发挥领域各主体关键作用，
加速 CAE 软件产品化进程。



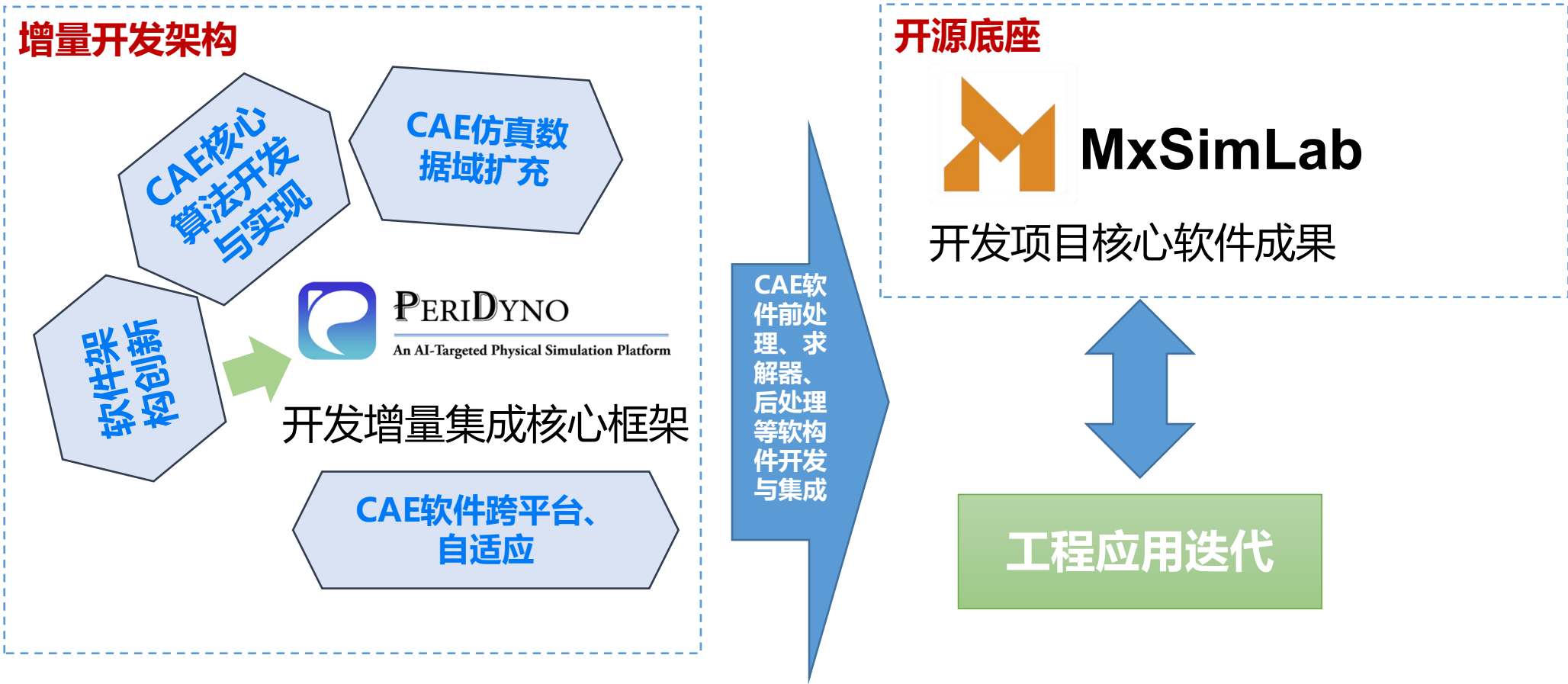
- 1. 软件开发：快速集成需求
- 2. 软件应用：丰富的软构件
- 3. 科普性学习：灵活、可视

