文章编号:1000 - 582X(2003)01 - 0028 - 06

运载火箭控制系统漏电诊断平台的设计与实现*

唐贤明[1,2], 樊友平[1], 毛万标[2], 韩荣军[1], 曾雪红[1], 黄席樾[1]
(1. 重庆大学 自动化学院, 重庆 400044; 2. 西昌卫星发射中心, 四川 西昌 615000)

摘　要:论述了自行研制设计的可视化漏电故障传播关系图分析系统的功能及程序结构与设计原理。该系统由故障诊断模块和人机交互模块组成,可使运载火箭控制系统漏电故障传播关系图的生成与分析直观而简便,只需用鼠标在屏幕上绘制出传播关系图,就能自动识别并进行定性和定量分析。该方法不仅是故障传播关系图的新绘制方法,也是全新的故障树数据输入法,是故障树算法的拓展和补充。最后给出了一个实例,应用图形输入法得到满意结果。

关键词:可视化分析系统; 结构化变量; 计算机辅助分析; 漏电故障传播关系图
中图分类号:TP306.3;TP302.1　　**文献标识码**:A

在进行运载火箭控制系统漏电故障诊断研究的过程中,为对所提出的诊断理论和方法进行验证,笔者设计并实现了用于漏电故障诊断的平台,并且在该平台上开发了用于 CZ - 3B 运载火箭控制系统的漏电故障诊断软件。由于 CZ - 3B 运载火箭控制系统的漏电故障传播关系图十分庞大,如果用传统字符式的输入和编辑方法,在输入时,稍有疏忽,就会导致错误输入;在编辑时,也非常不利于迅速定位需要编辑的对象,所以平台还加入了故障传播关系图的图形输入功能。除此之外,为使平台具有扩展性,还为平台设计了基于 COM 的诊断算法接口,符合该接口的算法可很方便地加入到平台里。

1　漏电故障诊断平台的整体结构和功能

整个平台是一套可视化程度高的故障分析系统。它可用鼠标在屏幕上很方便地绘制出漏电故障传播关系图,故障传播关系图的生成和分析非常直观简便。漏电故障诊断平台分为两部分:故障诊断模块和人机交互模块[1-2]。故障诊断模块是设计与开发工作的核心,主要负责故障诊断任务的建立、管理、运行,以及与人机交互模块进行通信;人机交互模块主要解决故障关系传播图的图形输入,充当操作人员与故障诊断模块的桥梁。模块关系见图 1。

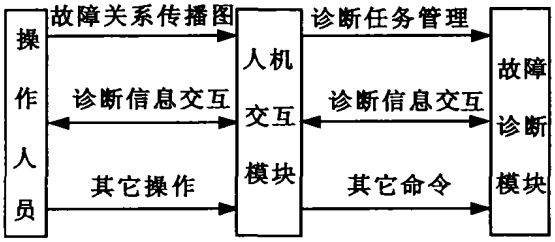


图 1　模块关系图

2　人机交互模块设计与实现

漏电故障关系传播图由 2 种基本单元组成,即部件和故障传播关系(以下简称关系)。从图形的角度看,两者具有一些相同属性和操作功能。例如每个部件或关系都应有一个唯一标识,它们都必须完成绘制操作等等。将这些相同的属性和操作放在一个类 CDrawObject 中,部件和关系均由它派生出来。另外,由于类 CDrawObject 没有现实对应,因此它是不能实现例化的,所以它的函数全部定义为纯虚函数,即它是一个抽象类。

部件所对应的类称为 CUnit,它是从 CDrawObject 继承而来。它的定义包括:CDrawObject 所定义的各项操作的实现;部件对应的图形的外观属性,如图形的大小、颜色、位置、图形描述文字的字体信息等等;故障诊断的相关属性,如 10 个在不同诊断条件下的权值、对

* 收稿日期:2002 - 09 - 08
基金项目:国防预研基金资助项目(2001281);教育部高等学校博士学科点专项科研基金资助项目(99061116)
作者简介:唐贤明(1949 -),男,安徽芜湖人,高级工程师,研究方向为智能控制与知识工程、复杂控制系统的智能故障诊断。

