

# ANSYS 软件及其二次开发工具

ANSYS 作为有限元领域的大型通用程序，在工业应用领域及科研方面均有深入的应用。其广泛而有效的分析工具能解决各类问题，如结构、流体、热、电磁问题等，同时 ANSYS 还为高级用户提供了多种二次开发工具，利用这些工具，用户可以高效地扩充 ANSYS 的功能。本章对 ANSYS 软件和其他商业软件，以及开源软件进行了简单介绍，然后分别对 ANSYS 的 4 个二次开发工具 APDL、UPFs、UIDL 及 Tck\Tk 进行了介绍。

## 本章要点：

- 各商业及开源有限元软件的介绍
- ANSYS 的 4 个二次开发工具

## 1.1 ANSYS 简介

若要对 ANSYS 进行二次开发，首先有必要对 ANSYS 软件的发展及其组成部分有所了解，对 ANSYS 软件的结构有了整体的把握才能有的放矢、目标明确的进行相关二次开发工作。本节简要介绍 ANSYS 公司及 ANSYS 软件的发展历程，然后介绍 ANSYS 12.0 软件的组成，最后给出一些其他商业及开源有限元软件的简单介绍。

### 1.1.1 ANSYS 的发展历程

ANSYS 公司于 1970 年在美国成立，创始人 John Swanson 博士是匹兹堡大学力学系教授，公司总部位于美国宾西法尼亚州的匹兹堡。经过四十年的发展，ANSYS 在有限元软件领域占据了举足轻重的地位，被世界各工业领域广泛接受，成为全球众多专业技术协会认可的标准分析软件。ANSYS 集成了力学、热学、电学、声学、流体等多个模块，可用于航空航天、汽车、电子电气、国防军工、铁路、造船、石油化工、能源电力、核工业、土木工程、冶金与成形以及生物医学等各个领域。

ANSYS 公司于 2003 年开发设计了新一代 CAE 仿真平台 ANSYS Workbench。ANSYS

Workbench 由各种功能丰富的模块组成, 有 Windows 风格的优化易用的界面, 能直接读入常用的各种格式的模型文件, 并具有良好的数据交换能力和强大的协同仿真环境。新版的 ANSYS 软件产品的灵活性、易用性和强大的功能都达到了数值分析软件的一个新高度。

近年来, 随着 ANSYS 公司实力的增强, 不断收购其他 CAE 软件公司, 进一步扩充了 ANSYS 软件的功能。例如, ANSYS 公司于 2006 年收购了在流体领域处于领先地位的美国 Fluent 公司, 于 2008 年收购了在电路和电磁仿真领域处于领导地位的美国 Ansoft 公司。通过整合, ANSYS 公司日渐成为全球最大的仿真软件公司。目前, ANSYS 整个产品线包括结构分析 (ANSYS Mechanical) 系列, 流体动力学 (ANSYS CFD (FLUENT/CFX)) 系列, 电子设计 (ANSYS Ansoft) 系列以及 ANSYS Workbench 和 EKM 等。本书以 Windows XP 32 位操作系统、ANSYS 12.0 为平台编写。

### 1.1.2 ANSYS 12.0 的组成

自从 ANSYS 引入 Workbench 后, ANSYS 软件产品主要由两部分构成, 即 ANSYS Workbench 及经典 ANSYS (ANSYS 12.0 将其称为 ANSYS Mechanical APDL)。Workbench 是 ANSYS 公司提出的协同仿真环境, Workbench 的目的是使所有与仿真工作相关的人、技术、数据在一个统一环境中协同工作, 各类数据之间的交流、通讯和共享皆可在这个环境中完成。ANSYS 12.0 的 Workbench 交互界面如图 1-1 所示。

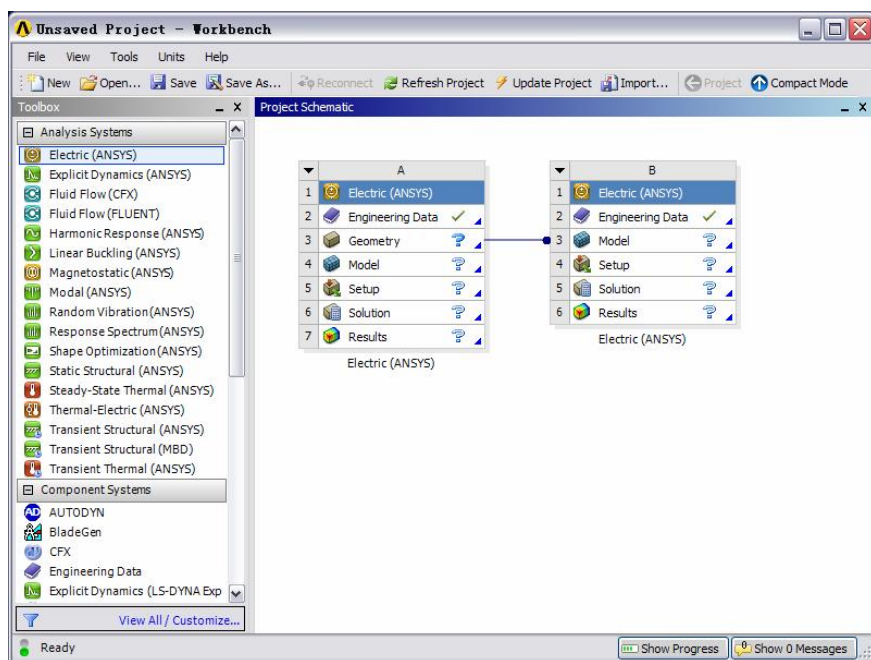


图 1-1 ANSYS 12.0 Workbench 协同仿真环境

Workbench 是一个将 ANSYS 的各个模块进行整合, 并与其协同仿真环境相结合的产物。图 1-2 是 ANSYS 的各个分析模块, 从图中可以发现, 利用 Workbench 几乎可以直接使用 ANSYS 所有产品的前后处理器及求解器, 比如 AUTODYN、BladeGen、CFX 等, 此外经典 ANSYS 也是 Workbench 的一个分析模块, 当然, Workbench 的强大之处不在于其对各模块的简单集成,

而是在于其先进的协同仿真环境思想。下面对图 1-2 中的各集成模块进行简单说明：



图 1-2 ANSYS 12.0 Workbench 集成的分析模块

- AUTODYN：显式有限元分析程序，用来解决固体、流体、气体及其相互作用的高度非线性动力学问题。
- BladeGen：旋转机械叶片建模程序。
- CFX：计算流体力学程序，支持各种可压缩、不可压缩流体的流动分析及复杂几何体的传热分析。
- Engineering Data：工程数据库，用于查看、编辑、增加分析所需的材料参数。
- Explicit Dynamics (LS-DYNA Export)：通用显式动力分析程序，适合求解结构的高速碰撞、爆炸和金属成形等非线性动力冲击问题，也可求解传热、流体及流固耦合问题。
- Finite Element Modeler：有限元模型生成器，支持各种软件有限元文件的导入，如 NASTRAN (\*.bdf、\*.dat、\*.nas)、ABAQUS (\*.inp)、Mechanical APDL (\*.cdb)、CFX (\*.def、\*.res) 以及 Mesh 文件 (\*.cmdb、\*.meshdat) 等。
- FLUENT：计算流体力学程序，支持各种可压缩、不可压缩流体的流动分析及复杂几何体的传热分析。
- Geometry：用于导入或新建几何模型。
- Mechanical APDL：经典 ANSYS，在经典 ANSYS 界面内操作。
- Mechanical Model：结构分析时用于设定材料参数、导入模型及网格划分。
- Mesh：网格划分器。
- Results：结果查看器。
- TurboGrid：专业旋转机械叶片网格划分器。
- Vista TF：旋转机械叶片设计辅助工具。

注意：集成在 Workbench 中的各模块有些可以独立于 Workbench 运行，如经典 ANSYS、FLUENT 等，有些则必须通过 Workbench 才能运行。

经典 ANSYS 自 ANSYS 诞生以来界面一直未发生太大变化，如图 1-3 所示，虽界面简介，但 ANSYS Mechanical APDL 功能却十分丰富。本书所谓的二次开发，包括 APDL 的二次开发及 UPFs 的二次开发均是针对经典 ANSYS 而言的，与 ANSYS Workbench 无关。

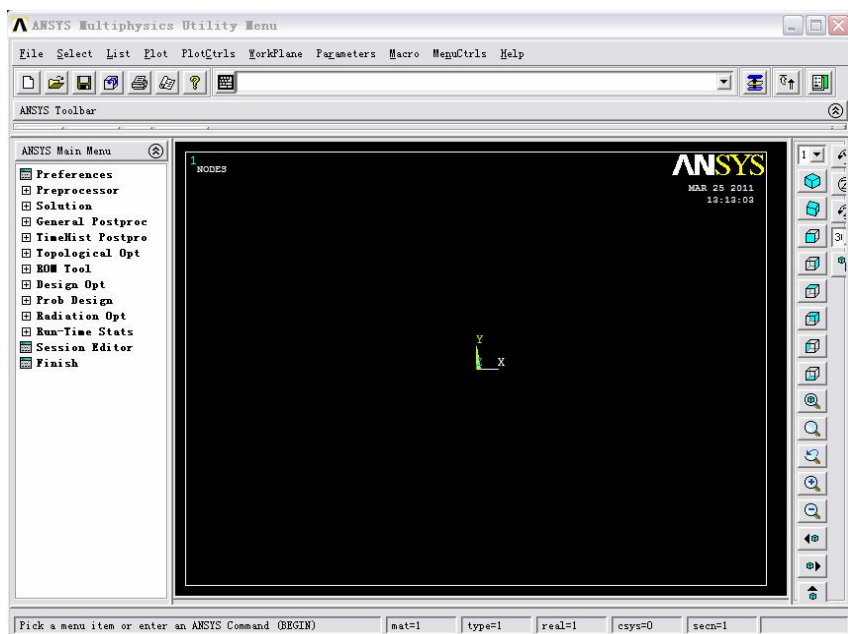


图 1-3 ANSYS 12.0 经典 ANSYS 交互界面

### 1.1.3 其他有限元软件

有限元是一套博大精深的科学方法，在其多年来的发展过程中诞生了许多有限元软件，除了众所周知的 ANSYS 之外，还有许多其他商业的及开源的软件，在这些有限元软件中很多都为用户提供了强大的二次开发接口，对于科研人员及工程技术人员而言，对这些软件有个大致了解是很有益处的。

#### 1. 商业软件

##### (1) ABAQUS

ABAQUS 是一款功能强大的有限元软件，是世界上最著名的非线性有限元分析软件之一，是由美国达索 SIMULIA 公司（原 ABAQUS 公司）开发、维护及销售的有限元分析软件。最新版本为 2010 年推出的 ABAQUS 6.10 版，该版本推出了众多新的功能，同时也改进了以前版本的很多功能。ABAQUS 可以解决从相对简单的线性分析到极富挑战性的非线性模拟等各种问题，拥有大量不同种类的单元类型、材料模型等，它不仅能够解决结构分析（应力/位移）问题，而且能够模拟和研究包括热传导、质量扩散、电子元件器的热控制（热电耦合分析）、声学、土壤力学（渗流—应力耦合分析）和压电分析等广阔领域中。

ABAQUS 是一个协同、开放、集成的多物理场仿真平台，由各个模块组合而成。ABAQUS 的分析模块有两个：ABAQUS/Standard，即通用分析模块，可求解绝大多数线性和非线性问题；ABAQUS/Explicit，显式分析模块，用于模拟瞬态问题。ABAQUS/CAE 是 ABAQUS 的交互式图形用户界面，具有强大的前后处理能力，其中子模块 ABAQUS/Viewer 用于后处理。ABAQUS/Aqua 是专门用于模拟海岸结构的模块，ABAQUS/Design 用于设计敏感性分析。此外还有其他模块，这里不再一一介绍，感兴趣的读者可访问 ABAQUS 中国官方网站 <http://www.abaqus.com.cn/> 获得更多了解。

ABAQUS 为用户提供了 FORTRAN 子程序二次开发接口, 类似于 ANSYS 的 UPFs (见 1.2 节), 它允许用户通过子程序以 FORTRAN 代码的形式来扩展主程序的功能, 给用户提供了强大而又灵活的用户子程序接口, 这些接口可使用户按照自己的要求灵活解决问题, 可大大地扩充 ABAQUS 的功能。如 UMAT 用户子程序用于在 ABAQUS/Standard 分析模块中实现用户自定义材料模型的开发, 又如 UEL 用户子程序用于开发用户自定义单元。此外, 通过 GUI 脚本可以创建新的图形用户界面和用户交互操作, 还可通过内核脚本 (Python 等语言) 实现前处理建模和后处理分析计算结果等。

### (2) MSC.Nastran

Nastran 是 1966 年美国国家航空航天局 (NASA) 为了满足当时航空航天工业对结构分析的迫切需求主持开发的大型应用有限元程序, 该程序功能强大, 得到了很好的评价。1971 年 MSC 公司 (<http://www.mscsoftware.com.cn>) 对原始的 Nastran 做了大量改进, 采用了新的单元库、增强了程序的功能、改进了用户界面、提高了运算精度和效率。特别对矩阵运算方法做了重大改进, 即而推出了自己的专利版本: MSC. Nastran。此后, 又有多家公司对 Nastran 进行改进, 但占据主导地位的仍是 MSC.Nastran。

MSC. Nastran 为用户提供了方便的模块化功能选项, 其主要功能模块有: 基本分析模块 (含静力、模态、屈曲、热应力、流-固耦合及数据库管理等)、动力学分析模块、热传导模块、非线性分析模块、设计灵敏度分析及优化模块、超级单元分析模块、气动弹性分析模块、高级对称分析模块以及用于二次开发的 DMAP 用户开发工具模块。

MSC. Nastran 具有开放的体系结构, 二次开发工具 DMAP 语言 (Direct Matrix Abstraction Program) 可深入 MSC.Nastran 的内核。一个 DMAP 模块可由成千上万个 FORTRAN 子程序组成, 并采用高效的矩阵处理方法。

2006 年, MSC 公司发布了 MD.Nastran, 该软件在继承了 MSC.Nastran 的基础上, 陆续集成了 Marc、Dytran、Sinda (热分析软件)、Dyna 和 Actran (声学分析软件) 等著名软件的先进技术, 大大增强了高级非线性、显式非线性、热分析、外噪声分析等功能。目前, 该软件最新版本是 MD.Nastran 2010。

### (3) MSC.Marc

MSC.Marc 是 MSC 公司推出的 (1999 年 MSC 公司收购了 MARC 公司) 一款功能齐全的高度非线性有限元软件, 具有极强的结构分析能力, 能满足学术界和工业界的多种需求。其应用领域已从开发初期的核电行业迅速扩展到国防、航空、航天、汽车、造船、铁道、石油化工、能源、电子元件、机械制造、材料工程、土木工程、医疗机械、冶金工艺和家用电器等诸多领域。

MSC.Marc 的主要模块有:

- 前后处理图形对话框 MSC.Marc/Mentat;
- 高度非线性有限元软件求解器 MSC.Marc;
- Marc 并行求解器 MSC.Marc Parallel;
- 六面体自动划分模块 MSC.Marc/Hexmesh。

此外 MSC.Marc 还拥有许多其他模块, 感兴趣的读者可访问 MSC 公司官方网站。

MSC.Marc 为用户提供了友好的二次开发子程序接口, 为分析复杂问题和二次开发带来了很大的方便。MSC.Marc 为用户提供了 100 多个 FORTRAN 用户子程序接口, 这些用户子程序

接口覆盖了除求解方法外 MSC.Marc 有限元分析的所有环节。利用这些子程序可以完成以下二次开发功能：定义加载、边界条件和状态变量；定义各向异性材料特性和本构关系；定义粘塑性和广义塑性材料；定义粘弹性材料；修改几何形状及定义输出量等。此外，Marc 还提供了对图形对话框界面 MSC.Marc/Mentat 进行二次开发的工具，利用它可以定制个性化的菜单，甚至可以进行菜单的汉化。

#### (4) ADINA

ADINA 是由国际上著名的美国麻省理工学院 K. J. Bathe 教授领导的 ADINA R&D 公司研究开发的商用工程软件，是基于有限元技术的大型通用分析仿真平台。该软件被广泛应用于各个工业领域的工程仿真计算，包括土木建筑、交通运输、机械制造、石油化工等各个领域。ADINA 有限元程序在处理结构非线性、流固耦合方面具有强大的优势，是国际上最重要的大型非线性有限元软件之一，对线性、非线性，静力、动力，传热，计算流体动力学，流固耦合等复杂问题具有强大优势，被业内人士认为是非线性有限元发展方向的代表。1981 年 ADINA 的非商业软件进入中国市场，为有限元在我国的应用起到了很好的推动作用，不但解决了许多急需解决的工程问题，其源代码也成为国内一些科研院所进行有限元程序研究开发的基础。ADINA 最新版本为 8.6，关于 ADINA 更多介绍参见 <http://www.adina.com.cn/>。

ADINA 系统主要包括下列六个模块：

- 用户界面 ADINA-AUI (ADINA User Interface)；
- 结构分析求解器 ADINA；
- 传热分析求解器 ADINA-T；
- 计算流体动力学 (CFD) 求解器 ADINA-F；
- 流体—结构耦合分析求解器 ADINA-FSI；
- 热—机械耦合分析求解器 ADINA-TMC。

ADINA System 是一个全集成系统，所有分析模块使用统一的 ADINA 用户界面 (ADINA-AUI)。

为满足用户需要，ADINA 提供了完善的二次开发环境。ADINA 8.X 版本具有完善的用户开发环境，能对材料本构关系、单元算法、单元失效准则、结构断裂判据和裂纹扩展规律以及边界条件进行二次开发。

在有限元理论几十年的发展历程中，诞生了许多优秀的商业有限元软件，它们各具特色，各有自己稳定的用户群体。除了本节列举的商业有限元软件之外还有许多其他软件，如 CMSOL、ALGOR 等，用户如有需要可查阅相关资料。

#### 2. 开源软件

所谓开源软件 (Open Source Softwares) 就是源代码公开，且可以被公众使用的软件，此外公众对软件的修改和发行也不受限制。比如大家熟悉的 Linux 操作系统，以及 OpenOffice 办公软件等都是典型的开源软件。在数值计算软件领域，也出现了很多以教学或研究为目的的开源软件，下面向读者介绍几个常用的有限元开源软件。

##### (1) FEAP

FEAP (Finite Element Analysis Program) 是美国加州大学伯克利分校土木与环境工程系 Robert L. Taylor 教授及其团队研制的通用有限元程序，研制目的是教学与科研，最新版本为 FEAP 8.3，官方网站是：<http://www.ce.berkeley.edu/projects/feap/>。该软件用 FORTRAN 语言编

写, 为开源软件, 但仍需收取较少的费用, 免费版本是 FEAPpv。

FEAP 支持众多操作系统如 Windows、Linux、UNIX 等。FEAP 拥有较快的计算速度, 软件本身带有后处理功能, 同时提供强劲的二次开发接口, FEAP 还有并行计算能力。FEAP 有完善网格划分功能, 广泛的线性、非线性求解算法, 可图形化显示网格划分并可用云图方式显示计算结果。包含众多单元类型, 如各种三维结构单元、温度单元、梁单元、平板单元及壳单元。FEAP 包括多种本构模型, 如线性及非线性弹性本构、粘弹性损伤本构模型、弹塑性模型等。

FEAPpv 是 FEAP 的个人版, FEAPpv 是免费的, 个人版可用来学习和研究有限元理论, FEAPpv 下载地址为 <http://www.ce.berkeley.edu/projects/feap/feappv/>。学习 FEAPpv 可参考 O.C. Zienkiewicz 及 Robert L. Taylor 等人的《The Finite Element Method》一书, 该书是与 FEAPpv 配套的书籍, 对于学习研究有限元理论很有帮助。

### (2) deal.II

deal.II 是一款基于 C++编写的开源有限元软件, 始于德国海德堡大学数值方法小组的工作, 其编写目的是促进先进有限元程序的快速发展, deal.II 主要用于学术研究, 也被用来解决实际工程项目。学习 deal.II 需要有 C++语言基础及较完善的有限元理论知识, 最新版本为 deal.II 7.0, 可到其官网下载: <http://www.dealii.org/>。deal.II 有如下特点:

- 具有统一的二次开发接口;
- 超强的自适应网格划分功能, 并可进行局部网格细分;
- 支持各种单元, 如各阶拉格朗日单元, 连续、非连续单元, 各阶 Nedelec 单元及 Raviart-Thomas 单元等;
- 完善的软件说明文档;
- 软件数据组织及算法结构清晰;
- 支持数种标准输出格式, 以方便进行后处理;
- 支持并行处理器, 支持多种操作系统。

### (3) libMesh

libMesh 是美国德克萨斯大学奥斯汀分校于 2002 年开始开发的用于求解偏微分方程的程序库, libMesh 有些方面类似于 deal.II, 比如同样基于 C++编写, 同样支持并行处理器, 同样擅长于自适应网格划分等, 其官方网站为: <http://libmesh.sourceforge.net/>。

libMesh 支持各种常用单元, 支持稳态及瞬态仿真分析。LibMesh 使用了许多已有的优秀的程序库, 如用支持并行计算的 PETSc 来求解线性方程组, 利用 SLEPc 程序库来解决特征值问题等。libMesh 程序库提供的组件包括: 各种通用 2D、3D 单元, 稀疏矩阵求解器, 网格划分器, 网格文件转换(支持输出各种格式网格文件)等。

