基于预处理技术和弧长法的非线性方程通用求解子程序

总结报告

清华大学土木工程系　土博零　陆新征

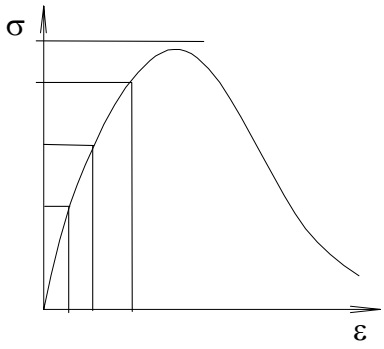
指导教师：江见鲸

2000 年 12 月 7 日

一、 课题研究背景

在结构有限元分析中，经常需要求解非线性方程组

$$F(\Delta x) = \Delta f \qquad (1)$$



这里向量 Δx 为有限元各个节点的位移，Δf 为节点外力。如果材料的力学性能复杂，特别是当有些材料的力学性能与其应力历史有关时，必须求解一系列的方程组 $F(\Delta x_i) = \Delta f_i$，并要求 Δx 和 Δf 都不能过大，以达到“跟踪”结构荷载—位移变化全过程的要求。由于实际计算中的无法预知计算结果，所以当计算接近极值点时，往往因为对节点外力 Δf 的设定不合适而导致求解失败或结果“跳跃”，即 Δx 或 Δf 过大，造成精度的丧失乃至结果的错误。目前结构分析中解决这一问题的方法主要有以下三种形式：

1、 不考虑下降段

当我们只关心峰值点时，当发现计算接近峰值点后，不断减小加载步长，这样，可以得到一个较精确的峰值点，但无法得到下降段。

2、 迭代强制收敛方法。

如果主要想了解的是下降段而不是峰值点的值，则当计算非常靠近极值点时，迭代如果不收敛或收敛很慢。这时就让它强制收敛，即当迭代次数大于预先设定的最大迭代次数 N_{cr} 时,则认为 N_{cr} 次迭代的结果就是非线形方程组的解，进行下一轮计算。这样可能会在峰值点附近精度不高，特别是当加载步长取得比较大的时候。

3、 线形搜索法（本方法是从结构分析软件 SAP-Structure Analysis Program 的说明中看到的，具体方法我也不是很了解）

当计算接近峰值点时，对每次计算的结果进行一个小的折减，使程序自动适应下降段。但如果曲线变化过于突然，或加载步长过大，则本方法失效。

为了达到对结构荷载—位移变化的全过程自动跟踪，我设计了一个基于弧长法的非线性方程通用求解子程序。由于当计算接近极值点时，刚度矩阵往往是病态的，为了提高在极值点附近的精度，在迭代求解过程中，使用了预处理技术，以提高求解的稳定性和精确性。

二、基本原理及方法

2.1、利用弧长法实现对求解过程的自动控制。

2.1.1、基本原理

结构分析的一般方程为 $F(\Delta x_i)=\Delta f_i$，为了达到自动控制的目的，我在该方程组中增加一个变量 λ，变原方程组为

$$F(\Delta x_i)=\lambda\Delta f_i \qquad (2)$$

这时，方程组共有 *n+1* 个未知量：$\Delta x\in R^n$，λ，而方程只有 *n* 个，因此，必须补充条件。

现在令补充条件为

$$\lambda^2\|\Delta f\|_2^2+\|\Delta x\|_2^2=S^2 \qquad (3)$$

则 *S* 即为附加的控制变量，我们称之为“弧长”。这时，方程组变为 $F^{'}\in R^{(n+1)\times(n+1)}$，通过引入 λ，从而实现了对计算过程的自动控制。

2.1.2、具体方法

将方程组（2）写成牛顿迭代形式，即：

$$K(x_k)\Delta x_k=\lambda_k\Delta f+R_k \qquad (4)$$

这里 $K(x_k)$ 即为 $F'(x_k)$，Δx_k 为第 k 次迭代的位移值，λ_k 为第 k 次迭代的 λ 值，R_k 为节点不平衡力。

令 $r=\begin{Bmatrix}\lambda\|\Delta f\|_2\\ \Delta x\end{Bmatrix}$，如图 2 所示，则附加条件（3）可以写成

$$\|r\|_2=S \qquad (5)$$

即在迭代中，i，i+1 点沿一圆弧进行，故称为弧长法。现在的问题是如何确定 $\Delta\lambda^{(i+1)}$ 与 $\Delta u^{(i+1)}$。按图 2 所示，将迭代向量表示为

$$r^{(i+1)}=r^{(i)}+\Delta u^{(i+1)} \qquad (6)$$

代入（5），得到

$$\left(r^{(i)}+\Delta u^{(i+1)}\right)\cdot\left(r^{(i)}+\Delta u^{(i+1)}\right)=S^2 \tag{7}$$

