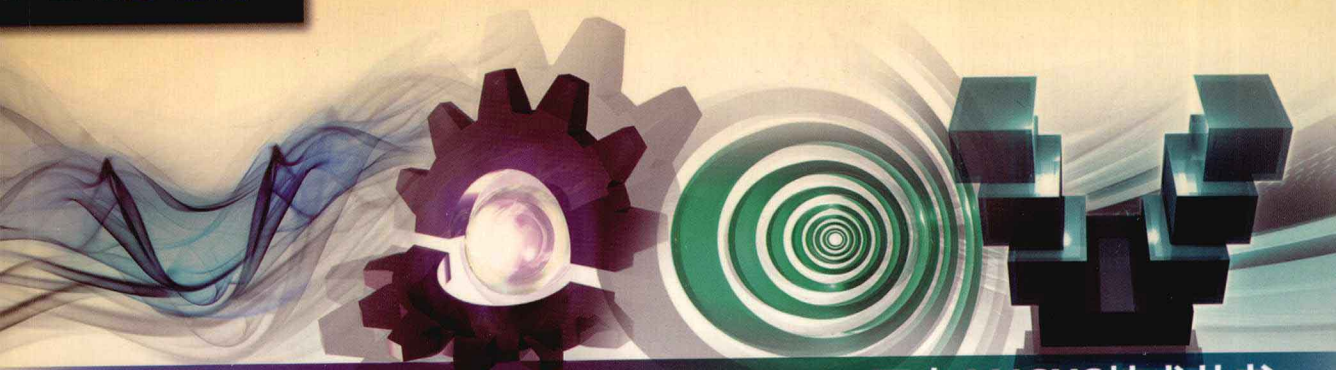


ANSYS原厂策划 万水精心出品

ANSYS高级仿真技术系列

ANSYS®



万水ANSYS技术丛书

ANSYS APDL 参数化有限元分析技术 及其应用实例

张涛 等编著



中国水利水电出版社
www.waterpub.com.cn

万水 ANSYS 技术丛书

ANSYS APDL 参数化有限元分析技术 及其应用实例

张 涛 等编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

APDL (参数化设计语言) 是 ANSYS 的高级分析技术之一, 也是 ANSYS 高级应用的基础, 它提供一种逐行解释性的编程语言工具, 可以很好地用于实现参数化的有限元分析、分析批处理、专用分析系统的二次开发和设计优化等, 是 ANSYS 不可缺少的重要技术。随着 ANSYS 技术的发展, APDL 技术将在多个方面起着更加重要的作用: 随着 ANSYS Workbench 应用环境的广泛使用, APDL 在 Workbench 环境下如何灵活运用成为 ANSYS 结构分析中高级用户的进一步需求; APDL 参数化大型项目尤其是桁架结构模型; APDL Math 新技术为中高级用户灵活调用 ANSYS 求解器提供了可能。

本书主要分两大部分介绍和学习参数化设计语言 APDL, 1~15 章主要介绍 APDL 语言的基本要素, 16~19 章重点介绍 APDL 的典型应用技术。

本书主要适合于已掌握 ANSYS 经典界面基本操作和 Workbench 工作环境的初级用户和部分中高级用户, 是一本学习 APDL 的技术资料, 是灵活掌握 ANSYS 专题分析技术的辅助资料, 也是 Workbench 用户使用 APDL 语言的一本工具书。通过对本书的学习, 读者会进一步提高有限元分析的分析手段和综合应用能力, 进一步提高 ANSYS 软件的使用深度。

图书在版编目 (C I P) 数据

ANSYS APDL参数化有限元分析技术及其应用实例 /
张涛等编著. — 北京: 中国水利水电出版社, 2013.3
(万水ANSYS技术丛书)
ISBN 978-7-5170-0664-0

I. ①A… II. ①张… III. ①有限元分析—应用软件
IV. ①0241.82-39

中国版本图书馆CIP数据核字 (2013) 第032767号

策划编辑: 杨元泓 责任编辑: 张玉玲 封面设计: 李 佳

书 名	万水 ANSYS 技术丛书 ANSYS APDL 参数化有限元分析技术及其应用实例
作 者	张 涛 等编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心 (零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市铭浩彩色印装有限公司
规 格	184mm×260mm 16 开本 17 印张 426 千字
版 次	2013 年 3 月第 1 版 2013 年 3 月第 1 次印刷
印 数	0001—5000 册
定 价	42.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社发行部负责调换
版权所有·侵权必究

前 言

历史上，人物和事件的组合催生新的技术进步并影响历史的发展这类现象屡见不鲜。20世纪中期的数值仿真领域就处在这一时期。那时，数值仿真已经开始走向了工程设计与研发的前端。从那时起，采用计算机辅助工程（CAE）技术来解决工程问题日趋重要。