- 1. 有求解器，变成完整软件
- 2. 有场景，变成定制软件
- 3. 有算法，快速实现、测试

- 1. 有模型，找软件
- 2. 不同软件结果对比测试
- 3. 替代商业软件

解决传统 CAE 软件开发分散独立、重复劳动的状态，实现在CAE 软件模块的快速开发、灵活组织及数据互通，满足不同任务驱动下对CAE软件学习、开发和应用的多层需求。

从Peridyno到MxSimLab

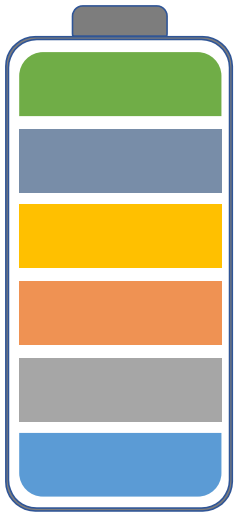


有关数值方法介绍

基于边的梯度光滑有限元
方法及低阶高精度单元

显式有限元的全流程
GPU并行计算方法

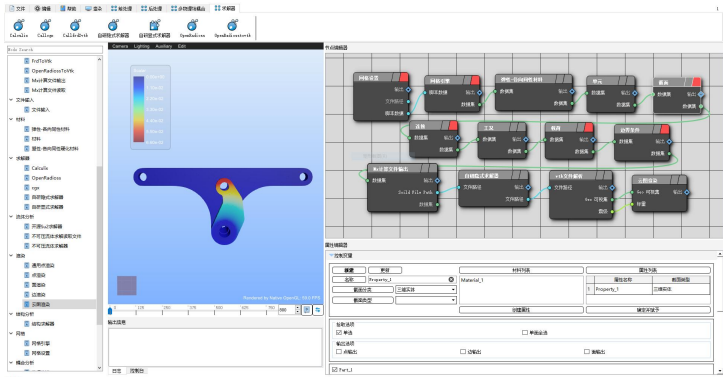
复杂结构接触界面的
高容错性的自动化并行接
触计算方法



多物理场耦合算法

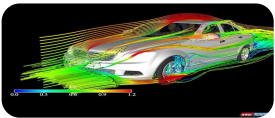
基于对偶粒子框架的流体
仿真方法

基于粒子系统的流体仿真
关键技术与优化



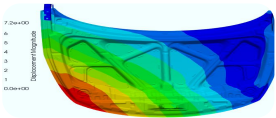
CAE软件仿真算法库

MxSimLab
(CAE软件)