又因 $$r^{(i)}\cdot r^{(i)}=S^2 \tag{8}$$

所以 $$\Delta u^{(i+1)}(\Delta u^{(i+1)}+2r^{(i)})=0 \tag{9}$$

又因为 $$\Delta u^{(i+1)}\cdot\Delta u^{(i+1)}=\left[\Delta x^{(i+1)}\right]^T\left[\Delta x^{(i+1)}\right]+\left(\Delta\lambda^{(i+1)}\left\|\Delta f\right\|_2\right)^2 \tag{10}$$

$$r^{(i)}\cdot r^{(i)}=\left[x^{(i)}\right]^T\left[x^{(i)}\right]+\left(\lambda^{(i)}\left\|\Delta f\right\|_2\right)^2 \tag{11}$$

最后得到方程

$$\left[\Delta x^{(i+1)}\right]^T(\left[\Delta x^{(i+1)}\right]+2\left[x^{(i)}\right]+\Delta\lambda^{(i+1)}\left\|\Delta f\right\|_2(\Delta\lambda^{(i+1)}\left\|\Delta f\right\|_2+2\lambda^{(i)}\left\|\Delta f\right\|_2)=0 \tag{12}$$

将$\left[\Delta x^{(i+1)}\right]$分为两部分：

$$\left[\Delta x^{(i+1)}\right]=\Delta\lambda^{(i+1)}\left[\Delta x^{(i+1)}\right]_I+\left[\Delta x^{(i+1)}\right]_{II} \tag{13}$$

其中：

$$\left[K^{(i)}\right]\left[\Delta x^{(i+1)}\right]_I=\left[\Delta f\right] \tag{14}$$，

$$\left[K^{(i)}\right]\left[\Delta x^{(i+1)}\right]_{II}=\left[R^{(i)}\right] \tag{15}$$

将（13）代入（12），得到关于$\Delta\lambda^{(i+1)}$的二次方程：

$$a\left(\Delta\lambda^{(i+1)}\right)^2+2\Delta\lambda^{(i+1)}+c=0 \tag{16}$$

系数

$$a=1+\left[\Delta x^{(i+1)}\right]_I^T\left[\Delta x^{(i+1)}\right]_I,$$

$$b=\lambda^{(i)}+\left[\Delta x^{(i+1)}\right]_I^T(\left[\Delta x^{(i+1)}\right]_{II}+\left[x^{(i)}\right])$$

$$c=\left[\Delta x^{(i+1)}\right]_{II}^T(\left[\Delta x^{(i+1)}\right]_{II}+2\left[x^{(i)}\right])$$

这样，解方程（16），就可以得到$\Delta\lambda^{(i+1)}$。

为了简化$\Delta\lambda^{(i+1)}$的求解过程，在程序编制过程中，我使用了切平面法，即用垂直于$r^{(i)}$的向量$\Delta u^{(i+1)}$来代替圆弧，即

$$r^{(i)}\cdot\Delta u^{(i+1)}=0 \tag{17}$$

写成矩阵形式为

$$\left[x^{(i)}\right]^T\left[\Delta x^{(i+1)}\right]+\lambda^{(i)}\Delta\lambda^{(i+1)}\left\|\Delta f\right\|_2^2=0 \tag{18}$$

同样将 $\Delta x^{(i+1)}$ 分为两部分，则可以解得

$$\Delta\lambda^{(i+1)} = -\left(\frac{\left[x^{(i)}\right]^T \left[\Delta x^{(i+1)}\right]_{II}}{\left[x^{(i)}\right]\left[\Delta x^{(i+1)}\right]_I + \lambda^{(i)}\left\|\Delta f\right\|_2} \right) \frac{1}{\left\|\Delta f\right\|_2} \qquad (19)$$