### (4) OpenSees

OpenSees 的全称是 Open System for Earthquake Engineering Simulation(地震工程模拟开放体系), 官网地址为 <http://opensees.berkeley.edu/>。它是由加州大学伯克利分校于 1999 年推出的用于结构和岩土地震反应模拟的开源软件。OpenSees 广泛用于太平洋地震工程研究中心和美国其他一些大学和科研机构的科研项目中, 较好地模拟了包括钢筋混凝土结构、桥梁、岩土工程在内的众多的实际工程和振动台试验项目, 证明其有较好的非线性数值模拟精度。OpenSees 可以实现的分析过程包括:

- 静力线弹性分析;
- 静力非线性分析;
- 模态分析;
- pushover 拟动力分析;
- 动力线弹性分析;
- 动力非线性分析;
- 地震作用下的可靠度及灵敏度的分析。

由于 OpenSees 是开源软件, 故自从 1999 年推出以来, 该软件不断进行升级和提高, 加入了许多新的材料和单元, 引入了许多业已成熟的 FORTRAN 库文件为己所用 (如 FEAP、FEDEAS 材料), 更新了高效实用的运算法则和判敛准则, 允许多点输入地震波纪录, 并不断提高运算中的内存管理水平和计算效率, 允许用户在脚本层面对分析进行更多控制。此外, OpenSees 支持并行运算。正式基于 OpenSees 的以上特点, 近年来国内对 OpenSees 的研究使用逐年增加。

除了以上提及的软件外, 还有许多其他值得研究的开源软件, 例如 OpenFEM、FEDEAS、FFEP 等, 感兴趣的读者可查阅相关文件。

## 1.2 ANSYS 二次开发工具

用于 ANSYS 二次开发的工具主要有 4 个, 即 APDL、UPFs、UIDL 及 Tcl/Tk。使用以上工具可以建立新的材料模型 (非线性弹性、弹塑性、粘弹塑性、蠕变、超弹性等各种材料模型), 构建新的单元类型, 定义摩擦准则, 参数化建模, 优化分析, 构建流程化的 ANSYS 分析平台, 建立符合用户专业需求的 ANSYS 用户界面等。

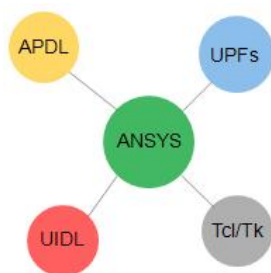


图 1-4 ANSYS 二次开发工具

APDL (ANSYS Parametric Design Language) ——ANSYS 参数化设计语言, 即通常所说的命令流; UPFs (User Programmable Features) ——用户可编程特性, 操作途径是对 ANSYS 核心 FORTRAN 代码进行修改, 对开发者有限元知识水平要求较高; UIDL (User Interface Design Language) ——用户界面设计语言; Tcl (Tool command language) ——工具命令语言, Tk 是基于 Tcl 的图形开发工具箱, 二者用于 ANSYS 界面开发, 比 UIDL 更加接近底层。

本节对这四种开发工具进行简单介绍, 由于本书重点是 APDL 及 UPFs, 故对于 UIDL 及 Tcl/Tk 本节给出简单应用实例, 本书其他部分不再介绍。

### 1.2.1 APDL 参数化设计语言

对于一些重复的结构分析，例如对模型的某一参数进行了修改，此时若全部重新建模、划分网格、加载、后处理，则需要很大的工作量。APDL 就是为解决这类问题而设计的，它是用来自动完成某些功能或建模的语言，使用非常方便，利用它对 ANSYS 进行二次开发可以极大地提高分析效率。

APDL 是 ANSYS Parametric Design Language 的简称，即 ANSYS 参数化设计语言，由类似于 FORTRAN 的语言部分和 1000 多条 ANSYS 命令组成。APDL 是一种解释性文本语言，有顺序、选择、循环及宏等结构。利用 APDL 将 ANSYS 命令组织起来，编写出参数化的用户程序，从而实现有限元分析的全过程，即建立参数化的实体模型、参数化的网格划分与控制、参数化的材料定义、参数化的载荷和边界条件定义、参数化的分析控制和求解以及参数化的后处理。本书第 2 章将对 APDL 基本知识进行介绍。

### 1.2.2 UPFs 用户可编程特性

UPFs 是 User Programmable Features（用户可编程特性）的简称，用户可以根据需要利用 UPFs 重新编译连接生成用户定制版本的 ANSYS 软件，例如创建新单元、定义新的材料属性、定义用户失效准则等，用户还可以编写自己的优化设计算法，甚至可以将整个 ANSYS 程序作为子程序调用。UPFs 是用户在 ANSYS 提供的 FORTRAN 源代码的基础上，修改其用户可编程子程序和函数（称为用户子程序），从源代码层次上对 ANSYS 进行二次开发的工具。用户需要在相应的 FORTRAN 语言编译器（ANSYS 同样支持非 FORTRAN 语言的编译器，如 C 语言，但需要 FORTRAN 语言外壳，且需要格外小心，不建议这种方式）的支持下，将编译修改后的源代码与 ANSYS 库相连形成用户版本的 ANSYS 可执行文件，另外还可以创建自己的外部命令。以下 ANSYS 产品支持 UPFs：

- ANSYS Multiphysics
- ANSYS Mechanical
- ANSYS Structural
- ANSYS Emag-Low Frequency
- ANSYS Emag-High Frequency
- ANSYS PrepPost

本书以 ANSYS 12.0 为例介绍。在安装 ANSYS 12.0 时，ANSYS 默认不安装 UPFs 二次开发相关文件，如图 1-5 所示，需要用户点选“ANSYS Customization Files”选项才能利用 UPFs 进行 ANSYS 二次开发，有些其他版本的 ANSYS 默认情况下也是不安装这些文件的，需要读者注意。

安装好后，ANSYS 12.0 会将相关的用于二次开发的 FORTRAN 文件存放在\Ansys Inc\v120\ansys\customize\user 路径下，如图 1-6 所示，用户可从中找到需要的用户子程序对其进行修改，然后编译连接（详见 3.5 节）即可生成用户自定义版本的 ANSYS。用户可以同时编译多个不同的用户子程序，当然，编译的用户子程序越多，花费的时间也就越长。

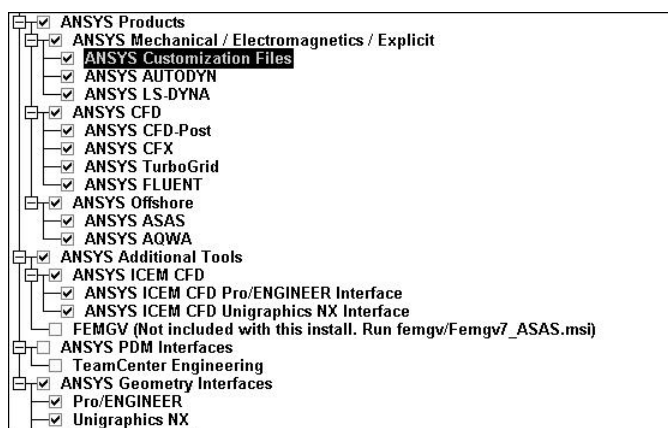


图 1-5 ANSYS 12.0 安装选项

提示: 在 ANSYS 11.0 及之前的版本中, ansys 文件夹中仅有 custom 一个二次开发文件夹, ANSYS 12.0 及其之后的版本增加了 customize 文件夹。

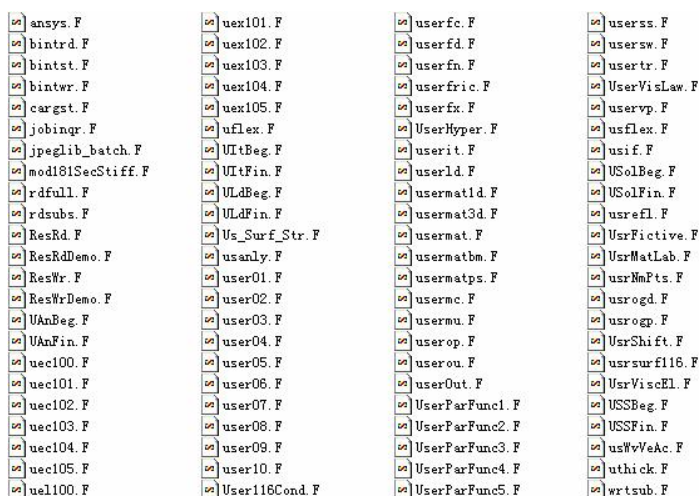


图 1-6 user 文件夹下的 FORTRAN 用户子程序

注意: 为防止 FORTRAN 用户子程序被错误修改而无法再次使用, 建议读者对以上 user 文件夹下的所有文件进行备份。

UPFs 的参考资料很少, 最重要的也是最权威的资料就是 ANSYS 软件帮助文档, 即 Guide to ANSYS User Programmable Features, 建议读者认真阅读本帮助文档。该文档在 ANSYS 12.0 帮助中的路径为: Mechanical APDL(formly ANSYS)>Programmer's Manual for Mechanical APDL>II, Guide to ANSYS User Programmable Features。ANSYS 12.0 之前的版本帮助文档中不含 Guide to ANSYS User Programmable Features, 读者可到网上下载阅读。本书第 3 章将介绍 UPFs 基本知识。

### 1.2.3 UIDL 界面设计语言

UIDL (User Interface Design Language) 即用户界面设计语言, 是 ANSYS 为用户提供的

专门进行程序界面设计的语言，允许用户改变 ANSYS 图形界面（GUI）中的一些组项，提供了一种允许用户灵活使用、组织设计 ANSYS 图形界面的强有力工具。

用 UIDL 编写的程序文件称为控制文件，必须以“\*.GRN”为扩展名，它可在 ANSYS 原有的菜单中添加自己设定的菜单项和控制程序。UIDL 控制文件由一个控制文件头和几个构造块构成，一个构造块对应着 GUI 的一个元件，每个构造块是一系列的 UIDL 命令，构造块分为菜单块和功能块两个类型，控制文件结构如图 1-7 所示。

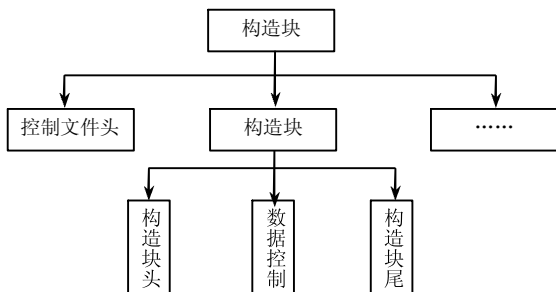


图 1-7 UIDL 控制文件结构

### 1. 控制文件头结构

控制文件头是一系列特定的 UIDL 命令，它们为 UIDL 处理器定义了控制文件的信息，要求每条命令行不得超过 80 字符，且必须以冒号（:）开头。典型的控制文件头如下所示：

```

:F UIMENU.GRN
:D Modified on % E%, Revision(SID) = % I%
:I      0,      0,      0
:!
```

说明如下：

- 控制文件头第一行必须有：F Filename，Filename 是 UIDL 控制文件名；
- 第二行必须有：D Description，Description 是对本文件的一些说明，一般由 ANSYS 自动添加；
- I 描述了 GUI 的位置信息，每个 0 必须出现在第 9、18、27 列，在运行过程中 ANSYS 会自动对其修改；
- ! 是分隔标记，可有可无，建议使用，以增强程序的可读性。

### 2. 构造块结构

构造块是 UIDL 控制文件的核心，它涵盖了菜单信息、命令信息以及帮助文件信息，按照其不同的功能类型可划分为菜单构造块、命令构造块和帮助构造块。构造块主要由 3 部分组成，分别是头部分(Block Header Section)、数据控制部分(Data Controls Section)和尾部分(Ending)。

头部分：

```

:N Men-MyProject
:S      0,      0,      0
:T Cmd
:A MyMenu
:C
:D My Own Project
```

数据控制部分:

```
Fnc_MyAnalysis
```

尾部分:

```
:E END
```

```
:!
```

说明如下:

- N 定义惟一的构造块名称;
- S 描述了构造块的位置信息, 每个 0 必须出现在第 9、16、23 列, ANSYS 在调用过程中会自动对其修改;
- T 描述构造块的类型, 可分为 Menu、Cmd、Help 等;
- A 对不同类型的构造块有不同功能, 在 Menu 块中为必需, 定义出现在 GUI 中的子菜单名称, 而在 Cmd 块中则为可选, 定义出现在菜单中的命令名称;
- D 描述构造块的信息;
- C 在 Cmd 块中为可选, 用于执行 ANSYS 命令、APDL 或用户函数等;
- 数据控制部分根据不同类型的构造块有不同的写法, 例如在菜单构造块中可以利用 Fnc\_MyAnalysis 调用用户函数等;
- 尾部分: E 标志着构造块的结束, 最后用 “!” 将各个构造块隔开。

### 3. ANSYS 调用 UIDL 的过程

以下均在 ANSYS 12.0 环境下进行说明。ANSYS 在启动时会自动在其安装目录下的\Ansys Inc\120\ansys\gui\en-us\UIDL 文件夹中搜寻 menulist120.ans 文件, 并调用其指向的 UIDL 文件。因此, 只需对 menulist120.ans 进行修改, 使其指向用户建立的 GRN 文件即可实现调用自己定制的 GUI 界面。打开 menulist120.ans 文件, 读者可发现 ANSYS 会调用 UIDL 文件夹下的 UIMENU.GRN、UIFUNC1.GRN 和 UIFUNC2.GRN。用户可以通过两种方式建立用户界面。第一种方法是将 UIMENU.GRN、UIFUNC1.GRN 和 UIFUNC2.GRN 拷贝至当前工作路径下, 对其重新编辑, 即可实现用户自定义 GUI 界面。第二种方法是建立新的 GRN 文件, 并在 menulist120.ans 中说明其路径。

**注意:** 不要直接对 ANSYS 的 UIDL 目录下的 GRN 文件进行修改, 否则可能会导致 ANSYS 无法正常启动。

### 4. 应用实例

本例通过建立一个数据输入窗口来介绍 UIDL 的应用。我们建立包含如下内容的 Input.GRN 文件 (\chp1\UIDL\Input.GRN):

```
:F Screwmenu.GRN      !建立名为 Screwmenu.GRN 的控制文件
:D Modified on %E%,Revision (SID) = %I%
:I      0,      0,      0 !0 位于 9, 18, 27 列
:!                                !以上四行是控制文件头文件
:N MenuRoot
:S      0,      0,      0 !0 位于 9, 16, 23 列
:T Menu                  !说明本模块是菜单模块
:A Ansys Main Menu      !主菜单菜单名
:D ANSYS Main MENU      !以上四行是模块首部
Men_UVBA_Main_T1        !以下是数据控制部分
Men_UVBA_Main_T2
Men_UVBA_Main_T3
```

```

Fnc_Preferences      !调用功能模块
Sep_                 !分割线
Men_Preproc          !调用菜单模块
Men_Solution
Men_GenlPost
Men_TimePost
K_LN (DROPTTEST)
Sep_
K_LN (DROPTTEST)
Men_DropTest
Sep_
Men_Topo
Men_ROM
Men_DesXplorerVT
K_LN (alpha)
Men_DesOpt
K_LN (ALPHA)
Men_DesOpt_al
Men_ProbDesign
Men_Aux12
Men_RunStat
Sep_
Fnc_UNDO
Sep_
Fnc_FINISH           !以上是主菜单的一般固定格式
Sep_                 !分割线
Men_input            !添加的螺旋结构参数化建模菜单
K_LN (UTILMENU)
Men_UtilMenu
Men_UVBA_Main_B1
Men_UVBA_Main_B2
Men_UVBA_Main_B3
:E END               !该行是模块尾行
:N Men_input
:S      0,      0,      0
:T Menu
:A Input Parameter
:D Input Parameter
:C) Multipro,'start',4 !创建对话框
:C) /prep7
:C) *cset,1,3,A,'Parameter A',5      !参数 A, 默认值为 5
:C) *cset,4,6,B,'Parameter B',10     !参数 B, 默认值为 10
:C) *cset,7,9,C,'Parameter C',15     !参数 C, 默认值为 15
:C) *cset,10,12,D,'Parameter D',20   !参数 D, 默认值为 20
:C) Multipro,'end'
:E END

```

将上述 Input.GRN 文件拷贝到前面所述的 ANSYS 软件 UIDL 目录下,并在此目录下的 menulist120.ans 文件中最后一行添加如下语句:

```
%ANSYS120_DIR%\gui\en-us\UIDL\Input.GRN
```

运行 ANSYS 后,在 ANSYS 主菜单(ANSYS Main Menu)最下端多了一项 Input Parameter,单击该菜单项会弹出如图 1-9 所示的参数输入对话框。输入参数后单击 OK 按钮,则参数 A、B、C、D 传入 ANSYS。

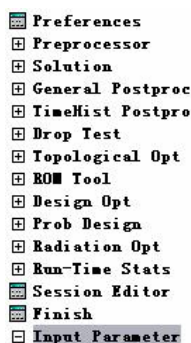


图 1-8 UIDL 控制文件结构

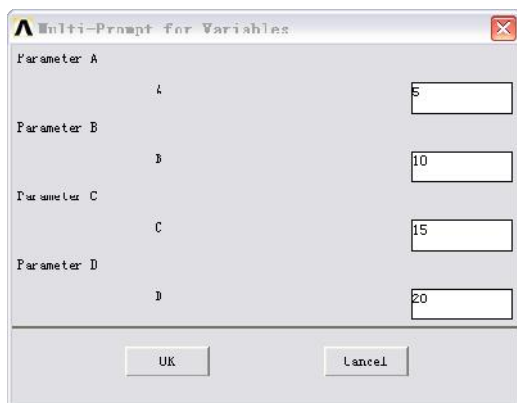


图 1-9 参数输入对话框

### 1.2.4 Tcl/Tk 语言

UIDL 是目前编写 ANSYS 用户界面大多采用的开发工具，但编写出的界面功能有限，有时不能很好地满足用户的要求。Tcl 为工具命令语言（Tool command language）的缩写，它是一种脚本语言，可以很容易地添加到其他应用程序中。Tk 是基于 Tcl 的图形开发工具箱，是 Tcl 的重要扩展部分。利用 Tcl/Tk 就可以构建出满足用户要求的界面。该语言是由伯克利大学 John Ousterhout 教授于 20 世纪 80 年代开发，由于其代码是开源的故近年来得到了快速发展，从 ANSYS 5.7 开始，已将 Tcl 融入经典 ANSYS 分析环境中，并被应用到 ANSYS 界面的开发中来，从 ANSYS 6.1 开始则完全兼容了 Tcl/Tk。

Tcl 是一种解释性语言，运行它需要额外的解释器，这种解释器能解释执行以文本文件方式储存的 Tcl 程序（\*.Tcl 及 \*.Tk），常用的编译器有 tclsh 和 wish，分别用来解释 Tcl 和 Tk，由于 Tk 是 Tcl 的扩展，因此 wish 解释器也可用于解释 Tcl 程序。

#### 1. 用户界面的编写

对话框是用户界面的常见特征，一个典型的对话框需要如下部分：

##### （1）顶层窗口的建立

Tk 遵循层次结构。对话框是一个窗口，应首先建立它的窗口才能放置各种组件。可以使用 Tk 的 toplevel 命令创建一个顶层窗口的框架，例如：

```
toplevel .topWin -width 150 -height 50    #创建一个名为 topWin，尺寸为 150×50
wm title .topWin "Dialog"               #使用窗口管理器设定 topWin 的标题为"Dialog"
```

注意，“#”字符后为注释。

##### （2）组件的添加

Tk 提供了建立多种组件的命令，简单的例如，button 创建一个命令按钮、entry 创建一个单行文本输入组件等，还可以调用一些标准对话框，如打开文件、颜色选择等，用户可以按照自己的需要进行添加。每个组件都有很多属性，如果在创建的时候用户不指定它的属性值，它将以默认的值进行创建。

##### （3）组建的布局

组件在窗口中的布局由布局管理器安排。Tcl/Tk 中包含三种管理器：

- 打包摆放管理器（pack）：使用一个组件作为父窗口，然后在其中安排多个组件，它

不是确切指定每个窗口的摆放位置，而是由编程人员来定义一些窗口应该如何安放，然后打包器会实施具体的操作。

- 栅格摆放管理器 (grid): 在一种能够调整行列尺寸的栅格中安排组件。
- 定位摆放管理器 (place): 可以指定窗口确切的位置和尺寸，或指定组件的相对位置和相对尺寸，在程序设计中可以根据具体的情况选定恰当的布局管理器。

#### (4) APDL 与 Tcl/Tk 结合

APDL 命令可以方便地与 Tcl/Tk 结合，可以通过调用 ANSYS API 函数来实现：

- ans\_sendcommand: 向 ANSYS 传递 APDL 命令；
- ans\_getvalue: 从 ANSYS 数据库中获取变量值；
- ans\_getvector: 从 ANSYS 数据库中获取向量值。

