非线性方程组的解法

江见鲸 陆新征
清华大学土木工程系
2005

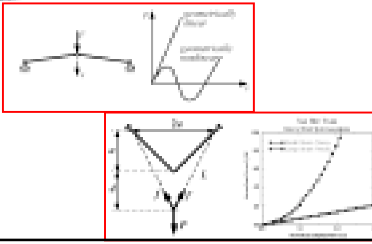
非线性问题分类

- 几何非线性
- 材料非线性
- 边界非线性

Nature is nonlinear;

the best design use nature as an ally

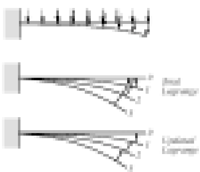
几何非线性



大应变问题

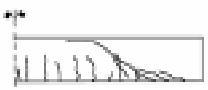
- 小应变 $\varepsilon = \frac{\partial u}{\partial x}$
- 大应变 $\varepsilon = \frac{\partial u}{\partial x} + \boxed{\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial v}{\partial x}\right)^2}$

大位移问题



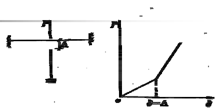
材料非线性

- •塑性
- •非线性弹性
- •粘弹性
- •粘塑性
- •断裂
- •损伤
- •徐变

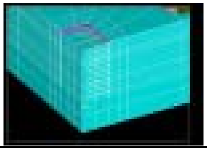


边界非线性

- 接触非线性



- 单元非线性



非线性方程（组）求解

- 每个非线性有限元问题，都包含有两类非线性方程（组）的计算
- 单元刚度矩阵
- 整体刚度矩阵

单元刚度矩阵

- 一般，已知 ${}^{t}F^{e}$ ${}^{t}\delta^{e}$ $\Delta\delta^{e}$
- 求 ${}^{t+\Delta t}F^{e}$

$$\left\{{}^{t+\Delta t}F^{e}\right\} = \left\{{}^{t}F^{e}\right\} + \left[\mathbf{K}^{e}\right]\left\{\Delta\delta^{e}\right\}$$

$$\left[\mathbf{K}^{e}\right] = f\left({}^{t}F^{e}, {}^{t}\delta^{e}, \Delta\delta^{e}\right)$$

- 对于仅材料非线性

$$\left[\mathbf{K}^{e}\right] = \int \mathbf{B}^{T}\mathbf{D}\mathbf{B}\mathrm{d}V = \int \mathbf{B}^{T} f({}^{t}\sigma^{e}, {}^{t}\varepsilon^{e}, \Delta\varepsilon^{e})\mathbf{B}\mathrm{d}V$$

整体刚度矩阵

- 已知 ${}^{t}F, {}^{t}\delta, \Delta F$
- 求 $\{{}^{t+\Delta t}\delta\} = \{{}^{t}\delta\} + [\mathbf{K}]^{-1}\{\Delta F\}$

$$[\mathbf{K}] = \sum [\mathbf{K}^{e}]$$

一般计算过程

- 形成单元刚度矩阵
- 计算整体刚度矩阵
- 计算位移增量
- 计算单元应变增量
- 计算单元应力增量（非线性求解1）
- 计算总内力
- 计算不平衡内力

非线性求解2

非线性方程组的解法

- 基本思路：
  分段线性化方法，将荷载划分成很多小步，逐步施加
- 具体操作方法：
  - 显式求解法（增量法）
  - 隐式求解法（迭代法）

显式求解法

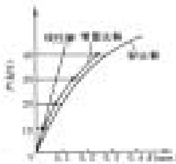
- 将荷载分成若干小步，逐步施加
- 认为在每个小步中，结构是线性的，同一荷载步的刚度矩阵相同
- 不同荷载步的刚度矩阵可以不同
- 用一系列的折线去近似曲线

欧拉折线法(Forward Euler Method)

$$[P_n] = \sum [\Delta P_i]$$

$$[\delta_n] = [\delta_{n-1}] + [\Delta\delta_n]$$

$$[\Delta\delta_n] = [K_{n-1}]^{-1}[\Delta P_i]$$



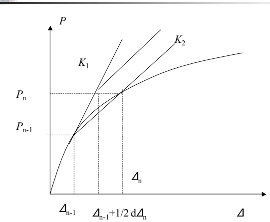
计算步骤（对于单元刚度矩阵）

- 已知当前应力、应变
- 根据当前应力应变求当前的切线刚度
- 根据当前切线刚度求单元刚度矩阵
- 根据单元刚度矩阵和应变增量求应力增量

计算步骤（对于整体刚度矩阵）

- 已知当前单元的切线刚度矩阵
- 形成整体刚度矩阵
- 根据荷载增量求位移增量

修正的欧拉折线法(Mid-point Method)