与式（13），（16）相比，该式简化了很多。

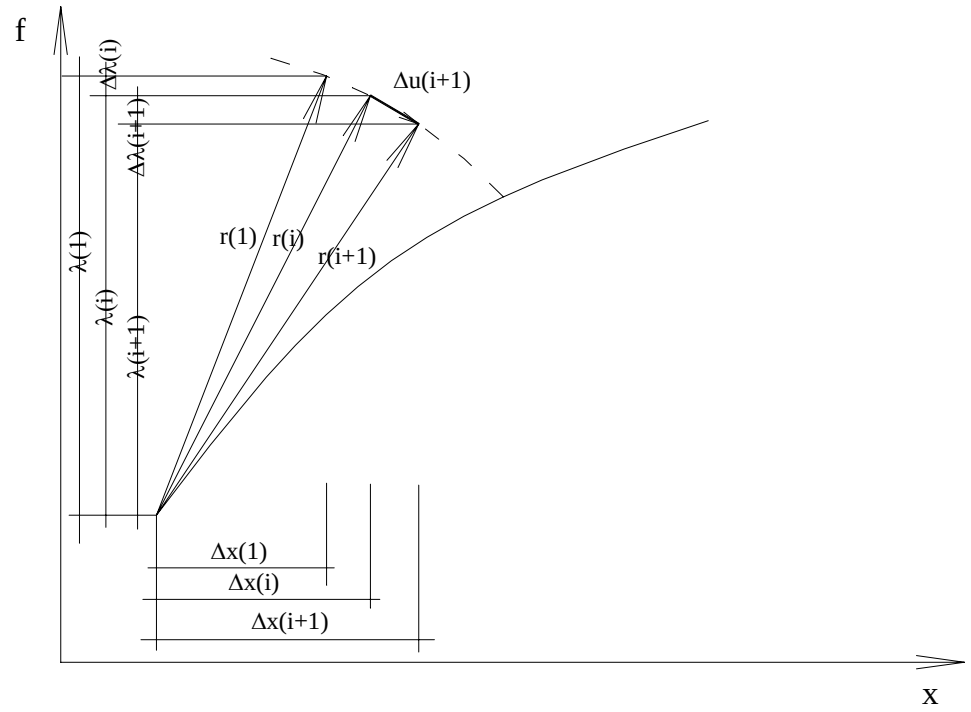


图 2、弧长法迭代过程示意图

在实际计算中，除了要知道 $\lambda^{(i)}$ 的数值以外，还需要知道刚度矩阵的正定性，如果刚度矩阵不是正定的，则节点荷载 $\Delta f_{i+1} = -\Delta f_i$。

2.1.2、 弧长法的具体实现方法

在实际计算中，先进行矩阵正定性判别，利用 Lancsoz 方法，进行两步 Lancsoz 过程，得到相应的 Ritz 值 θ_1，θ_n。因为 $\theta_2 \approx \lambda_n$，$\lambda_n$ 为刚度矩阵最小的特征值。如果 $\theta_2 < 0$，则认为刚度矩阵为非正定。