对于一个复杂产品的设计，环境、结构、热、流场、电磁等多种因素共同影响了其性能。而早期 CAE 技术只能对其某一方向进行仿真分析建模，例如有限元模型、计算流体力学模型、计算电磁学模型等，这是在计算成本与收益之间做一个权衡的结果。然而，在过去的十年中，随着计算机性能的提升，CAE 技术发生了革命性的变化。具体地讲，各个学科都有了成熟的产品来解决相关领域的问题，例如计算固体力学方面有 ANSYS Mechanical、MSC Nastran、Abaqus 等；计算流体力学方面有 CFX、Fluent 等；计算电磁学有 Ansoft 等。

随着自主研发能力的增强，国内外企业、科研院所对设计分析人员的要求已经从具备单一学科设计分析能力转变到具备多学科综合设计分析能力。这对设计分析人员提出了更高的要求，需要学习更多的学科知识、更多的软件工具。CAE 企业 ANSYS 公司意识到市场的需求，从 2002 年起逐渐兼并了 Fluent、CFX、Icepak 和 Ansoft 等仿真工具，致力于多物理场仿真分析工具的开拓。ANSYS Workbench 即是 ANSYS 公司在 2002 年为了整合自身产品并最终实现多物理场耦合而提出的框架体系，目前在国内外客户中已经广泛使用。它的典型特点是：

- 多物理场耦合

多物理场耦合为 ANSYS 产品的最大特色，充分体现了 CAE 领域的发展趋势。它具备结构、热、流体、电磁单场求解器和多场耦合求解器。在 ANSYS Workbench 框架下，用户可以方便地实现流—固、流—固—热、电—热等耦合场分析。Workbench 解决了不同软件之间仿真载荷及数据传递的问题。

- 统一前后处理

Workbench 具有强大的 CAD 软件接口、易用的网格划分工具和后处理功能，工程设计人员利用 Workbench 可以只学习一套模型处理、网格划分工具，输出不同的求解器网络格式，进行相应的仿真分析。

- 多学科参数优化

通过 Workbench 体系能够将项目中的几何、材料、载荷和计算结果等进行参数化，然后利用 ANSYS Design Explorer 模块进行试验设计（DOE）、目标驱动优化设计（Goal-Driven Optimization）、最小/最大搜索（Min/Max Search）、六西格玛分析（Six Sigma Analysis）等多学科参数优化设计。

APDL（参数化设计语言）作为 ANSYS Mechanical 的高级分析技术之一，在这一发展过程中也起到了重要的作用，是 ANSYS 中高级用户不可缺少的重要技术。具体来说，APDL 技术将在以下几个方面起着重要的作用：

- 随着 ANSYS Workbench 应用环境的广泛使用，而 Workbench 并不能直接实现 Mechanical 求解器的所有建模、高级求解和后处理功能，因此 APDL 在 Workbench

环境下如何灵活运用成为 ANSYS 结构分析中高级用户的进一步需求。

- 大量输入参数、不同数据源的大型项目，尤其是大量使用梁、管、质点单元模型，仍然适合于 APDL 技术进行参数化建模和项目管理。
- 非 ANSYS Workbench 网格划分工具作为 Mechanical 的前处理工具时，需要使用 APDL 对模型进行载荷工况管理、求解器设置和后处理。
- 研究人员、高级有限元分析人员认为修改结构矩阵而将 Mechanical 作为一个求解器来使用的情况下，即运用 APDL Math 新技术时，需要使用 APDL 技术。

综上，本书将献给以上四方面的 ANSYS 用户。本书主要分两大部分介绍和学习参数化设计语言 APDL，1~15 章主要介绍 APDL 语言的基本要素，16~19 章重点介绍 APDL 的典型应用技术。其中，APDL 的基本要素包括支持 APDL 的菜单操作、变量、数组与表参数及其用法、数据文件的读写、数据库信息的访问、数学表达式、使用函数编辑器和加载器、矢量与矩阵运算、APDL Math、内部函数、流程控制、宏与宏库，以及定制用户图形界面。这些技术要素是 APDL 编程语言的组成部分，他们可以很好地将 ANSYS 的命令按照一定顺序组织起来，并利用参数实现数据的交换和传递，实现有限元分析过程的参数化和批处理。特别地，APDL Math 是 13.0 版本以来 Mechanical APDL 模块中的重要新功能发布，也是 ANSYS 走向开放的重要一步。APDL Math 扩展了 APDL 脚本环境，用于调用 ANSYS 软件强大的矩阵运算功能和求解器。APDL 的应用除包括参数化的建模、加载、求解、后处理等基本技术外，还包括专用分析系统的开发、界面系统开发、基于 APDL 的优化设计技术，以及 Workbench 中 APDL 的使用。其中 Workbench 中 APDL 的使用对于 Workbench 用户提高分析深度及水平，提升分析效率有着重要的作用。本书对这些技术要素逐一进行介绍，并提供大量典型实例，帮助读者真正掌握和理解这些技术并能举一反三。