使用举例：

```
ans_sendcommand "*set,ansVar,$tkVar"    #将变量 tkVar 赋给 ANSYS 数据库中变量 ansVar
set tkVar [ans_getvalue PARM,ansVar,VALUE] #将 ANSYS 数据库中变量 ansVar 赋给变量 tkVar
set tkArray [ans_getvector ansArray]      #将 ANSYS 数据库中数组 ansArray 赋给变量 tkArray
```

#### (5) 窗口的撤销

当窗口对话结束的时候，应该释放掉它占用的资源，对于 toplevel 来说也是如此，因为它不具有在结束对话时自动撤销资源的能力，所以必须由用户在例如 OK 和 Cancel 按钮的事件响应过程中加入窗口撤销语句，例如 destroy .topWin。

#### 2. 程序的调用

使用 Tcl/Tk 编写的界面文件通常以 tcl 为扩展名，这里假定源文件名 filename.tcl，在 ANSYS 中可以通过以下方式来调用该文件：

- 命令行方式下有三种：①~tcl,'source filename.tcl'；②~tk,'source filename.tcl'；③~eui,'source filename.tcl'。
- 宏文件方式。

#### 3. 应用实例

下面给出一个简单的例子，创建并运行 createnode.tcl，用于在 x-y 平面指定位置创建一个节点，如图 1-10 所示。具体步骤如下：

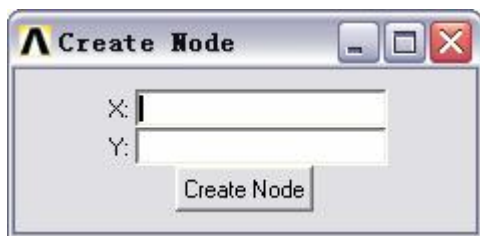


图 1-10 Tcl/Tk 创建节点对话框

- 打开 ANSYS，注意当前工作路径；
- 打开记事本程序，输入以下代码 (\chap1\Tcl\_Tk\createnode.tcl)：

```
destroy .toplevel    #撤销原 toplevel 窗口
set t [toplevel .toplevel] #新建名为 toplevel 的窗口，并存入变量 t 内
wm title $t "Create Node" #为窗口设定标题 "Create Node"
set xLab [label $t.lx -text "X:"] #建立名为 xLab 的标签
```

```

set yLab [label $t.ly -text "Y:"]           #建立名为 yLab 的标签
set xEntry [entry $t.ex -textvariable x]    #建立名为 xEntry 的文本框，值将被传入 x
set yEntry [entry $t.ey -textvariable y]    #建立名为 yEntry 的文本框，值将被传入 y
set b1 [button $t.b1 -command "CreateNode" \ #字符 “\” 为续行符
-text "Create Node"] #建立命令按钮 b1，按下按钮则执行 Create Node 过程
grid $xLab -row 0 -column 0                 #设置 xLab 标签位置
grid $xEntry -row 0 -column 1              #设置 xEntry 标签位置
grid $yLab -row 1 -column 0               #设置 yLab 标签位置
grid $yEntry -row 1 -column 1             #设置 yEntry 标签位置
grid $b1 -row 2 -column 0 -columnspan 2   #设置命令按钮 b1 位置
#CreateNode 过程
proc CreateNode {} {
    global x y                             #设置全局变量
    set NextNode [ans_getvalue "NODE,0,NUM,MAXD"] #从 ANSYS 数据库获得下一节点号
    ans_sendcommand "/prep7"               #前处理命令
    ans_sendcommand "n,$NextNode,$x,$y"    #建立节点命令
}

```

- 保存 createnode.tcl 于当前工作路径下；
- 在 ANSYS 中执行以下命令：

```
~tcl, 'source createnode.tcl'
```

本节简要介绍了 ANSYS 二次开发的工具。ANSYS 四种开发工具侧重点各不相同：UPFs 主要是通过编写一些新的程序实现 ANSYS 所不具有的特性，完成最复杂的二次开发工作，比如说构建新单元，复杂数据库交互，外围命令定制等；APDL 主要和 ANSYS 的分析部分频繁接触，对 ANSYS 的命令进行重新组合，以完成一系列分析任务；UIDL 主要是控制 GUI 界面；Tcl/Tk 用于创建更高级的界面。但是，各个工具间不是相互独立的，而是相互联系、协同工作的。例如，UPFs 在很多情况下也借助了 APDL 命令来调用其功能，同样也能在 UIDL 中嵌入 APDL 命令，以构建比较复杂的 GUI 二次开发工作。

## 参数化建模及分析

APDL 是重要的二次开发工具，具有各种参数定义、基本的数值计算、流程控制、宏程序与子函数的使用、数据文件的输入输出等功能。通过 APDL 可实现较复杂大型机械装备有限元建模时分析模型尺寸、材料性能、载荷、边界条件施加的位置和网格密度等的参数化，从而扩展传统有限元的分析范围，可完成许多通用性很强的参数化建模及工程分析。

有限元分析的标准过程包括：定义模型及其载荷、求解和解释结果，假如求解结果表明有必要修改设计，那么就必须改变模型的几何结构或载荷并重复上述步骤。特别是当模型较复杂或修改较多时，这个过程可能很浪费时间。APDL 用建立智能分析的手段为用户自动完成上述循环的功能。也就是说，程序的输入可设定为根据指定的函数、变量及选出的分析标准作决定。它允许复杂的数据输入，使用户对任何设计或分析属性有控制权，例如，几何尺寸、材料、边界条件和网格密度等，扩展了传统有限元分析范围以外的能力，并扩充了更高级运算包括灵敏度研究、零件参数化建模、设计修改及设计优化。为用户控制任何复杂计算的过程提供了极大的方便。

本章 4.1 节首先介绍 APDL 参数化建模及分析的基础知识，4.2 节和 4.3 节将分别结合两个例子介绍 APDL 参数化建模的过程，加深对参数化建模的理解，4.4 节结合采矿过程专用应力分析程序介绍参数化通用分析程序的应用。

### 本章要点：

- APDL 参数化建模
- APDL 通用分析程序

### 4.1 APDL 参数化建模及分析基础

本节首先介绍 APDL 参数化建模及分析的基本知识。

### 4.1.1 APDL 参数化建模基础

虽然可以利用一些几何建模软件,如 Solidworks、AutoCAD、Pro/E 以及 Catia 等建立几何模型,再以通用格式导入 ANSYS 软件,但这种方式不适合循环重复性的操作。比如需要对几何参数进行细微调整,则需要重新建立模型并导入 ANSYS,这是相当麻烦的。而利用 APDL 进行参数化建模可以解决这个问题,从而大大提高分析效率。

利用 APDL 进行参数化建模的一般步骤如下:

- Step 1** 初始化 ANSYS 系统。一般执行 FINISH, /CLEAR 清除内存,开始新的分析,还可指定工作文件名和标题。
- Step 2** 定义参数并赋值。这是参数化创建几何模型的关键步骤,一般必须在进入前处理器之前把需要的几何尺寸参数全部定义好,这样便于管理,不会导致混乱。不要在程序流程的不同位置随时定义参数,那样会使参数的定义太散乱,容易出现参数同名、不知道创建了的参数个数以及每个参数的作用混乱等问题。
- Step 3** 进入前处理器利用参数创建几何模型。注意:创建几何对象时一定要利用参数名作为几何尺寸进行输入,不可用参数值进行输入,否则几何尺寸固定不变,失去参数化建模的意义;另外,尽量采用人工控制编号的方法创建几何对象,并配合组件进行操作处理,如布尔运算、复制、移动、镜像等。
- Step 4** 在前处理器中划分单元网格模型。注意:尽量根据几何拓扑特征划分映射或者扫描单元网格,这些网格比较规则,容易控制单元形状,但自由网格常用于参数化建模过程,由于它可以实现任意拓扑的网格划分,适应性非常好。

了解了创建参数化几何模型的基本过程之后,一般需要整理模型的几何尺寸和定位尺寸,整理模型的几何尺寸链,既不要有多余的尺寸也不要缺少尺寸,然后再给每个需要采用参数进行定义的尺寸指定一个参数名,固定不变的尺寸可以直接采用尺寸数值进行处理,也可以将全部尺寸定义成参数,然后只调整需要变化尺寸的参数。

### 4.1.2 APDL 通用分析程序

在实际工作中,经常开发一些通用分析程序实现某一类专用功能,把这些专用的分析程序编写成一个个的宏文件,然后由一个主分析程序控制整个流程,在需要的时候调用所需的宏文件。当所有分析程序和宏文件都编好后,可以在 Toolbar 上建立按钮进行专用分析程序的驱动和调用,实现专用分析流程的过程控制。

## 4.2 建立参数化弹簧模型

螺旋体结构的弹簧在工程中应用广泛,借助 ANSYS 对其进行分析,具有重要的现实意义,本节结合一个例子介绍螺旋体结构的参数化建模过程。

### 4.2.1 问题描述

CAD 模型的创建是有限元分析的第一步,也是关键的一步,但 ANSYS 本身并没有对螺旋结构的建模提供直接的方法。本节利用 ANSYS 提供的二次开发工具 APDL 实现弹簧参数化

建模，为此类结构的建模提供一种通用有效的方法。

建立弹簧首先要建立螺旋线，再沿着螺旋线拉伸生成弹簧丝。螺旋线的参数方程：

$$\begin{cases} x = r \cos(t) \\ y = r \sin(t) \\ z = ht \end{cases} \quad (4-1)$$

式中， $r$  —螺旋半径；

$h$  —螺距；

$t$  —参数。

在柱坐标下，可以直接建立螺旋线的关键点，无需使用上述参数方程。

#### 4.2.2 模型建立

参数化建模的关键是几何特征参数化。螺旋结构的主要特征是螺旋半径、螺距；另外，螺旋数目对模型的建成亦很重要，首先应将这三个变量参数化。例如：

```
rd=25      !定义螺旋半径 rd 为 25
```

有了这个参数，当需要改变螺旋半径时，就无需重写整个程序，只需将以上的赋值语句修改，例如将 25 改为 30，即可实现程序按半径 30 重新建模，提高了建模效率。

具体 APDL 语句如下 (\chp4\spring\spring.inp)：

(1) 初始化 ANSYS 环境

清空内存，开始新的分析：

```
finish
/clear
```

(2) 定义几何尺寸参数

参数化尺寸控制：

```
rd=25      !定义螺旋半径 rd 为 25
lj=3       !定义螺距 lj 为 3
ls=10      !定义螺数 ls 为 10 圈
rs=2       !定义弹簧丝径 rs=2
```

(3) 曲线的离散

由于 ANSYS 并未提供连续曲线的直接建模方法，故本文将把此类连续曲线离散化，即将线离散成点，自底向上构造此类曲线的模型。在柱坐标下，将模型离散化会产生若干关键点，需把这些关键点存储起来以备建立螺旋线时使用，用 ANSYS 提供的数组变量可以方便地实现这一点，相关命令如下：

```
*dim,theta,array,ls*3+1      !定义数组 theta，长度可根据螺数 ls 调整
*vfill,theta(1),ramp,0,360/3 !给数组 theta 赋值
```

(4) 建立螺旋线

首先进入前处理器：

```
/prep7
*afun,deg      !角度
csys,1         !柱坐标系
```

下面将根据前面定义的数组在 ANSYS 前处理器中创建关键点，再把这些关键点连接成线，由此建立起螺旋线的几何模型。创建关键点的相关命令如下：

```
*do,i,1,ls*3+1
  k,i,rd,theta(i),lj*(i-1)
```

```
*enddo
```

以上命令用到了循环控制结构 DO-ENDDO，可以方便地控制数组元素生成关键点，以下还将用这个结构控制曲线的生成，创建曲线的相关命令如下：

```
*do,i,1,ls*3  
  l,i,i+1  
*enddo
```

至此，螺旋线的建模过程已完成，结果见图 4-1。

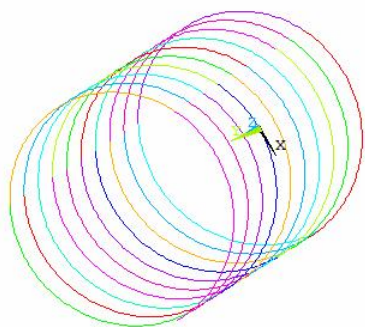


图 4-1 螺旋线模型

#### (5) 生成弹簧丝

创建工作平面，并在各工作平面上创建圆面，结果如图 4-2 所示：

```
*do,i,0,ls*3-1  
  lwplan,-1,i+1,0  
  cyl4,0,0,rs  
*enddo
```

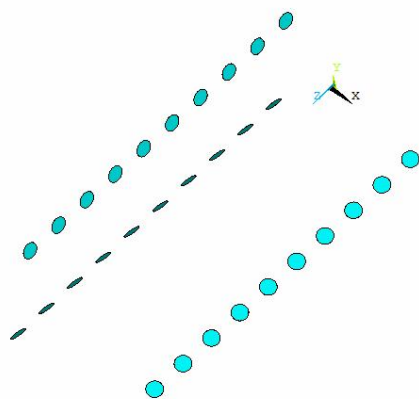


图 4-2 创建各拉伸圆面

由各圆面沿各段螺旋曲线拉伸生成螺旋体：

```
*do,i,1,ls*3  
  vdrag,i,,,,,i,,,,,  
*enddo
```

至此，弹簧建模完成，如图 4-3 所示。

在参数化的建模过程中可以简单地修改其中的参数，本例中，将螺旋半径 rd 改为 15，螺

距改为 5，螺数改为 20，弹簧丝径 rs 改为 1，得到另一模型，如图 4-4 所示。

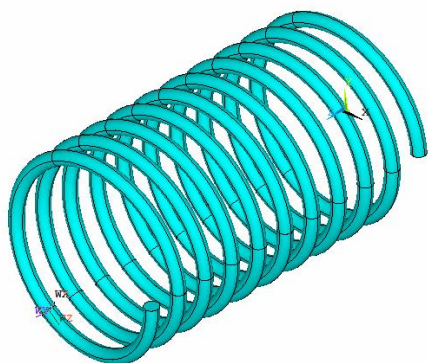


图 4-3 弹簧模型

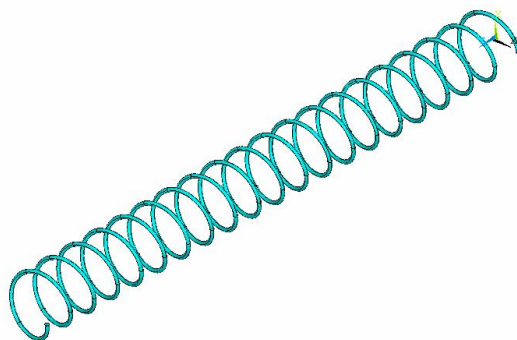


图 4-4 弹簧模型之二

#### (6) 划分网格

进行自由网格划分即可：

```
et,1,45  
type,1  
vmesh,all
```

网格划分结果如图 4-5 所示。

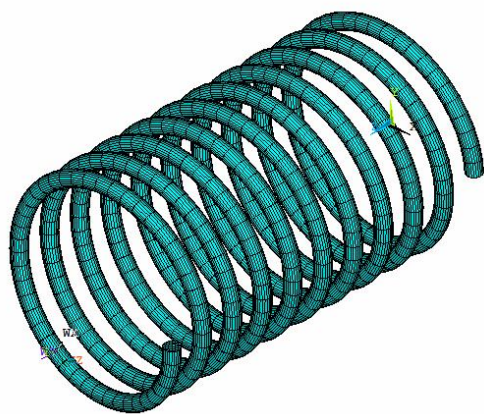


图 4-5 弹簧网格划分结果

## 4.3 建立渐开线圆柱齿轮模型

本节通过渐开线圆柱齿轮建模过程来进一步学习建立参数化有限元模型的过程。

### 4.3.1 问题描述

齿轮机构用于传递任意两轴之间的运动和动力，它是现代机器中最重要的传动机构之一。用有限元法对圆柱齿轮的轮齿进行变形和应力分析意义重大。渐开线齿轮是一种极为常用的齿

轮, 本节我们将要利用 APDL 参数化设计语言建立以下尺寸的渐开线圆柱齿轮: 端面模数  $m=6\text{mm}$ , 齿数  $Z=28$ , 斜齿圆柱齿轮分度圆柱上的螺旋角  $\beta=12^\circ$ 。

图 4-6 为渐开线形成原理示意图, 当一条直线 CK 沿一圆周作纯滚动时直线上任意点 K 的轨迹 AB, 就是该圆的渐开线。该圆为渐开线的基圆, 直线 KC 为渐开线的发生线, 角  $\alpha$  为渐开线上 K 点的展角, 角  $\theta$  为 K 点的压力角。渐开线的极坐标方程式为:

$$\begin{cases} R_k = R_b / \cos \alpha \\ \theta = \tan \alpha - \alpha \end{cases} \quad (4-1)$$

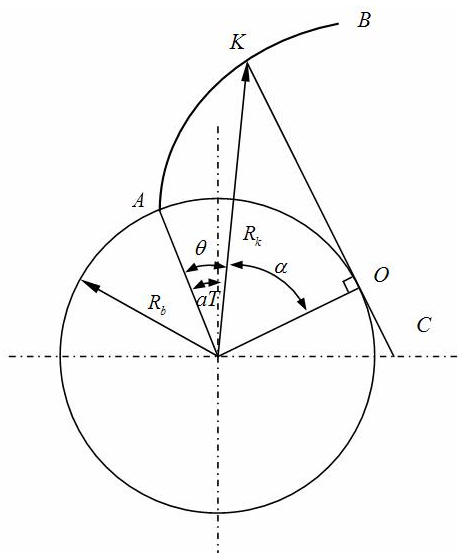


图 4-6 渐开线形成原理

式中:  $R_k$  为 K 点到圆心的距离;  $R_b$  为基圆半径。

每一对对应的  $R_k$  值和  $\theta$  值确定渐开线的一个点。渐开线上各点 (如 K 点) 的坐标为:

$$\begin{cases} X_k = R_k * \sin(aT - \theta) \\ Y_k = R_k * \cos(aT - \theta) \end{cases} \quad (4-2)$$

由于在 ANSYS 中不可以利用方程画曲线, 所以想要画出一条渐开线很困难, 也很难控制, 这就决定了渐开线齿廓只能用简单的线来做拟合, 不能直接生成渐开线, 我们采用小段直线来拟合, 即 ANSYS 生成的齿廓不是真正的渐开线, 而是由一段段小直线组成的曲折线, 近似于渐开线。齿廓渐开线分得段数越多, 拟合得就越逼真。但限于计算机配置以及考虑到运算速度, 拟把一个齿廓渐开线分成 20 段, 即在齿根圆的点与齿顶圆的点之间插入 19 个点, 从齿根圆到齿顶圆是一个半径等变的过程。

### 4.3.2 模型建立

采用自底向上的建模方法建立斜齿圆柱齿轮实体模型, 首先定义关键点, 再依次创建线、面和体。下面是用 APDL 语言参数化建模命令程序, 该命令流见 \chp4\gear\involute helical gear.inp。

(1) 初始化 ANSYS 环境

清空内存, 开始新的分析:

```
finish
/clear
```

## (2) 定义几何尺寸参数

参数化尺寸控制:

```
*afun,rad      !将角度单位转化为弧度
pi=acos(-1)    !圆周率 pi=-1 的反余弦值
*afun,deg      !把角度单位转化为度数
z=28           !齿数
m=6            !模数
beta=12        !分度圆柱上的螺旋角 β=12°
b0=14          !齿的支撑部分厚度
b=42           !齿宽
bb=54          !齿轮宽度
do=20          !齿的支撑部分开孔直径
n_hole=6       !齿的支撑部分开孔个数
d1=30          !齿轮轴孔直径
d2=50          !齿轮轮毂直径
d3=90          !齿的支撑部分开孔分度圆直径
d4=132         !齿的支撑部分最大直径
b1=10          !键槽宽度
h1=33          !键槽高度
r=z*m/2        !分度圆半径
alpha=360/z/4  !分度圆上一个齿厚对应的半角
rb=r*cos(20)   !基圆半径, 标准压力角为 20°
theta=(tan(20)-20*pi/180)*180/pi !标准压力角所对应的展角
anglet=alpha+theta
ra=m*(z+2)/2   !齿顶圆半径
```

## (3) 生成齿面

进入前处理器, 建立关键点:

```
/prep7
ai=20          !定义渐开线齿廓上 21 个点
*do,fai,rb,ra,(ra-rb)/ai
alfa=acos(rb/fai)
thet=(tan(alfa)-alfa*pi/180)*180/pi
x2=fai*sin(anglet-thet) !渐开线关键点 x 坐标
y2=fai*cos(anglet-thet) !渐开线关键点 y 坐标
k,fai*ai/(ra-rb),x2,y2 !定义渐开线关键点
*enddo
k1=kp(rb*sin(anglet-thet),rb*cos(anglet-thet),0)
k2=kp(ra*sin(anglet-thet),ra*cos(anglet-thet),0)
```