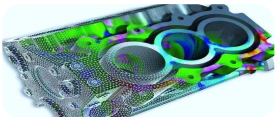


造型设计

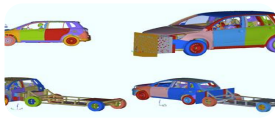
人体损伤



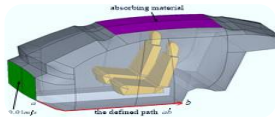
结构设计



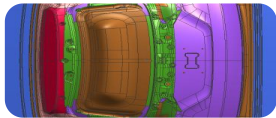
寿命评估



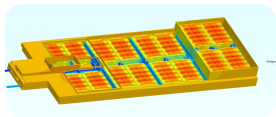
性能设计



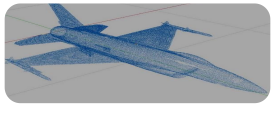
噪声评估



工艺设计



电芯管理



战机设计



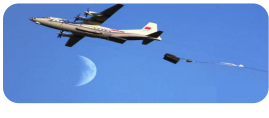
舰船撞击



装甲防护



载具倾覆



战车空投



战机鸟撞



返回舱着陆



登月舱着陆



从Peridyno到MxSimLab

□ 基于数据层 (Field)、模块层 (Module) , 节点层 (Node) 以及场景层 (SceneGraph) 的四层架构的通用物理仿真引擎

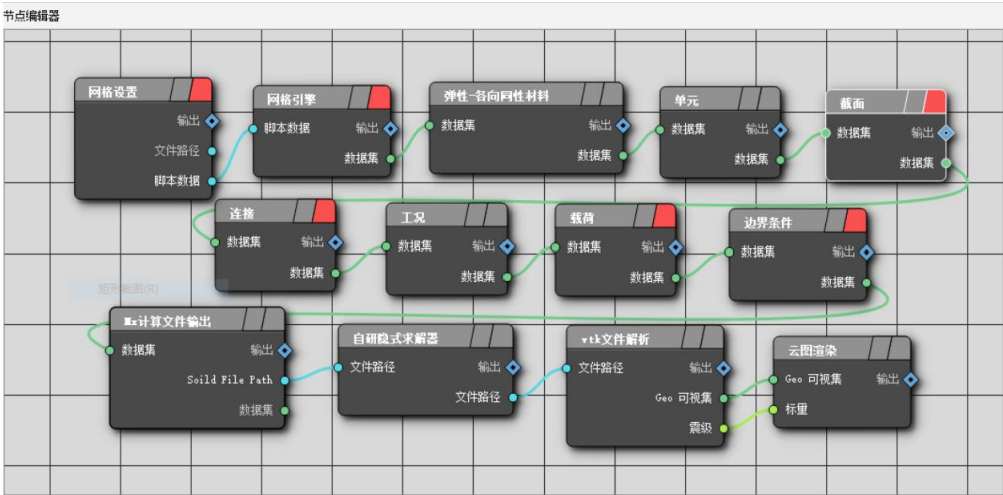
SceneGraph: CAE仿真场景搭建, 支持节点动态增减、编辑等

Node: CAE仿真流程搭建及管理, 支持模块管理、遍历、扩展等

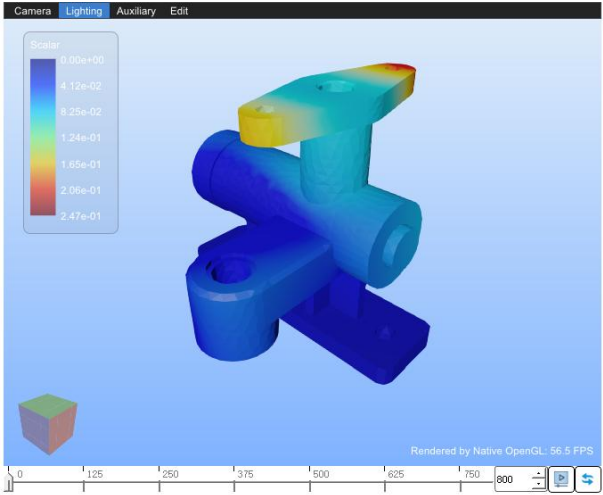
Module: 仿真计算功能模块化表示, 支持碰撞、约束、动力学及通用计算等

Field: 异构平台运行态仿真数据层次化表示, 支持反射、多态、数据传递等

作用?