由于时间仓促，加之本书内容新、书中涉及面广及作者水平有限，书中不足甚至错误之处在所难免，恳请广大读者批评指正。

作者
2013 年 2 月

目 录

前言

第 1 章 APDL 参数化语言概论	1
第 2 章 参数与参数菜单系统	2
2.1 参数概念与类型	2
2.2 参数的命名规则	2
2.3 参数化操作环境介绍	3
第 3 章 变量参数及其用法	6
3.1 变量的定义与赋值	6
3.1.1 利用*SET 命令进行变量定义与赋值	6
3.1.2 利用赋值号“=”进行变量定义与赋值	7
3.1.3 利用变量定义菜单或命令输入窗口进行变量定义与赋值	7
3.1.4 在启动时利用驱动命令进行变量定义与赋值	8
3.2 删除变量	8
3.3 数值型变量值的替换	9
3.4 字符参数的用法	9
3.4.1 字符参数的常见用法	9
3.4.2 强制字符参数执行替换	10
3.4.3 抑制发生字符参数替换	11
3.4.4 使用字符参数的限制	11
3.5 数字或字符参数的动态替换	11
3.6 列表显示变量参数	12
3.7 存储与恢复变量	13
第 4 章 数组参数及其用法	14
4.1 数组参数类型与概念	14
4.2 定义数组参数	15
4.3 赋值数组参数	17
4.3.1 利用*SET 命令或“=”给单个或多个数组元素赋值	17
4.3.2 利用*VEDIT 命令或按其等价菜单方式编辑数组	18
4.3.3 利用*VFILL 命令或者其等价菜单方式填充数组向量	18
4.4 列表显示数组参数	20
4.5 曲线图形显示数组参数列矢量	20
4.6 删除数组参数	23
4.7 存储与恢复数组参数	23
第 5 章 表参数及其用法	24
5.1 表参数的概念、定义、删除与赋值	24
5.2 曲线图形显示表参数的列矢量	27
5.3 表插值及表载荷应用实例	27
第 6 章 参数与数据文件的写出与读入	37
6.1 使用*VWRITE 写出数据文件	37
6.2 使用*VREAD 命令读取数据文件填充数组	40
6.3 使用*TREAD 命令读取数据文件并填充 TABLE 类型数组	41
第 7 章 访问 ANSYS 数据库数据	45
7.1 提取数据库数据并赋值给变量	45
7.1.1 *GET 提取命令	45
7.1.2 与*GET 等价的内嵌提取函数	47
7.1.3 对象信息查询函数	51
7.1.4 系统信息查询函数/INQUIRE	54
7.1.5 获取_STATUS 和_RETURN 参数值	55
7.2 批量提取数据库数据并赋值给数组	58
第 8 章 数学表达式	62
第 9 章 使用函数编辑器与加载器	64
9.1 使用函数编辑器	64
9.2 使用函数加载器	70
9.3 使用函数边界条件加载及其应用实例	71
9.3.1 使用函数边界条件加载	71
9.3.2 使用函数边界条件加载应用实例	71
第 10 章 矢量与矩阵运算	80
10.1 矢量与矩阵运算设置	80
10.2 矢量运算	83