由关键点生成线:

```
*do,ii,k1,k2-1
l,ii,ii+1
*enddo
```

合并线:

```
lcomb,all,,0
```

生成一段齿根圆弧线:

```
flst,2,2,8
fitem,2,0,0,0
fitem,2,r-1.25*m,0,0
circle,p51x,,,90,,
```

生成基圆到齿根圆间齿廓曲线, 取已建成的渐开线一段:

```
kwpave,k1
csys,4
lsymm,y,1,,,0,0
lovlap,2,3
ldele,4,5,,1
ldele,7,,,1
```

生成一条完整的齿廓曲线:

```
wpcsys,-1,0
csys,0
nummrg,kp,,,10w
lcomb,all,,0
```

镜像齿廓曲线:

```
lsymm,x,1,,,0,0
```

生成齿顶圆的圆弧线: 用 3 点生成:

```
k,10,0,ra,0 !定义一个轮齿齿顶圆中间点
k3=kp(-ra*sin(anglet-thet),ra*cos(anglet-thet),0)
larc,k3,k2,10
```

生成一个圆环面:

```
cyl4,,,d4/2-10,90-360/z/2,r-1.25*m,90+360/z/2
adele,1
lovlap,1,2,4
lfillt,1,10,0.38*m, !齿根圆角半径 pf=0.38m
lfillt,2,8,0.38*m,,
ldele,9,,,1
lcomb,4,10,0
lcomb,8,11,0
```

生成一个完整面:

```
al,all
```

生成 z 个齿面:

```
csys,1
agen,z,1,,,0,360/z,0,,0
```

面相加操作:

```
aadd,all
al=arnext(0)
```

至此 z 个齿所构成的完整齿面已生成, 见图 4-7 渐开线齿面二维模型。

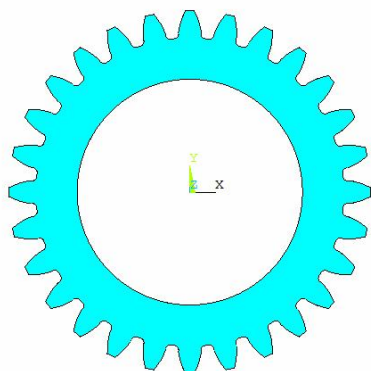


图 4-7 渐开线齿面二维模型

## (4) 生成齿圈体

创建螺旋线:

```

k,501,r,0,0
*if,beta,eq,0,then
k,502,r,0,b
*else
k,502,r,b*360/(2*pi*r/tan(beta)),b
*endif
l,501,502
ll=_return

```

由面挤出体:

```

vdrag,a1,,,,,ll
csys,0

```

至此, 由  $z$  个齿所构成的完整齿圈体已生成, 见图 4-8 渐开线齿圈体三维模型。

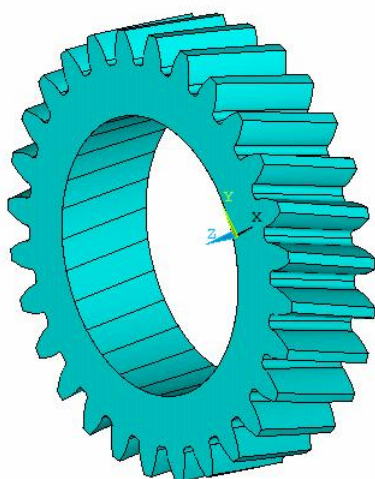


图 4-8 渐开线齿圈体三维模型

## (5) 生成齿的支撑部分

按照以下方法建立齿的支撑部分:

```

wpoff,0,d3/2,-b0
cyl4,,,do/2,,,b0
csys,1
vgen,n_hole,1,,,0,360/n_hole,0,0
vsel,u,volu,,3
cm,v1,volu
allsel
vsbv,3,v1
v1=vlnext(0)
csys,0

```

## (6) 在中心孔上开键槽

键槽是齿轮重要的结构, 用于齿轮与轴的紧固, 按照以下方法建立:

```

wpof,0,-d3/2,-(bb-b0)/2
block,-b1/2,b1/2,-(h1-d1/2),0,0,bb
vsbv,v1,1 !体相减

```

```
wpstyle,,,,,,,,,0 !关闭工作平面
```

### (7) 划分网格

直接采用自由网格划分方法即可:

```
et,1,solid92      !定义单元类型 1  
esize,5,0         !设定划分单元大小  
mshkey,0  
mshape,1,3d  
vmesh,all         !自由划分网格
```

最终建模结果如图 4-9 所示。

在定义几何尺寸参数部分改变齿轮结构参数为: 端面模数  $m=6.5\text{mm}$ , 齿数  $z=32$ , 斜齿圆柱齿轮分度圆柱上的螺旋角  $\beta=-12^\circ$ , 齿的支撑部分开  $N\_HOLE=4$  个孔, 执行上述命令流程可以得到另一个齿轮模型, 见图 4-10 齿轮参数化模型之二。

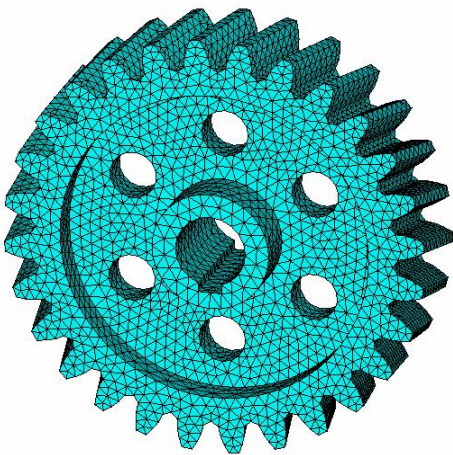


图 4-9 最终齿轮模型

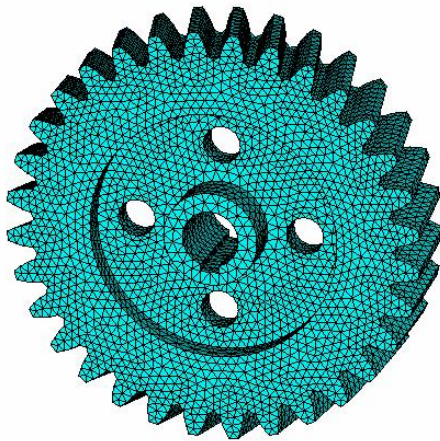


图 4-10 改变参数的模型二

在参数化的建模过程中可以简单地修改其中的参数, 从而进行参数化分析, 得到反复分析各种尺寸、不同载荷大小的多种设计方案或者序列性产品, 极大地提高分析效率, 减少分析成本。另外, APDL 也是 ANSYS 设计优化的基础, 只有创建了参数化的分析流程才能对其中的设计参数执行优化改进, 达到优化设计目的。总之, APDL 扩展了有限元分析能力, 提供了建立标准化零件库、序列化分析、设计修改、设计优化以及更高的数据分处理能力, 包括灵敏度研究等。

## 4.4 采矿过程专用应力分析程序

APDL 二次开发的另一个重要应用是建立专用分析程序, 基于 APDL 可以实现参数化有限元建模、参数化加载、参数化求解和参数化结果分析与处理, 实现 ANSYS 有限元分析全过程的参数化批处理。特别适合于开发标准分析过程的模块化、参数化和自动化, 可以很好地用于参数化有限元分析的专用分析系统的二次开发等, 大大提高工作效率。

本节将具有各功能的代码编写成一个宏文件, 然后由一个主分析程序控制整个流程, 在需要的时候调用相应的宏文件。当所有的分析程序和宏文件都编好后形成一个专用分析程序

时,可以在 Toolbar 上建立按钮进行专用分析程序的驱动和调用,实现专用分析流程的过程控制。本节结合采矿过程,建立一个功能相对简单的专用应力分析程序,该分析程序可以灵活地调整参数,如采场尺寸、开采步数、每步开采大小、材料参数等,此外本节还详细介绍了初始应力场的处理方法。本节所有宏文件在\chp4\mining 文件夹下。

#### 4.4.1 问题描述

##### 1. 基本问题

如图 4-11 所示,我们将问题简化为平面应变情形,采用 PLANE182 号单元(该单元支持初始应力场)。假定有一厚  $h_c$  (单位为米,下同)的煤层待开采,现研究开采区域围岩应力分布及地表沉陷的情况。

从地表到煤层底板所在地的材料见图 4-11,依次为风化层(高度用  $h_{fh}$  表示)、软质岩( $h_{rz}$ )、泥岩( $h_n$ )、煤层( $h_c$ )及砂岩( $h_s$ )。采掘长度为  $l_c$ ,模型总长度是  $l$ ,初始开采点  $s$  的坐标为  $(spx,hs)$ ,开采步数为  $nstep$  (假定每步开采距离相同)。

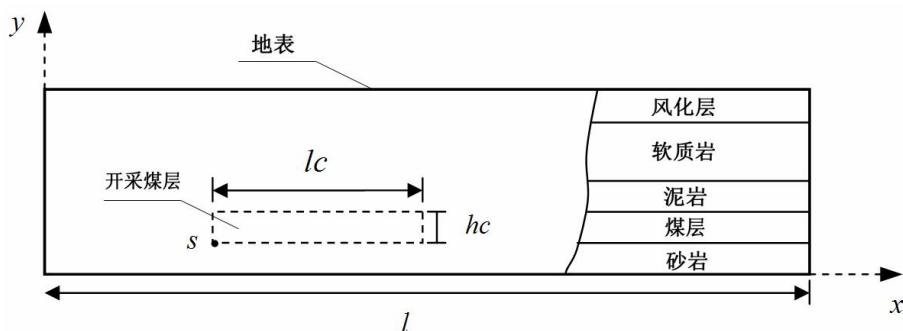


图 4-11 采矿过程模型

为了简化问题,所有地层均采用线弹性材料模型,表 4-1 列出了本例使用的各层参数。读者也可以采用其他路径相关的弹塑性材料,但不能使用 Drucker-Prager 材料模型,Drucker-Prager 支持 LINK1、LINK8、PIPE20、BEAM23、BEAM24、PLANE42、SOLID45、PIPE60、SOLID62、SOLID65、PLANE82、PLANE83、SOLID92 及 SOLID95 单元,不支持 PLANE182 单元,且 Drucker-Prager 本构模型支持的单元均不支持 inistate 命令(与初始应力场相关,后面将会介绍到)。

表 4-1 本例使用的各地层材料参数

岩层名称	弹性模量 $E$ (MPa)	泊松比 $\mu$	密度 $\rho$ ( $\text{kg/m}^3$ )
风化层	2000	0.15	2200
软质岩	4700	0.2	2300
泥岩	8500	0.2	2400
煤层	4900	0.3	1300
砂岩	14300	0.2	2500

## 2. 初始应力场

在开始应力分析前，首先应处理初始应力场的问题。大家都知道，煤层在开采前，由于长期的地质运动造成了相对稳定的状态，即认为在初始地应力（包括竖向应力和水平应力）的作用下，煤岩体处于稳定状态。岩体工程中的开挖实际上是将开挖面上的天然地应力部分地解除，从而使工程附近的地应力场发生变化（扰动），在应力场变化的同时也有相应的变形（位移）发生。换句话说，初始应力场不引起变形和位移，变形和位移仅仅是由开采扰动引起的。

本节仅考虑由重力场引起的竖向初始应力，即自重应力场。为了得到理想的分析结果，我们要做的工作是平衡掉初始应力场（重力引起的）产生的位移，即得到一个位移为零的初始状态。如果自重应力场就是初始应力场，直接将重力载荷施加于有限元模型，并施加相应的边界约束，计算得到在重力载荷下的应力场，使用 `inistate, write` 命令存入到 `.ist` 的文件中。再用 `inistate, read` 命令将得到的初始应力场文件和重力载荷一起施加于原始有限元模型，便可得到一个没有位移的初始应力场。可以模拟重力作用下位移稳定的土体应力场。有了应力场以后，可以进行单元的杀死（单元生/死技术），这样才有应力释放现象的产生。

在本节中，初应力文件的作用就是将重力产生的单元节点的位移调整为零，本身不改变应力值。为了进一步说明初始应力场的重要性，结合一个简单的例子，来比较说明有无初始应力场的区别。

图 4-12 是重力场作用下，不考虑初始应力场采场竖向位移分布图，图 4-13 是考虑初始应力场后采场竖向位移分布图。显然，结合实际采场顶板的冒落现象可见后者的结果是合理的，而前者采空区上方大范围的位移变化是直接由重力场产生的，而实际上在长期的地质稳定过程中，重力的这种影响早已平衡掉了，实际位移应该是由开挖扰动引起的。

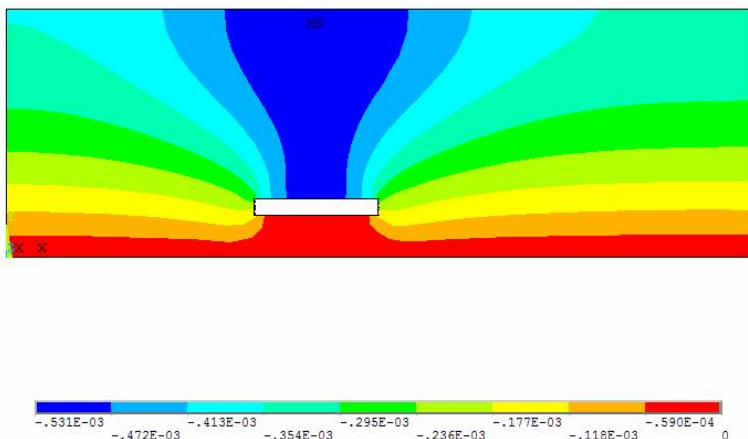


图 4-12 不考虑初始应力场时的竖向位移分布

如前面所述，ANSYS 12.0 关于初始应力场的命令是 `inistate`，使用该命令的一种方式组合使用 `inistate, write` 及 `inistate, read`。前者生成初始应力场文件，后者读入初始应力场文件，从而平衡掉由初始应力状态产生的位移。注意，`inistate` 还可用于初始应变（包括初始弹性应变及初始塑性应变）。此外还需指出，`inistate` 仅用于最新技术单元，不适用于传统单元。

**注意：**使用 `usermat` 定义的材料模型不支持 `inistate` 命令，假如 ANSYS 主程序调用了 `usermat`，则初应力文件及 `INISTATE, READ` 命令不起作用。

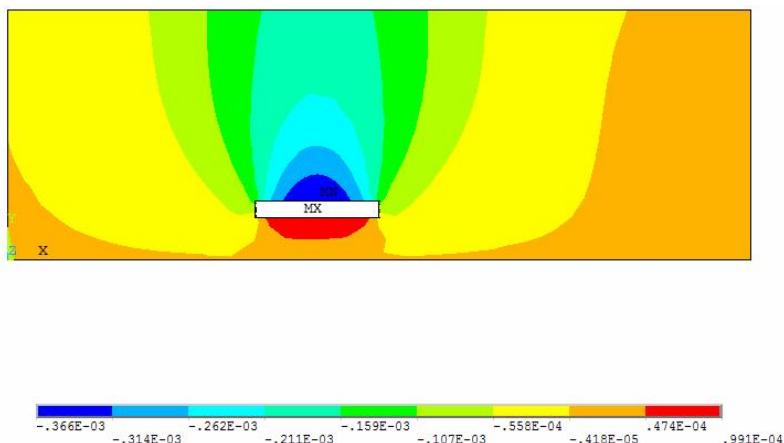


图 4-13 考虑初始应力场时的竖向位移分布

#### 4.4.2 参数化模型

实现上述问题的分析需要建立以下过程：参数化建模，求解和后处理。首先，在计算机某一指定目录下创建专用分析宏文件（本例将宏文件存储在 c:\mining 文件夹下）来实现不同的有限元分析功能，该专用分析程序包括以下宏文件，名称和功能分别如下：

- **model.mac**: 参数化有限元建模程序，实现参数化定义模型各尺寸，各岩层材料参数，网格划分密度等，划分网格后最终存储有限元模型 db 文件；
- **initial.mac**: 生成自重应力场初始应力文件，保存为 ist 文件；
- **solution.mac**: 读入初始应力场文件，开始各步的开挖求解；
- **displac\_y.mac**: 绘制指定载荷步结束时的竖向位移云图。

下面将分别介绍 model.mac 及 initial.mac。

##### 1. model.mac 宏文件

清空内存，开始新的分析：

```
finish
/clear
```

模型几何参数、开挖步数、网格划分密度等参数的定义：

```
l=200      !模型长度
hfh=15     !风化层高度
hrz=20     !软质岩高度
hn=10      !泥岩高度
hc=5       !煤层高度
hs=10      !砂岩高度
lc=100     !采掘长度
nstep=4    !开采步数
spx=50     !起始开采位置
msize=2    !网格划分大小
```

进入前处理器，设定单元类型，定义各层材料参数，建立几何模型，划分网格，最后保存模型文件：

```
/prep7
et,1,182
keyopt,1,3,2      !平面应变选项
```

```

!设置材料参数
mp,ex,1,2000e6          !风化层材料参数
mp,prxy,1,0.15
mp,dens,1,2200
mp,ex,2,4700e6          !软质岩材料参数
mp,prxy,2,0.2
mp,dens,2,2300
mp,ex,3,8500e6          !泥岩材料参数
mp,prxy,3,0.2
mp,dens,3,2400
mp,ex,4,4900e6          !煤层材料参数
mp,prxy,4,0.3
mp,dens,4,1300
mp,ex,5,14300e6         !砂岩材料参数
mp,prxy,5,0.2
mp,dens,5,2500
!建立关键点
k,1,0,0
k,2,1,0
k,3,1,hfh+hrz+hn+hc+hs
k,4,0,hfh+hrz+hn+hc+hs
k,5,0,hs
k,6,spx,hs
k,7,spx+lc,hs
k,8,1,hs
k,9,0,hs+hc
k,10,spx,hs+hc
k,11,spx+lc,hs+hc
k,12,1,hs+hc
k,13,0,hn+hc+hs
k,14,1,hn+hc+hs
k,15,0,hrz+hn+hc+hs
k,16,1,hrz+hn+hc+hs
!由关键点生成面
a,1,2,8,7,6,5
a,5,6,10,9
a,6,7,11,10
a,7,8,12,11
a,9,10,11,12,14,13
a,13,14,16,15
a,15,16,3,4
!给各个面上的单元赋属性
asel,,,1
aatt,5,,1,,
asel,,,2,4,1
aatt,4,,1,,
asel,,,5
aatt,3,,1,,
asel,,,6
aatt,2,,1,,

```

```

asel,,,7
aatt,1,,1,,
allsel,all
!网格划分
lsl,all
lesize,all,msize
allsel,all
amesh,all
!边界条件
dl,1,,ALL,
dl,2,,UX
dl,12,,UX
dl,14,,UX
dl,17,,UX
dl,20,,UX
dl,6,,UX
dl,9,,UX
dl,16,,UX
dl,19,,UX
dl,22,,UX
!重力场
acel,0,9.8
!存储模型
save,,
finish

```

## 2. initial.mac 宏文件

清空内存:

```

finish
/clear

```

进入求解器, 求解并输出初始应力场文件:

```

resume,,           !读入模型文件
finish
!进入求解器
/solu
time,1
autots,on
pred,on
nlgeom,on
nsubst,20,1000,10
outres,all,all
inistate,write,1,,,,,s !写初应力文件
solve
save,,             !保存
finish

```

INISTATE,WRITE 命令的用法如下:

INISTATE, WRITE, *FLAG*, , , *CSID*, *Dtype*

其中: FLAG=1 表示打开初状态输出开关(默认), FLAG=0 则关闭; CSID 决定初状态采用的坐标系(默认), CSID=0 表示在全局笛卡尔坐标系中输出初状态, CSID=-1 表示在材料坐标系中输出初状态, CSID=-2 表示在单元坐标系中输出初状态(对于杆、梁和分层单元);

Dtype 表示想要输出的变量，S 表示输出初应力，EPEL 表示输出初弹性应变，EPPL 表示输出初塑性应变。使用 INISTATE,WRITE 命令后默认生成 filename.ist 文件，存放在当前工作路径下，其中 filename 为当前 ANSYS 任务的文件名。

后面还会用到 INISTATE,READ，这里一并说明：

INISTATE, READ, *Fname*, *Ext*, *Path*

该命令用于读入初状态文件，Fname 是文件名，Ext 是扩展名，Path 为初状态文件存放路径（默认为当前工作路径）。实际应用时使用以下命令即可：

INISTATE, READ,,,

#### 4.4.3 开挖求解及后处理

开挖前首先应读入初始应力场文件，以平衡掉重力场产生的位移，开挖共分为 nstep 步。

##### 1. solution.mac 宏文件

使用 DO 循环执行各开挖步的求解，命令流如下：

```
finish
/clear
resume,,      !读入模型文件
/solu
inistate,read,, !读入初始应力文件
*do,k,1,nstep  !执行各开挖步的求解
    time,k
    autots,on
    pred,on
    nlgeom,on
    nsubst,20,1000,10
    outres,all,all
    nsel,s,loc,x,spx+(k-1)*lc/nstep,spx+k*lc/nstep  !选择本开挖步要挖掉的节点
    nsel,r,loc,y,hs,hs+hc
    esln,s
    ekill,all
    allsel,all
    solve
*enddo
save,,
finish
```