部件的功能描述、位置描述以及存放部件外观照片的图形文件的地址等等；其它操作，如对图形进行放大和缩小、保存各种属性到文件中等等。

关系所对应的类称为 CLink，它也是从 CDrawObject 继承而来。它的定义包含：CDrawObject 所定义操作的实现；关系所对应图形的外观属性，例如连续直线段的宽度、颜色、各个端点位置等等；故障诊断的相关属性，如它的起点和终点所联系的对象，关系是单向还是双向等等。一个故障关系传播图中有大量的部件和关系，管理这些对象使用了 MFC Collection Classes 中的 CTypedPtrlist，它基于模板的是双向链表，有次序性（需依序处理），链有头尾，可从头尾或从链表的任何位置插入元素，速度快。定义以下 2 种链表类型来分别管理 CUnit 和 CLink：

typedef CTypedPtrList < CObList, CUnit * > CUnitList;

typedef CTypedPtrList < CObList, CLink * > CLinkList;

要实现故障关系传播图的图形输入，就是要可视的创建部件和关系，使图能够实时显现，并且能够实现删除、修改、拖动等功能。这些功能与一般的 CAD 软件的功能是一样，但故障模型图与普通的 CAD 软件不同之处在于图中的对象之间不仅有位置上的关系，还存在故障逻辑上的关系，自然还存在一些与故障诊断相关的算法。

2.1 对象的图形创建

部件的图形创建十分简单，在软件界面的工具条上选定操作后，在绘图区按一下鼠标左键，平台捕获 WM_LBUTTONDOWN 鼠标消息后，调用鼠标左键处理函数，该函数从鼠标消息里获取按键时鼠标的位置，据此生成一个部件的图形，然后即可对它进行各种操作，如修改它的各种属性、改变位置等等。

但在创建关系时，由于关系的图形是一根或多根连续不封闭的直线段，所以仅经过一次按键是无法完成创建关系的过程，必须经过 2 次以上的按键才能完成。同时，关系的创建是有一定条件约束的，即关系图形的起点与终点必须与图形对象相连，起点或终点在空白区域的关系是没有物理实际意义的，所以在创建关系时应该确保生成的关系图形是有效的。

这些与创建关系相关的操作自然可以定义在关系的类中，但从面向对象的角度看，这些操作是不应该定义在关系的类中，因为在调用这些操作函数时，并没有存在一个有效的关系，因此设计了一个专门负责创建关系的类（ClinkCreator）。从实际角度看，在整个软件的运行空间里，应该只有一个 ClinkCreator 对象来负责关系的创建，而不必每生成一个 Clink 对象就要先生成一个 ClinkCreator 对象来负责创建工作，这可以通过设计一个 ClinkCreator 的全局对象来实现。但在软件的运行期内，仅仅是在进行图形修改或漏电故障诊断的操作，而没有创建新的 Clink 对象，此时在保留一个 ClinkCreator 对象显然是没有必要的。因此当没有提出创建 Clink 对象的操作时，软件的运行空间中不存在 ClinkCreator 类的实例，当我们第一次要求创建 Clink 对象时，程序会实例化一个 ClinkCreator 对象，并且使该对象一直存在并负责后续创建 Clink 对象的操作。

2.2 绘图网格功能

在许多的 CAD 软件中都具有一种绘图网格功能。在这项功能启动的情况下，软件的绘图区有很多横向和纵向的网格线，用来帮助对齐图形对象。在拖动或绘制图形对象时，软件会自动将其与最近的网格线交叉点对齐。这种功能使绘图时可以很方便地对齐线段、图形，因而使整个图形更加美观整洁。

实现该功能时，要记录所有网格线交叉点在绘图区的坐标，由于网格线交叉点数目庞大，通常不直接记录这些交叉点的坐标，而是依据网格线的横向间隔与纵向间隔动态计算来获取交叉点的坐标。该项功能与图形的绘制密切相关，在绘制图形的过程中，程序要处理的 Windows 消息有 WM_LBUTTONDOWN、WM_LBUTTONUP、WM_MOUSEMOVE 等等，这些由鼠标发送的消息里都包含了鼠标发送该消息时在绘图区的坐标，在相关的消息处理函数里，将这些坐标调整为与其最近的网格线交叉点的坐标，即可实现绘图网格功能。具体做法是：将消息中的原始横坐标对网格线的横向间隔求余，余数如果小于横向间隔的 1/2，则新的横坐标为原始横坐标减去余数，但如果余数大于横向间隔的 1/2，则新的横坐标为原始横坐标加上网格线的横向间隔与余数之差。纵向坐标调整类似。这样得到的坐标就是距鼠标消息中的坐标最近的网格线交叉点。