10.2.1	矢量间运算 (*VOPER 命令)	83	第 14 章	宏文件与宏库	114
10.2.2	矢量函数 (*VFUN 命令)	86	14.1	APDL 宏及其功能	114
10.2.3	矢量-变量运算 (*VSCFUN 命令)	87	14.2	宏文件命名规则	115
10.2.4	矢量插值运算 (*VITRP 命令)	88	14.3	宏搜索路径	116
10.3	矩阵运算	89	14.4	创建宏文件的方法	117
10.3.1	矩阵间运算 (*MOPER 命令)	89	14.4.1	使用 *CREATE 创建宏文件	117
10.3.2	拷贝或转置数组矩阵 (*MFUN 命令)	91	14.4.2	使用 *CFWRITE 创建宏文件	120
10.3.3	计算傅里叶级数 (*MFOURI 命令)	92	14.4.3	使用 /TEE 创建宏文件	121
第 11 章	APDL Math	94	14.4.4	使用菜单 Utility Menu>Macro> Create Macro 创建宏文件	122
11.1	APDL Math 使用过程	94	14.4.5	用文本编辑器创建宏文件	122
11.2	矩阵和向量大小	95	14.5	宏的局部变量	123
11.3	提取复标量值	95	14.5.1	宏命令行的输入变量	123
11.4	自由度排序	96	14.5.2	宏内部使用的局部变量	125
11.5	创建用户自定义超单元	97	14.6	运行宏	126
11.6	矩阵运算使用建议	98	14.7	宏嵌套: 在宏内调用其他宏	127
11.7	APDL Math 实例	98	14.8	使用宏库文件与运行宏库中的宏	128
11.7.1	实例 1: 模态分析之后验证模态 振型的正交性	98	14.9	在宏中使用组和组件	130
11.7.2	实例 2: 由 .full 文件读取矩阵和 载荷向量并求解	99	14.10	加密宏文件	131
11.7.3	实例 3: 完全法谐响应扫频分析	99	14.10.1	准备加密宏	131
11.7.4	实例 4: 由 .full 文件进行非对称 模态分析	100	14.10.2	生成加密宏	132
11.7.5	实例 5: 由 .hbmatt 文件进行阻尼 模态分析	100	14.10.3	运行加密宏	132
11.7.6	实例 6: 由 .sub 文件导入、修改 并生成新的 .sub 文件	101	第 15 章	定制用户化图形交互界面	133
第 12 章	内部函数	102	15.1	单参数输入对话框	133
第 13 章	流程控制	104	15.2	多参数输入对话框	134
13.1	*GO 无条件分支	104	15.3	调用 ANSYS 程序已有的对话框	136
13.2	*IF-*IFELSE-*ELSE-*ENDIF 条件 分支	105	15.4	宏中实现拾取操作	137
13.3	*DO-*ENDDO 循环	108	15.5	程序运行进度对话框	138
13.4	*DOWHILE 循环	110	15.6	宏运行的消息机制	139
13.5	*REPEAT 重复一个命令	110	15.7	定制工具条与缩写	141
13.6	流程控制命令快速参考	112	15.7.1	定制用户化工具条按钮	142
			15.7.2	存储与恢复工具条按钮	143
			15.7.3	嵌套工具条缩写	145
			第 16 章	基于 APDL 的常规应用及其实例	146
			16.1	ANSYS 程序的启动参数与启动文件	146
			16.2	驱动可执行文件	147
			16.3	利用工具条按钮调用宏	148
			16.4	读入和写出数据文件并实现多载荷步 瞬态动力学求解实例	149

16.5	参数化建模：创建标准零件/模型的通用宏	152	19.2.1	Mechanical 调用 Mechanical APDL 求解器原理	219
16.6	参数化建模：连续变厚度板壳模型	158	19.2.2	Mechanical 驱动 Mechanical APDL 方式	220
16.7	施加随坐标变化的压力载荷	160	19.2.3	Mechanical APDL 文件系统	221
16.8	施加表载荷进行载荷插值求解	163	19.2.4	使用 Commands 准备工作	221
第 17 章	基于 APDL 的专用分析程序二次开发实例	167	19.3	Mechanical 使用 Commands 对象	222
第 18 章	基于 APDL 的有限元优化技术及其应用	180	19.3.1	Commands 重要工具	222
18.1	基于 APDL 的优化设计概念	180	19.3.2	Geometry 分支	226
18.2	基于 APDL 的设计优化过程	181	19.3.3	Remote Points	228
18.2.1	创建分析文件	182	19.3.4	Connection 分支	229
18.2.2	执行优化过程	185	19.3.5	Analysis 分支	232
18.2.3	查看设计序列结果	195	19.3.6	Solution 分支	234
18.2.4	验证最优或者选择的可行性优化设计序列	197	19.3.7	Commands 输入输出参数	236
18.3	基于 APDL 的常见设计优化实例	199	19.4	Workbench 中消声器声场分析	238
18.3.1	数学问题的极小值	199	19.4.1	引言	238
18.3.2	桁架轻型化优化设计	204	19.4.2	问题定义	239
第 19 章	Workbench 中 APDL 的使用	218	19.4.3	操作步骤	239
19.1	Workbench 中使用 Mechanical APDL 的场景	218	19.4.4	关键 Commands 说明	246
19.2	Commands 使用预备知识	219	19.4.5	进一步讨论	251
			附录 A	APDL 命令	252
			附录 B	优化设计命令	255
			附录 C	APDL 通道命令	257

APDL 参数化语言概论

APDL 是 ANSYS Parametric Design Language 的缩写，即 ANSYS 参数化设计语言，它是一种类似 FORTRAN 的解释性语言，提供一般程序语言的功能，如参数、宏、标量、向量及矩阵运算、分支、循环、重复以及访问 ANSYS 有限元数据库等，另外还提供简单界面定制功能，实现参数交互输入、消息机制、界面驱动和运行应用程序等。

利用 APDL 的程序语言与宏技术组织管理 ANSYS 的有限元分析命令，就可以实现参数化建模、施加参数化载荷与求解以及参数化后处理结果的显示，从而实现参数化有限元分析的全过程，同时这也是 ANSYS 批处理分析的最高技术。在参数化的分析过程中可以简单地修改其中的参数达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。同时，以 APDL 为基础用户可以开发专用有限元分析程序，或者编写经常重复使用的功能小程序，如特殊载荷施加宏、按规范进行强度或刚度校核宏等。

