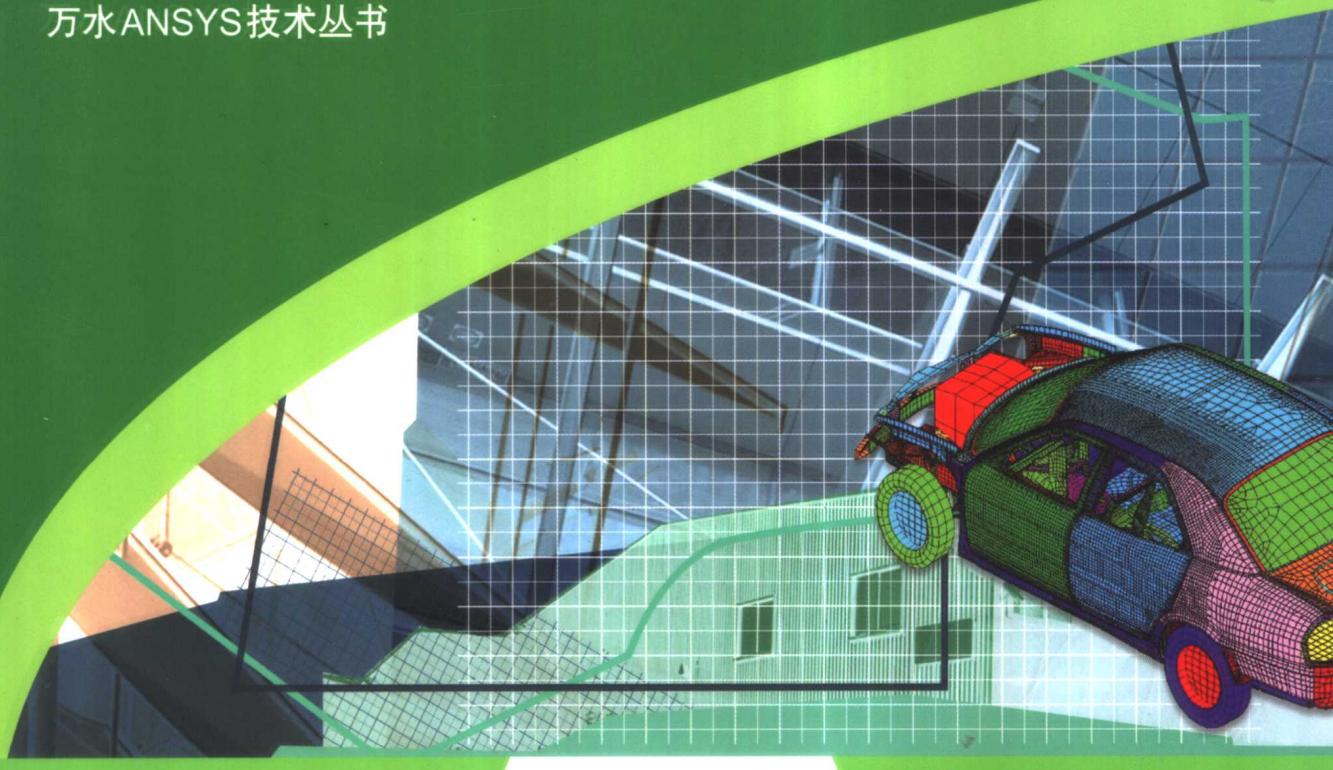
万水ANSYS技术丛书



APDL
参数化有限元分析技术及其应用实例

博弈创作室　编著

中国水利水电出版社
www.waterpub.com.cn

万水 ANSYS 技术丛书

APDL 参数化有限元分析技术及其应用实例

博弈创作室　编著

中国水利水电出版社

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以及设计优化等，是ANSYS不可缺少的重要技术，所有ANSYS使用人员都应该掌握它，丰富自己的分析手段，提高工作效率。APDL技术一直被认为是成为一名ANSYS高级用户的重要标志，也是广大ANSYS用户的永恒追求和至高目标。

本书主要分两大体系介绍学习参数化设计语言APDL，前十四章主要介绍APDL语言的基本要素，从第十五章到十七章重点介绍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、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的编程语言的组成部分，他们可以很好地将ANSYS的命令（代表不同的有限元分析处理指令和系统信息操作指令）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。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，界面系统开发以及必须基于APDL的优化设计技术。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。