单元刚度矩阵

- 已知应力，应变，应变增量 $[\mathbf{K}_{n-1}]$
- 根据当前应力应变求切线刚度矩阵
- 求中点应力 $\{\sigma_n'\} = \{\sigma_{n-1}\} + [\mathbf{K}_{n-1}]\frac{\{\mathrm{d}\varepsilon_n\}}{2}$
- 根据中点应力和应变 $\{\sigma_n'\}$，$\{\varepsilon_{n-1}\} + \frac{\{\mathrm{d}\varepsilon_n\}}{2}$ 求此时的切线刚度矩阵 $[\mathbf{K}_{n-1}']$
- 得到最终单元应力 $\{\sigma_n\} = \{\sigma_{n-1}\} + [\mathbf{K}_{n-1}']\{\mathrm{d}\varepsilon_n\}$

整体刚度矩阵

- 已知当前荷载P_{n-1}，位移Δ_{n-1}，荷载增量dP_n
- 根据当前荷载和位移求得当前切线刚度矩阵$[K_{n-1}]$
- 根据当前切线刚度矩阵求解当增加$1/2dP_n$增量时的位移增量$d\Delta_n'$
- 根据$P_{n-1}+1/2dP_n$和$\Delta_{n-1}+d\Delta_n'$求得此时切线模量$[K_{n-1}']$
- 总位移增量为： $d\Delta_n=[K_{n-1}']^{-1}dP_n$

评述

- 显式求解法（增量法）的误差是逐步累积的
- 通过减小步长的方法可以提高精度
- 不需要迭代，不存在收敛问题
- 一些研究者推荐在单元刚度矩阵部分采用显式求解，在整体刚度矩阵部分采用隐式求解

隐式方法

- 根据计算方法
  - 割线刚度法
  - 切线刚度法（Full Newton-Raphson)
  - 等刚度法
  - 改进的切线刚度法(Modified NR)
- 同样有单元刚度矩阵和整体刚度矩阵的迭代问题。

割线刚度法

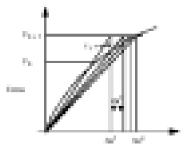
$$[\delta_1]=[K_0]^{-1}[P]$$

$$[\delta_1]\to[K_1]$$

$$[\delta_2]=[K_1]^{-1}[P]$$

……

$$[\delta_n]=[K_{n-1}]^{-1}[P]$$



评述

- 概念简单，对于比例加载的全量模型尤其适用
- 当出现卸载或往复荷载时可能不适合

切线刚度法(Full NR)

$$[\delta_1]=[K_0]^{-1}[P]$$

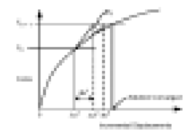
$$[\delta_1]\to[P_1^{\text{internal}}]\to[K_1]$$

$$[\Delta P_1]=[P]-[P_1^{\text{internal}}]$$

$$[\Delta\delta_2]=[K_1]^{-1}[\Delta P_1]$$

$$[\delta_2]=[\delta_1]+[\Delta\delta_2]$$

……



评述

- 理论上收敛速度最快
- 每次迭代都要形成刚度矩阵
- 当接近屈服时，或者出现软化时，切线刚度矩阵斜率特别小甚至不是正定矩阵，往往带来收敛的困难

等刚度法

$$[\delta_1]=[K_0]^{-1}[P]$$

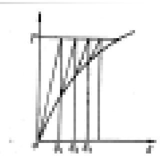
$$[\delta_1]\to[P_1^{\text{internal}}]$$

$$[\Delta P_1]=[P]-[P_1^{\text{internal}}]$$

$$[\Delta\delta_2]=[K_0]^{-1}[\Delta P_1]$$

$$[\delta_2]=[\delta_1]+[\Delta\delta_2]$$

……

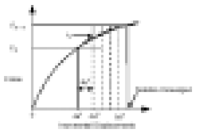


评述

- 收敛速度相对较慢
- 不需每次都形成新的刚度矩阵
- 在切线刚度矩阵奇异时效果有时要优于切线刚度法

改进的切线刚度法(Modified NR)