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，达到最优化设计目标。

ANSYS 12.0 之后的版本中，ANSYS 结构有限元求解器有两套前后处理体系：一个是传统 ANSYS 界面，称为 Mechanical APDL；一个是 Workbench 界面，称为 ANSYS Mechanical。二者使用同一求解器，但是前后处理技术不同。与传统的 Mechanical APDL 界面相比，Workbench 界面的特点是：与三维 CAD 软件（如 UG NX、PROE 等）的几何接口解读能力更强，能够将 CAD 软件中的参数直接读取到 Workbench 中；专门针对于仿真分析几何模型处理的功能，包括细节特征清理、薄壁结构抽中面等；Mechanical 界面载荷施加使用工程化的语言对几何对象进行载荷施加，降低了软件的使用门槛；后处理图形显示效果更加优越。由于这些优势，越来越多的工程师开始使用 ANSYS Workbench 界面。但是，有一些情况下，Workbench 中求解设置、载荷施加和后处理仍然有部分功能不能直接实现，而需要借助 APDL 命令的使用。因此，APDL 知识仍然是中高级分析人员的必要技能。

总之，APDL 扩展了传统有限元分析范围之外的能力，提供了建立标准化零件库、序列化分析、设计修改、设计优化以及更高级的数据分析处理能力，包括灵敏度研究等，同时也提供了 Workbench 用户分析的深度和广度。

参数与参数菜单系统

2.1 参数概念与类型

参数是指 APDL 中的变量与数组。变量参数有两种类型：数值型和字符型；数组参数有三种类型：数值型、字符型和表，其中表是一种特殊的数值型数组，允许自动进行线性插值。

在 APDL 中任何参数都不需要单独声明参数的类型。数值型参数，无论整型还是实型都按照双精度数进行存储，被使用但未赋值的参数程序将其默认为一个接近 0 的极小值（大约为 2^{-100} ）。字符型参数存储字符串，赋值方法是将字符串括在一对单引号中（字符串最大长度不超过 8 个字符）。

与其他编程语言完全类似，参数可以作为任何命令的值域或在交互界面的输入框中替代各种具体的数值和字符串。当前面的参数值发生改变时，重新执行带参数的操作或者命令就会执行新参数值的处理。例如，将 1 赋给参数 kpx，将 10 赋给参数 kpy，将 -5 赋给参数 kpz，然后执行命令 `k,1,kpx,kpy,kpz`，相当于定义坐标为(1,10,-5)的关键点 1，定义关键点 1 的完整命令流如下：

```
kpx=1
kpy=10
kpz=-5
/prep7
k,1,kpx,kpy,kpz
```

如果修改上述命令流中的 kpx,kpy,kpz 的赋值大小，后边定义的关键点 1 的位置则相应改变，这就是参数化定义模型的思想。

2.2 参数的命名规则

参数名称必须遵循以下规则：

- (1) 必须以字母开头，长度不超过 32 个字符。

(2) 只能包含字母、数值和下划线。

(3) 一般不能以下划线开头，以下划线开头的参数为系统隐含参数（在 ANSYS 系统中不显示，只是编写代码的人员自己知道），只能用于 GUI 和宏中。

(4) 以下划线（_）结尾命名的参数可以用*STATUS 命令成组列表显示，也可以成组利用*DEL进行删除。

(5) 不能使用宏专用的局部参数名：ARG1~ARG9 和 AR10~AR99。

(6) 不能使用*ABBR 命令定义的缩写。

(7) 不能使用 ANSYS 标识字（Label）。

- 自由度标识字：TEMP、UX、PRES 等。
- 通用标识字：ALL、PICK、STAT 等。
- 用户定义标识字：如用 ETABLE 命令定义的。
- 数组类型标识字：CHAR、ARRAY、TABLE 等。
- ANSYS 的函数名称：SQRT、ABS、SIN 等。
- ANSYS 的命令名称：K、LSTR、N 等。
- 已经定义的组件与部件名称（Component and Assembly）。

下面举例说明一些有效和无效的参数名称。

有效参数名称：

Radius1

Length

Width

Radius_of_Hole

无效参数名称：

ABC123456789012345678901234567890（长度超过 32 个字符）

S@B（含非法字符“@”）

UX（系统的自由度标识字）

12add3（以数值开头）

2.3 参数化操作环境介绍

在正式学习参数的用法之前，先要熟悉 ANSYS 中与参数相关的菜单系统，如图 2-1 所示为参数化操作菜单，如图 2-2 至图 2-4 所示为参数化操作菜单的下级子菜单。对照各级菜单路径，该参数化操作菜单所包含子菜单的意义与功能如下：

Utility Menu>Parameters>

- Scalar Parameters...：定义/删除/编辑变量。
- Get Scalar Data...：*GET 提取数据库数据并赋值给变量。
- Array Parameters：定义/删除/编辑数组或表（如图 2-2 所示）。
 - Define/Edit...：定义/删除/编辑数组。
 - Fill...：*VFILL 填充数组。
 - Read From File...：读数据文件到数组。