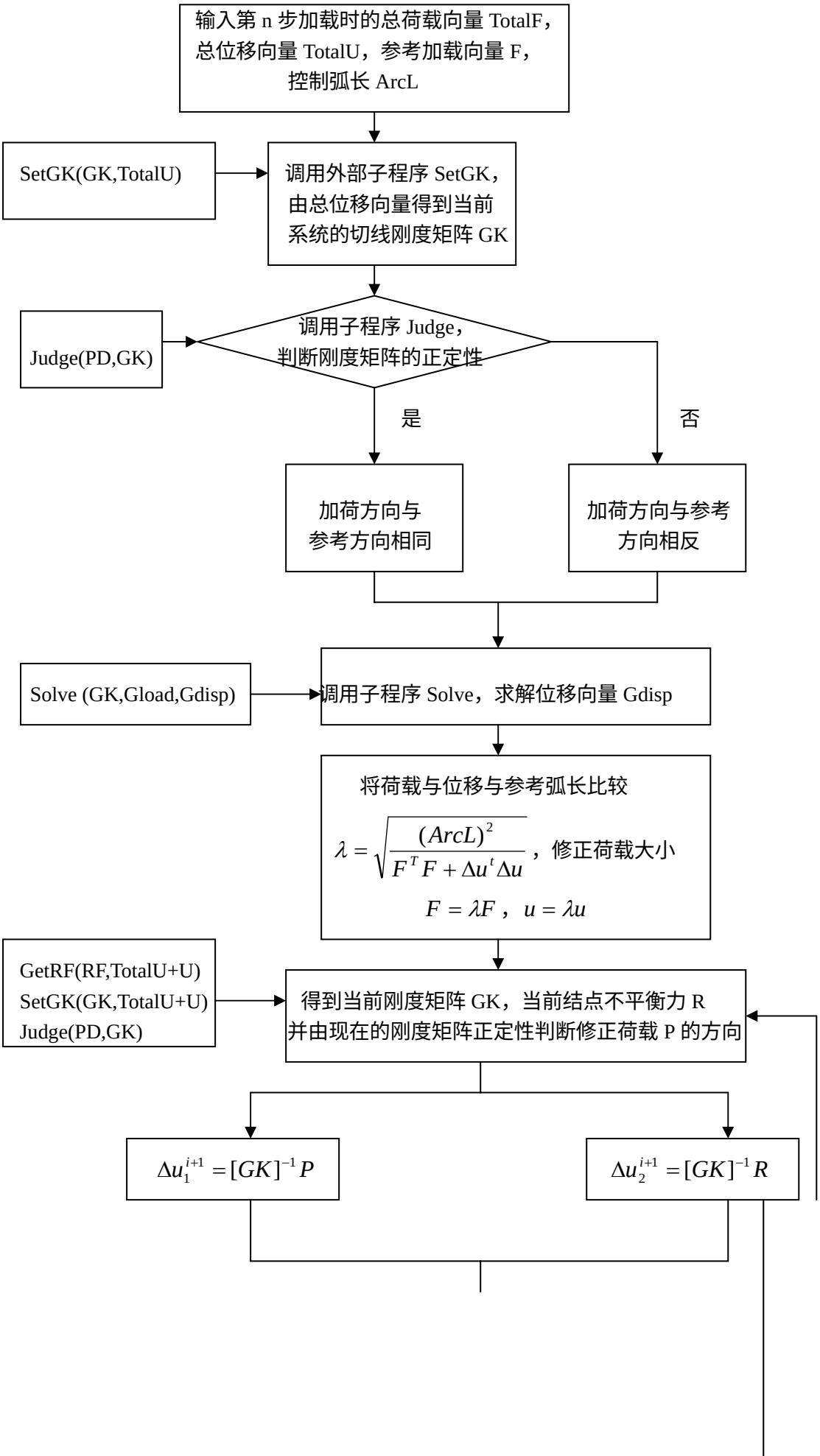
弧长法程序模块由三个子程序组成：

1：subroutine ArcLeng(TotalP,TotalU,F,ArcL)，功能：弧长法迭代核心程序

2：subroutine Judge(PD,GK)，功能：判断刚度矩阵的正定性

3：subroutine Solve(GK,Gload,Gdisp)，功能：求解迭代方程组

子程序 ArcLeng 的流程图如图 3 所示：



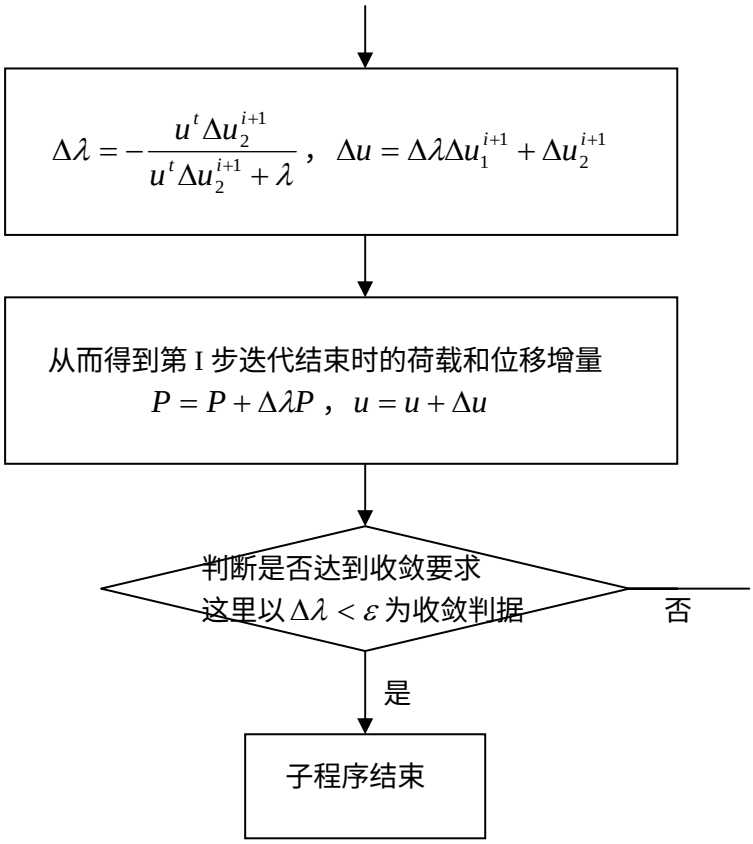


图 3：子程序 ArcLeng 的流程图

子程序 Judge 使用 Lanczos 方法来判断刚度矩阵是否正定。具体做法是：

进行两步 Lanczos 过程，得到矩阵 $B \in R^{2\times 2}$，再得到 B 的特征值 θ_1，θ_2。因为 $\theta_2 \approx \lambda_n$，$\lambda_n$ 为刚度矩阵最小的特征值。如果 $\theta_2 < 0$，则认为刚度矩阵为非正定。

子程序 Solve 使用预处理技术求解线性方程组，如果刚度矩阵奇异，则施加一个小阻尼，此时 $GK = GK + \lambda I$， $\lambda = 10^{-10} \min val(abs(GK))_n$，其特点将在下节讨论。

2.2、利用预处理技术实现对方程组的精确求解

2.2.1、关于使用 Lancsoz 方法求解的尝试（具体方法参见阶段报告一）

在选择求解方法之初，我根据有关资料上的介绍，选定 Lancsoz 方法来求解线性方程组。基本思想是：

1、先用 Lancsoz 方法处理矩阵，将矩阵转化为三对角阵。

2、对三对角阵进行预处理，使其无穷范数相等

3、求解预处理后的三对角阵。

在实际分析中发现，利用预处理子程序 PreCon 的确可以有效降低矩阵的条件数，对三对角矩阵效果似乎更明显。但是，同时也应注意到，由于 Lanczos 过程计算中不可避免的带来数值上的误差，当矩阵的条件数特别大时，预处理子程序的效果往往会被 Lanczos 过程的

