光学自动设计程序汇编

国防工业出版社

光学自动设计程序汇编

南京大学数学系计算数学专业 编

国防工业出版社

内 容 简 介

本书内容包括以几何象差加权平方和为评价函数的各种光学自动设计的程序(算法)。书后附有光斑评价函数和膜系自动设计的程序。本书可供光学设计、膜系设计的技术人员参考。

光学自动设计程序汇编

南京大学数学系计算数学专业 编

*

国防工业出版社出版

北京市书刊出版业营业许可证出字第 074 号

新华书店北京发行所发行　各地新华书店经售

上海商务印刷厂排版　国防工业出版社印刷厂印装

*

787×1092 1/16　印张 7　164 千字

1978 年 9 月第一版　1978 年 9 月第一次印刷　印数：00,001－11,000 册

统一书号：15034·1731　定价：0.77 元

目　　录

编 写 说 明

我们在编写了《光学系统自动设计中的数值方法》(以下简称《数值方法》)一书之后，就着手编制有关算法的程序，并进行了计算。现将各程序编辑成册，供光学界参考使用。

为了便于读者阅读和使用，特作如下的说明：

1．对各程序(两个附录中的程序除外)通用的标识符，在这里一并说明如下：

ml	系统的面数(不包括光阑和象面)
$ml1$	$ml+1$(即将象面包括进去的面数)
mp	光阑前面的面数
n	可改变的结构参数的个数
nr	可改变的结构参数中曲率半径的个数
m	象差的个数
$nfun$	计算象差的过程被调用的次数的计数器，每调用一次它加 1
$u0$	孔径角
$w0$, w	视场角($u0=0$)或物高($u0\neq 0$)，其中 $w0$ 为输入时的存贮单元
h	入瞳半径，当物在有限远时作为输入物距的存贮单元
$l0$	物距
ql	光阑到它前面一面的距离
ee	挑选视场的乘常数
ea	挑选孔径的乘常数
$csu0$	$\cos(u0)$
$tgu0$	$\mathrm{tg}\,(u0)$
$ci2$	$\cos^2 i$
upl	象方孔径角 u'
lpl	最后一面到理想象面的距离 l'
foc	焦距，当物在有限远时表示横向放大率
eps	小量 ε
$ep1$	小量 $\varepsilon 1$
$ep2$	小量 $\varepsilon 2$
r	存放曲率半径的数组
c	存放曲率的数组(在有些程序里，它在输入时作为存放曲率半径之用，而数组 r 就不用了)
d	存放间隔和厚度的数组
bnd	存放 D 光折射率 n_d 的数组
bnc	存放 C 光折射率 n_c 的数组
bnf	存放 F 光折射率 n_f 的数组

dn　　存放 n_f-n_c 的数组

s、$s0$　　存放各象差的数组

sp　　存放各象差目标值的数组

$w1$　　存放各象差的权因子的数组(不等式法和适应法除外)

v　　存放用于指定结构参数 r, d 中可变者之序号的整型数组

对程序非通用的标识符将在各程序的前面另作说明。

2．每个程序的前面都附有算法的概述，对算法作简单的介绍，而算法的详细叙述请查阅《数值方法》一书，但是在《数值方法》中对直交化方法的叙述，在计算中发现有问题，故该方法的叙述以本书为准。

3．关于初始数据的准备，如果以各种象差作为评价标准，在程序开头要输入的有下列各量：

① 简单变量 ml、mp、$w0$ (物在无穷远时为 0)、$w0$ (物在无穷远时为视场，物在有限远时为物高)、h(物在无穷远时为入射光的高度，物在有限远时为物距)、ee、ea、ql、n、nr、m 为各方法程序均要输入的量，另外根据各方法的不同要求输入某些有关的量，例如小量 ε。

② 数组 $r[1:ml1]$：各面的初始曲率半径，输入时按顺序第一面为 $r[1]$，而象面作为最后一面放在 $r[ml1]$，如果某一面是平面，例如象面就是平面，则输入的数据写成 0。