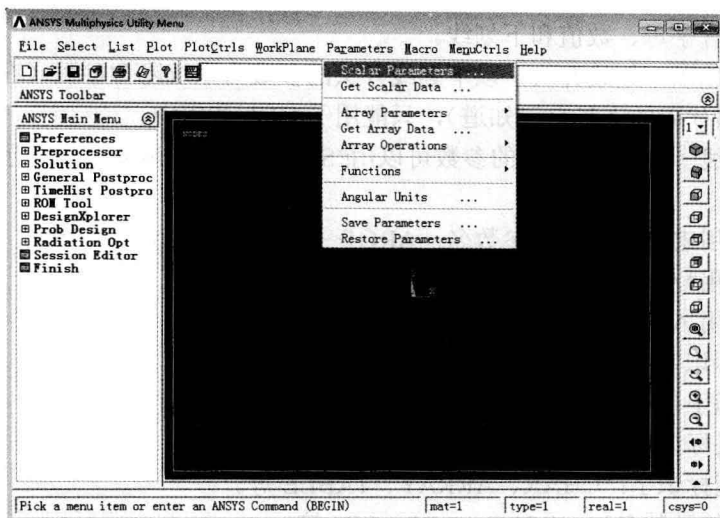


图 2-1 参数化操作菜单

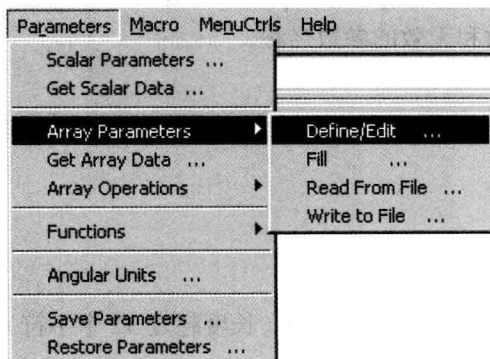


图 2-2 定义/编辑/填充/读入/写出数组与表菜单

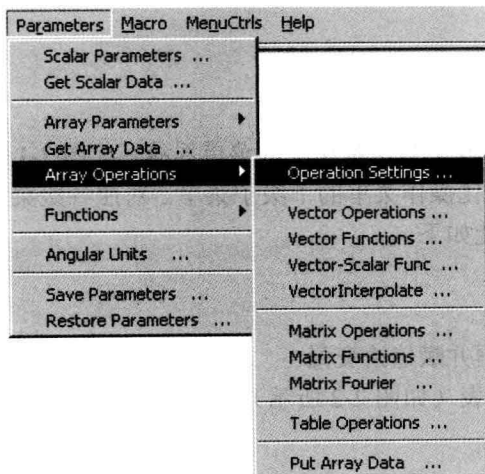


图 2-3 数组（矢量/矩阵）运算菜单

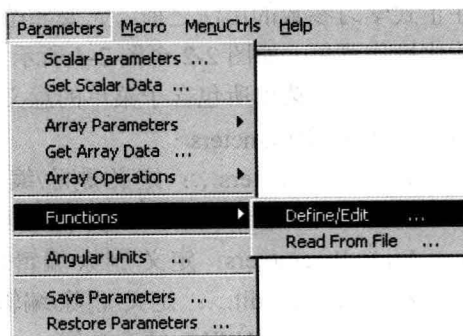


图 2-4 自定义函数菜单

- Write to File...: 写数组到数据文件。
- Get Array Data...: *VGET 提取数据库数据并赋值给数组或表。
- Array Operations: 数组运算 (如图 2-3 所示)。
 - Operation Settings...: 操作设置。
 - Vector Operations...: 矢量运算。
 - Vector Functions...: 矢量函数。
 - Vector-Scalar Func...: 矢量-变量函数。
 - Vector Interploate...: 矢量插值。
 - Matrix Operations ...: 矩阵运算。
 - Matrix Functions...: 矩阵函数。
 - Matrix Fourier...: 矩阵傅里叶运算。
 - Table Operations...: 表运算。
 - Put Array Data...: 输出数组数据到结果数据库。
- Functions: 编辑与读写自定义函数 (如图 2-4 所示)。
 - Define/Edit...: 定义/编辑函数 (函数编辑器)。
 - Read From File...: 从函数文件中读取函数 (函数加载器)。
- Angular Units...: 内部三角函数的角度单位。
- Save Parameters...: 存储参数。
- Restore Parameters...: 恢复参数。

变量参数及其用法

3.1 变量的定义与赋值

变量定义与赋值有以下 6 种途径：

- (1) 利用*SET 命令进行定义与赋值。
- (2) 利用赋值号“=”进行定义与赋值。
- (3) 利用菜单路径 Utility Menu>Parameters>Scalar Parameters 或命令输入窗口进行定义与赋值。
- (4) 在启动时利用驱动命令进行定义与赋值。
- (5) 利用*GET 及其等效函数提取 ANSYS 数据库数据进行定义与赋值（参见 7.1 节中的相关介绍）。
- (6) 利用*ASK 命令进行定义与赋值。