数值误差“吃掉”，最后没有出现有些文献中声称的那样明显的效果，因此，在这个预处理方法上，还需要一些讨论。以 Hilbert 矩阵为例，在双精度下，当矩阵 n>5 时，预处理 Lanczos 的结果精度对普通高斯消去已经没有什么优势了。

因此，我认为基于 Lancsoz 方法求解的尝试没有取得预想效果，需要用新的方法代替。

2.2.2、A_3 预处理技术

取预处理矩阵 $M=A_3$，即取原矩阵的三条主对角线元素为预处理矩阵，对原矩阵进行预处理。为了比较其效果，这里以 Hilbert 矩阵为例，比较其效果。

方程 $Hx = b$， H 为 Hilbert 矩阵，分别用普通高斯消去法和预处理后的消去法得到结果 $\tilde{x}$，记误差为 $\left\|b - A\tilde{x}\right\|_2$，结果如表 1 所示：

表 1、预处理效果比较

矩阵的阶数	未处理结果的误差	预处理结果的误差
10	1.164153218269348E-010	1.091393642127514E-010
20	2.607703208923340E-008	1.117587089538574E-008
30	1.981854438781738E-006	9.313225746154785E-008

可见，预处理具有一定的效果。而且此方法的代价并不大，可以作为一个实用方法使用。

2.2.3、线性方程组求解

由于计算中实际的刚度矩阵都是稀疏的，因此，我在这里直接使用数学软件包 IMSL 中的稀疏矩阵求解子程序 DLSLXG 作为预处理后刚度矩阵的求解子程序。经比较发现，DLSLXG 的解题速度比直接编程计算的速度快一倍以上，当矩阵特别大时，效果更加明显。

三、具体算例

为了验证该程序的有效性，我特选择了一个经典的非线性结构分析题目，一个串联双弹簧系统进行了分析。系统如图 4 所示：

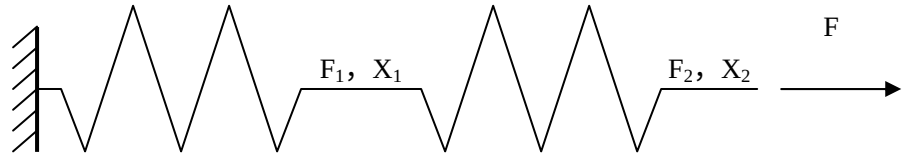


图 4：双弹簧系统

此系统在数学上为一二元非线性方程组，未知量为 X_1,X_2，刚度矩阵 $GK \in R^{2\times 2}$。

每根弹簧的“力-位移”关系为

$\frac{F^i}{F_0^{\ i}} = [0.5\left(\frac{x^i}{x_0^{\ i}}\right)^3 - 2.25\left(\frac{x^i}{x_0^{\ i}}\right)^2 + 2.75\left(\frac{x^i}{x_0^{\ i}}\right)]$，如图 5 所示。

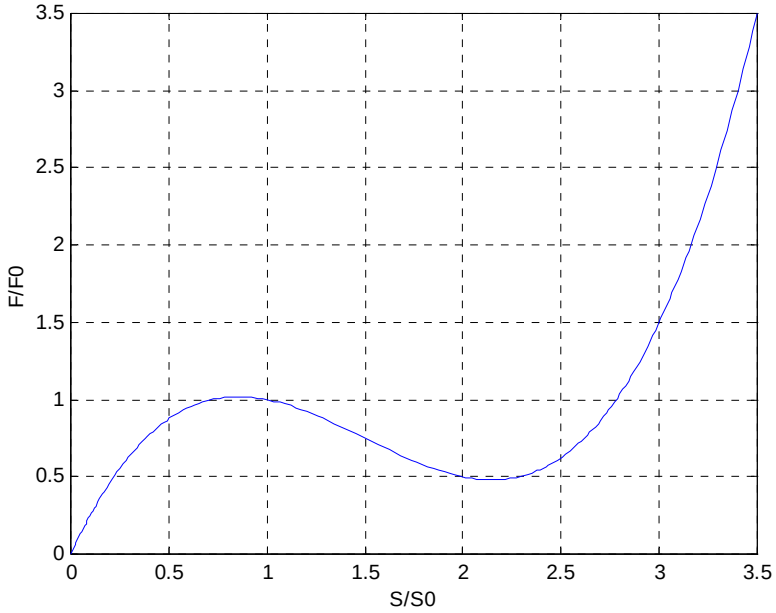


图 5：弹簧的力-位移关系

设第一根弹簧的屈服力 $F_0^1 = 2F_0^2$，在第二根弹簧端部施加荷载，得到的荷载（F）-位移(X_2)全曲线如图 6 所示。

由图 6 可以看出，本系统在求解时，具有相当的复杂性。出现了 4 个极值点，且导数符号也变化了 4 次。一般的牛顿法或位移控制法均失败或求解效率很低。但利用本程序。可以迅速稳定的得到结果。迭代次数一般只需 2-3 次（极值点附近需要 6-7 次）就可以达到 10^{-7} 精度，因此，本程序具有较好的使用用途。