③ 数组 $d[1:ml1]$：$d[1]$ 为入瞳到第一面的距离，$d[ml1]$ 为象距 l'，它们是在程序中算出来的，输入时可为任意数，其余 $d[2]\sim d[ml]$ 输入时顺序为各面间的初始厚度或间隔。

④ 数组 bnd、bnc, $bnf[1:ml1]$：$bnd[1]$ 应是第一个透镜对 D 光的折射率，其后按顺序为空气或透镜对 D 光的折射率，$bnd[ml]$ 和 $bnd[ml1]$ 都应该是 1，因为在象面的两侧皆为空气。对 bnc、bnf 的数据也是如此安排。

⑤ 数组 $v[1:n]$：前 nr 个为 $r(c)$ 中可变者的下标(从小到大)，后 $n-nr$ 个为 d 中可变者的下标(从小到大)。

⑥ 数组 sp, $w1[1:m]$ 也要按各象差的顺序输入，它们的顺序请参阅本书第一章。

4．书中所有方法的程序均以象差加权平方和作为评价函数，另将光斑评价函数作为附录，供读者参考，如用它来评价时，则要更换程序中有关计算评价函数的过程，并对主程序作适当的改造。

5．膜系的自动设计问题也编制了一部分程序，试算的效果较好，故也收入本书作为附录。

6．我们用 $*w1$ 表示输入，$*w2$ 表示输出，因为输入、输出的格式比较繁琐，而且完全可以自行安排，所以将它略去了(只有在过程 $ZHUIJI$ 中有一处例外)，在符号 $*w1$ 或 $*w2$ 之后的括号中所列出的为输入或输出量表。

我们在编制光学系统设计及光学多层膜系设计程序和进行试算过程中，得到中国科学院长春光学精密机械研究所四室、中国科学院南京天文光学仪器厂光学组和计算机房、南京电影机械厂以及北京工业学院等单位的同志们的大力支持和帮助，在此谨表示谢意。

由于我们从事研究光学系统自动设计的工作为时不久，加上水平有限，书中必然会存在一些缺点和错误，热诚地希望读者提出宝贵的意见。

编　者

一九七七年十一月

第一章　计算象差的过程

一、程 序 简 介

我们是采用象差与其目标值之差再加权的平方和来作为一个光学系统的评价函数，即

$$\Phi=\sum_{i=1}^{m}[w_i(s_i-s_{目标i})]^2,$$

其中 w 是权因子，s 表示实际算出的象差，$s_{目标}$ 表示各象差的目标值，m 是象差的个数。

在本程序中权因子和目标值都是在主程序一开始用输入的办法给定的。计算一个象差则用专门的过程——XCH。下面将 XCH 作一简单介绍。

在 XCH 中计算的象差有：7 种初级象差系数(它们为 $S1$, $S2$, $S3$, $S4$, $S5$, $C1$, $C2$)，全孔径和一个带孔径的轴上球差 LA，正弦差 OSC，波色差 OPD，全视场和一个带视场的细光束弧矢场曲 x_s，细光束象散 x_t-x_s，子午彗差 $Coma_T$，轴外球差 LAT 和相对畸变 $Dist$，共 23 种象差。另外为了便于控制焦距和后工作距离，我们将象方孔径角 u' 和理想象面到系统最后一面的距离 l' 也作为象差参加评价。因此在 XCH 中计算象差的个数为 25 个($m=25$)。这 25 个象差在过程中用一个一维数组 s 来存放，并且 s 是作为形式参数出现在过程说明中。s 各分量所代表的象差如下：

$s[1]$——$S1$
$s[2]$——$S2$
$s[3]$——$S3$
$s[4]$——$S4$
$s[5]$——$S5$
$s[6]$——$C1$
$s[7]$——$C2$
$s[8]$——u'
$s[9]$——l'
$s[10]$——LA_1
$s[11]$——OSC_1
$s[12]$——OPD_1
$s[13]$——LA_2
$s[14]$——OSC_2
$s[15]$——OPD_2
$s[16]$——x_{s_1}
$s[17]$——$(x_t-x_s)_1$
$s[18]$——LAT_1
$s[19]$——$Coma_{T_1}$