本书主要适合于已掌握基本操作的ANSYS初级用户和部分中、高级用户，是一本学习APDL的技术资料，也是灵活掌握ANSYS专题分析技术的辅助资料。通过对本书的学习读者会进一步提高有限元分析的丰富分析手段和综合应用能力。

图书在版编目（CIP）数据

APDL参数化有限元分析技术及其应用实例 / 博弈创作室编著. —北京：中国水利水电出版社，2004.3

（万水ANSYS技术丛书）

ISBN 7-5084-2016-0

Ⅰ. A… Ⅱ. 博… Ⅲ. 有限元分析—APDL语言—程序设计 Ⅳ. ①0241.82 ②TP312

中国版本图书馆CIP数据核字（2004）第009888号

书　　名	APDL参数化有限元分析技术及其应用实例
作　　者	博弈创作室　编著
出版　发行	中国水利水电出版社（北京市三里河路6号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）63202266（总机）　68331835（营销中心）　82562819（万水）
经　　售	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	北京市天竺颖华印刷厂
规　　格	787×1092mm　16开本　14.75印张　333千字
版　　次	2004年2月第1版　2004年2月第1次印刷
印　　数	0001—5000册
定　　价	24.00元

前　言

自从计算机问世以来，人类对计算机和相关的应用技术越来越依赖，甚至变成了不可缺少的工具。在各行各业进行产品设计分析过程中，从方案选择、方案实施、产品原型和实验，到最后产品定型和市场化，计算机技术始终服务着我们，CAD/CAE/CAM 以及 PDM 等是最常见的计算机应用技术。其中，CAE（Computer Aided Engineering）即计算机辅助工程，在产品设计过程中根据提出的产品原型建立相应的计算机仿真模型和虚拟工作环境，利用计算机仿真程序检验产品的各种功能特性，测试产品能否实现预期的功能，并力求达到良好的经济效益和社会效益。在产品概念设计阶段、产品设计前期以及定型验算和实验研究的全部过程，CAE 技术已经成为必不可少的分析和优化技术，为产品的可靠性和高效性提供技术保证，同时避免大量实验验证过程，降低设计成本，提高设计效率，缩短开发周期，最终达到提高产品竞争力的目的，取得生存优势。

相对于传统设计分析手段，如按规范设计、实验方法、经验方法设计等设计方法，CAE 技术具有极大的优势。实验方法要求设计人员必须制造出不同实验产品，然后进行实验测试和评估，通过实验查找设计中的不足并不停改进，需要反反复复，周期不可控制，而且人为因素和实验的随机性和误差等不可确定性影响因素很多，使实验的正确性和成功率难以得到保证，另外实验和模型的费用往往十分昂贵。经验方法对于行业新人来说十分困难，因为经验产生的条件和背景他们往往不是特别清楚，致使在利用经验的过程中难免不出现失误，这就成为产品可靠性和安全性的巨大隐患。总之，传统的许多分析方法越来越难以满足我们社会高速发展带来的丰富而又多变的需要，难以满足高社会经济效益和高性能提出的精益设计的要求，而 CAE 技术正是设计分析技术的革命，为广大工程设计人员提供一个很好计算设计分析平台，完成产品设计的仿真工作。在当今世界上，美国 ANSYS 公司是最具盛名的 CAE 提供商之一，它提供 CAE 设计分析程序 ANSYS 系列产品，功能完善，在世界上得到广泛应用。

在 ANSYS 得到广泛应用的同时，许多技术人员对 ANSYS 程序的了解和认识还不够系统全面，这会给我们的工作和研究带来许多隐患和障碍。一方面，分析人员需要花费大量的精力去学习 ANSYS，熟悉应用环境和系统资源，掌握关心的技术，研究其运用的规律和技巧，研究分析不同问题的技巧。这个过程本身是很难的，因为 ANSYS 只是我们工作和研究的工具之一，我们需要的功能也往往只是其中的一个部分，如何在最短时间掌握所关心的技术才是我们的关键所在；另一方面，ANSYS 的技术资料较少，学习培训的机会也少，许多分析人员感到无从下手，不知道如何学习 ANSYS，自己掌握到什么程度才能顺利正确地完成自己的分析工作。这些都是广大分析人员面临的问题，为适应需要特此撰写本书，供大家参考，限于水平有限，难免有不足之处，敬请各位读者指教。

