

结构动力学模型修正

频域修正方法

李丹

2026 年 5 月 18 日

引言与频域修正概述

结构参数化建模

基于模态参数的目标函数

优化算法

数值算例：四层剪切框架

小结与思考题

引言与频域修正概述

模型修正指通过实测数据校准结构数值模型的参数，使模型预测的振动特性与真实结构一致。它本质上是一个反演问题：由有限的实测动力特性反求模型参数的最优估计。由于测量信息有限，该反演问题往往是病态的，需要借助约束与正则化获得物理合理的解。

本讲采用参数修正法，以结构物理参数（如层间刚度）为调整对象。修正后的模型保持明确的物理意义，刚度参数下降可对应结构损伤。基本流程为：建立初始模型，计算其模态，与实测对比，调整参数，反复迭代直至吻合。

按所用数据，模型修正分为频域与时域两类。频域方法基于固有频率、振型等模态信息，计算效率较高、实现相对简单，且能充分利用成熟的模态测试技术；时域方法直接基于响应时程，通常转化为状态估计问题求解。

频域修正的基本思想

调整有限元模型参数 θ ，使模型预测的固有频率 $\omega_i(\theta)$ 与振型 $\phi_i(\theta)$ 逼近实验测量结果。这一过程归结为一个 **优化问题**，本讲核心是其目标函数的构造与求解算法。

结构参数化建模

两种参数化方式

直接参数化以物理参数本身为待修正量：

$$\mathbf{K}(\boldsymbol{\theta}) = \sum_{i=1}^{n_{\text{elem}}} \mathbf{K}_i^e(\theta_i)$$

参数物理含义明确，便于结果解读。

相对参数化以相对初始模型的修正系数为待修正量：

$$\mathbf{K}(\boldsymbol{\theta}) = \mathbf{K}_0 + \sum_{i=1}^{n_{\theta}} \theta_i \mathbf{K}_i$$

参数无量纲、数值条件好，便于设定先验与正则化，本讲算例采用此方式。

算例结构：四层剪切框架

第 s 层实际层间刚度 $k_s = k_s^{\text{nom}}(1 + \theta_s)$, 质量与刚度矩阵

$$\mathbf{M} = \text{diag}(m_1, \dots, m_4), \quad \mathbf{K}(\boldsymbol{\theta}) = \begin{pmatrix} k_1+k_2 & -k_2 & 0 & 0 \\ -k_2 & k_2+k_3 & -k_3 & 0 \\ 0 & -k_3 & k_3+k_4 & -k_4 \\ 0 & 0 & -k_4 & k_4 \end{pmatrix}$$

装配规则：第 s 层连接楼层 $s-1$ 与 s 。 θ_s 为相对名义值的修正量 ($\theta_s=0$ 即名义/初始模型), 待由实测模态识别。

MATLAB: 质量与刚度矩阵装配

```
1 function [M, K] = shear_frame_KM(theta, k_nom, mass)
2 % k_s = k_nom(s)*theta(s)
3 n = numel(theta);
4 k_story = k_nom .* (1 + theta);
5 K = zeros(n);
6 for s = 1:n
7     if s == 1
8         K(1,1) = K(1,1) + k_story(1);
9     else
10         K(s-1,s-1) = K(s-1,s-1) + k_story(s);
11         K(s, s) = K(s, s) + k_story(s);
12         K(s-1,s) = K(s-1,s) - k_story(s);
13         K(s, s-1) = K(s, s-1) - k_story(s);
14     end
15 end
16 M = diag(mass(:));
17 end
```

基于模态参数的目标函数

数值模型的模态由广义特征值问题给出：

$$(\mathbf{K}(\boldsymbol{\theta}) - \omega_i^2 \mathbf{M}) \boldsymbol{\phi}_i = \mathbf{0}$$

$\omega_i(\boldsymbol{\theta})$ 与 $\boldsymbol{\phi}_i(\boldsymbol{\theta})$ 隐式依赖待修正参数 $\boldsymbol{\theta}$ 。