$s[20]$——$Dist_1$

$s[21]$——x_{s_2}

$s[22]$——$(x_t-x_s)_2$

$s[23]$——LAT_2

$s[24]$——$Coma_{T_2}$

$s[25]$——$Dist_2$

在 XCH 中各种象差的计算步骤大致如下：

1．进行光阑倒描求出理想的入瞳位置。如果光阑是在第一面之前 ($mp=0$)，则跳过这一步；

2．同时追迹两条近轴光线算出 u', l', $S1$, $S2$, $S3$, $S4$, $S5$, $C1$, $C2$；

3．追迹二条实际的轴上光线(一条全孔径，一条带孔径)，算出各自的 LA, OSC, OPD；

4．追迹三条全视场 (带孔径) 的轴外光线 (主光线、上光线和下光线)，算出 x_{s_1}, $(x_t-x_s)_1$, LAT_1, $Coma_{T_1}$, $Dist_1$；

5．追迹三条带视场(全孔径或带孔径)的轴外光线，算出 x_{s_2}, $(x_t-x_s)_2$, LAT_2, $Coma_{T_2}$, $Dist_2$。

这里要说明三点：

1．实际光线都是子午面上的，因此所附的用于追迹实际光线的过程——$ZHUIJI$ 仅仅是子午面上的，只有 x、y 两个坐标。

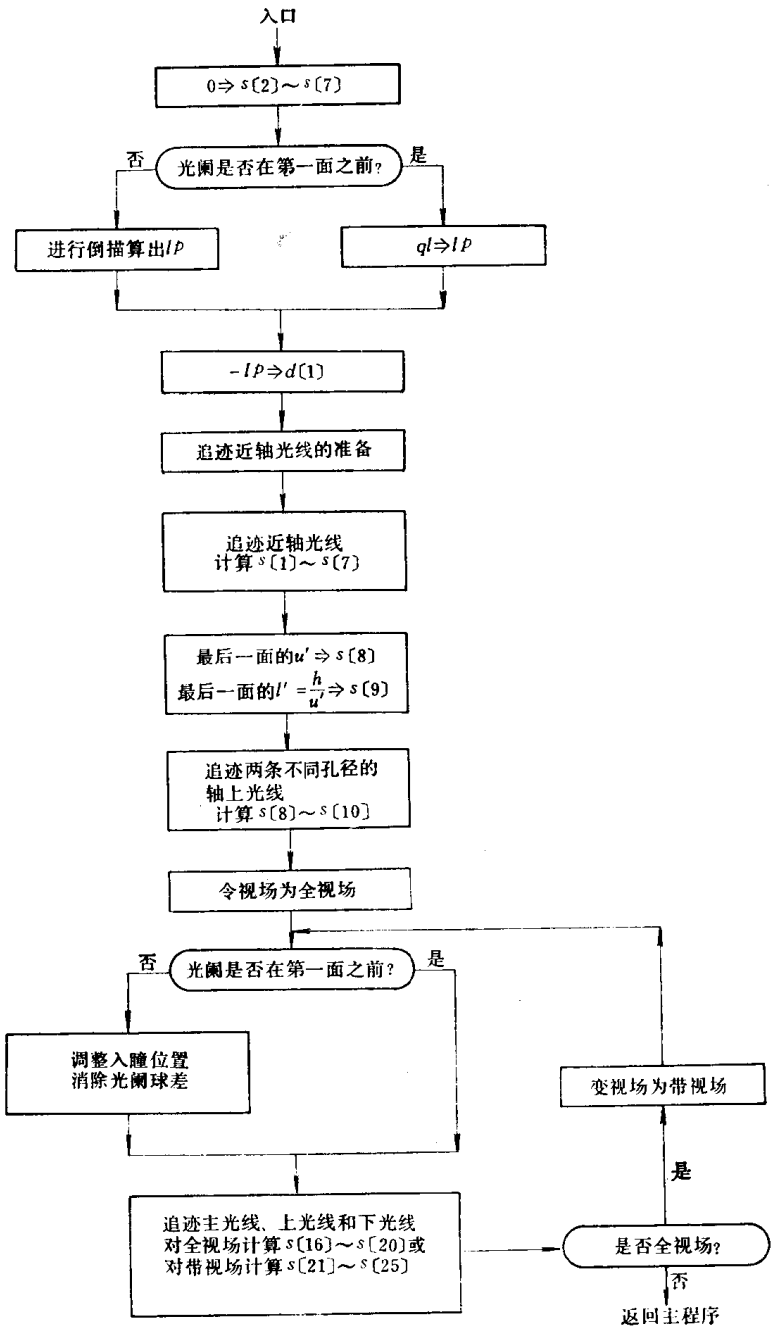
2．在进行一条实际光线的追迹时，如果这条光线在某面通不过，则在过程 $ZHUIJI$ 中安排有信息的打印，然后返回主程序，这时全程量 $ci2$ 为负。