作者

2003 年 11 月

目　　录

第一章　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。

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。

第二章　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，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，只能由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：如 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：

1．有效参数名称：

Radius1

Length

Width

Radius_of_Hole

2．无效参数名称：

ABC12345678901234567890123456789O（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，首先熟悉 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、图 2-3 和图 2-4 所示分别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：

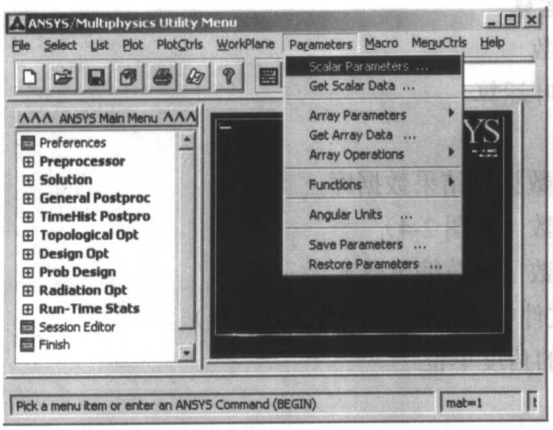


图 2-1　参数化操作菜单

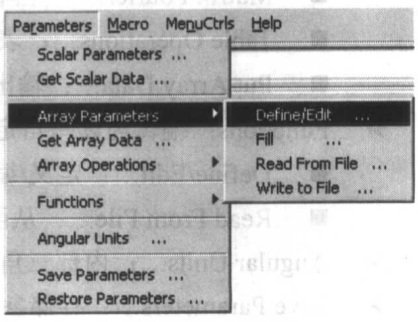


图 2-2　定义/编辑/填充/读入/写出数组与表菜单

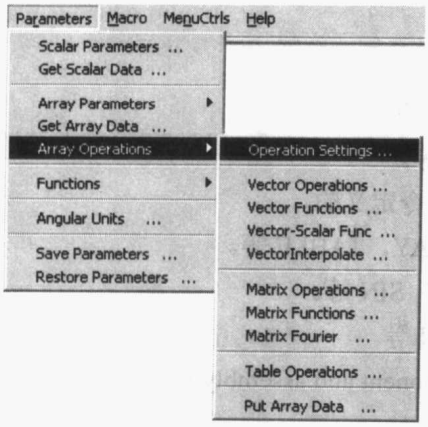


图 2-3 数组（矢量/矩阵）运算菜单

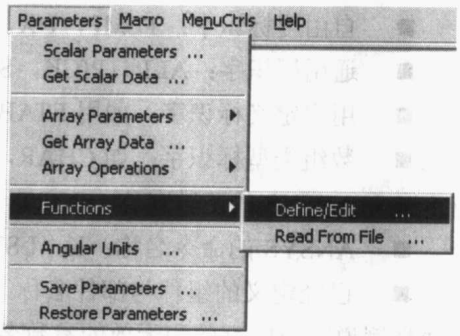


图 2-4 自定义函数菜单

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）
  - 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
  - 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）
  - 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...：矩阵傅利叶运算
  - 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）
  - 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

第三章　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．利用命令*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（参见第 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　利用命令*SET 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（即 Width 赋值为 12）

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

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

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

*SET,A(1),1,2,3,4（即数组元素赋值 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是通过内部调用*SET 命令实现参数定义与赋值，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执行）