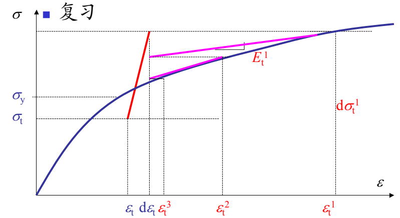
- 不是每次迭代都改变刚度矩阵，而是迭代一定次数后，再改变刚度矩阵



清华大学研究生课程——《钢筋混凝土有限元》

弹塑性分析的应力更新算法

- 复习



清华大学研究生课程——《钢筋混凝土有限元》

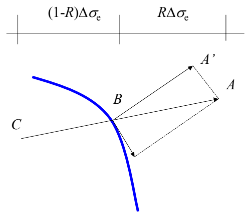
应力更新显式算法

- 计算步骤
  - 上一步单元已经屈服，加载后仍未屈服，使用弹性本构矩阵D_e
  - 上一步单元已经屈服，加载后仍然屈服，使用弹塑性本构矩阵D_{ep}
  - 上一步单元处于弹性，加载后单元屈服，处于过渡状态，使用加权组合方法$D=mD_e+(1-m)D_{ep}$

$$m=\frac{\varepsilon_s-\bar{\varepsilon}_{k-1}}{\bar{\varepsilon}_k-\bar{\varepsilon}_{k-1}}$$

清华大学研究生课程——《钢筋混凝土有限元》

图示



清华大学研究生课程——《钢筋混凝土有限元》

应力更新隐式算法

- Backward Euler 算法

(输入 $\sigma_{ij}^{n-1},\varepsilon_{ij}^{p\,n-1},\Delta\varepsilon_{ij}^{n}$)

首先计算弹性应力增量: $\sigma_{ij}^{t}=\sigma_{ij}^{n-1}+D_{e,ijkl}\Delta\varepsilon_{kl}^{n}$

判断是否屈服: $f_A^p=F^p(\sigma_{ij}^t,\varepsilon_{ij}^{p\,n-1}),\Delta\lambda_A=0$

如果大于屈服应力. $f_A^p>0$

计算屈服面法线方向: $m_{ij}=\dfrac{\partial G^p(\sigma_{ij}^t)}{\partial\sigma_{ij}}$

返回屈服面: $F^p(\sigma_{ij}^t-\Delta\lambda_B D_e\, m_{ij},\varepsilon_{ij}^{p\,n-1})=0 \;\Rightarrow\; \Delta\lambda_B$

清华大学研究生课程——《钢筋混凝土有限元》

积分步骤

判断是否屈服: $f_B^p=F^p(\sigma_{ij}^t-\Delta\lambda_B D_e\, m_{ij},\varepsilon_{ij}^{p\,n-1}+\Delta\lambda_B m_{ij})$

迭代计算 (i) 直至 $\left|\Delta\lambda_A-\Delta\lambda_B\right|<e$

计算新的塑性增量: $\Delta\lambda=\Delta\lambda_A-f_A^p\dfrac{\Delta\lambda_B-\Delta\lambda_A}{f_B^p-f_A^p}$

计算新的投影方向: $m_{ij}^{(i)}=\dfrac{\partial G^p(\sigma_{ij}^t-\Delta\lambda D_e\, m_{ij}^{(i-1)})}{\partial\sigma_{ij}}$

计算屈服函数: $f^p=F^p(\sigma_{ij}^t-\Delta\lambda D_e\, m_{ij}^{(i)},\varepsilon_{ij}^p+\Delta\lambda m_{ij}^{(i)})$

修正迭代变量:

$f_B^p<0 \;\Rightarrow\; f_B^p=f^p,\;\; \Delta\lambda_B=\Delta\lambda$

$f_B^p\geq 0 \;\Rightarrow\; f_A^p=f_B^p,\;\; \Delta\lambda_A=\Delta\lambda_B,\;\; f_B^p=f^p,\;\; \Delta\lambda_B=\Delta\lambda$

迭代结束

清华大学研究生课程——《钢筋混凝土有限元》

收敛标准

- 判断对象
  - 力收敛标准
  - 位移收敛标准
  - 能量收敛标准
- 判断标准
  - 相对误差
  - 绝对误差
- 范数
  - 无穷范数
  - 一范数
  - 二范数

清华大学研究生课程——《钢筋混凝土有限元》

范数

- 无穷范数 $\|V\|_\infty=\max|V_i|$
- 一范数 $\|V\|_1=\sum|V_i|$
- 二范数 $\|V\|_2=\sqrt{\sum V_i^2}$