3．程序中有两个全程量 ee 和 ea 是在主程序一开始时通过输入进行赋值，ee 是用于挑选零视场和全视场时的一个带孔径以及一个带视场，ea 用于挑选带视场时的上(下)光线的孔径。一般 ee 取 0.707，ea 如果取 1 时，初始结构不会产生光线通不过的情况，则就取 1，否则取 0.85 或 0.707。

二、主要标识符说明

xk、yk	光线与系统中各面交点的坐标
xa、ya	光线的方向余弦
dd	光线在相继两个面之间的路径
coi	$\cos i$
$cp2$	$\cos^2 i'$
no	界面前面介质的折射率
npl	界面后面介质的折射率
je	拉-赫不变量
jj	第一面到入瞳的距离
$lppl$	最后一面到出瞳的距离
$i1$	第一条近轴光线的入射角
$i2$	第二条近轴光线的入射角
$u1$	第一条近轴光线与 x 轴的夹角

三、框　　图



四、程　　序

```
procedure XCH(s); array s;
begin
  integer k, l;
  real xk, yk, xa, ya, dd, coi, op, cp2, n0, npl, je, jj, lp, la, lppl, i1, i2, u1, u2, h1,
       h2, up, dnpl, dnp0, ck, dk, wa, au, eu, sa, sb, sc, sd, se, ta, tb, xt, ca, cb, ce,
```

```
        sw, cw, tw;
  procedure ZHUIJI (ck, dk);
    value ck, dk; real ck, dk;
  begin
    real et, xd, aa;
    xd:=xk-dk;
    et:=-(xd*xa+yk*ya);
    aa:=(xd*xd+yk*yk-et*et)*ck-2*(xd+et*xa);
    ci2:=xa*xa-aa*ck;
    if ci2<0 then goto LZ1;
    coi:=*sqrt(ci2);
    dd:=et+aa/(xa+coi);
    xk:=xd+dd*xa; yk:=yk+dd*ya;
    et:=n0/npl; cp2:=1-et*et*(1-ci2);
    if cp2≥0 then goto LZ2;
    ci2:=cp2;
LZ1:*w2(0, '6h sin i>1');
    goto LZ3;
LZ2: op:=*sqrt(cp2)-et*coi;
    xa:=et*xa-op*(xk*ck-1);
    ya:=et*ya-op*yk*ck;
LZ3:
  end;
  nfun:=nfun+1;
  for k:=1 step 1 until 7 do s[k]:=0;
  if mp=0 then
    begin lp:=ql; goto LX1 end;
  u1:=-0.5; h1:=-u1*ql; d[1]:=0; npl:=1;
  for k:=mp step -1 until 1 do
    begin
      n0:=npl;
      npl:=if k>1 then bnd[k-1] else 1;
      i1:=-(h1*c[k]+u1);
      u1:=i1*(1-n0/npl)+u1;
      h1:=h1-d[k]*u1
    end;
  up:=-u1; lp:=h1/up;
LX1: d[1]:=-lp; d[ml1]:=0; u1:=u0;
  if u0≠0 then
```

```
    begin
      h1:=l0*u0; cb:=l0-lp;
      h:=cb*tgu0; ca:=cb/w;
      u2:=-*sign(ca)/*sqrt(ca*ca+1);
      je:=w*u0
    end else
    begin
      h1:=h; u2:=tgu0; je:=-h*u2
    end;
  dnpl:=0; jj:=je*je; h2:=lp*u2;
  npl:=1; s[5]:=je*u2*u2;
  for k:=1 step 1 until ml do
    begin
      n0:=npl; npl:=bnd[k];
      ck:=c[k]; dk:=d[k+1];
      sa:=h1*n0; i1:=h1*ck-u1;
      sc:=1-n0/npl; sb:=i1*sc;
      u1:=u1+sb; h1:=h1-dk*u1;
      se:=sa*sb*(i1-u1);
      s[1]:=s[1]+i1*se;
      dnp0:=dnpl; dnpl:=dn[k]/npl;
      ce:=sa*(dnpl-dnp0);
      s[6]:=s[6]+i1*ce; i2:=h2*ck-u2;
      sb:=i2*sc; u2:=u2+sb;
      s[5]:=s[5]+h2*n0*i1*sb*(i2-u2);
      h2:=h2-dk*u2;
      sb:=i2*se; s[2]:=s[2]+sb;
      sd:=jj*sc*ck/n0; s[4]:=s[4]+sd;
      s[7]:=s[7]+i2*ce;
      s[3]:=s[3]+(if i1=0 then -sd else sb*i2/i1)
    end;
  s[5]:=s[5]-je*u2*u2;
  s[8]:=upl:=u1;