2.3 诊断过程中的人机交互接口

由于采用的是序贯诊断方式，在诊断过程中诊断任务会输出中间提示信息，用户根据提示信息的指示进行测量，然后将测量结果反馈给诊断算法，以指示诊断任务继续运行。当诊断任务等待用户的反馈信息时，它是处于等待状况，而人机交互接口则处于运行状态，当用户输入测量结果后，诊断任务又继续运行，而人机交互接口则处于等待状态，直到下一次需要用户输入测量结果。以上所描述的过程涉及到人机交互和故障诊断，如果将两部分混为一体，必然导致逻辑关系

含糊不清,人机交互接口程序和故障诊断程序之间相互重叠,这样的程序结构不利于排错,也不利于扩展与改善。更好的方法是使用多线程。

2.4 检查漏电故障关系传播图的有效性

在图形编辑过程中,常常会更改和删除一些图形单元。由于图形单元之间存在着故障的逻辑关系,而这些关系也许会由于编辑图形单元而被破坏。因此必须在诊断前检查漏电故障关系图是否有效。

通常在以下几个方面进行检查:

1)图中是否有图形单元。图中没有部件或关系显然是无法诊断的,所以首先检查存放部件对象的队列和存放关系对象的队列是否为空。

2)是否存在无效关系。虽然在关系的创建中,会保证关系图形的起点与终点是与图形对象相连,但有可能关系所关联的对象在后来的操作中被删除,这样的关系就是无效的。所以有必要检查是否存在无效关系。

3)查图形是否为连通图。故障关系传播图应该是连通的,如果模型本身是不连通的,那就应将模型视为由多个故障关系传播图组成,而分别进行故障诊断。检查图形的连通性有多种方法,笔者采用了一种基于深度优先搜索的方法。符号 VERTICES(G)将返回图 G 中的节点数,算法如下:

输入 图 G。

输出 是否为连通。

a.对于每个节点 v,MARK(v)←0,FATHER(v)←0,同时 n←0。

b.任选一个节点 s,设为起始节点,v←s。

c.MARK(v)←1,n←n+1。

d.如果 v 的所有"子"节点的 MARK 值均为 1,则转到 e;否则任选一个未被访问的"子"节点 t,FATHER(t)←v,v←t 转到 c。

e.如果 FATHER(v)≠0,则 v←FATHER(v)转到 d;否则转到 f。

f.如果 n 与 VERTICES(G)一致,则图是连通的,否则图不连通。

2.5 图形单元的动态调整

在绘制故障关系传播图时,通常会改变图形单元的位置。如果没有图形单元的动态调整功能,图形单元的位置变换后,图形单元之间的故障逻辑关系在图形外观上就显得支离破碎,例如图 2(b)是图 2(a)中节点 1 移动后的图形。

虽然图 2(a)和图 2(b)在故障逻辑关系上还是一样的,但显然图 2(b)非常容易引起错误。所以设计图形单元的动态调整功能,即当一个图形单元位置变化后,与之相关联的图形单元的位置也发生相应的变化。

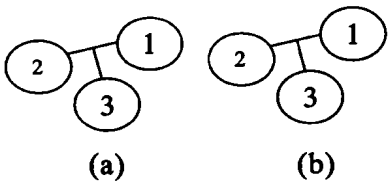


图 2 图形单元的动态调整

部件和关系在位置变化时的约束条件不同。部件可以自由移动。如设一个关系不能整体移动,对关系位置的调整是通过调整其直线段的端点实现的,首尾两个端点的移动必须在图形单元之间,其它端点可以自由移动。

当调整部件 U 的位置时,与部件 U 相联的所有关系 L_i(i=1,2,…,n)都应该有相应的调整,调整方式为:L_i 与 U 相交的那段直线段的端点将跟随部件 U 做相应的调整,L_i 中的其它直线段端点保持不变。然后与 L_i 相联的部分关系也需要作相应的调整,调整方式为:关系 L_i 位置发生变化,即端点为 a、b 的直线段发生变化,所有与 a、b 的直线段相交的关系 L_j(j=1,2,…,n)都应有相应调整,即 L_j 的相交点将跟随 a、b 的直线段变化,而 L_j 中的其它直线段端点保持不变。