3.1.1 利用*SET 命令进行变量定义与赋值

*SET 命令定义和赋值参数的格式如下：

```
*SET, Par, VALUE, VAL2, VAL3, VAL4, VAL5, VAL6, VAL7, VAL8, VAL9, VAL10
```

其中，Par 是参数名；VALUE 是参数的赋值，可以是数值或字符串；VAL2~VAL10 也是参数的赋值，可以是数值或字符串。

利用该命令定义和赋值参数的实例如下：

```
*SET,Width,12 (Width 赋值为 12)
```

```
*SET,EX_Mat1,2.1E11 (EX_Mat1 赋值为 2.1E11)
```

```
*SET,Length,Width (Length 赋值为 Width, 即 Length 等于 12)
```

```
*SET,File_name,'Good' (File_name 赋值为'Good')
```

```
*SET,A(1),1,2,3,4 (数组元素赋值 A(1)=1, A(2)=2, A(3)=3, A(4)=4)
```

3.1.2 利用赋值号“=”进行变量定义与赋值

“=”可以直接用来定义和赋值变量，它作为一种速记符实际上是通过内部调用*SET命令实现参数的定义与赋值，其标准格式如下：

Name = Value

其中，Name 是参数名；Value 是赋给参数的数值或字符，字符值必须放在一对单引号中，长度不超过 8 个字符。

对应 3.1.1 节中的实例，下面是利用“=”方式定义的方法：

Width=12

EX_Mat1=2.1E11

Width=12

Length =Width

File_name='Good'

A(1)=1

A(2)=2

A(3)=3

A(4)=4

3.1.3 利用变量定义菜单或命令输入窗口进行变量定义与赋值

在 ANSYS 命令输入窗口中可以直接按照*SET 命令或“=”格式定义并赋值变量，如图 3-1 所示就是定义 Width=12 的方法（注意，所有的命令都可以在该命令输入窗口中执行）。

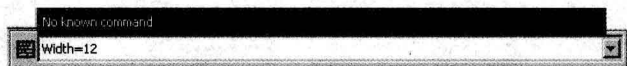


图 3-1 在命令输入窗口中定义并赋值变量

另外一种是利用菜单路径 Utility Menu>Parameters>Scalar Parameters 进行定义与赋值变量的方法。选择该菜单路径，弹出如图 3-2 所示的定义/赋值/删除变量对话框，在对话框中的 Selection 文本输入框中利用“=”格式输入变量定义与赋值表达式，然后单击 Accept 按钮，定义成功的变量将显示在 Items 的列表框中（这里显示的变量包括其他所有方法定义的变量）。

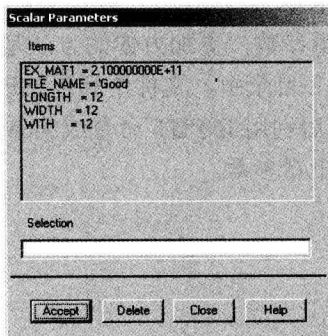


图 3-2 定义/赋值/删除变量对话框

3.1.4 在启动时利用驱动命令进行变量定义与赋值

在交互图形界面启动 ANSYS 时，弹出如图 3-3 所示的启动设置界面，图中粗线方框中就是启动变量输入文本框，按照格式“-Para1 Value1 -Para2 Value2 ...”在其中进行变量定义与赋值，图 3-3 中定义了两个变量，即 Width=12 和 Radius=4。