振动测试提取前 n_{mode} 阶实测频率 ω_i^{EXP} 与振型 $\boldsymbol{\phi}_i^{\text{EXP}}$ 。数值振型取出测量自由度对应分量后与之对比，二者在同一参考分量处归一化为 1，保证可比。

目标函数：频率残差与 MAC

综合频率与振型差异：

$$\min_{\theta} f(\theta) = \sum_{i=1}^{n_{\text{mode}}} \left[\left(\frac{\omega_i^{\text{EXP}} - \omega_i(\theta)}{\omega_i^{\text{EXP}}} \right)^2 + \left(\frac{1 - \sqrt{\text{MAC}_i}}{\sqrt{\text{MAC}_i}} \right)^2 \right]$$

模态保证准则

$$\text{MAC}_i = \frac{|(\phi_i^{\text{EXP}})^T \phi_i(\theta)|^2}{[(\phi_i^{\text{EXP}})^T \phi_i^{\text{EXP}}][\phi_i^T \phi_i]} \in [0, 1]$$

共线时 $\text{MAC} \rightarrow 1$ ，正交时 $\text{MAC} \rightarrow 0$ 。

目标函数：直接振型差与正则化

直接对比振型，得到便于最小二乘求解的形式：

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \sum_{i=1}^{n_{\text{mode}}} \left[\left(\frac{\omega_i^{\text{EXP}} - \omega_i(\boldsymbol{\theta})}{\omega_i^{\text{EXP}}} \right)^2 + \|\boldsymbol{\phi}_i^{\text{EXP}} - \boldsymbol{\phi}_i(\boldsymbol{\theta})\|^2 \right]$$

为缓解病态，引入正则化：

$$f_{\text{reg}}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \beta R(\boldsymbol{\theta}), \quad R(\boldsymbol{\theta}) = \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2 \text{ 或 } \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_1$$

ℓ_2 抑制参数变化并光滑解； ℓ_1 促进稀疏，适合损伤识别。

残差向量：把目标函数写成最小二乘

把上式每一项作为残差的一个分量，按顺序堆叠成残差向量

$$\mathbf{r}(\boldsymbol{\theta}) = \begin{pmatrix} r_1 \\ \vdots \\ r_{n_r} \end{pmatrix} = \begin{pmatrix} \frac{\omega_1^{\text{EXP}} - \omega_1(\boldsymbol{\theta})}{\omega_1^{\text{EXP}}} \\ \vdots \\ \frac{\omega_{n_m}^{\text{EXP}} - \omega_{n_m}(\boldsymbol{\theta})}{\omega_{n_m}^{\text{EXP}}} \\ \phi_1^{\text{EXP}} - \phi_1(\boldsymbol{\theta}) \\ \vdots \\ \phi_{n_m}^{\text{EXP}} - \phi_{n_m}(\boldsymbol{\theta}) \end{pmatrix}$$

于是目标函数就是平方和

$$f(\boldsymbol{\theta}) = \frac{1}{2} \sum_{p=1}^{n_r} r_p(\boldsymbol{\theta})^2 = \frac{1}{2} \|\mathbf{r}(\boldsymbol{\theta})\|_2^2$$

其中 n_r 为残差总数（频率项 n_m 个，振型项 $n_m \times$ 测点数）。

雅可比矩阵的展开 (核心)

残差对参数的全部一阶偏导构成**雅可比矩阵** $\mathbf{J} = D\mathbf{r} \in \mathbb{R}^{n_r \times n_\theta}$:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial r_1}{\partial \theta_1} & \frac{\partial r_1}{\partial \theta_2} & \cdots & \frac{\partial r_1}{\partial \theta_{n_\theta}} \\ \frac{\partial r_2}{\partial \theta_1} & \frac{\partial r_2}{\partial \theta_2} & \cdots & \frac{\partial r_2}{\partial \theta_{n_\theta}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_{n_r}}{\partial \theta_1} & \frac{\partial r_{n_r}}{\partial \theta_2} & \cdots & \frac{\partial r_{n_r}}{\partial \theta_{n_\theta}} \end{pmatrix}$$