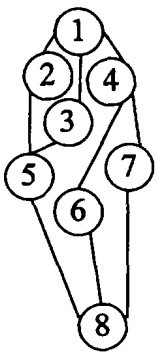
当调整关系 L 位置时,与关系 L 相联的部分关系 L_i(i=1,2,…,n)都应该有相应的调整,调整方式为:L 的端点为 a、b 的直线段发生变化,所有与 a、b 的直线段相交的关系 L_i 都应有相应调整,L_i 的相交点将跟随 a、b 的直线段变化,而 L_i 中的其它直线段端点保持不变。然后与 L_i 相联的部分关系也需要作相应调整,调整方式为:关系 L_i 位置发生变化,即端点为 a、b 的直线段发生变化,所有与 a、b 的直线段相交的关系 L_j(j=1,2,…,n)都应有相应调整,即 L_j 的相交点将跟随 a、b 的直线段变化,而 L_i 中的其它直线段端点保持不变。

3 故障诊断模块设计与实现

3.1 诊断模块中的数据结构[3]

在人机交互模块中,部件和关系分别采用不同的对象表示,从定义中可以看出故障传播关系是存放在关系对象中,部件和关系通过队列进行管理。如此安排是为了使人机交互模块中的各项功能,尤其是绘图功能的实现可以参考其它 CAD 软件的编制原理,从而简化编程工作。另一个好处是由于诊断模块支持多种诊断算法,人机交互模块中采用尽可能直观的数据结构,而让诊断算法自身负责产生它所需要的数据结构。

漏电故障诊断的算法是以故障关系传播图的领接矩阵为基础的，在具体实现上采用了领接链表来描述图的领接矩阵。图 4 与图 3 的领接链表相对应。



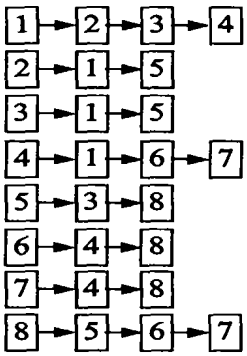


图 3　漏电故障关系传播模型　图 4　图 3 对应的领接链表

图 G 的领接链表形式定义如下：

CList < int, int > * G = new CList < int, int > [n];

其中 n 是图的节点数，CList < int, int > 是以 int 为单元的队列，一阶 CList < int, int > 的数组就代表了图。图的领接链表生成算法如下：

输入　部件对象队列 U_q 和关系对象队列 L_q；

输出　图的领接链表 G。

1) U_q 和 L_q 的访问索引均置于队列头部，U_q 中有 n 个部件，令 G 为一维 n 阶队列数组，每个队列均为空，$i \leftarrow 0$。

2) 如果 U_q 的访问索引已经到达尾部，转到 6)；否则，从 U_q 中依序取出部件对象 u，$i \leftarrow I+1$，转到 3)。

3) 如果 L_q 的访问索引已经到达尾部，转到 5)；否则，从 L_q 中依序取出关系对象 l，转到 4)。

4) 如果 l 与 u 关联，将 l 关联的另一个对象 v 加入到队列 $G[i]$中，转到 3)；否则直接转到 3)。

5) L_q 的访问索引置于队列头，转到 2)。

6) 结束

在实现中采用的树的数据结构定义如下：

```
struct Edge{
    int Prev;
    int Next;
};
typedef CList < Edge, Edge& > CTree;
```

Prev 和 Next 代表部件，结构 Edge 是节点序列对表示树的边，CTree 表示树。

最小路的数据结构就更简单，定义如下：

```
typedef CList < int, int > CPath;
```

该队列中的元素表示了部件，最小路集用一维 CPath 数组表示。

为了查询方便，在以上数据结构中用来表示部件的不是部件对象指针，也不是部件对象的唯一表示，而是部件对象在部件队列中的索引。

3.2　诊断任务的多线程实现[4]

在 MFC 中，线程分为 2 种：用户界面线程和辅助线程。用户界面线程常用于接收用户的输入，处理相应的事件和消息，它有自己的消息循环。辅助线程主要用于后台处理，不能直接处理用户输入信息。当诊断平台运行时，用户界面线程开始运行，当要进行故障诊断时，用户界面线程会创建一个辅助线程来执行诊断任务。设用户界面线程为 T_1，辅助线程为 T_2。T_1 的执行步骤如下：