##### 2. displac\_y.mac 宏文件

注意，在后处理中绘制云图时应该将开挖掉的部分除去，即不显示开挖掉的部分。

```
finish
/clear
resume,,      !读入模型文件
/post1
esel,s,live    !不显示被开挖掉的部分
set,last
plnsol,u,y
```

#### 4.4.4 分析实例

修改 ANSYS 系统文件\ANSYS Inc\vl20\ansys\apdl\start120.ans (即 ANSYS 安装路径中的启动文件), 利用文本编辑器打开 start120.ans 文件, 在该文件的最后面添加一系列命令, 其中需要利用/PSEARCH 命令设置采矿过程专用分析程序的 4 个宏文件的存放路径, 本例我们存放在 c:\mining 文件夹下。另外利用\*ABBR 命令定义工具条 (Toolbar) 相关按钮, 通过工具条上的按钮调用在 c:\mining 文件夹下的各宏文件。

在 start120.ans 文件最后添加下列命令:

```
/psearch,c:\mining
*abbr,mining_model,model
*abbr,mining_initial stress,initial
*abbr,mining_solution,solution
*abbr,mining_displacement,displac_y
```

正常启动 ANSYS 12.0, 可见工具条 (Toolbar) 上相对默认状态增加了 4 个按钮, 如图 4-14 所示。

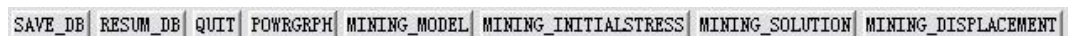


图 4-14 工具条

新增的各按钮分别对应的功能如下:

- 按钮 **MINING\_MODEL**: 运行宏文件 model.mac;
- 按钮 **MINING\_INITIALSTRESS**: 运行宏文件 initial.mac;
- 按钮 **MINING\_SOLUTION**: 运行宏文件 solution.mac;
- 按钮 **MINING\_DISPLACEMENT**: 运行宏文件 displac\_y.mac。

下面演示采矿过程专用分析程序的操作流程:

首先, 创建有限元模型。单击工具条上的 **MINING\_MODEL** 按钮, 即可自动创建有限元模型, 划分好网格之后的模型如图 4-15 所示。若想改变参数, 如修改各岩层的高度, 打开 c:\mining 文件夹下的 model.mac 进行相应的修改即可。

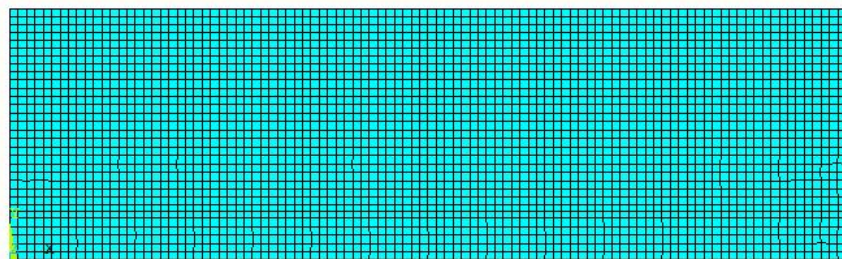


图 4-15 网格划分模型

接着单击工具条上的 **MINING\_INITIALSTRESS** 按钮, 求解只有重力作用下的应力场并输出初始应力场文件。只有重力作用下的初始应力场如图 4-16 所示。

然后单击工具条上的 **MINING\_SOLUTION** 按钮进行各开挖步的求解。

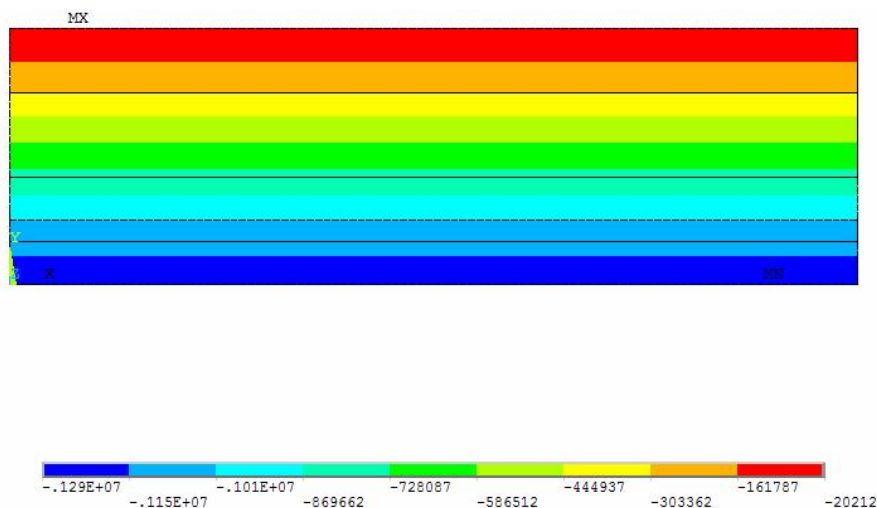


图 4-16 竖向初始应力场

最后进行后处理，查看竖向位移场，读者可修改 c:\mining 文件夹下的 displac\_y.mac 来确定要查看的子步数。单击工具条上的 MINING\_DISPLACEMENT 按钮，输出所有开挖步结束后的位移云图，如图 4-17 所示。

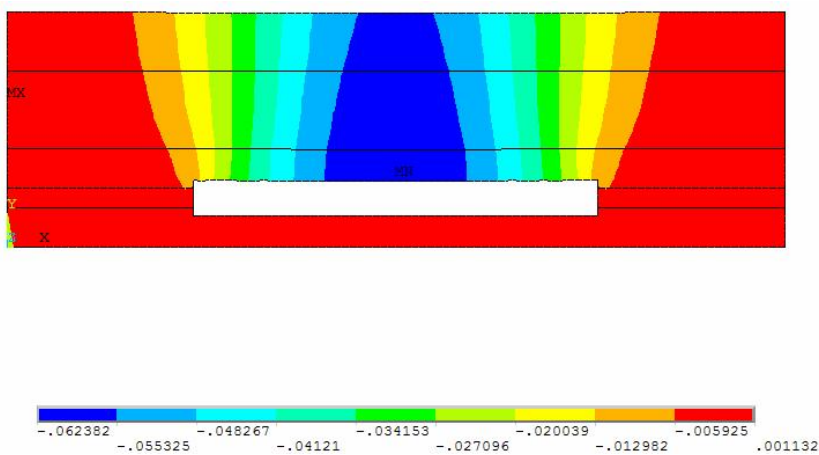


图 4-17 开挖完成后的竖向位移场

# 复杂弹塑性材料模型的二次开发

第 8 章介绍了简单材料模型的二次开发，包括线弹性模型（`usermat`）、超弹性模型（`UserHyper`）、损伤材料模型（`usermat`）、蠕变材料模型（`usercreep`）以及层合板材料的破坏准则（`userfc`）等。然而，工程中常用的材料很多都是弹塑性的，比如用于模拟金属材料的基于 Tresca 屈服准则或 von Mises 屈服准则的塑性模型，以及用于岩土工程的基于 Mohr-Coulomb 屈服准则或 Drucker-Prager 屈服准则的塑性模型，`usermat` 用户子程序能够胜任这类复杂弹塑性本构模型的二次开发任务。

本章首先介绍塑性理论的基础知识，接着介绍一个 `usermat` 用户子程序的应用实例。该例子是基于 Drucker-Prager 屈服准则的塑性模型，然后介绍其二次开发过程并给出了一个边坡稳定性分析应用实例。提醒读者注意，本章介绍的塑性材料模型是基于小变形理论（几何线性理论）的，而且仅考虑率无关塑性。

## 本章要点：

- 塑性理论基础知识
- Drucker-Prager 本构模型的开发

## 9.1 塑性理论基础

在介绍复杂塑性本构模型的二次开发之前，有必要介绍塑性理论的基本知识。介绍过程中只给出表达式，不给出推导过程，关于塑性理论更详尽的资料，读者可参考其他资料。塑性理论的任务是对具有弹塑性特性材料的应力应变关系给出理论表述，主要由三部分组成，即屈服准则、强化准则以及流动法则。屈服准则用于确定何时开始塑性变形（例如，应力达到一定的值时），本节介绍四种常用的屈服准则：Tresca 准则、von Mises 准则、Mohr-Coulomb 准则和 Drucker-Prager 准则。强化准则用于确定后继屈服面的演化规律，常见的强化准则有：理想塑性（无强化）、各向同性强化、随动强化。流动法则给出塑性应变增量的大小和方向，主要分为关联流动法则（屈服函数  $f$  与塑性势  $Q$  相同，适合于金属材料）和非关联流动法则（屈

服函数  $f$  与塑性势  $Q$  不相同, 适用于岩土类材料)。本节介绍的塑性理论中, 塑性应变不依赖时间, 即所谓的率无关塑性。

### 9.1.1 屈服准则

一般情况下, 屈服准则是应力的函数:

$$f(\sigma_{ij}) = k(\kappa) \quad (9-1)$$

式中,  $f$  表示某屈服函数;  $k$  是下一小节讨论的强化参数  $\kappa$  的函数。从物理角度分析, 任何屈服准则都不应依赖于所选取的坐标系方向, 因此只能是三个应力不变量的函数:

$$I_1 = \sigma_{ij} \quad (9-2)$$

$$I_2 = \frac{1}{2} \sigma_{ij} \sigma_{ij} \quad (9-3)$$

$$I_3 = \frac{1}{3} \sigma_{ij} \sigma_{jk} \sigma_{ki} \quad (9-4)$$

对于金属材料, Bridgeman 经过大量试验观察, 得出金属的塑性变形与静水压力无关的结论, 因此金属的屈服函数与  $I_1$  无关, 且只与应力偏量不变量有关, 于是有:

$$f(J_2, J_3) = k(\kappa) \quad (9-5)$$

式中,  $J_2$  和  $J_3$  分别是应力偏量的第二和第三不变量, 应力偏量:

$$\sigma'_{ij} = \sigma_{ij} - \frac{1}{3} \delta_{ij} \sigma_{kk} \quad (9-6)$$

下面首先介绍金属常用的屈服准则—Tresca 屈服准则。

#### 1. Tresca 屈服准则

Tresca 屈服准则认为当最大剪应力达到一定值时即开始屈服。若主应力为  $\sigma_1$ 、 $\sigma_2$  和  $\sigma_3$ , 且

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \quad (9-7)$$

则当

$$\sigma_1 - \sigma_3 = Y(\kappa) \quad (9-8)$$

时开始屈服, 其中  $Y$  是强化参数  $\kappa$  的函数, 可由试验确定。考虑其他各种可能的最大剪应力值, 可给出以下 6 种情况 (将 (9-8) 式右端移项, 然后用大写的  $\Phi$  表示屈服函数):

$$\begin{aligned} \Phi_1 &= \sigma_1 - \sigma_3 - Y(\kappa) \\ \Phi_2 &= \sigma_2 - \sigma_3 - Y(\kappa) \\ \Phi_3 &= \sigma_2 - \sigma_1 - Y(\kappa) \\ \Phi_4 &= \sigma_3 - \sigma_1 - Y(\kappa) \\ \Phi_5 &= \sigma_3 - \sigma_2 - Y(\kappa) \\ \Phi_6 &= \sigma_1 - \sigma_2 - Y(\kappa) \end{aligned} \quad (9-9)$$

可以看出, 在主应力空间  $\sigma_1 \sigma_2 \sigma_3$  内, 该准则可以用一无限长的正六边形柱体表示 (图 9-1), 轴线与  $\sigma_1 = \sigma_2 = \sigma_3$  的点确定的空间斜线重合, 而且由于静水压力不影响屈服, 因此柱体的每个正截面都相同。定义  $\sigma_1 + \sigma_2 + \sigma_3 = 0$  的平面为  $\pi$  平面, 则 Tresca 屈服面在  $\pi$  平面上的投影如图 9-2 所示。

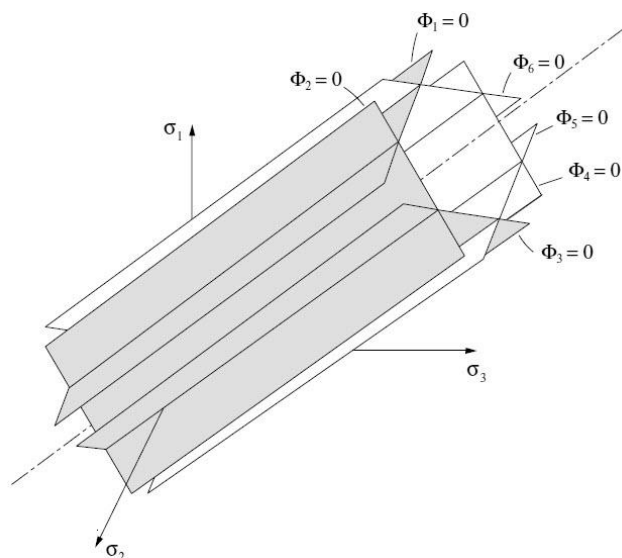
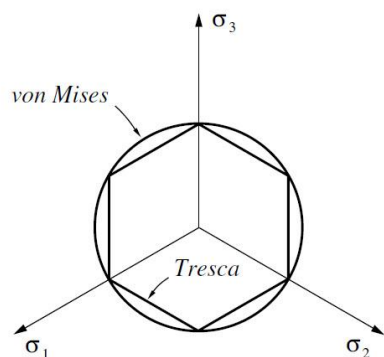


图 9-1 主应力空间 Tresca 屈服面

图 9-2 Tresca 和 von Mises 屈服面在  $\pi$  平面上的投影

## 2. von Mises 屈服准则

von Mises 在 1913 年提出，当  $J_2$  达到极限值时将产生屈服：

$$\sqrt{J_2} = k(\kappa) \quad (9-10)$$

应力偏量的第二不变量  $J_2$  可以写成如下形式：

$$\begin{aligned} J_2 &= \frac{1}{2} \sigma'_{ij} \sigma'_{ij} = \frac{1}{6} [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2] \\ &= \frac{1}{2} (\sigma_x'^2 + \sigma_y'^2 + \sigma_z'^2) + \tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2 \end{aligned} \quad (9-11)$$

屈服准则 (9-9) 式可进一步写成

$$q = \sqrt{3J_2} = \sqrt{3}k \quad (9-12)$$

$q$  称为 Mises 等效应力。通过研究简单应力状态下材料的屈服, 可以得到常量  $k$  的物理意义, 例如, 在纯剪情况 ( $\sigma_1 = -\sigma_2$ ,  $\sigma_3 = 0$ ) 下, 应用式 (9-9) 和式 (9-10) 时要求  $k$  必须等于屈服剪应力; 而在单向拉伸的情形 ( $\sigma_2 = \sigma_3 = 0$ ) 下, 则要求  $\sqrt{3}k$  是单向的屈服应力。

图 9-3 所描述的 von Mises 屈服面是一圆柱体, 它在  $\pi$  平面上的投影如图 9-2 所示, 是一个半径为  $\sqrt{2}k$  的圆, 而  $\pi$  平面上 Tresca 的屈服轨迹是从原点到顶点的距离为  $\sqrt{\frac{2}{3}}Y$  的六边形。

对于单向应力状态, 只要适当选择常数  $Y$  就可以使这两个屈服准则彼此一致并同实验相符。常量  $Y$  可以有不同的选择, 习惯上取  $Y = \sqrt{3}k$  以使圆通过六边形的顶点, 这里  $Y$  是简单拉伸时的屈服应力。在纯剪时这两个准则差别最大, 这时 von Mises 准则给出的屈服应力是 Tresca 准则的  $2/\sqrt{3}$  ( $\approx 1.15$ ) 倍。对于大多数金属, von Mises 准则与实验数据更加符合, 但由于 Tresca 准则在理论应用上比较简单, 故也经常使用。

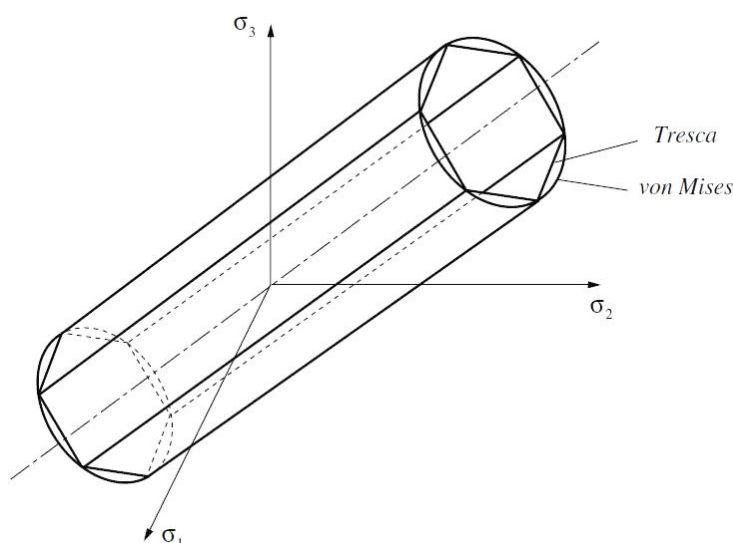


图 9-3 主应力空间 Tresca 和 von Mises 屈服面

### 3. Mohr-Coulomb 屈服准则

前面两种屈服准则均是和静水压力无关的, 适合于描述金属类材料。对于土、岩石以及混凝土之类的材料而言, 它们的屈服特性与静水压力有很大的关系。下面将要给出一个经典的考虑静水压力影响的屈服准则, 即 Mohr-Coulomb 屈服准则。

该准则假设宏观塑性屈服由材料颗粒间的摩擦滑移产生, 通过归纳总结 Coulomb 摩擦定律, 发现当剪应力  $\tau$ , 正应力  $\sigma_n$  满足下式时发生屈服:

$$\tau = c - \sigma_n \tan \phi \quad (9-13)$$

式中,  $c$  是粘聚力,  $\phi$  是内摩擦角, 正应力  $\sigma_n$  假设以拉伸为正。

Mohr-Coulomb 屈服准则可以通过图 9-4 所示的 Mohr 圆表示。从图中可以看出, 在由所有应力集构成的 Mohr 圆中, 由最大主应力  $\sigma_{\max}$  和最小主应力  $\sigma_{\min}$  构成的 Mohr 圆与直线  $\tau = c - \sigma_n \tan \phi$  相切, 三个 Mohr 圆均在该直线下方的应力状态处于弹性状态, 在该直线上方

的应力状态处于塑性状态, 因此也将该直线称做临界线。

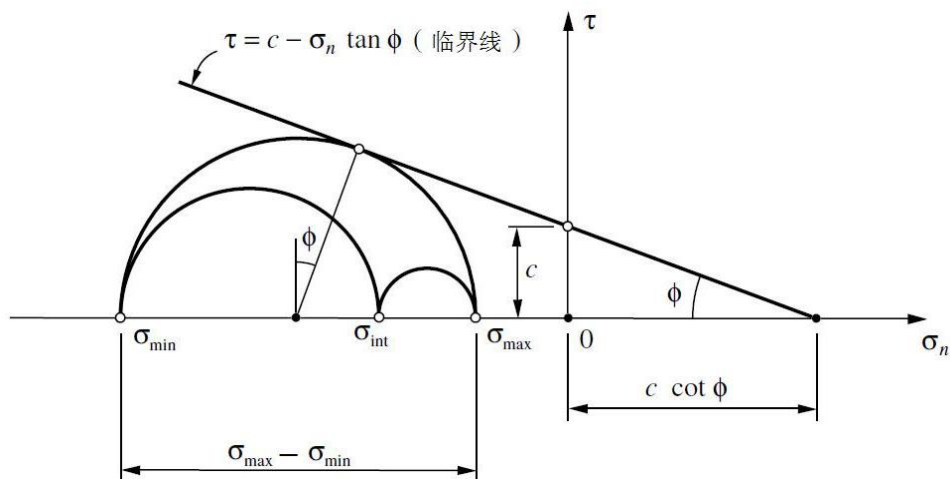


图 9-4 Mohr-Coulomb 屈服准则的 Mohr 圆表示法

(9-13) 式等价于

$$\frac{\sigma_{\max} - \sigma_{\min}}{2} \cos \phi = c - \left( \frac{\sigma_{\max} + \sigma_{\min}}{2} + \frac{\sigma_{\max} - \sigma_{\min}}{2} \sin \phi \right) \tan \phi \quad (9-14)$$

上式可进一步写成

$$(\sigma_{\max} - \sigma_{\min}) + (\sigma_{\max} + \sigma_{\min}) \sin \phi = 2c \cos \phi \quad (9-15)$$

因此, Mohr-Coulomb 屈服准则可以写成

$$\Phi = (\sigma_{\max} - \sigma_{\min}) + (\sigma_{\max} + \sigma_{\min}) \sin \phi - 2c \cos \phi \quad (9-16)$$