为了比较初始参考弧长对计算的影响，我又改变弧长进行了一系列的试算，结果如图 7 所示。可以看出，当参考弧长大与一定限度时，其对整个求解过程的控制能力减弱，结果出现“ 跳跃 ”。（如图 7 中 3、4、5 组曲线）。因此，选择合适的初始参考弧长，使计算效率和精度上有一个较好的平衡点，对弧长法有重要意义。

图 8 比较了各个不同的线性方程组求解程序对结果的影响，应用了预处理技术后的求解子程序在峰值点附近有更好的性能，参考弧长对极值附近的控制更加稳定有效。（注，为了便于比较，将三条曲线绘于一张图内。为了清楚起见，第二，第三条曲线分别平移了 0.25，0.5。实际上三条曲线几乎完全相同。）

图 9 为现在常用结构计算程序的结果与本程序计算结果的对比，取相同的加载步长。可见，本方法对整个过程计算效果最好。不但完整的反映出了整个变化过程，精度也相当的高。

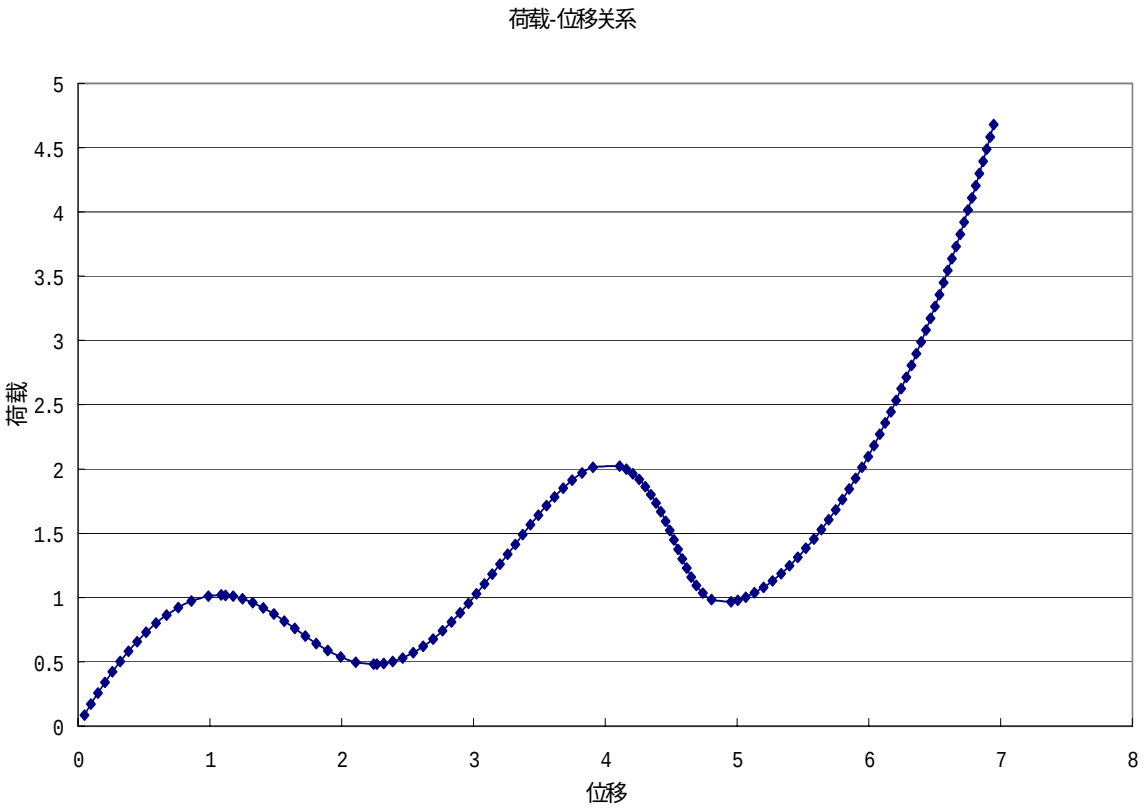


图 6：荷载，位移全曲线