节点图



仿真可视图

群聊: MxSimLab交流群



该二维码7天内(7月16日前)有效, 重新进入将更新

从Peridyno到MxSimLab

1 CAE仿真与渲染分离

- 统一渲染接口
- 多后端渲染器支持
- 统一通信协议
- 海量仿真数据支持

2 CAE仿真与交互分离

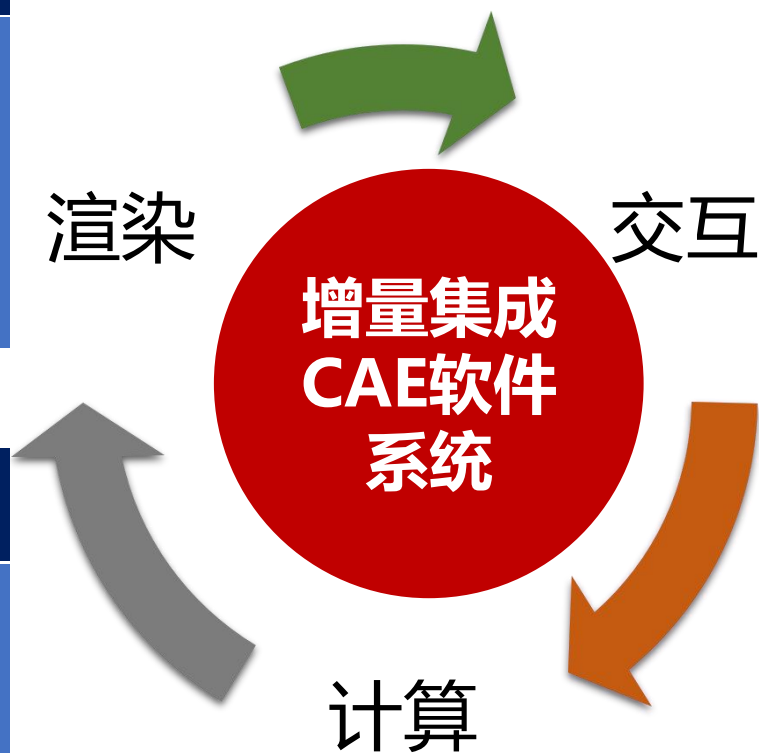
- 算法模块开发平台
- 交互设计平台
- 基于Web的云端平台
- 移动平台

3 CAE仿真与底层分离

- CPU
- GPU
- 集群环境
- 统一数据接口

4 CAE仿真模块复用

- 增量模块开发
- 增量模块集成
- 增量模块拓展
- 模块共享与复用

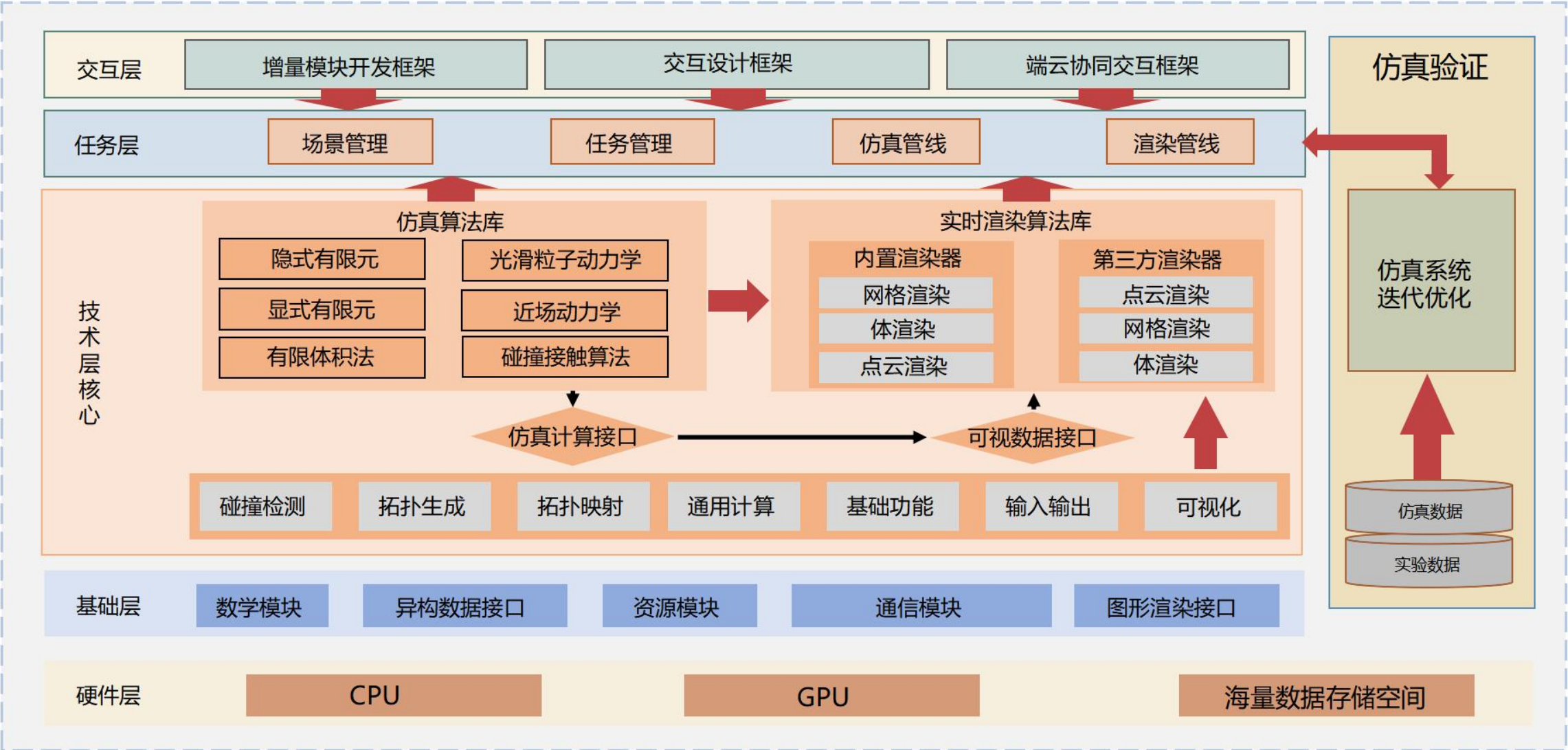


解决复杂CAE任务的增量开发、集成与模块复用难题，达到了仿真计算代码零成本迁移、多平台复用的目标。



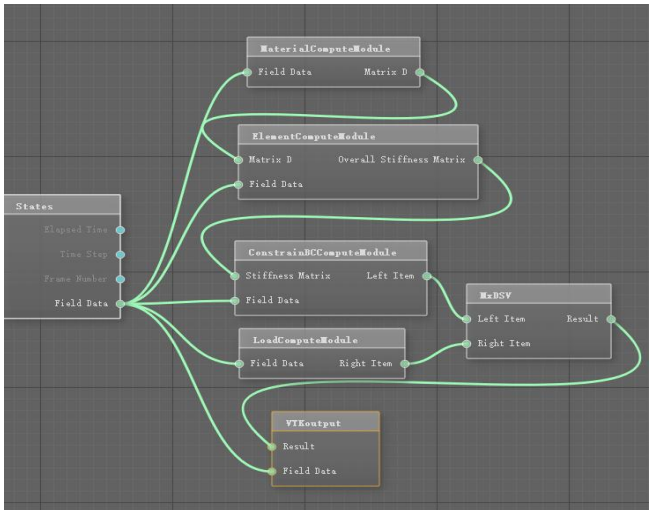
从Peridyno到MxSimLab

云原生增量集成式CAE软件系统架构CAE

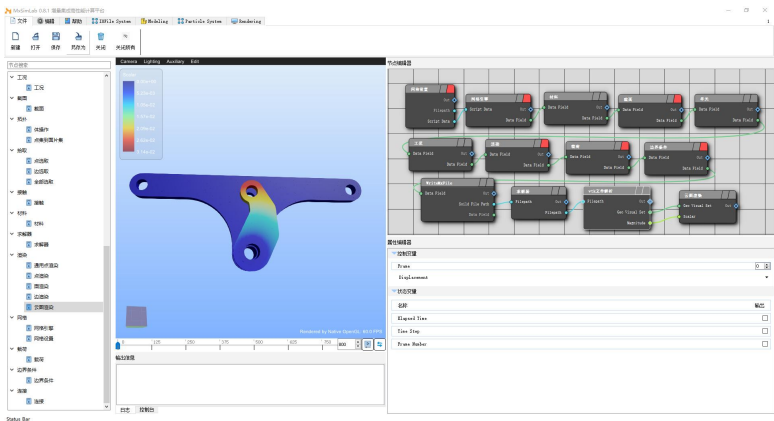


从Peridyno到MxSimLab

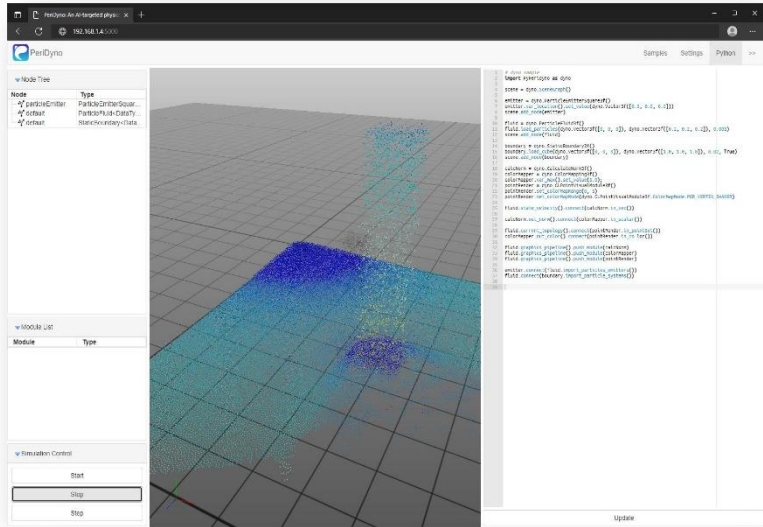
面向对象：学生、科研人员



面向对象：工程研发人员



面向对象：应用人员



增量模块开发框架

- 轻量级GLFW框架
- 大量参考实例
- 大量测试样例

交互设计框架

- 交互设计QT框架
- 复杂场景搭建
- 二次开发与集成

云应用框架

- 端云协同WT框架
- 仿真方案动态加载
- 端云协同