类似于 Tresca 屈服准则的处理方法, 有:

$$\begin{aligned} \Phi_1 &= \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3) \sin \phi - 2c \cos \phi \\ \Phi_2 &= \sigma_2 - \sigma_3 + (\sigma_2 + \sigma_3) \sin \phi - 2c \cos \phi \\ \Phi_3 &= \sigma_2 - \sigma_1 + (\sigma_2 + \sigma_1) \sin \phi - 2c \cos \phi \\ \Phi_4 &= \sigma_3 - \sigma_1 + (\sigma_3 + \sigma_1) \sin \phi - 2c \cos \phi \\ \Phi_5 &= \sigma_3 - \sigma_2 + (\sigma_3 + \sigma_2) \sin \phi - 2c \cos \phi \\ \Phi_6 &= \sigma_1 - \sigma_2 + (\sigma_1 + \sigma_2) \sin \phi - 2c \cos \phi \end{aligned} \quad (9-17)$$

以上表示方法称为多组面表示法。以上各式构成了一个锥形的屈服面 (由于静水压力影响屈服面, 所以屈服面为锥形而不是柱形), 如图 9-5 所示。对于上式中的第一式, 当  $\sigma_1 = \sigma_2 = \sigma_3$  时有:

$$(\sigma_1 + \sigma_3) \sin \phi = 2c \cos \phi \quad (9-18)$$

从而有:

$$\sigma_1 = \sigma_2 = \sigma_3 = c \cot \phi \quad (9-19)$$

因此顶点  $O$  位于  $\sigma_1 = \sigma_2 = \sigma_3 = c \cot \phi$  处。

#### 4. Drucker-Prager 屈服准则

Drucker-Prager 屈服准则是 Drucker 和 Prager 于 1952 年给出的对 Mohr-Coulomb 屈服准

则的光滑近似。它是在 von Mises 屈服准则中引入了一个附加项，从而考虑了静水压力的影响，即

$$\alpha I_1 + \sqrt{J_2} = k' \quad (9-20)$$

该准则的屈服面是圆锥形的，如图 9-6 所示。当  $\alpha=0$  时，Drucker-Prager 屈服准则退化为 von Mises 屈服准则。

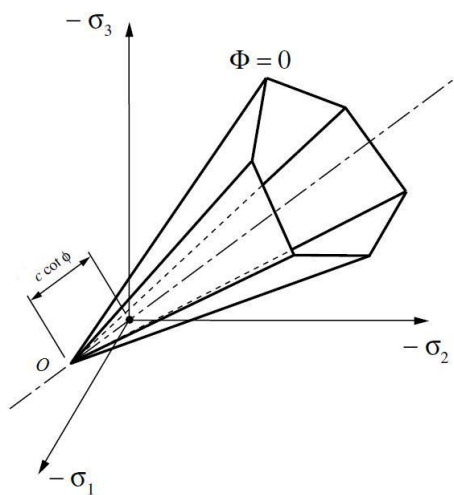


图 9-5 主应力空间 Mohr-Coulomb 屈服面

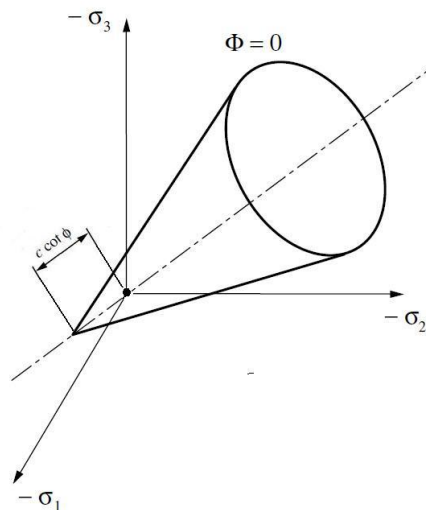


图 9-6 主应力空间 Drucker-Prager 屈服面

为了使 Drucker-Prager 屈服准则和 Mohr-Coulomb 屈服准则相互吻合，有两个常用的近似处理方法：每个截面上 Drucker-Prager 圆与 Mohr-Coulomb 六边形的外顶点吻合（如图 9-7 所示），则需要满足

$$\alpha = \frac{2 \sin \phi}{\sqrt{3}(3 - \sin \phi)} \quad k' = \frac{6c \cos \phi}{\sqrt{3}(3 - \sin \phi)} \quad (9-21)$$

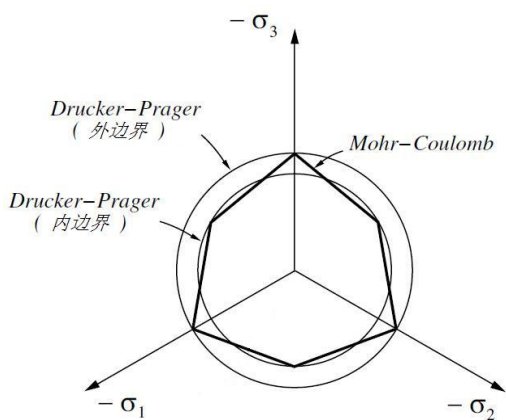


图 9-7 Mohr-Coulomb 和 Drucker-Prager 屈服面在  $\pi$  平面上的投影

每个截面上 Drucker-Prager 圆与 Mohr-Coulomb 六边形的内顶点吻合 (如图 9-7 所示), 则需要满足

$$\alpha = \frac{2 \sin \phi}{\sqrt{3}(3 + \sin \phi)} \quad k' = \frac{6c \cos \phi}{\sqrt{3}(3 + \sin \phi)} \quad (9-22)$$

### 9.1.2 强化准则

在初始屈服之后, 产生进一步塑性变形的应力值, 将依赖于当前的塑性应变的大小, 称为工作强化或应变强化, 随后的屈服面以某种方式依赖于塑性变形, 因此屈服面在塑性变形的各个阶段都是变化的。下面介绍几种常见的应变强化类型:

#### 1. 理想塑性

理想塑性不考虑强化, 其屈服应力的大小与变形的程度无关, 即屈服面保持固定。图 9-8 给出 von Mises 理想塑性屈服面示意图, 并给出了与其对应的单轴循环加载理想塑性应力—应变曲线。

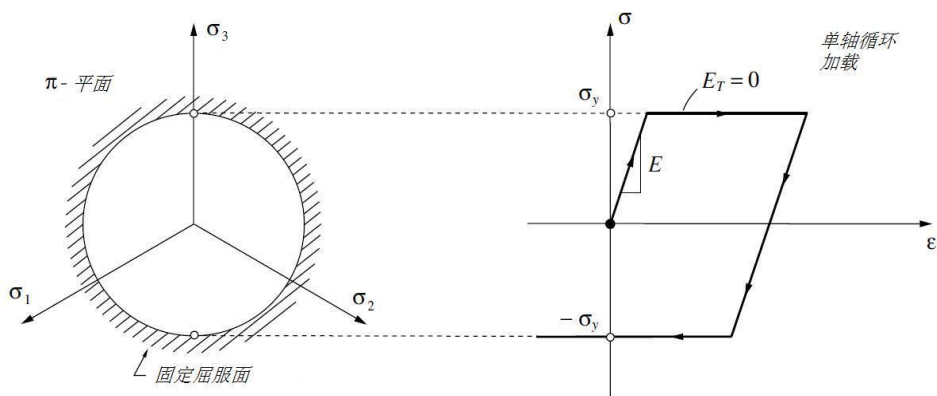


图 9-8 理想塑性

#### 2. 各向同性强化

如果随后的屈服面是初始屈服曲面无平移的均匀扩展, 那么这种应变强化模型就是各向同性强化。图 9-9 给出了 von Mises 各向同性强化示意图, 并给出了与其对应的单轴循环加载各向同性强化塑性应力—应变曲线。

强化参数  $\kappa$  用来表示屈服变量  $k$  与塑性变形的关系, 可以确定屈服面逐渐扩展的规律。参数  $\kappa$  常用的度量方法是, 考虑  $\kappa$  与总塑性变形程度相关, 引入等效塑性应变增量:

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}} \{ (d\varepsilon_{ij})_p (d\varepsilon_{ij})_p \} \quad (9-23)$$

在考虑与静水压力无关的屈服准则, 比如 Tresca 和 von Mises 屈服准则时, 由于  $(d\varepsilon_{ii})_p = 0$ , 所以偏塑性应变增量与塑性应变增量相同, 即  $(d\varepsilon'_{ij})_p = (d\varepsilon_{ij})_p$ 。因此, 我们可以把式 (9-23) 写成

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}} \{ (d\varepsilon'_{ij})_p (d\varepsilon'_{ij})_p \} \quad (9-24)$$

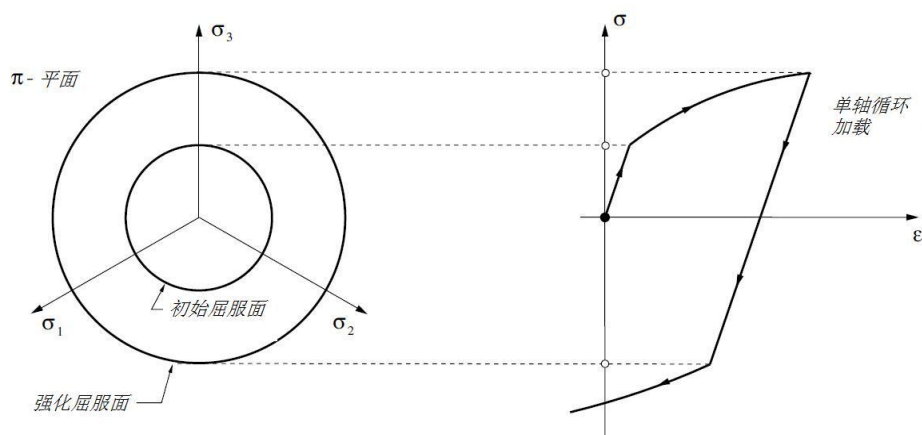


图 9-9 各向同性强化塑性

因此，不考虑静水压力时，强化参数可以写成

$$\kappa = \bar{\varepsilon}_p \quad (9-25)$$

式中， $\bar{\varepsilon}_p$  是在应变路径上积分  $d\bar{\varepsilon}_p$  的结果。

### 3. 随动强化

如果随后的屈服面保持着原来的形状和方位，而只在应力空间中像刚体一样平移，则称为随动强化。随动强化将会导致在循环加载下从实验观察到的 Bauschinger 效应，图 9-10 给出了 von Mises 模型随动强化示意图，并给出了与其对应的单轴循环加载随动强化塑性应力—应变曲线。

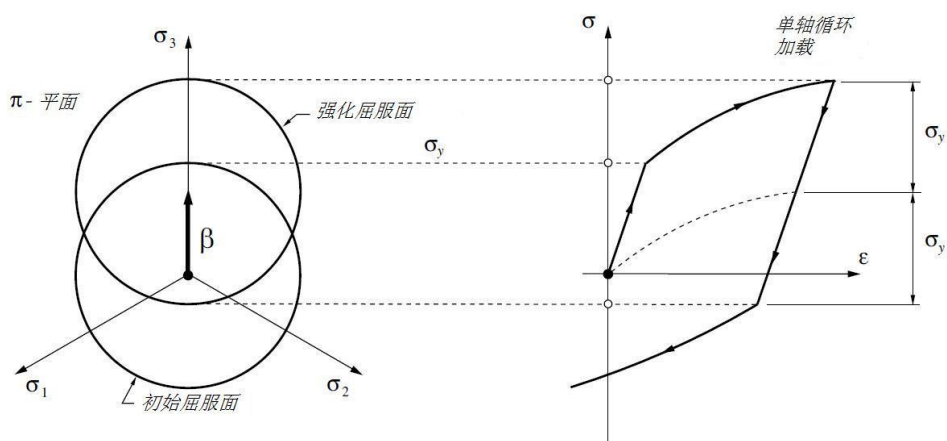


图 9-10 随动强化塑性

本章不讨论随动强化，仅讨论理想塑性和各向同性强化塑性。

### 9.1.3 流动法则

在介绍流动法则之前首先讨论加、卸载条件。对于式 (9-1)， $f = k$  的应力状态表示塑性

状态；当  $f < k$  时则表示处在弹性状态。在  $f = k$  的塑性状态时，由于应力增加的变化而引起的屈服函数的变化是

$$df = \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} \quad (9-26)$$

于是，有以下三种可能：

- $df < 0$ ：出现弹性卸载（弹性特征），应力点回到屈服面内。
- $df = 0$ ：中性加载（对于理想塑性材料为塑性特征），应力点保持在屈服面上。
- $df > 0$ ：塑性加载（对于应变强化材料为塑性特征），应力点保持在扩张了的屈服面上。还可以证明，对于稳定材料初始屈服面以及所有后继屈服面都必须是凸的。

材料在初次屈服之后，材料特性是部分弹性和部分塑性的。在任意应力增量过程中，应变增量分为弹性分量和塑性分量两部分，即

$$d\varepsilon_{ij} = (d\varepsilon_{ij})_e + (d\varepsilon_{ij})_p \quad (9-27)$$

其中，弹性应变增量可用下式表示

$$(d\varepsilon_{ij})_e = \frac{d\sigma'_{ij}}{2\mu} + \frac{(1-2\nu)}{E} \delta_{ij} d\sigma_{kk} \quad (9-28)$$

式中， $\mu$  为拉梅常数， $E$  为弹性模量， $\nu$  为泊松比。

为了导出塑性应变分量和应力增量之间的关系，必须对材料特性进行进一步的假设。这里给出一个塑性理论的重要假定：塑性应变增量与塑性势  $Q$  成正比，因此有

$$(d\varepsilon_{ij})_p = d\lambda \frac{\partial Q}{\partial \sigma_{ij}} \quad (9-29)$$

式中， $d\lambda$  是比例常数，称为塑性乘子。由于在屈服之后由上式控制着塑性流动，故称为流动法则。假如取  $Q = f$ ，则上述流动法又可称为相关联的流动法则，否则称为非关联流动法则。

关联流动法则可写成

$$(d\varepsilon_{ij})_p = d\lambda \frac{\partial f}{\partial \sigma_{ij}} \quad (9-30)$$

对于 von Mises 屈服准则而言

$$(d\varepsilon_{ij})_p = d\lambda \frac{\partial \sqrt{J_2}}{\partial \sigma_{ij}} = d\lambda \frac{1}{2\sqrt{J_2}} \cdot \frac{\partial J_2}{\partial \sigma_{ij}} = d\lambda \frac{\sigma'_{ij}}{2\sqrt{J_2}} \quad (9-31)$$

上式就是经典的 Prandtl-Reuss 流动法则。

将式 (9-28) 和式 (9-29) 代入式 (9-27)，可得到关联流动时弹塑性变形和应力间的关系式

$$d\varepsilon_{ij} = \frac{d\sigma'_{ij}}{2\mu} + \frac{(1-2\nu)}{E} \delta_{ij} d\sigma_{kk} + d\lambda \frac{\partial f}{\partial \sigma_{ij}} \quad (9-32)$$

关联流动法则应用于金属材料已得到了广泛的认可，但对于岩石和土壤类材料，使用关联流动法则确是不合适的，因为关联流动法则可能会过高地估计体积膨胀。但简单起见，本章仅介绍一个关联流动法则的例子。

### 9.1.4 写成矩阵形式

现在我们把前面介绍的塑性理论表达式（关联流动法则）转换成易于实施数值计算的矩阵形式。首先把式（9-1）定义的屈服准则改写成

$$\Phi(\boldsymbol{\sigma}, \kappa) = f(\boldsymbol{\sigma}) - k(\kappa) \quad (9-33)$$

微分上式有

$$d\Phi = \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} d\boldsymbol{\sigma} + \frac{\partial \Phi}{\partial \kappa} d\kappa = 0 \quad (9-34)$$

或

$$\mathbf{a}^T d\boldsymbol{\sigma} - A d\lambda = 0 \quad (9-35)$$

其中  $\mathbf{a}^T$  定义如下：

$$\mathbf{a}^T = \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \left[ \frac{\partial \Phi}{\partial \sigma_x} \quad \frac{\partial \Phi}{\partial \sigma_y} \quad \frac{\partial \Phi}{\partial \sigma_z} \quad \frac{\partial \Phi}{\partial \tau_{xy}} \quad \frac{\partial \Phi}{\partial \tau_{yz}} \quad \frac{\partial \Phi}{\partial \tau_{xz}} \right] \quad (9-36)$$

$A$  的定义如下：

$$A = -\frac{1}{d\lambda} \frac{\partial \Phi}{\partial \kappa} d\kappa \quad (9-37)$$

矢量  $\mathbf{a}$  叫流动矢量。式（9-32）即可写成矩阵形式

$$d\boldsymbol{\varepsilon} = \mathbf{D}_e^{-1} d\boldsymbol{\sigma} + d\lambda \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \quad (9-38)$$

上式中  $\mathbf{D}_e$  为材料弹性应力应变矩阵。将上式用  $d_D^T = \mathbf{a}^T \mathbf{D}_e$  来乘，并用式（9-35）消去  $\mathbf{a}^T d\boldsymbol{\sigma}$ ，于是得到如下形式的塑性乘子  $d\lambda$ ：

$$d\lambda = \frac{1}{H + \mathbf{a}^T \mathbf{D}_e \mathbf{a}} \mathbf{a}^T \mathbf{D}_e d\boldsymbol{\varepsilon} \quad (9-39)$$

为了数值计算方便，我们把上式中的  $\frac{1}{H + \mathbf{a}^T \mathbf{D}_e \mathbf{a}}$  记为 ABETA，把  $\mathbf{a}^T \mathbf{D}_e d\boldsymbol{\varepsilon}$  记为 AGASH。

将式（9-39）代入（9-38）式，我们得到整个弹塑性增量的应力应变关系为

$$d\boldsymbol{\sigma} = \mathbf{D}_{ep} d\boldsymbol{\varepsilon} \quad (9-40)$$

其中

$$\mathbf{D}_{ep} = \mathbf{D}_e - \frac{d_D d_D^T}{A + d_D^T \mathbf{a}}; \quad d_D = \mathbf{D}_e \mathbf{a} \quad (9-41)$$

实际上  $\mathbf{D}_{ep}$  就是 8.1 节所介绍的一致切线算子矩阵（见式（8-20）一致切线算子的定义式）。至此， $\mathbf{D}_{ep}$  表达式中，还需要确定标量  $H$ ，一般情况下， $H$  可由单轴应力—塑性应变曲线确定。

### 9.1.5 用应力不变量表示的屈服函数

为了便于数值计算，可以把屈服函数改写成适合于数值计算的用应力不变量表示的形式，其主要优点是可以利用计算机对普遍形式的屈服函数和流动法则编出程序，而对于任意种以上介绍的屈服准则，只需要三个常数就能确定。

主应力偏量  $\sigma'_1$ 、 $\sigma'_2$  和  $\sigma'_3$  是作为三次方程

$$t^3 - J_2 t - J_3 = 0 \quad (9-42)$$

的根给出的。注意到有以下三角恒等式：

$$\sin^3 \theta - \frac{3}{4} \sin \theta + \frac{1}{4} \sin 3\theta = 0 \quad (9-43)$$

将  $t = r \sin \theta$  代入式 (9-42) 中，有

$$\sin^3 \theta - \frac{J_2}{r^2} \sin \theta + \frac{J_3}{r^3} = 0 \quad (9-44)$$

比较式 (9-43) 和式 (9-44)，可得

$$r = \frac{2}{\sqrt{3}} \sqrt{J_2} \quad (9-45)$$

$$\sin 3\theta = -\frac{4J_3}{r^3} = -\frac{3\sqrt{3}}{2} \frac{J_3}{(J_2)^{3/2}} \quad (9-46)$$

由式 (9-46) 在  $\pm\pi/2$  范围内对于  $3\theta$  求出第一个根，用  $\theta$  代替第三不变量  $J_3$  来表示屈服函数是方便的。注意到  $\sin(3\theta + 2n\pi)$  的周期性，我们立刻得到三个（且只有三个） $\sin \theta$  的可能值，并由它们确定三个主应力的值，依次把  $\sin \theta$  的三个值代入  $t = r \sin \theta$  中，即给出三个主应力偏量值。用式 (9-45) 来替代  $r$ ，并加上平均静水应力分量给出的主应力分量为

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = 2\sqrt{\frac{J_2}{3}} \begin{pmatrix} \sin(\theta + \frac{2\pi}{3}) \\ \sin \theta \\ \sin(\theta + \frac{4\pi}{3}) \end{pmatrix} + \frac{I_1}{3} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (9-47)$$

式中， $\sigma_1 > \sigma_2 > \sigma_3$  且  $\frac{\pi}{6} \geq \theta \geq -\frac{\pi}{6}$ 。 $\theta$  实际上类似于 Lode 参数。本节所述的 4 个屈服准则，都可以用  $I_1$ 、 $J_2$  和  $\theta$  表示，下面分别介绍：

#### (1) Tresca 屈服准则

用式 (9-47) 中的  $\sigma_1$  和  $\sigma_3$  代入式 (9-8)，得出

$$2\sqrt{\frac{J_2}{3}} \left[ \sin\left(\theta + \frac{2\pi}{3}\right) - \sin\left(\theta + \frac{4\pi}{3}\right) \right] = Y(\kappa) \quad (9-48)$$