第 p 行对应第 p 个残差, 第 j 列对应第 j 个参数。本讲算例中 $n_\theta = 4$ (四层刚度系数), $n_m = 3$, 测点 4 个, 故 $n_r = 3 + 3 \times 4 = 15$, $\mathbf{J}$ 为 15×4 矩阵。

雅可比的分块结构

按残差类型, $\mathbf{J}$ 自然分为频率块与振型块:

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_\omega \\ \mathbf{J}_\phi \end{pmatrix}, \quad (\mathbf{J}_\omega)_{ij} = \frac{\partial}{\partial \theta_j} \left(\frac{\omega_i^{\text{EXP}} - \omega_i(\boldsymbol{\theta})}{\omega_i^{\text{EXP}}} \right) = -\frac{1}{\omega_i^{\text{EXP}}} \frac{\partial \omega_i}{\partial \theta_j}$$

频率块 (本例 3×4) 显式为

$$\mathbf{J}_\omega = - \begin{pmatrix} \frac{1}{\omega_1^{\text{EXP}}} \partial_{\theta_1} \omega_1 & \cdots & \frac{1}{\omega_1^{\text{EXP}}} \partial_{\theta_4} \omega_1 \\ \frac{1}{\omega_2^{\text{EXP}}} \partial_{\theta_1} \omega_2 & \cdots & \frac{1}{\omega_2^{\text{EXP}}} \partial_{\theta_4} \omega_2 \\ \frac{1}{\omega_3^{\text{EXP}}} \partial_{\theta_1} \omega_3 & \cdots & \frac{1}{\omega_3^{\text{EXP}}} \partial_{\theta_4} \omega_3 \end{pmatrix}$$

振型块 $\mathbf{J}_\phi$ (12×4) 由 $-\partial \phi_i / \partial \theta_j$ 逐分量排列。代码中用中心差分数值计算 $\mathbf{J}$ 。

MATLAB: 模态求解与残差

```
1 function [omega, Phi] = modal_solve(K, M, nmode)
2 [V, D] = eig(K, M);
3 [lam, idx] = sort(real(diag(D)), 'ascend');
4 V = real(V(:, idx));
5 omega = sqrt(lam(1:nmode));   Phi = V(:, 1:nmode);
6 for i = 1:nmode
7     [~, p] = max(abs(Phi(:,i)));
8     Phi(:,i) = Phi(:,i) / Phi(p,i);   % max|comp| = 1
9 end
10 end
11
12 function r = resid_vec(theta, data)   % f = 0.5*||r||^2
13 [M,K] = shear_frame_KM(theta, data.k_nom, data.mass);
14 [omega,Phi] = modal_solve(K, M, data.nmode);
15 Phi = Phi(data.meas,:);   r_omega = zeros(data.nmode,1);   r_shape = [];
16 for i = 1:data.nmode
17     s = data.ref(i);
18     phi_num = Phi(:,i)/Phi(s,i);
19     phi_exp = data.Phi_exp(:,i)/data.Phi_exp(s,i);
20     r_omega(i) = (data.omega_exp(i)-omega(i))/data.omega_exp(i);
21     r_shape = [r_shape; (phi_exp-phi_num)];
22 end
23 r = [r_omega; r_shape];
24 end
```

优化算法

模型修正归结为无约束优化

$$\min_{\theta \in \mathbb{R}^n} f(\theta), \quad \theta^* = \arg \min f, \quad p^* = f(\theta^*)$$

迭代算法构造序列 $\{\theta^{(n)}\}$, 每步形如

$$\theta^{(n+1)} = \theta^{(n)} + t^{(n)} \Delta \theta^{(n)}$$

$\Delta \theta^{(n)}$ 为搜索方向, $t^{(n)} \geq 0$ 为步长。算法差异在于用几阶导数确定方向、如何确定步长。终止判据 $\|\nabla f(\theta^{(n)})\| < \varepsilon$ 。