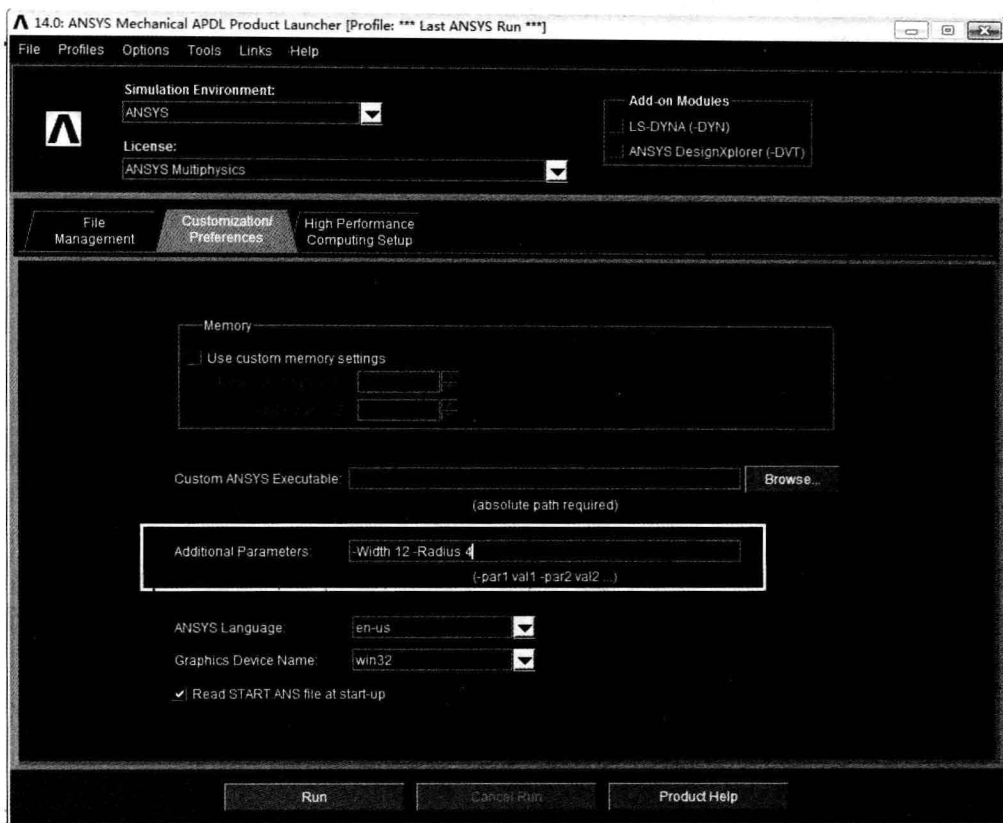


图 3-3 启动设置界面（粗线方框中为启动变量输入文本框）

如果采用命令驱动 ANSYS 时，在 ANSYS 的运行命令之后按照格式“-Para1 Value1 -Para2 Value2 ...”进行变量定义与赋值。对应图 3-3 所示的命令方式如下：

```
ansys140 -Width 12 -Radius 4
```

如果启动时需要定义大批变量参数，更加方便的方法是在 start140.ans（正常安装情况下位于...\\ANSYS Inc\\v140\\ansys\\apdl 目录中）文件中利用*SET 或“=”进行变量或者数组参数定义；或者定义一个参数文件，然后利用/INPUT 命令或菜单路径 Utility Menu>File>Read Input from 读入该文件，定义并赋值大批量参数。

3.2 删除变量

删除变量有下列三种基本方法：

(1) 菜单删除：选择菜单 Utility Menu>Parameters>Scalar Parameters，弹出如图 3-2 所示

的对话框，选中 Items 变量列表中的变量，然后单击 Delete 按钮。

(2) *SET 命令赋空值删除，对于字符参数则赋值为"（空字符串）。

例如，要删除 Width 变量，执行命令：

```
*SET,Width,
```

(3) "="命令赋空值删除，对于字符参数则赋值为"。

例如，要删除 Width 变量和 File_name 字符变量，执行命令：

```
Width=
```

```
File_name=
```



注意

给变量赋 0 并不会删除该变量，而是等于 0。同理，给字符变量赋单引号中为空格也不会删除该变量。

3.3 数值型变量值的替换

一般首先定义一序列的数值型变量，然后用他们替代命令或者交互界面输入域中的具体数值，程序会自动用变量的数值进行替代并执行命令。如果使用的变量并没有提前定义并赋值，程序会自动赋给它一个接近 0 的值 (2^{-100})，并且不会发出任何警告信息。除/TITLE、/STITLE、*ABBR 和/TLABEL 命令之外的大多数情况下，如果变量在命令中使用之后被重新定义或赋值，前面使用该变量的命令不会自动更新。

参见下面的实例：

```
X=3
```

```
Y=4
```

```
C=SQRT(X*X+Y*Y)    !平方和的平方根
```

```
X=8                !对变量 X 重新赋值并不会改变 C 的运算结果
```

3.4 字符参数的用法

在 ANSYS 中字符主要有文件名及其扩展名、命令名称及其字符值域、分析标题、宏文件名等，在一般情况下都是直接使用规定的名称字符串，有时需要利用字符参数替代这些具体的名称、扩展名、标题或子标题、字符值域等，在程序运行中用字符参数的赋值进行替代。字符参数的定义与赋值必须用一对单引号 (') 将赋值字符串括起来。例如，下面是存储 ANSYS 数据库的程序，每次运行时只需改变文件名变量的赋值就可以实现存储任何名称的数据库文件：

```
File_Name='My_First_DB'    !定义并赋值文件名变量 File_Name
```

```
SAVE, File_Name,DB        !SAVE 命令中引用文件名变量并发生替代
```

3.4.1 字符参数的常见用法

(1) 替代命令中的字符值域，如上述 SAVE 命令的实例。

(2) 在利用*USE 命令或选择菜单 Utility Menu>Macro>Execute Data Block 调用宏时替代宏名的字符参数。实例如下：