将上式展开可得

$$2\sqrt{J_2} \cos \theta - Y(\kappa) = 0 \quad (9-49)$$

#### (2) von Mises 屈服准则

因为 von Mises 屈服准则本来就仅与  $J_2$  有关，所以在这种情况下表达式没有变化，根据式 (9-10)，有

$$\sqrt{J_2} - k(\kappa) = 0 \quad (9-50)$$

#### (3) Mohr-Coulomb 屈服准则

用式 (9-47) 中的  $\sigma_1$  和  $\sigma_3$  代入 (9-17) 式，可得

$$\frac{1}{3}I_1 \sin \phi + \sqrt{J_2} \left( \cos \theta - \frac{1}{\sqrt{3}} \sin \theta \sin \phi \right) - c \cos \phi = 0 \quad (9-51)$$

#### (4) Drucker-Prager 屈服准则

该准则的表达式没有什么变化:

$$\alpha I_1 + \sqrt{J_2} - k' = 0 \quad (9-52)$$

其中, 参数  $\alpha$  和  $k'$  可直接由式 (9-21) 或式 (9-22) 获得。

为了计算 (9-41) 式的一致切线算子矩阵  $\mathbf{D}_{ep}$ , 首先要用适于数值计算的形式来表示流动矢量  $\mathbf{a}$ , 我们可以得出

$$\mathbf{a}^T = \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \frac{\partial \Phi}{\partial I_1} \frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial \Phi}{\partial \sqrt{J_2}} \frac{\partial \sqrt{J_2}}{\partial \boldsymbol{\sigma}} + \frac{\partial \Phi}{\partial \theta} \frac{\partial \theta}{\partial \boldsymbol{\sigma}} \quad (9-53)$$

式中

$$\boldsymbol{\sigma}^T = \{\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{xz}\} \quad (9-54)$$

微分式 (9-46), 我们得到

$$\frac{\partial \theta}{\partial \boldsymbol{\sigma}} = \frac{-\sqrt{3}}{2 \cos 3\theta} \left[ \frac{1}{(J_2)^{3/2}} \frac{\partial J_3}{\partial \boldsymbol{\sigma}} - \frac{3J_3}{(J_2)^2} \frac{\partial \sqrt{J_2}}{\partial \boldsymbol{\sigma}} \right] \quad (9-55)$$

将上式代入 (9-53) 式, 并利用 (9-46) 式, 则可写出

$$\mathbf{a} = C_1 \mathbf{a}_1 + C_2 \mathbf{a}_2 + C_3 \mathbf{a}_3 \quad (9-56)$$

式中

$$\begin{aligned} \mathbf{a}_1^T &= \frac{\partial I_1}{\partial \boldsymbol{\sigma}} = \{1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0\} \\ \mathbf{a}_2^T &= \frac{\partial \sqrt{J_2}}{\partial \boldsymbol{\sigma}} = \frac{1}{2\sqrt{J_2}} \{\sigma_x \quad \sigma_y \quad \sigma_z \quad 2\tau_{xy} \quad 2\tau_{yz} \quad 2\tau_{xz}\} \\ \mathbf{a}_3^T &= \frac{\partial J_3}{\partial \boldsymbol{\sigma}} = \left\{ \left( \sigma'_y \sigma'_z - \tau_{xy}^2 + \frac{J_2}{3} \right), \left( \sigma'_x \sigma'_z - \tau_{yz}^2 + \frac{J_2}{3} \right), \left( \sigma'_x \sigma'_y - \tau_{xz}^2 + \frac{J_2}{3} \right), \right. \\ &\quad \left. 2(\tau_{yz} \tau_{xz} - \sigma'_x \tau_{xy}), 2(\tau_{xz} \tau_{xy} - \sigma'_y \tau_{yz}), 2(\tau_{xy} \tau_{yz} - \sigma'_z \tau_{xz}) \right\} \end{aligned} \quad (9-57)$$

和

$$\begin{aligned} C_1 &= \frac{\partial \Phi}{\partial I_1} \\ C_2 &= \frac{\partial \Phi}{\partial \sqrt{J_2}} - \frac{\tan 3\theta}{\sqrt{J_2}} \frac{\partial \Phi}{\partial \theta} \\ C_3 &= \frac{-\sqrt{3}}{2 \cos 3\theta} \frac{1}{(J_2)^{3/2}} \frac{\partial \Phi}{\partial \theta} \end{aligned} \quad (9-58)$$

对于不同的屈服准则,  $C_1$ 、 $C_2$  和  $C_3$  可进一步求得, 如表 9-1 所示。

只有常数  $C_1$ 、 $C_2$  和  $C_3$  需要用屈服面来确定, 即由一个屈服面改变为另一个屈服面时, 需要改变这 3 个常数。表 9-1 给出了前面介绍的 4 种屈服准则对应的常数, 其他屈服准则采用类似的方法推导即可得到。

表 9-1 不同屈服准则对应的  $C_1$ 、 $C_2$  和  $C_3$  值

屈服准则	$C_1$	$C_2$	$C_3$
Tresca	0	$\alpha \cos \theta (1 + \tan \theta \tan 3\theta)$	$\frac{\sqrt{3}}{J_2} \frac{\sin \theta}{\cos 3\theta}$
von Mises	0	$\sqrt{3}$	0
Mohr-Coulomb	$\frac{1}{3} \sin \phi$	$\cos \theta [(1 + \tan \theta \tan 3\theta) + \sin \phi (\tan 3\theta - \tan \theta) / \sqrt{3}]$	$\frac{\sqrt{3} \sin \theta + \cos \theta \sin \phi}{2J_2 \cos 3\theta}$
Drucker-Prager	$\alpha$	1.0	0

需要注意, 对于 Tresca 屈服准则和 Mohr-Coulomb 屈服准则,  $\theta = \pm 30^\circ$  的角点处流动矢量  $\mathbf{a}$  无法确定, 因为此时表 9-1 中  $C_2$  和  $C_3$  的值就变得不确定了。对于  $|\theta| \leq 29^\circ$  的所有值, 采用表 9-1 中的值即可, 对于角点附近的值, 依据不同的屈服准则采用不同的处理方法:

(1) 对于 Tresca 屈服准则

当  $\theta = \pm 30^\circ$  时

$$C_1 = 0, C_2 = \sqrt{3}, C_3 = 0 \quad (9-59)$$

(2) 对于 Mohr-Coulomb 屈服准则

当  $\theta = 30^\circ$  时

$$C_1 = \frac{1}{3} \sin \phi, C_2 = \frac{1}{2} \left( \sqrt{3} - \frac{\sin \phi}{\sqrt{3}} \right), C_3 = 0 \quad (9-60)$$

当  $\theta = -30^\circ$  时

$$C_1 = \frac{1}{3} \sin \phi, C_2 = \frac{1}{2} \left( \sqrt{3} + \frac{\sin \phi}{\sqrt{3}} \right), C_3 = 0 \quad (9-61)$$

按照以上处理方法, 可使角点处的塑性流动方向惟一, 这种处理方法相当于“削去”了屈服面的角点。

本节介绍了塑性理论的基本知识, 下一节将结合本节介绍基于 Drucker-Prager 关联流动屈服准则的二次开发。

## 9.2 Drucker-Prager 本构模型的开发

本节依据前一节介绍的塑性理论知识, 介绍 Drucker-Prager 本构模型, 采用关联流动法则, 本节仅考虑较简单的二维平面应变模型。8.1 节详细介绍 usermat 用户子程序, usermat 用户子程序有两个主要任务: 一是应力更新过程, 即给定应变增量  $\Delta \boldsymbol{\varepsilon}$  计算得到应力增量  $\Delta \boldsymbol{\sigma}$ , 从而得到新的应力  $\boldsymbol{\sigma}$ ; 另一个是给出一致切线算子矩阵  $\mathbf{D}_{ep}$ 。这里不再介绍 usermat。

### 9.2.1 一致切线算子矩阵

对于弹性迭代步, 一致切线算子矩阵实际上就是弹性应力应变矩阵  $\mathbf{D}_e$ 。下面介绍塑性问

题的一致切线算子矩阵  $\mathbf{D}_{ep}$  的计算方法。

对于本章介绍的 4 种本构模型，一致切线算子矩阵的通用表达式如下：

$$\mathbf{D}_{ep} = \mathbf{D}_e - \frac{d_D d_D^T}{A + d_D^T a}; d_D = \mathbf{D}_e \mathbf{a} \quad (9-62)$$

9.1 节是基于三维模型介绍的，由于本节基于平面模型，因此首先需要对相关公式进行简化调整。最主要的调整就是要去除在平面应变模型条件下等于 0 的应力分量或应变分量。因此，只有 4 个不为零的应力或应变分量：

$$\boldsymbol{\sigma}^T = \{\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy}\}, \varepsilon_z = 0 \quad (9-63)$$

对于平面应变问题，弹性应力应变矩阵  $\mathbf{D}_e$  可表示为

$$\mathbf{D}_e = \begin{bmatrix} \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & \frac{E\nu}{(1+\nu)(1-2\nu)} & \frac{E\nu}{(1+\nu)(1-2\nu)} & 0 \\ \frac{E\nu}{(1+\nu)(1-2\nu)} & \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & \frac{E\nu}{(1+\nu)(1-2\nu)} & 0 \\ \frac{E\nu}{(1+\nu)(1-2\nu)} & \frac{E\nu}{(1+\nu)(1-2\nu)} & \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & 0 \\ 0 & 0 & 0 & \frac{E}{2(1+\nu)} \end{bmatrix} \quad (9-64)$$

流动矢量  $\mathbf{a}$  变成

$$\mathbf{a}^T = \left\{ \frac{\partial \Phi}{\partial \sigma_x} \quad \frac{\partial \Phi}{\partial \sigma_y} \quad \frac{\partial \Phi}{\partial \sigma_z} \quad \frac{\partial \Phi}{\partial \tau_{xy}} \right\} \quad (9-65)$$

进一步有

$$\mathbf{a} = C_1 \mathbf{a}_1 + C_2 \mathbf{a}_2 + C_3 \mathbf{a}_3 \quad (9-66)$$

其中

$$C_1 = \alpha, C_2 = 1, C_3 = 0 \quad (9-67)$$

和

$$\begin{aligned} \mathbf{a}_1^T &= \{1 \quad 1 \quad 1 \quad 0\} \\ \mathbf{a}_2^T &= \frac{1}{2\sqrt{J_2}} \{\sigma_x \quad \sigma_y \quad \sigma_z \quad 2\tau_{xy}\} \end{aligned} \quad (9-68)$$

上式中

$$J_2 = \frac{1}{2}(\sigma_x'^2 + \sigma_y'^2 + \sigma_z'^2) + \tau_{xy}^2 \quad (9-69)$$

为了计算式 (9-62) 的一致切线算子矩阵，我们需要求出  $d_D$ ，将式 (9-64) 中的  $\mathbf{D}_e$  各分量代入  $d_D = \mathbf{D}_e \mathbf{a}$  中，可得

$$d_D = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} = \begin{pmatrix} \frac{E}{1+\nu} a_1 + M \\ \frac{E}{1+\nu} a_2 + M \\ Ga_3 \\ \frac{E}{1+\nu} a_4 + M \end{pmatrix} \quad (9-70)$$

其中,  $a_1$ 、 $a_2$ 、 $a_3$ 、 $a_4$  为流动矢量  $\mathbf{a}$  的分量, 即

$$\mathbf{a}^T = \{a_1 \quad a_2 \quad a_3 \quad a_4\} \quad (9-71)$$

$M$  为中间变量

$$M = \frac{E\nu(a_1 + a_2 + a_4)}{(1+\nu)(1-2\nu)} \quad (9-72)$$

$G$  为剪切模量

$$G = \frac{E}{2(1+\nu)} \quad (9-73)$$

然后将式 (9-70) 代入式 (9-62) 即可得到一致切线算子  $\mathbf{D}_{ep}$ 。

### 9.2.2 应力更新过程

如 8.1.3 小节所述, 弹塑性问题一般是通过增量迭代来解决的。`usermat` 用户子程序的一大任务是应力更新过程, 即根据传入的应变增量计算应力增量。对于弹塑性模型, 要保证更新后的应力状态满足一定要求。例如, 如果在增量过程中发生了屈服, 那么更新后的应力状态点应该落在更新后的屈服面上, 而屈服面的大小和增量过程中发生的塑性应变大小相关, 塑性应变大小又要按照流动法则确定。

常用的应力更新算法是回退算法, 分为两步: 一是弹性预测; 二是对弹性预测的应力进行修正, 确保应力状态点不超过屈服面, 称为塑性修正。

#### 1. 弹性预测步

弹性预测步假定该增量步的应变全为弹性应变, 按广义虎克定律计算得到应力增量预测值 (试应力):

$$d\boldsymbol{\sigma}_e^r = \mathbf{D}_e d\boldsymbol{\varepsilon}^r \quad (9-74)$$

将更新后的应力状态 (试应力对应的应力状态) 代入以下 Drucker-Prager 屈服函数式

$$\alpha I_1 + \sqrt{J_2} = k' \quad (9-75)$$

式中, 参数  $\alpha$  和  $k'$  见式 (9-21) 或式 (9-22), 本书采用式 (9-21), 即 Drucker-Prager 圆与 Mohr-Coulomb 的外顶点相吻合 (见图 9-7)。我们记上式左边部分为  $\bar{\sigma}$ , 即

$$\bar{\sigma} = \alpha I_1 + \sqrt{J_2} \quad (9-76)$$

称为等效应力 (不同于 Mises 等效应力)。

#### 2. 塑性修正步

在介绍塑性修正步之前, 首先说明本节采用的强化准则是线性各向同性强化, 假定强化

与等效塑性应变  $\bar{\varepsilon}_p$  有关, 可采用下式进行:

$$k'_{r-1} = k'_0 + H\bar{\varepsilon}_p^{r-1} \quad (9-77)$$

$$k'_0 = \frac{6c \cos \phi}{\sqrt{3}(3 - \sin \phi)} \quad (9-78)$$

式中,  $H$  是强化参数,  $\bar{\varepsilon}_p^{r-1}$  是第  $r-1$  次迭代产生的等效塑性应变, 在 `usermat` 中以状态变量的形式存储: `statev(2)`。

塑性修正具体步骤如下:

**Step 1** 塑性修正分为两种情况, 新的屈服和后继屈服。可通过下式进行判断:

$$\bar{\sigma}^{r-1} > k'_{r-1} \quad (9-79)$$

$\bar{\sigma}^{r-1}$  是第  $r-1$  次迭代产生的等效应力 (式 9-76), 在 `usermat` 中以状态变量的形式存储 (关于状态变量, 见 8.3 节): `statev(1)`。上述判断有以下两种结果:

$$(1) \bar{\sigma}^{r-1} > k'_{r-1}$$

后继屈服, 说明高斯积分点此前已经屈服。继续检查是否有  $\bar{\sigma}_e^r > \bar{\sigma}_e^{r-1}$ , 其中,  $\bar{\sigma}_e^r$  是弹性试应力对应的等效应力。假如是则表明本迭代步发生了屈服, 因此应力必须退回到屈服面上 (图 9-11)。假如  $\bar{\sigma}_e^r \leq \bar{\sigma}_e^{r-1}$ , 则表示本迭代步为弹性卸载步, 直接跳转至 Step4。

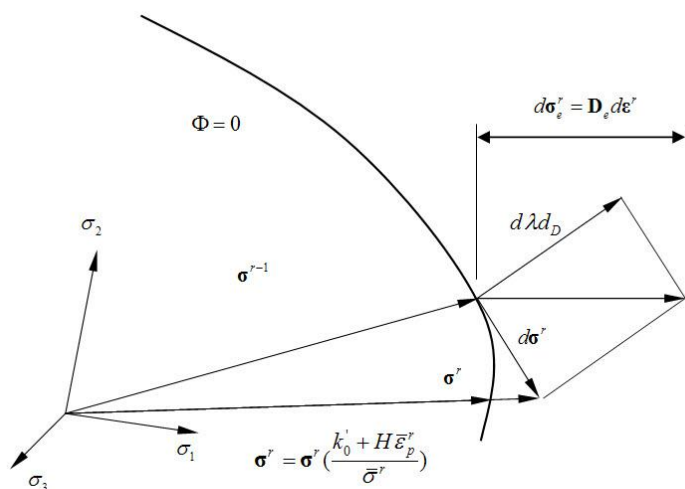


图 9-11 后继屈服塑性修正示意图

$$(2) \bar{\sigma}^{r-1} \leq k'_{r-1}$$

新的屈服, 说明上一迭代步高斯积分点未屈服。继续检查是否有  $\bar{\sigma}_e^r > k'_{r-1}$ , 假如是则表明本迭代步发生了屈服, 因此应力必须退回到屈服面上 (图 9-12)。假如  $\bar{\sigma}_e^r \leq k'_{r-1}$ , 则表示本迭代步仍是弹性的, 跳转至 Step4。

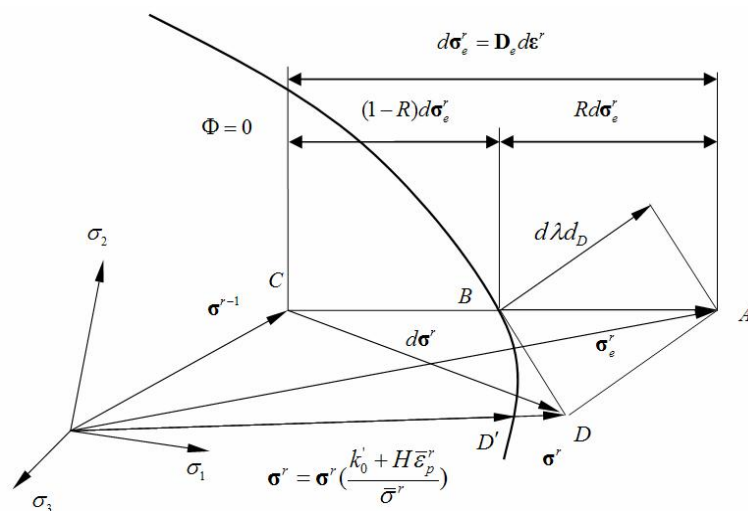


图 9-12 新的屈服塑性修正示意图

**Step 2** 对于已经屈服的高斯点，只要计算总应力中满足屈服准则的部分

$$\sigma^r = \sigma^{r-1} + (1-R)d\sigma_e^r \quad (9-80)$$

如图 9-11 和图 9-12 所示，对于后继屈服而言  $R=1$ ，对于新的屈服而言， $R = \frac{AB}{AC} = \frac{\bar{\sigma}_e^r - k_{r-1}'}{\bar{\sigma}_e^r - \bar{\sigma}^{r-1}}$ 。

**Step 3** 其余部分的应力  $Rd\sigma_e^r$ ，必须采用某种方式有效地消去，如图 9-11 所示，因为允许塑性变形，点  $A$  必须回到屈服面上来。这可形象地描述如下：在  $C$  点加载，应力点在弹性区内移动，直到与屈服面相交于  $B$  点，超过这一点的弹性点，将得到由  $A$  点确定的最终应力状态（弹性预测步）。但是，为了满足屈服准则，应力点就不能移动到屈服面以外。因此，应力点只可能在屈服面上移动，直到同时满足平衡方程和本构关系为止。由式 (9-39)、(9-40) 和 (9-41) 有

$$d\sigma^r = Dd\varepsilon^r - d\lambda d_D \quad (9-81)$$

或

$$\sigma^r = \sigma^{r-1} + d\sigma_e^r - d\lambda d_D \quad (9-82)$$

上式给出了满足弹塑性条件的总应力  $\sigma^r$ 。从图 9-11 中可以看出，假如载荷增量取得不够小，则最终应力点  $D$  可能会偏离屈服面。事实上，只要调节矢量  $\sigma^r$  的大小，就可以使  $D$  退回到屈服面上来。由应力  $\sigma^r$  产生的等效应力记为  $\bar{\sigma}^r$ ，并要指出，当  $D$  点位于屈服面上时，这个值就等于  $k_0' + H\bar{\varepsilon}_p^r$ ，相应的调节式为

$$\sigma^r = \sigma^r \left( \frac{k_0' + H\bar{\varepsilon}_p^r}{\bar{\sigma}^r} \right) \quad (9-83)$$

通过上式对  $\sigma^r$  进行调节，以使  $D$  点回到屈服面上。上式表示  $\sigma^r$  的缩减，说明各个应力分量都将按比例减小。

如果载荷增量较大，且应力点在屈服面的大曲率附近，上述方法就不能精确预测屈服面上的最终点  $D$  点。为了解决这个问题，把超过的应力部分  $Rd\sigma_e^r$  分成相等的  $m$  个部分，则每一