1)创建事件 E_1 和 E_2，E_1 和 E_2 的初始状态设为无信号，清空进程变量 v_3。

2)创建 T_2 来执行诊断任务。

3)令 E_1 设为有信号。

4)等待 E_2 有信号。

5)E_2 变为有信号，等待结束。

6)如果进程变量 v_1 为 0，转到 9)；否则转到 7)。

7)将提示信息从进程变量 v_2 取出，输出到屏幕，并等待用户输入。

8)将用户反馈信息存放在进程变量 v_3 中，转到 3)。

9)将提示信息从进程变量 v_2 取出，输出到屏幕，并且释放 E_1 和 E_2。

T_2 的执行步骤如下：

1)等待 E_1 有信号。

2)E_1 变为有信号时，等待结束。

3)根据进程变量 v_3 进行诊断。

4)如果诊断结束，转到 7)；否则转到 5)。

5)需要向用户提供提示信息时，令进程变量 v_1 为 1，提示信息存放在进程变量 v_2 中。

6)令 E_1 为无信号、E_2 为有信号，转到 1)。

7)进程变量 v_1 为 0，诊断结果存放在进程变量 v_2 中，令 E_2 为有信号，线程结束。

3.3　基于 COM 的诊断算法接口[5-6]

故障诊断平台 COM 客户程序所有的诊断算法都采用进程内服务器的形式，所有诊断算法服务器都要满足实现接口 IDiagnose，IDiagnose 的定义如下：

```
import "unknwn.idl";
……
```

```
[
……
uuid ( E8A18AEE - 5D0C - 11D6 - 9A24 - BF7964DCF929),
……
]
interface IDiagnose : IUnknown{
typedef struct {
  ……
  }tagUnit;
  typedef struct {
  ……
  }tagLink;
   HRESULT GetInfo( [out , string] wchar-t * pName,
   [out , string] wchar-t * pVersion,
   [out , string] wchar-t * pProducer);
   HRESULT Initialize( [in]short  NumUnits,
[in , size-is(NumUnits)]tagUnit Units[],
[in]short  NumLinks,
[in , size-is(NumLinks)]tagLink Links[]);
   HRESULT Promote( [out , string] wchar-t * pPromotion),
[out]boolean  bOver);
   HRESULT Feedback([in]boolean bYes);
   HRESULT Finalize();
};
```

其中 GetInfo 是获取算法的名称、版本、开发商。Initialize 是进行算法初始化，它的参数传入了一维 tagUnit 数组和一维 tagLink 数组以及它们的大小。由于 IDL 语言是不支持对象，用来表示图的 CUnit 和 CLink 的队列不能直接通过接口传递。首先用 IDL 定义了 2 个结构 tagUnit 和 tagLink，它们的数据成员与类 CUnit 和 CLink 的数据成员是基本一致的(tagUnit 和 tagLink 使用的数据类型是 IDL 所规定的类型)，在调用该接口前要将 CUnit 和 CLink 的队列转化为 tagUnit 和 tagLink 的数组。Promote 是算法向外提供的提示信息和指示算法是否结束。Feedback 是用来输入用户的反馈信息。Finalize 是算法结束时所执行的一些必要的操作，如释放内存、文件等等。

将实现接口的 COM 对象的全局唯一标识符存放在 Windows 注册表中 HKEY-LOCAL-MACHINE \ Software \ CQU-Automation \ Diagnoser \ Extensions \，当诊断平台开始运行时，会搜索注册表中的该位置来确定系统中可用的算法服务器，然后通过 Idiagnose 的全局唯一标识符获取指向该接口的指针，通过 GetInfo 获取算法的基本信息。进行诊断时，平台根据用户选择的算法来创建相应诊断线程。

4 故障诊断平台应用实例及结论

应用 Visual C + + 语言实现了计算机辅助运载火箭控制系统漏电故障关系传播图的图形输入，并应用于漏电故障的诊断分析上，取得了令人满意的结果。下面以运载火箭控制系统的电源 I 负母线漏电故障诊断为例，给出了部分应用结果。图 5 是漏电故障诊断程序运行后，用鼠标点击工具按钮绘制出电源 I 负母线漏电故障关系传播图后屏幕的硬拷贝，图中的小长方形及编号代表线路中的元部件，在绘图过程中位置是自动调整、变化的。图 6 和图 7 是绘制故障关系传播图时编辑部件属性和部件关系后屏幕的硬拷贝。