  s[9]:=d[ml1]:=lpl:=h1/u1;
  lppl:=h2/u2;
  foc:=if u0≠0 then u0/u1 else h/u1; eu:=1;
  for l:=12, 15 do
    begin
      xk:=s[l]:=0; yk:=h*eu; ya:=-u0*eu;
```

```
        xa:=*sqrt(1-ya*ya); npl:=1;
        for k:=1 step 1 until ml1 do
          begin
            n0:=npl; npl:=bnd[k]; dk:=d[k];
            ZHUIJI(c[k], dk);
            if ci2<0 then goto LXE;
            if k>1 then s[l]:=s[l]+(dk-dd)*dn[k-1]
          end;
        s[l-2]:=la:=yk*xa/ya;
        s[l-1]:=1+eu*upl/ya-la/(lpl-lppl);
        eu:=ee
      end;
      l:=15; wa:=w;
LX2:  if u0=0 then
      begin
        cw:=* cos(wa);
        sw:=-* sin(wa);
        tw:=sw/cw
      end;
    if mp=0 then goto LX4;
LX3:xk:=yk:=0;
    if u0≠0 then
      begin
        sw:=*sign(cb)/*sqrt(cb*cb+wa*wa);
        cw:=cb*sw; sw:=wa*sw
      end;
    xa:=cw; ya:=sw; npl:=1;
    for k:=1 step 1 until mp do
      begin
        n0:=npl; npl:=bnd[k];
        ZHUIJI(c[k], d[k]);
        if ci2<0 then goto LXE
      end;
    n0:=npl; ZHUIJI(0, ql);
    la:=yk*xa/ya;
    if *abs(la)<0.001 then goto LX4;
    lp:=lp+la*ya/(up*sw)*0.5;
    d[1]:=-lp;
    if u0≠0 then
```

```
      begin
        cb:=l0-lp; h:=cb*tgu0
      end;
    goto LX3;
LX4: au:=0;
LX5: xk:=0; yk:=au*h;
    if u0≠0 then
      begin
        tw:=wa-yk;
        sw:=*sign(cb)/*sqrt(cb*cb+tw*tw);
        cw:=cb*sw; sw:=tw*sw
      end;
    xa:=cw; ya:=sw; npl:=1;
    for k:=1 step 1 until ml1 do
      begin
        n0:=npl; npl:=bnd[k];
        ck:=c[k]; ZHUIJI(ck, d[k]);
        if ci2<0 then goto LXE;
        if au≠0 ∨ k=ml1 then goto LX6;
        if k=1 then
          sa:=ta:=if u0≠0 then dd*(l0-xk)/(xk-lp)
                          else_10 16
              else
          begin sa:=sb-dd; ta:=tb-dd end;
        sb:=npl*op*ck;
        tb:=npl*cp2/(sb*ta+n0*ci2)*ta;
        sb:=npl/(sb*sa+n0)*sa;
        if k=ml then
          begin
            xt:=lpl-xk;
            s[l+1]:=sa:=xt-sb*xa;
            xt:=xt-tb*xa;
            s[l+2]:=xt-sa
          end;
LX6: end;
    if au≥0 then
      begin
        if au=0 then
          begin
```

```
            au:=eu; sc:=yk;
            s[l+5]:=1-yk/(foc*tw)
          end else
          begin
            au:=-au; sd:=yk; ta:=ya/xa
          end;
        goto LX5
      end;
    s[l+3]:=(sd-yk)/(ta-ya/xa)-xt;
    s[l+4]:=sc-0.5*(sd+yk);
    if l=15 then
      begin
        l:=l+5; wa:=w*ee; eu:=ea;
        goto LX2
      end;
  LXE:
  end;
```