下面依次推导：梯度下降、牛顿法、高斯-牛顿、LM。

预备：向量求导的定义与线性型

对标量函数 $g(\delta)$ ，其对向量 $\delta \in \mathbb{R}^n$ 的导数定义为逐分量偏导排成的列向量（梯度）：

$$\frac{\partial g}{\partial \delta} = \left(\frac{\partial g}{\partial \delta_1}, \frac{\partial g}{\partial \delta_2}, \dots, \frac{\partial g}{\partial \delta_n} \right)^T$$

规则一（线性型）： 设 $g = \mathbf{b}^T \delta = \sum_{k=1}^n b_k \delta_k$ 。对第 j 个分量求偏导， $\sum_k b_k \delta_k$ 中只有 $k=j$ 项含 δ_j ：

$$\frac{\partial g}{\partial \delta_j} = \frac{\partial}{\partial \delta_j} \sum_k b_k \delta_k = b_j \implies \boxed{\frac{\partial(\mathbf{b}^T \delta)}{\partial \delta} = \mathbf{b}}$$

预备：二次型的求导（逐元素）

规则二（二次型）： 设 $q = \delta^T \mathbf{A} \delta = \sum_{i=1}^n \sum_{k=1}^n \delta_i A_{ik} \delta_k$ 。对第 j 个分量求偏导，含 δ_j 的项分两类（ $i=j$ 与 $k=j$ ）：

$$\frac{\partial q}{\partial \delta_j} = \underbrace{\sum_k A_{jk} \delta_k}_{i=j \text{ 的项}} + \underbrace{\sum_i \delta_i A_{ij}}_{k=j \text{ 的项}} = (\mathbf{A} \delta)_j + (\mathbf{A}^T \delta)_j$$

把 $j = 1, \dots, n$ 排成列向量：

$$\frac{\partial(\delta^T \mathbf{A} \delta)}{\partial \delta} = (\mathbf{A} + \mathbf{A}^T) \delta \xrightarrow[\mathbf{A}=\mathbf{A}^T]{\text{对称时}} \boxed{2\mathbf{A} \delta}$$

本讲中 Hessian、 $\mathbf{J}^T \mathbf{J}$ 等均对称，故用 $2\mathbf{A} \delta$ 。

预备：Cauchy-Schwarz 不等式

对任意向量 $\mathbf{a}, \mathbf{d} \in \mathbb{R}^n$:

$$|\mathbf{a}^T \mathbf{d}| \leq \|\mathbf{a}\|_2 \|\mathbf{d}\|_2, \quad \text{等号当且仅当 } \mathbf{d} \parallel \mathbf{a}$$

特别地，在 $\|\mathbf{d}\|_2 = 1$ 的方向中，

$$\mathbf{a}^T \mathbf{d} \geq -\|\mathbf{a}\|_2, \quad \text{最小值在 } \mathbf{d} = -\mathbf{a}/\|\mathbf{a}\|_2 \text{ 取得}$$

这给出“负梯度是下降最快方向”的依据。以上三条（向量/二次型求导、Cauchy-Schwarz）是后续所有推导的工具，下面逐步用到。

梯度下降法：搜索方向（详细推导）

在 $\theta^{(n)}$ 作一阶 Taylor 展开

$$f(\theta^{(n)} + t\mathbf{d}) = f(\theta^{(n)}) + t \nabla f(\theta^{(n)})^T \mathbf{d} + o(t)$$

函数下降量主项为 $t \nabla f^T \mathbf{d}$ 。在所有单位方向 $\|\mathbf{d}\|_2 = 1$ 中求最快下降，即最小化 $\nabla f^T \mathbf{d}$ 。由 Cauchy-Schwarz,

$$\nabla f^T \mathbf{d} \geq -\|\nabla f\|_2 \|\mathbf{d}\|_2 = -\|\nabla f\|_2$$

下界 $-\|\nabla f\|_2$ 在 $\mathbf{d} = -\nabla f / \|\nabla f\|_2$ 时取得。故取

$$\Delta \theta^{(n)} = -\nabla f(\theta^{(n)}), \quad \nabla f^T \Delta \theta = -\|\nabla f\|_2^2 < 0$$