开源的CAE软件增量集成开发平台，满足科研、教学、工程等多场景需求。



群聊：MxSimLab交流群



该二维码7天内(7月16日前)有效，重新进入将更新

- 1 有 限 元 法 基 础
- 2 PeriDyno到MxSimLab
- 3 M x S i m L a b 教 学

群聊: MxSimLab交流群



该二维码7天内(7月16日前)有效, 重新进入将更新

湖南大学-蔡勇 / MxSimLab

Watch 8Star 6Fork 1

代码Issues 1Pull Requests 2Wiki统计流水线服务

加入 Gitee

与超过 1000 万 开发者一起发现、参与优秀开源项目，私有仓库也完全免费：)

免费加入

已有帐号? 立即登录

develop 分支 15 标签 0

文件Web IDE克隆/下载

简介

penghaiyi fix compilation, fix QFileDialog and use ... d1c7c73 5天前 1144 次提交

cmake

data

examples

external

python

src

tests

mutinst

gitattributes

gitignore

gitmodules

CMakeLists.txt

Doxyfile

LICENSE

README.md

Update cmake modules

Merge remote-tracking branch 'peridyno/master' into develop

fix compilation, fix QFileDialog and use constDataPtr

Merge remote-tracking branch 'peridyno/master' into develop

Update installing configurations to support peri-cloud

fix compilation, fix QFileDialog and use constDataPtr

Fix the bug in Quat

first construct mxsimlab version --zbgw

update translation

mid check version

Include tinyobjloader as an external library

Merge remote-tracking branch 'peridyno/master' into develop

Initial commit

Initial commit

Merge branch 'master' of gitee.com:caiyong_GPU/mx-sim-lab into release

8个月前

5天前

5天前

5天前

25天前

5天前

26天前

3个月前

2个月前

13天前

2个月前

5天前

2年前

3个月前

1个月前

README.md

说明

工业CAE仿真软件集成平台

暂无标签

C++ 等 6 种语言

Apache-2.0

发行版

暂无发行版

贡献者 (26)

近期动态

群聊: MxSimLab 交流群

该二维码 7 天内 (7月16日前) 有效, 重新进入将更新

MxSimLab交流微信群，欢迎加入

https://gitee.com/caiyong_GPU/mx-sim-lab

MxSimLab功能模块





谢谢!



2023年7月