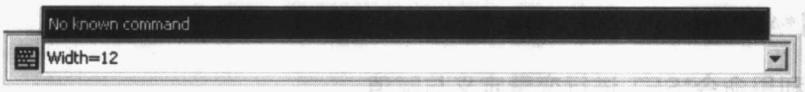


图 3-1 在命令输入窗口定义并赋值变量

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）。

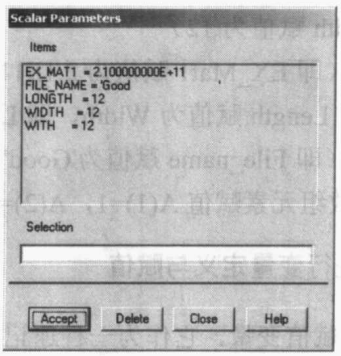


图 3-2 定义/赋值/删除变量对话框

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。

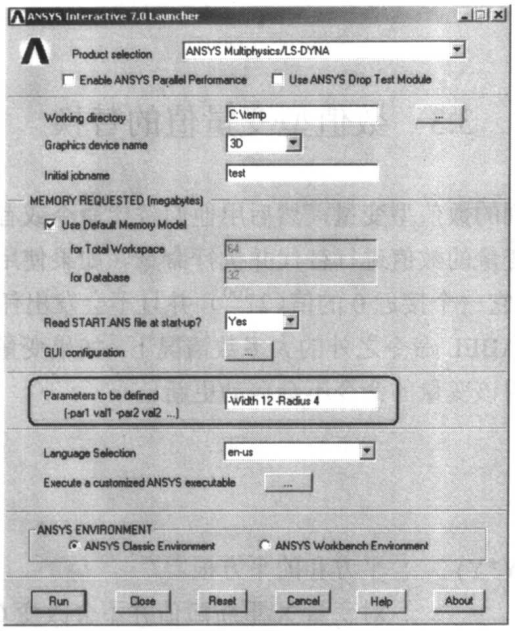


图 3-3　启动设置界面（粗线方框中为启动变量输入文本框）

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：

ansys70 –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 start70.ans（正常安装情况下位于.../ANSYS Inc/V70/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的方法：

1．菜单删除：选择菜单 Utility Menu>Parameters>Scalar Parameters，弹出如图 3-2 所示对话框，选中 Items 变量列表中的变量，然后单击 Delete 按钮删除之。

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'    ! 首先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：

```
Macro_Name='Solve_Case1'    ! Macro_Name 字符变量记录宏文件名
*USE, Macro_Name            ! *USE 调用 Macro_Name 变量指定的宏
```

3. 替代*ASK 命令中的提示字符串。实例如下：

```
String_Query='Please Enter With'    ! String_Query 字符变量记录提示信息
```

```
*ASK, Par, Query, DVAL          ! *ASK 弹出对话框显示 String_Query
                                ! 定义的信息字符串，并输入变量赋值
```

4. *CFWRITE 命令是将命令行写出到*CFOPEN 命令打开的文件中，可用于写一个分配给该文件的字符参数。实例如下：

```
*CFWRITE,File_name='abc'
```

5. 替代*CFOPEN、*VREAD、PARSAV 与 PARRES、/OUTPUT 等命令及其他们对应的菜单中的文件名与扩展名。

6. 用于*IF 和*ELSEIF 命令的 VAL1 和 VAL2 值域，用于比较字符串是否相同或不同，所以 Oper 值域只能使用 EQ (等于) 和 NE (不等于)。实例如下：

```
A='True'
B='False'
*IF,A,EQ,B,THEN    ! 如果 A 与 B 相同，那么……
```

7. 用于*MSG 命令的 VAL1~VAL8 值域，字符串描述符%C 用于在格式行中指明字符数据。

8. *VREAD 命令或菜单 Utility Menu>Parameters>Array Parameters>Read from File 用于从某个文件中读取字符参数并生成一个字符数组参数。在*VREAD 命令后的格式行中采用 FORTRAN 的字符描述符 A 定义字符串格式。

9. *VWRITE 命令或菜单 Utility Menu>Parameters>Array Parameters>Write to File 用于来以某种格式化的顺序把字符参数数据写到一个文件中。在*VREAD 命令后的格式行中采用 FORTRAN 的字符描述符 A 定义字符串格式。

3.4.2　强制字符参数执行替换

把字符参数名括在两个百分号%中可以实现强制替换，主要目的是实现在字符串中插入变化的子字符串。强制替换只能使用在下列场合：

1. 在/TITLE 命令的标题字符串中。
2. 在/STITLE 命令的子标题字符串中，与/TITLE 命令一样。
3. 在/TLABEL 命令的注释字符串中。
4. 在/SYP 命令的 ARG1 - ARG8 值域中，将命令传递到操作系统。
5. 在*ABBR 命令的缩写值域中，实现定义缩写。
6. 在命令的文件名或扩展名值域中，如/FILNAM、RESUME、/INPUT、/OUTPUT 和 FILE 等。
7. 在命令的任何 32 位字符域，如目录路径。
8. 在任何命令名值域中替代命令名，也可在值域 1 中作为一个“未知命令”的宏名。

实例 1：替换文件名

```
St1='My_'
St2='Model'
/FILNAM,%St1%%St2%
```