第二章　阻尼最小二乘法

一、算　　法

光学系统的评价函数，一般写成

$$\Phi(\vec{x})=\sum_{i=1}^{m} f_i^2(x_1, x_2, \cdots\cdots, x_n),$$

式中m表示象差个数，n表示可变的结构参数的个数。

光学系统的自动设计就是求$\vec{x}^*=[x_1^*, x_2^*, \cdots\cdots, x_n^*]^T$使$\Phi(\vec{x}^*)$的极小。为此，取一计算的出发点——初始点$\vec{x}_0$，把象差函数$f_i(\vec{x})$在这个点的附近按泰勒公式展开，并只取线性项。记展开的结果为$\tilde{f}_i(\vec{x})$：

$$\tilde{f}_i(\vec{x})=f_{i0}+\frac{\partial f_{i0}}{\partial x_1}\Delta x_1+\cdots\cdots+\frac{\partial f_{i0}}{\partial x_n}\Delta x_n,\ i=1, 2, \cdots\cdots, m, \tag{1}$$

其中$\Delta x_j=x_j-x_j^0$，$j=1, 2, \cdots\cdots, n$，称为初始结构参数的线性校正量，f_{i0}，$\frac{\partial f_{i0}}{\partial x_j}$分别表示$f_i$及$\frac{\partial f_i}{\partial x_j}$在初始点处的值。

用$\tilde{f}_i$近似表示f_i，并采用向量和矩阵的记号，将评价函数写成

$$\Phi(\vec{x})=(\vec{f}(\vec{x}^0)+A\Delta\vec{x})^T(\vec{f}(\vec{x}^0)+A\Delta\vec{x})。 \tag{2}$$

为了求它的极小点，令其一阶偏导数$\frac{\partial\Phi}{\partial x_j}$等于零，得到$n$个未知量的线性方程组，即最小二乘法的法方程：

$$A^TA\Delta\vec{x}=-A^T\vec{f}=-\vec{g}, \tag{3}$$

其中$\vec{g}$即grad Φ。解这个法方程就得到校正量$\Delta\vec{x}$，将校正量加到初始点上就得到一个新点

$$\vec{x}=\vec{x}^0+\Delta\vec{x}。 \tag{4}$$

这个点可作为下一次迭代的出发点。这就是最小二乘法。由于象差并非结构参数的线性函数，因此不能保证$\Phi(\vec{x})<\Phi(\vec{x}^0)$，而且矩阵$A^TA$的条件数往往很坏，所以要用加阻尼因子的办法来改善矩阵的条件，即将评价函数加上一个阻尼项$p\Delta\vec{x}^T\Delta\vec{x}$：