即负梯度方向是下降最快的方向。

梯度下降法：梯度 $\nabla f = \mathbf{J}^T \mathbf{r}$ 的推导

目标 $f = \frac{1}{2} \sum_{p=1}^{n_r} r_p^2$ 。对第 j 个参数求偏导，用复合函数求导：

$$\frac{\partial f}{\partial \theta_j} = \sum_{p=1}^{n_r} r_p \frac{\partial r_p}{\partial \theta_j} = \sum_{p=1}^{n_r} r_p \mathbf{J}_{pj} = (\mathbf{J}^T \mathbf{r})_j$$

把 $j = 1, \dots, n_\theta$ 排成列向量即

$$\nabla f = \mathbf{J}^T \mathbf{r}, \quad \underbrace{\mathbf{J}^T}_{n_\theta \times n_r} \underbrace{\mathbf{r}}_{n_r \times 1} = \underbrace{\nabla f}_{n_\theta \times 1}$$

本例 $\mathbf{J}^T$ 为 4×15 ， $\mathbf{r}$ 为 15×1 ，梯度 ∇f 为 4×1 。

梯度下降法：步长与收敛性

步长由**回溯线搜索**确定，要求满足充分下降条件：

$$f(\theta^{(n)} + t\Delta\theta^{(n)}) \leq f(\theta^{(n)}) + \alpha t \nabla f^T \Delta\theta^{(n)}$$

从 $t = 1$ 起，不满足则 $t \leftarrow \beta t$ (逐步回缩)， $\alpha \in (0, 0.5)$ ， $\beta \in (0, 1)$ 。

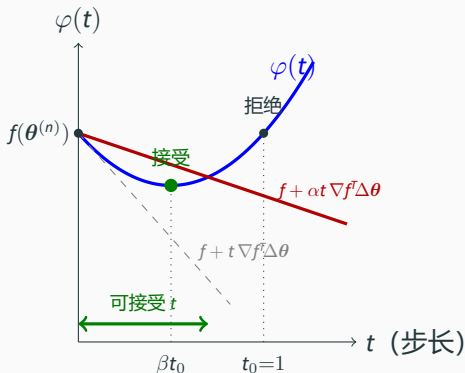
对二次型 $f = \frac{1}{2}\theta^T \mathbf{A}\theta$ ，精确线搜索下

$$\frac{f(\theta^{(n+1)}) - p^*}{f(\theta^{(n)}) - p^*} \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2, \quad \kappa = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}$$

收敛是**线性的**，条件数越大越慢。每步只需一次梯度，稳健但慢。

回溯线搜索：图示

沿搜索方向把 f 看作步长 t 的一元函数 $\varphi(t) = f(\theta^{(n)} + t\Delta\theta^{(n)})$ 。



曲线低于红色**充分下降线**即接受；初始 $t_0=1$ 落在其上方被拒绝，按 $t \leftarrow \beta t$ 回缩，直到进入可接受区间。

MATLAB: 梯度下降与回溯线搜索

```
1  for it = 1:maxit
2      [r, J] = fun(theta);
3      f = 0.5*(r'*r);      g = J'*r;           % grad = J' r
4      if norm(g) < tol, break; end
5      d = -g;              t = 1;             % steepest-descent dir
6      while true           % backtracking line search
7          r_t = fun(theta + t*d);
8          if 0.5*(r_t'*r_t) <= f + alpha*t*(g'*d) || t < 1e-12
9              break;
10         end
11         t = beta*t;
12     end
13     theta = theta + t*d;
14 end
```

牛顿法：目标函数的二阶 Taylor 展开

在 $\theta^{(n)}$ 处对目标函数 f 作二阶 Taylor 展开，得 f 在该点附近的二次近似

$$f(\theta^{(n)} + \delta) \approx f(\theta^{(n)}) + \nabla f(\theta^{(n)})^T \delta + \frac{1}{2} \delta^T \mathbf{H} \delta, \quad \mathbf{H} = \nabla^2 f(\theta^{(n)})$$