清华大学研究生课程——《钢筋混凝土有限元》

相对误差和绝对误差

- 相对误差

$$e=\frac{\|\delta V\|}{\|V\|}$$

- 绝对误差

$$e=\frac{\|\delta V\|}{\text{constant}}$$

力收敛标准

- 一般以不平衡力作为分析对象，以节点反力或荷载作为参考标准

$$\frac{\|节点不平衡力\|}{\|支座反力或荷载\|}<误差容限$$

- 一般推荐使用2范数，也有使用无穷范数的（MARC）

位移收敛标准

- 一般以位移增量为分析对象，以同级荷载作用下总节点位移为参考标准

$$\frac{\|某次迭代位移改变量\|}{\|同级荷载节点总位移\|}<误差容限$$

- 一般使用无穷范数

能量收敛标准

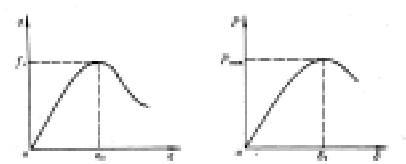
- 一般以某次迭代的应变能增量为分析对象，以同级荷载作用下总应变能为参考标准

$$\frac{\|某次迭代应变能改变量\|}{\|同级荷载总应变能\|}<误差容限$$

注意事项

- 当结构出现屈服时，或者接触分析中发生分离时，往往荷载变化很小而位移变化很快，这时位移标准要好于力标准
- 当结构出现跳跃时，或者接触刚刚发生时，往往位移变化很小和荷载迅速增大。这时力标准要好于位移标准
- 当结构发生分离或者卸载到零时，往往外力已经非常小（几乎为零）。这时使用相对误差判断往往会进行很多不必要的计算，应使用绝对误差判断。

考虑负刚度的计算方法



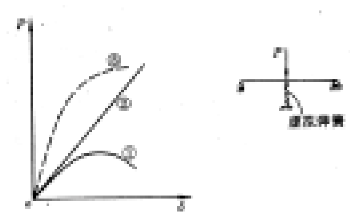
负刚度带来的主要问题

- 刚度矩阵不再是正定矩阵，很多迭代算法的收敛性不再得到保证
- 刚度矩阵存在奇异点（切线刚度为零），求逆可能失败
- 出现“分叉点”现象，在分叉点两侧来回跳动无法收敛
- 出现损伤集中，应变集中问题

解决负刚度问题的主要思路

- 人为引进一些刚度元素，使整体刚度矩阵仍然保持或接近正定
  - 虚拟弹簧法
- 采取某种措施使程序能够自动降低荷载大小
  - 位移控制法
  - 弧长法
- 放松在极值点附近的收敛检查
  - 增量法
  - 强制迭代法

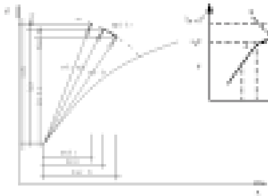
虚拟弹簧法

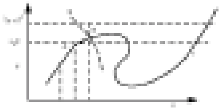


位移控制法

- 以位移来加载
- 控制关键节点的位移变化来决定施加荷载的大小
- 使用Lagrange法或罚函数法来控制指定节点位移
  - 注：Lagrange法可能会导致刚度矩阵不对称
  - 罚函数法可能会增大矩阵病态

弧长法





特点

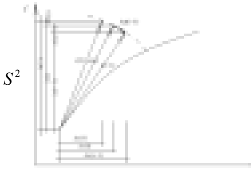
- 同时控制位移和荷载增量，能较好地跟踪整个结构的反应
- 比较适用于几何非线性问题，特别是稳定问题

基本方程

$$F(\Delta\mathbf{x}) = \Delta\mathbf{f}$$

$$F(\Delta\mathbf{x}) = \lambda\Delta\mathbf{f}$$

$$\lambda^2\|\Delta\mathbf{f}\|_2^2 + \|\Delta\mathbf{x}\|_2^2 = S^2$$

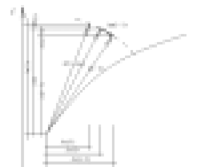


计算过程

$$\mathbf{K}(\mathbf{x}_k)\Delta\mathbf{x}_k = \lambda_k\Delta\mathbf{f} + \mathbf{R}_k$$

$$\mathbf{r} = \begin{Bmatrix} \lambda\|\Delta\mathbf{f}\|_2 \\ \Delta\mathbf{x} \end{Bmatrix}$$

$$\|\mathbf{r}\|_2 = S$$



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

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

$$\mathbf{r}^{(i)}\cdot\mathbf{r}^{(i)} = S^2$$

$$\Delta\mathbf{u}^{(i+1)}\cdot\left(\Delta\mathbf{u}^{(i+1)} + 2\mathbf{r}^{(i)}\right) = 0$$

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

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

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

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

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

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

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

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

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

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

$$\mathbf{r}^{(i)} + \Delta\mathbf{u}^{(i+1)} = 0$$

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

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