部分大小为

$$\frac{Rd\sigma_e^r}{m} \quad (9-84)$$

$m$  为小于

$$8\left(\frac{\bar{\sigma}_e^r - k'_{r-1}}{k'_0}\right) + 1 \quad (9-85)$$

的、最接近的整数值。然后依次将这  $m$  个部分回退到屈服面上。

**Step 4** 按照 (9-74) 式计算应力增量即可。

### 9.2.3 usermat 代码

下面给出以上所述 Drucker-Prager 本构模型的具体代码 (\chp9\slope\usermat.F):

```
*deck,usermat      USERDISTRIB  parallel      gal
      subroutine usermat(
&                  matId, elemId, kDomIntPt, kLayer, kSectPt,
&                  ldstep, isubst, keycut,
&                  nDirect, nShear, ncomp, nStatev, nProp,
&                  Time, dTime, Temp, dTemp,
&                  stress, statev, dsdePl, sedEl, sedPl, epseq,
&                  Strain, dStrain, epsPl, prop, coords,
&                  var0, defGrad_t, defGrad,
&                  tsstif, epsZZ,
&                  var1, var2, var3, var4, var5,
&                  var6, var7, var8)
c*****

#include "impcor.inc"
c
      INTEGER
&          matId, elemId,
&          kDomIntPt, kLayer, kSectPt,
&          ldstep, isubst, keycut,
&          nDirect, nShear, ncomp, nStatev, nProp
      DOUBLE PRECISION
&          Time,      dTime,      Temp,      dTemp,
&          sedEl,      sedPl,      epseq,      epsZZ
      DOUBLE PRECISION
&          stress (ncomp ), statev (nStatev),
&          dsdePl (ncomp,ncomp),
&          Strain (ncomp ), dStrain (ncomp ),
&          epsPl (ncomp ), prop (nProp ),
&          coords (3),      rotateM (3,3),
&          defGrad (3,3),      defGrad_t(3,3),
&          tsstif (2)
c
c
c
      DOUBLE PRECISION var0, var1, var2, var3, var4, var5,
&          var6, var7, var8
c
      INTEGER
```

```

&          I, J, MSTEP
DOUBLE PRECISION
&          STRES(4), STRAN(4), YOUNG, POSSI, EPSTN, EFFST,
&          COHES, HARDS, PREYS, DESIG(4), SIGMA(4), UNIAX,
&          FRICT, ESPRE, ESCUR, RFACT, REDUC, SGTOT(4),
&          AGASH, DLAMD, BGASH, CURYs, dStress(4), DILAT,
&          BRING, ROOT3, EMATX(4,4), DEVIA(4), SINT3, STEFF,
&          THETA, VARJ2, YIELD, AVECT(4), ABETA, DVECT(4)
YOUNG=prop(1)          !杨氏模量 E
POSSI=prop(2)          !泊松比  $\nu$ 
FRICT=prop(3)          !内摩擦角  $\phi$ 
COHES=prop(4)          !粘聚力 c
HARDS=prop(5)          !强化参数 H
EFFST=statev(1)        !状态变量 1: 等效应力, 式 (9-75)
EPSTN=statev(2)        !状态变量 2: 等效塑性应变
ROOT3=SQRT(3.0d0)      ! $\sqrt{3}$ 
UNIAX=6.0D0*COHES*COS(FRICT*0.017453292)/          !计算式 (9-77)
&          (ROOT3*(3.0D0-SIN(FRICT*0.017453292)))
c          计算弹性应力应变矩阵
CALL DMATX(prop,nProp,EMATX)
c          将上一迭代步结束时的应力 stress 传入 STRES
DO I=1,4
    STRES(I)=stress(I)
END DO
c          初始化试应力增量
DO I=1,4
    DESIG(I)=0.0D0
END DO
DO I=1,4
    DO J=1,4
        STRES(I)=STRES(I)+EMATX(I,J)*dStrain(J)    !计算试应力
        DESIG(I)=DESIG(I)+EMATX(I,J)*dStrain(J)    !试应力增量
    END DO
END DO
DO I=1,4
    dStress(I)=DESIG(I)
END DO
PREYS=UNIAX+EPSTN*HARDS          !式 (9-76)
c          计算不变量
CALL INVAR(prop,nProp,STRES,DEVIA,SINT3,STEFF,THETA,VARJ2,YIELD)
ESPRE=EFFST-PREYS              !用于判断是否是新的屈服
IF(ESPRE.GE.0.0D0) GO TO 50      !如果 ESPRE>0, 则是后继屈服, 跳转至语句 50
ESCUR=YIELD-PREYS              !用于判断当前迭代步是否发生屈服
IF(ESCUR.LE.0.0D0) GO TO 60     !如果 ESCUR<0, 则未发生屈服, 跳转至语句 60
c          发生了屈服
write(*,"(A30)") "Plastic"      !在 Output 窗口中显示
RFACT=ESCUR/(YIELD-EFFST)      !R (新的屈服)
GO TO 70
50    ESCUR=YIELD-EFFST          !用于判断当前迭代步是否发生屈服
    IF(ESCUR.LE.0.0D0) GO TO 60  !如果 ESCUR<0, 则未发生屈服, 跳转至语句 60
    RFACT=1.0D0                 !R (后继屈服)
70    MSTEP=ESCUR*8.0D0/UNIAX+1.0D0 !获得划分步数 m, 式 (9-85)
    MSTEP=INT(MSTEP)            !最接近的整数值
    REDUC=1.0D0-RFACT           !1-R
    DO I=1,4

```

```

        SGTOT(I)=stress(I)+REDUC*DESIG(I)      !式 (9-80)
        DESIG(I)=RFACT*DESIG(I)/MSTEP        !式 (9-84)
    END DO
c    对 m 应力缩减进行循环, 共 m 步
    DO I=1,MSTEP
        !计算不变量和屈服函数
        CALL INVAR(prop,nProp,SGTOT,DEVIA,SINT3,STEFF,THETA,VARJ2,
&            YIELD)
        !计算流动向量 a
        CALL YIELDF(DEVIA,prop,nProp,SINT3,STEFF,THETA,VARJ2,AVECT)
        !计算计算 Da
        CALL FLOWPL(prop,nProp,AVECT,ABETA,DVECT)
        AGASH=0.0D0
        DO J=1,4
            AGASH=AGASH+AVECT(J)*DESIG(J)      !见式 (9-39) 的 AGASH
        END DO
        DLAMD=AGASH*ABETA                      !塑性乘子增量  $d\lambda$ , 式 (9-39)
        IF(DLAMD.LT.0.0D0) DLAMD=0.0D0
        BGASH=0.0D0
        DO J=1,4
            BGASH=BGASH+AVECT(J)*SGTOT(J)
            SGTOT(J)=SGTOT(J)+DESIG(J)-DLAMD*DVECT(J)
        END DO
        EPSTN=EPSTN+DLAMD*BGASH/YIELD          !等效塑性应变
    END DO
    CALL INVAR(prop,nProp,SGTOT,DEVIA,SINT3,STEFF,THETA,VARJ2,YIELD)
    CURYS=UNIAX+EPSTN*HARDS
    BRING=1.0D0
    IF(YIELD.GT.CURYS) BRING=CURYS/YIELD
    DO I=1,4
        stress(I)=BRING*SGTOT(I)
    END DO
    EFFST=BRING*YIELD
    statev(1)=EFFST
    statev(2)=EPSTN
    CALL YIELDF(DEVIA,prop,nProp,SINT3,STEFF,THETA,VARJ2,AVECT)
    CALL FLOWPL(prop,nProp,AVECT,ABETA,DVECT)
    DO I=1,4
        DO J=1,4
            dsdePl(I,J)=EMATX(I,J)-ABETA*DVECT(I)*DVECT(J)
        END DO
    END DO
60    DO I=1,4
        stress(I)=stress(I)+dStress(I)
    END DO
    statev(1)=EFFST
    statev(2)=EPSTN
    DO I=1,4
        DO J=1,4
            dsdePl(I,J)=EMATX(I,J)
        END DO
    END DO
    RETURN
    END
c

```

```

c
      SUBROUTINE INVAR(prop,nProp,stress,DEVIA,SINT3,STEFF,THETA,VARJ2
        &
        ,YIELD)
#include "impcom.inc"
c*****
c
c****  本子程序用于计算应力不变量和当前屈服函数的值
c
c*****
      INTEGER          nProp
      DOUBLE PRECISION
      &
      prop(nProp), stress(4), SINT3, STEFF, THETA,
      &
      VARJ2, YIELD, ROOT3, SMEAN, DEVIA(4), VARJ3,
      &
      PHIRA, SNPHI
      ROOT3=SQRT(3.0D0)
      SMEAN=(stress(1)+stress(2)+stress(3))/3.0D0
      DEVIA(1)=stress(1)-SMEAN
      DEVIA(2)=stress(2)-SMEAN
      DEVIA(3)=stress(3)-SMEAN
      DEVIA(4)=stress(4)
      VARJ2=DEVIA(4)*DEVIA(4)+0.5D0*(DEVIA(1)*DEVIA(1)+DEVIA(2)*
      &
      DEVIA(2)+DEVIA(3)*DEVIA(3))
      VARJ3=DEVIA(3)*(DEVIA(3)*DEVIA(3)-VARJ2)
      STEFF=SQRT(VARJ2)
      IF(STEFF.EQ.0.0D0) THEN
        SINT3=0.0D0
      ELSE
        SINT3=-3.0D0*ROOT3*VARJ3/(2.0D0*VARJ2*STEFF)
      END IF
      IF(SINT3.LT.-1.0)SINT3=-1.0D0
      IF(SINT3.GT.1.0) SINT3=1.0D0
      THETA=ASIN(SINT3)/3.0D0
      PHIRA=prop(3)*3.14159265358979D0/180.0D0
      SNPHI=SIN(PHIRA)
      YIELD=6.0D0*SMEAN*SNPHI/(ROOT3*(3.0D0-SNPHI))+STEFF
      RETURN
      END

c
      SUBROUTINE YELDF(DEVIA,prop,nProp,SINT3,STEFF,THETA,VARJ2,AVECT)
#include "impcom.inc"
c*****
c
c****  本子程序计算流动向量 a
c
c*****
      INTEGER          nProp, I
      DOUBLE PRECISION
      &
      DEVIA(4), prop(nProp), SINT3, STEFF, THETA,
      &
      VARJ2, VECA1(4), VECA2(4), VECA3(4),ROOT3,
      &
      FRICT, TANTH, TANT3, SINTH, COSTH, COST3,
      &
      SNPHI, CONS1, CONS2, CONS3,AVECT(4)
      FRICT=prop(3)
      TANTH=TAN(THETA)
      TANT3=TAN(3.0D0*THETA)
      SINTH=SIN(THETA)

```

```

COSTH=COS(THETA)
COST3=COS(3.0D0*THETA)
ROOT3=SQRT(3.0D0)

c
c**** 计算 a1
c
  VECA1(1)=1.0D0
  VECA1(2)=1.0D0
  VECA1(3)=1.0D0
  VECA1(4)=0.0D0

c
c**** 计算 a2
c
  DO I=1,3
    VECA2(I)=DEVIA(I)/(2.0D0*STEFF)
  END DO
  VECA2(4)=DEVIA(4)/STEFF

c
c**** 计算 a3
c
  VECA3(1)=DEVIA(2)*DEVIA(3)+VARJ2/3.0D0
  VECA3(2)=DEVIA(1)*DEVIA(3)+VARJ2/3.0D0
  VECA3(3)=DEVIA(1)*DEVIA(2)-DEVIA(4)*DEVIA(4)+VARJ2/3.0D0
  VECA3(4)=-2.0*DEVIA(3)*DEVIA(4)
  SNPHI=SIN(FRICT*3.14159265358979D0/180.0D0)
  CONS1=2.0D0*SNPHI/(ROOT3*(3.0D0-SNPHI))
  CONS2=1.0D0
  CONS3=0.0D0
  DO I=1,4
    AVECT(I)=CONS1*VECA1(I)+CONS2*VECA2(I)+CONS3*VECA3(I) !式 (9-66)
  END DO
  RETURN
END

c
  SUBROUTINE FLOWPL(prop,nProp,AVECT,ABETA,DVECT)
#include "impcom.inc"
c*****
c
c**** 本子程序用于计算 Da
c
c*****
  INTEGER          nProp, I
  DOUBLE PRECISION
  &                prop(nProp), AVECT(4), ABETA, DVECT(4),
  &                YOUNG, POISS, HARDS, FMUL1, FMUL2,
  &                DENOM
  YOUNG=prop(1)
  POISS=prop(2)
  HARDS=prop(6)
  FMUL1=YOUNG/(1.0D0+POISS)
  FMUL2=YOUNG*POISS*(AVECT(1)+AVECT(2)+AVECT(4))/((1.0+POISS)*
  & (1.0D0-2.0D0*POISS))
  DVECT(1)=FMUL1*AVECT(1)+FMUL2
  DVECT(2)=FMUL1*AVECT(2)+FMUL2
  DVECT(3)=FMUL1*AVECT(3)+FMUL2

```

```

DVECT(4)=AVECT(4)*YOUNG/(2.0D0*(1.0D0+POISS))
DENOM=HARDS
DO I=1,4
    DENOM=DENOM+AVECT(I)*DVECT(I)
END DO
ABETA=1.0D0/DENOM      !见式 (9-39) 的 ABETA
RETURN
END

c
SUBROUTINE DMATX(prop,nProp,EMATX)
#include "impcom.inc"
c*****
c
c****  本子程序用于计算弹性应力应变矩阵
c
c*****
      INTEGER          nProp,K1,K2
      DOUBLE PRECISION
&          prop(nProp), EMATX(4,4),
&          YOUNG, POISS, CONST1, CONST2, CONST3

      YOUNG=prop(1)
      POISS=prop(2)
      CONST1=YOUNG*(1.0D0-POISS)/((1.0D0+POISS)*
&      (1.0D0-2.0D0*POISS))
      CONST2=YOUNG*POISS/((1.0D0+POISS)*(1.0D0-2.0D0*POISS))
      CONST3=YOUNG/(2.0D0*(1.0D0+POISS))
      EMATX(1,1)=CONST1
      EMATX(1,2)=CONST2
      EMATX(1,3)=CONST2
      EMATX(1,4)=0.0D0
      EMATX(2,1)=CONST2
      EMATX(2,2)=CONST1
      EMATX(2,3)=CONST2
      EMATX(2,4)=0.0D0
      EMATX(3,1)=CONST2
      EMATX(3,2)=CONST2
      EMATX(3,3)=CONST1
      EMATX(3,4)=0.0D0
      EMATX(4,1)=0.0D0
      EMATX(4,2)=0.0D0
      EMATX(4,3)=0.0D0
      EMATX(4,4)=CONST3
      RETURN
END

```

#### 9.2.4 边坡稳定性分析

编译连接以上 usermat 用户子程序后,即可利用用户自定义的 Drucker-Prager 本构模型进行分析。本小节介绍一个边坡稳定性分析实例。几何模型如图 9-13 所示:

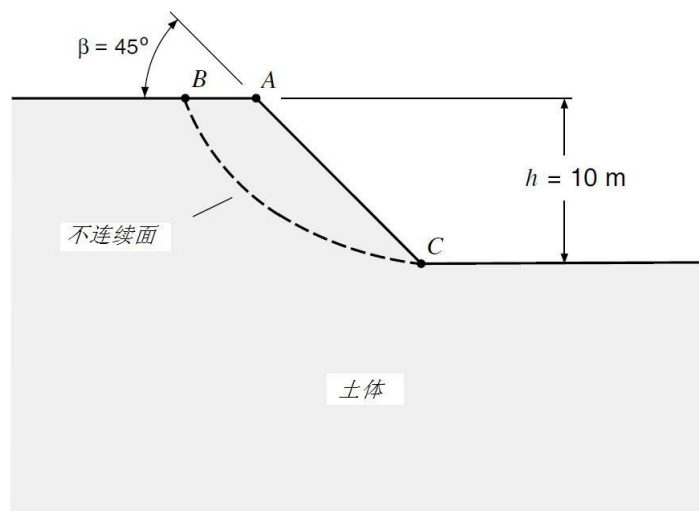


图 9-13 边坡稳定性分析几何模型

下面给出边坡稳定性分析 APDL 命令流 (\chp9\slope\slope.inp)。首先提醒读者注意，usermat 用户子程序仅用于最新技术单元（见表 3-2），不支持传统单元，这里使用二维的 PLANE183 单元。用户想要使用理想塑性材料时，只需将强化参数 H 置为 0 即可。

```

finish
/clear
!
!进入前处理器
/prep7
!参数设置
nLN      = 15                !网格划分密度
E        = 20000E3           !弹模 E
v        = 0.3               !泊松比 v
PHI       = 30                !内摩擦角 phi
c         = 30E3              !粘聚力 c
H         = 200000            !强化参数 H
density= 6000                !密度
accel    = 9.8                !重力加速度
!单元
et, 1, 183                    !183 号单元
keyopt, 1, 3, 2               !平面应变
!用户自定义材料模型
tb, user, 1, 1, 6              !6 个材料常数
tbddata,1,E,v,PHI,c,H
mp, dens,1, density
tb, state,1,,2                !2 个状态变量
!创建关键点
k, 1, 0, 0
k, 2, 0, 40
k, 3, 35, 40
k, 4, 45, 30
k, 5, 75, 30
k, 6, 75, 0
!生成面
a, 1, 2, 3, 4, 5, 6

```

```

/pnum , line, 1
lplot
!网格划分大小
lesize,1 , , ,nLN,0.5 , , ,1
lesize,2 , , ,2*nLN, 0.5 , , ,1
lesize,3 , , , nLN, , , ,1
lesize,4 , , ,2*nLN, 1.5 , , ,1
lesize,5 , , ,nLN,1.5 , , ,1
lesize,6 , , ,nLN, , , ,1
amesh,all          !网格划分

```

网格划分完成后的模型如图 9-14 所示:

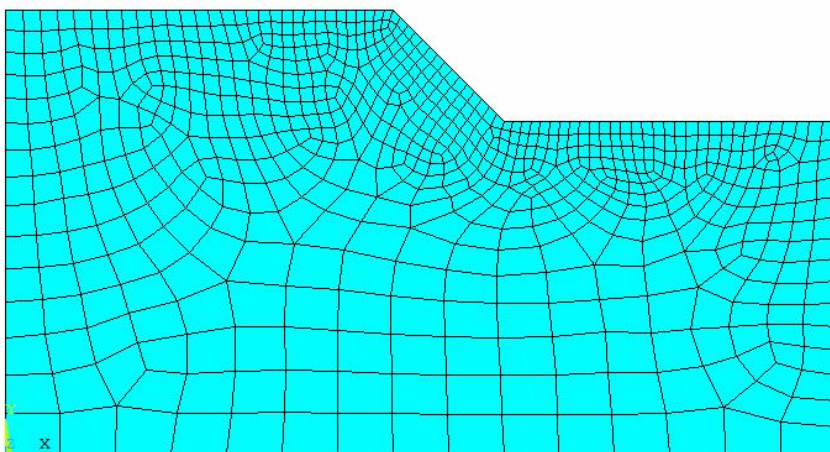


图 9-14 边坡稳定性分析有限元模型

继续设置边界条件, 并进行求解设置:

```

acel, 0, accel          !重力加速度
dl, 1, , ux,
dl, 5, , ux,
dl, 6, , all,
allsel,all
/replot
finish
!
!进入求解器
/solu
!求解设置
time,1
autots,on              !打开自动载荷步
pred,off              !关闭线性预测
nsubst,50,1000,20     !载荷步设置
outres,all,all
outres,svar,all        !保存所有状态变量到数据库
solve
!
!进入后处理器
/post1
set,last
/efacet,1

```

plnsol,svar,2 !显示状态变量 2,即等效塑性应变分布云图

以上命令流中，我们在 post1 后处理器中输出等效塑性应变分布云图，如图 9-15 所示：

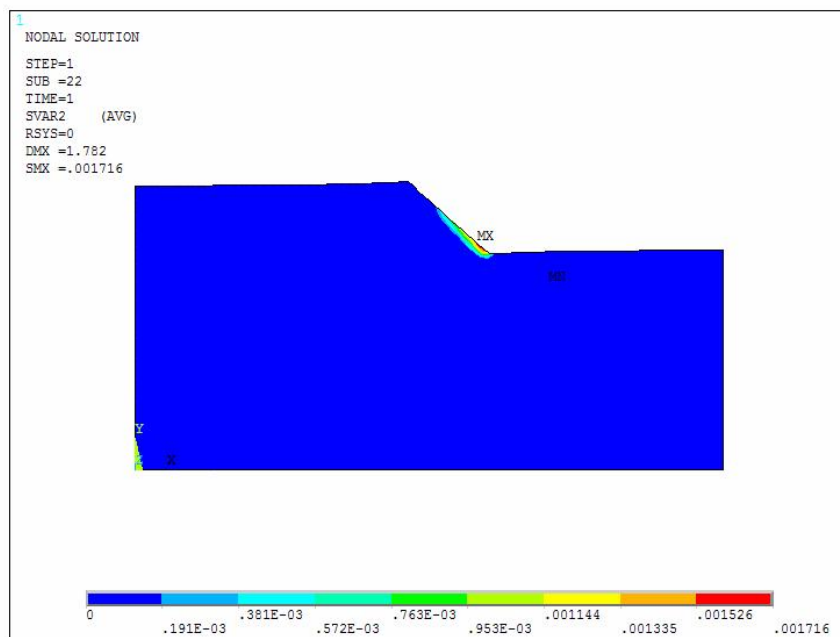


图 9-15 等效塑性应变分布云图

当材料密度为  $6000\text{kg/m}^3$ ，重力加速度为  $9.8\text{m/s}^2$  时，在图 9-14 所示的不连续面内即有塑性区形成，但等效塑性应变值很小，最大值仅为 0.001716。