$$\Psi(\vec{x})=\Phi(\vec{x})+p\Delta\vec{x}^T\Delta\vec{x}, \tag{5}$$

其中p为正数，相应的法方程为

$$(A^TA+pI)\Delta\vec{x}=-\vec{g}。 \tag{6}$$

因为加了p之后将对$\Delta\vec{x}$起控制作用，所以称p为阻尼因子，这个方法就称为阻尼最小二乘法。

阻尼因子如何选取，对阻尼最小二乘法来说是比较重要的。如果取得过小就不能保证评价函数下降；如果取得过大，所得$\Delta\vec{x}$又太小，收敛较慢，所以要取得适当。在计算中，阻尼

因子 p 是试用二次插值的办法来进行自动选取。因为对一个确定的 p 值，通过解法方程(6)就能得到一个 $\Delta\vec{x}$，令 $\vec{x}=\vec{x}^0+\Delta\vec{x}$，就可算出 $y=\Phi(\vec{x})$。于是对同一个 $\vec{x}^0$，y 就是 p 的函数。我们将 y 近似地看作 $\frac{1}{p}$ 的二次函数来进行插值。$y_0=\Phi(\vec{x}^0)$ 相应于 $\frac{1}{p}=0$，再给出两个相异的 p_1，p_2，算出相应的评价函数值 y_1，y_2。在 $\left(\frac{1}{p}, y\right)$ 平面上，如果判明经过 $(0, y_0)$，$\left(\frac{1}{p_1}, y_1\right)$，$\left(\frac{1}{p_2}, y_2\right)$ 的二次曲线是凸的，则算出这个插值曲线的极小点所相应的 p 作为我们优选的 p。在计算中既要求每次迭代所得到的新点其函数值一定要比上一次的小，而且还要选取在这次迭代中所有算出的函数值中之最小者。另外，我们是采用 LDL^T 分解(乔累斯基分解)来解法方程，这是为了在每次迭代中只要算一次微分矩阵 A，既节省计算量又节省内存。

计算步骤：

1．选定初始结构参数 $\vec{x}^0$，并计算相应的评价函数值 $\Phi(\vec{x}^0)$；

2．计算 $\vec{g}(\vec{x}^0)=\mathrm{grad}\,\Phi(\vec{x}^0)$，$B=A^TA(\vec{x}^0)$；

3．如果 $\|\vec{g}(\vec{x}^0)\|<\varepsilon$，则停止计算，否则进入下一步；

4．自动选取 p，并得到一个好的 $\Phi(\vec{x})<\Phi(\vec{x}^0)$。这一步包含多次解方程 $(B+pI)\Delta\vec{x}=-\vec{g}$，从而得 $\vec{x}=\vec{x}^0+\Delta\vec{x}$，并由此计算评价函数值 $\Phi(\vec{x})$。如果 p 变得很大后仍然选不到好的 p 值使 $\Phi(\vec{x})$ 下降，则计算终止；

5．将得到的 $\vec{x}$ 作为 $\vec{x}^0$，转向第 2 步。

二、主要标识符说明

p、$p1$、$p2$	都是阻尼因子
y、$y0$、$y1$、$y2$	都是评价函数值
ni	迭代次数计数器
cx	差商代导数时曲率的增量
dx	差商代导数时间隔或厚度的增量
kp	曲率的阻尼因子与间隔的阻尼因子之间的比例系数
fb, fc	都是存放评价函数各项未平方前之值的数组
A	微分矩阵 A
AA	A^TA
dl	存放矩阵 (A^TA+pI) 的对角线元素的数组
$b1$	法方程的右端(即 $\vec{g}$)
b	法方程的解(即 $\Delta\vec{x}$)
x	新点的坐标
SUM	算评价函数的过程
YFP	给定一个 $p2$ 算出一个 y_2 的过程