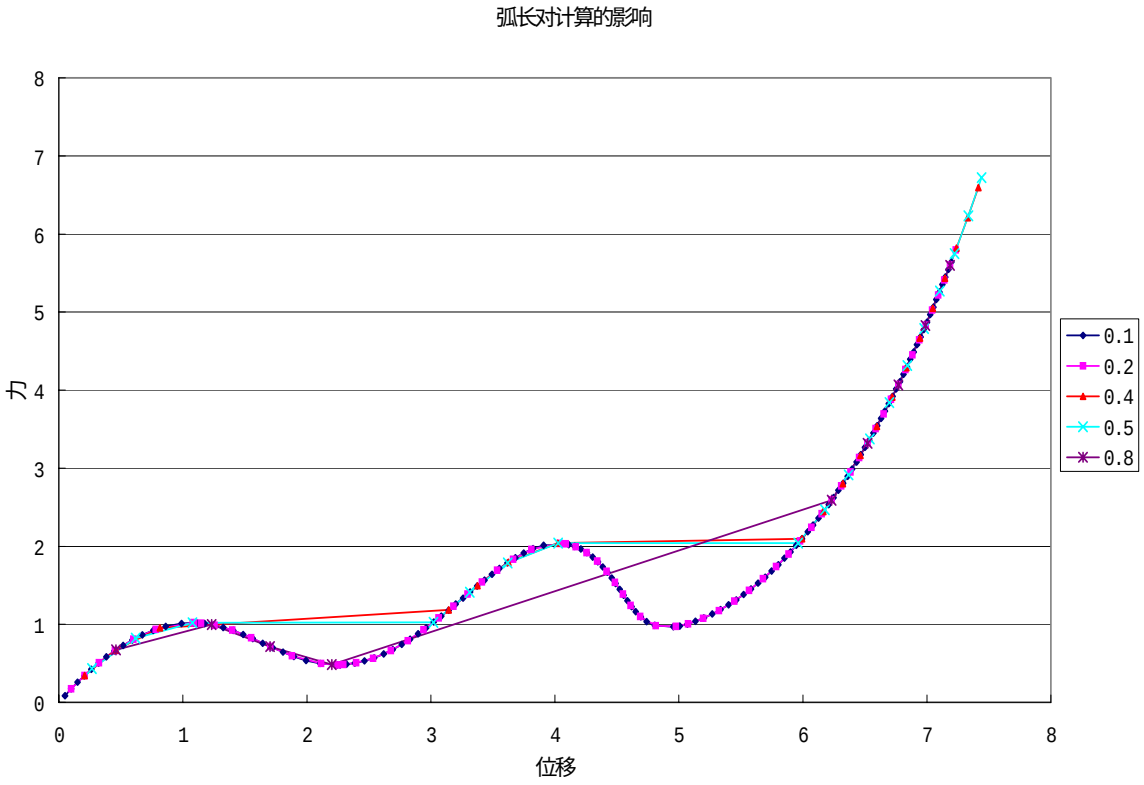


图 7：弧长对计算的影响

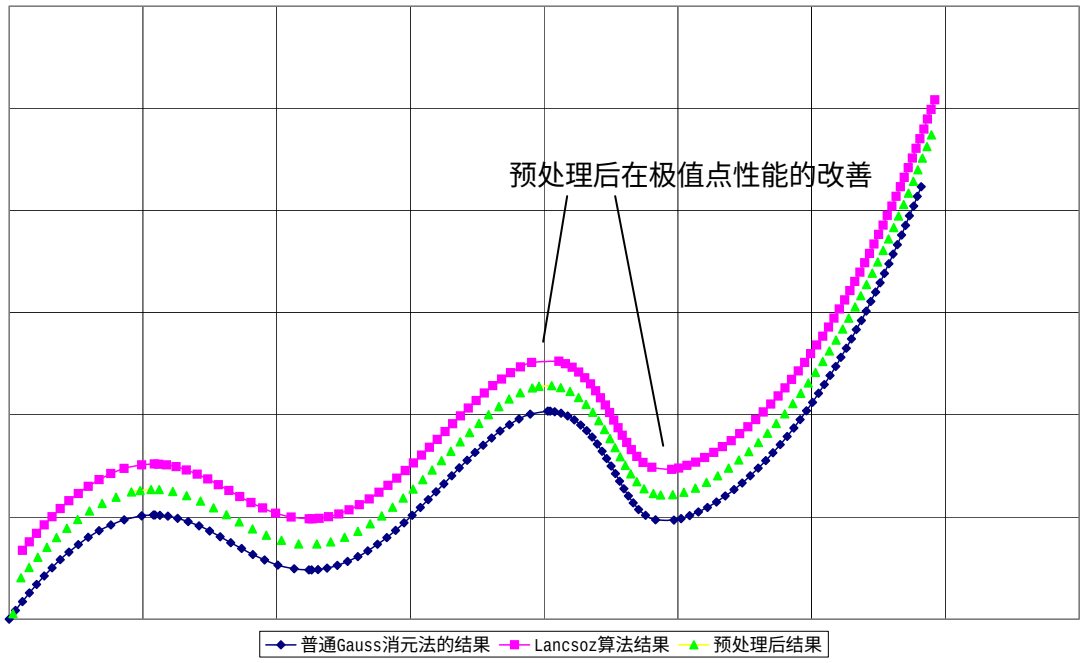


图 8、各种方程求解方法的比较的结果比较

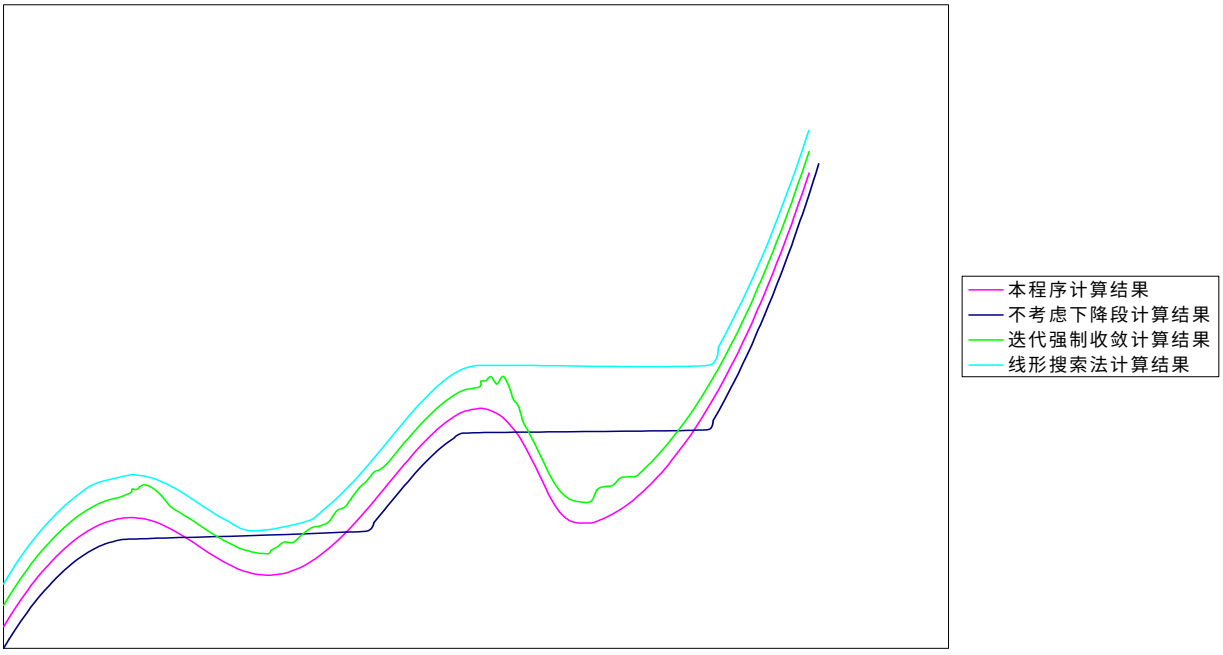


图 9 计算结果对比

另外，还有以下点需要说明：

1、 这里列出的其他方法的结果，都是用结构计算软件包得到的。其当迭代计算次数相同时，这些软件包的计算速度比我的程序要高一些（一般用时为我的程序的80%-90%），我认为，这是这些商品软件开发时进行了细致的优化的结果。

2、 如果计算时步长取得很小，线形搜索法也可以得到与本程序相同的结果，但是，其计算工作量几乎要大一倍（在计算本题时，为了得到同等精度的结果，本程序用了

150 步，线形搜索法用了超过 300 步，计算时间也要长不少）。

3、 实际计算中，在结构刚度变化平缓的阶段，可以适当放大计算步长，也就是说，本程序还有进一步优化的潜力。

四、 结论与讨论

1、弧长法作为求解非线性方程的一种有效方法，一直受到人们的关注和重视。提出了包括球面弧长法，柱面弧长法，切平面弧长法，常数线性化法等许多方法。本文利用弧长法中较简便的切平面法，结合结构计算问题特点稍加改进，并使之通用化，得到了较好的结果。

2、Lanczos 方法尽管在理论上具有较好的数值稳定性，但是，经过我的实践，我感觉此方法在求解线性方程组上的优势并不是很明显。还需要更强的预处理技术配合，才能充分发挥其优势。

3、但是，从以上分析也可以看出，由于在结构分析中经常要分析矩阵的正定性，因此，利用 Lanczos 过程求得矩阵最小特征值的近似值，进而判断矩阵的正定性，仍然使一个十分有效的方法。

4、在采用了 A_3 预处理技术和稀疏矩阵求解子程序后，线性方程组求解的速度和精度都得到了一定的提高，

5、我们可以认为，本程序的开发是成功的。在解决复杂问题时，有着比较明显的优势。如果能够加入对参考弧长的自动选取，则可以进一步提高本程序的效率。