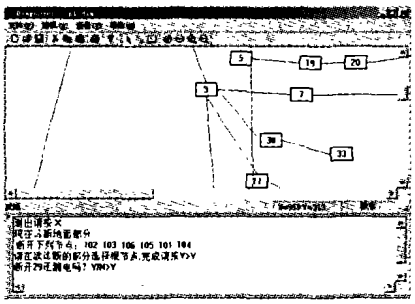


图 5 故障诊断平台运行主窗

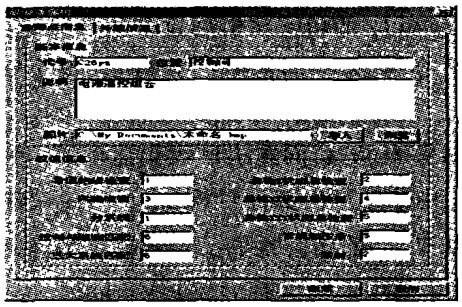


图 6 部件属性的窗口

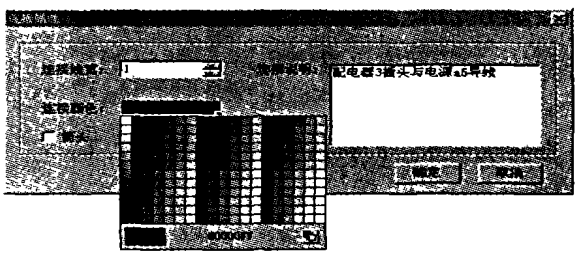


图 7 关系属性的窗口

参考文献：

[1] 华小洋，胡宗武，范祖尧. 多状态模糊故障树分析软件 MFFTAP[J]. 太原重型机械学院学报，1996,17(3): 23 - 27.

[2] GAMMA ERICH,JOHNSON RALPH, HELM RICHARD, et al. Design Patterns: Elements of Reusable Object - Oriented Software[M]. 北京:机械工业出版社,2000.

[3] 朱继洲. 故障树原理和应用[M]. 西安：西安交通大学出

版社，1988.
[4] 滕永盛. Windows95 下多线程中的同步机制[J]. 微型机与应用，1998，15(9)：10－13.
[5] 楼伟进. COM/DCOM/COM＋组件技术[J]. 计算机应用，2000，11(4)：31－33.
[6] 刘长有. 基于 COM 的控制系统仿真程序的设计与实现[J]. 计算机运用，2001，12(7)：64－65.

Designing and Realization of Leak Current Fault Flatroof for Lauch Vihicle Controlling System

TANG Xian-ming[1,2], *FAN You-ping*[1], *MAO Wan-biao*[2],
HANG Rong-jun[1], *ZENG Xue-hong*[1], *HUANG Xi-yue*[1]
(1. College of Automation, Chongqing University, Chongqing 400044, China;
2. Xichang Satellite Launching Centre, Xichang 615000, China)

Abstract: A visualized analysis system of leak current fault transmission graph is developed. Its functions, program structure and programming principle are described. The system is composed of fault diagnosing module and man-machine exchanging module, which make the generation and analysis of leak current fault transmission graph for lauch vehicle controlling system simple and visualized. The system could automatically recognize the picture of transmission and analyze both qualitatively and quantitatively only by drawing it on the monitor by the mouse. Finally an example is given, where a satisfying result is acquired via drawing inputting method which is totally new and expanding－complementing the fault tree alglorithm.
Key words: visualized analysis system; structurelized variable; computer aided analysis; leak current fault transmission relationship graph

（责任编辑　张　苹）

（上接第 17 页）

Optimization Method of Turbine Impeller's Stress and Deformation Calculation

YE Kang-feng, *WANG Bing-le*
(College of Mechanical Engineering, Chongqing University, Chongqing 400044, China)

Abstract: Taking into the feature of computer computing method, using Stress solution of elasticity theory and according to different method of solving of system of second order ordinary differential equations, this article provides two numerical value computing method as for the stress and deformation of the turbine impeller which is within the scope of elasticity. One combines the initial value computing method of system of ordinary differential equations (Runge－Kutta Method) with optimization method, another combines the boundary value computing method of system of ordinary differential equations-difference method－with optimization method and three points interpolation method. The proposed method can eliminate the deficiencies of Secondary Calculation method and is particularly suited for programmable computer－based solution. The sample show that results gotten by two methods is nearly equal to the precise results, so they are practicable. They completes the quick and precise calculating of stress and deformation. They have some general meaning, large commonality and the project employing value.
Key words: turbine impeller; stress; deformation; numerical value computing method

（责任编辑　成孝义）