其中 Hessian

$$\mathbf{H}_{ij} = \frac{\partial^2 f}{\partial \theta_i \partial \theta_j}, \quad \mathbf{H} \in \mathbb{R}^{n_\theta \times n_\theta} \text{ 对称}$$

牛顿法的思想：用上式右端这个 f 的二次近似代替 f ，直接跳到该二次近似的极小点。

牛顿法：由驻点条件导出牛顿方程

令 f 的二次近似（上式右端）取极小，即其对 δ 的梯度为零。视 $f(\theta^{(n)})$ 、 ∇f 、 $\mathbf{H}$ 为常量，用预备规则逐项求导：

$$\frac{\partial}{\partial \delta} (\nabla f^T \delta) = \nabla f, \quad \frac{\partial}{\partial \delta} \left(\frac{1}{2} \delta^T \mathbf{H} \delta \right) = \mathbf{H} \delta$$

两者相加令其为零，得**牛顿方程组**（关于 δ 的线性方程组）

$$\nabla f + \mathbf{H} \delta = 0 \quad \Longrightarrow \quad \boxed{\mathbf{H} \delta = -\nabla f}$$

当 $\mathbf{H} \succ 0$ 时该二次近似严格凸，唯一解 $\Delta \theta_{\text{nt}}^{(n)} = -\mathbf{H}^{-1} \nabla f$ 。

实际**不求逆**，而是解线性方程组 $\mathbf{H}\delta = -\nabla f$ 。 $\mathbf{H} \succ 0$ 时用 Cholesky 分解：

$$\mathbf{H} = \mathbf{L}\mathbf{L}^T \quad (O(n_\theta^3/3)), \quad \mathbf{L}\mathbf{y} = -\nabla f \text{ (前代)}, \quad \mathbf{L}^T\delta = \mathbf{y} \text{ (回代)}$$

组装 Hessian 需 n_θ^2 个二阶偏导 ($O(n_\theta^2)$)，分解 $O(n_\theta^3)$ ，高维代价显著。

收敛性：若 $\mathbf{H}(\theta^*) \succ 0$ 且 Hessian Lipschitz 连续，最优解附近为**二次收敛** $\|\theta^{(n+1)} - \theta^*\| \leq C\|\theta^{(n)} - \theta^*\|^2$ ；对二次函数一步到位。

远离最优解时 $\mathbf{H}$ 可能非正定，纯牛顿步未必下降甚至发散。改进为阻尼牛顿法：

- $\mathbf{H}$ 非正定时修正为 $\mathbf{H} + \mu\mathbf{I} \succ 0$ （取 $\mu > -\lambda_{\min}(\mathbf{H})$ ）后再解方程组；
- 在牛顿方向上加回溯线搜索确定步长 t （满足充分下降条件），
$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} + t \Delta \boldsymbol{\theta}_{\text{nt}}^{(n)}。$$

既保留接近最优解时的二次收敛，又保证全局下降的稳健性。

MATLAB: 阻尼牛顿法 (核心)

```
1 [r,J] = fun(theta); f = 0.5*(r'*r); g = J'*r;
2 H = zeros(nparam); % Hessian via FD of grad
3 for k = 1:nparam
4     tp=theta; tp(k)=tp(k)+hstep; tm=theta; tm(k)=tm(k)-hstep;
5     H(:,k) = ( gradient_of(fun,tp) - gradient_of(fun,tm) )/(2*hstep);
6 end
7 H = 0.5*(H + H'); % symmetrize
8 mu = max(0, 1e-8 - min(eig(H))); % shift to keep PD
9 d = -(H + mu*eye(nparam)) \ g; % solve H d = -g
10 % --- backtracking line search on t, then theta = theta + t*d ---
```

高斯-牛顿法：最小二乘的导数结构

对 $f = \frac{1}{2} \|\mathbf{r}\|_2^2$ ，前已得 $\nabla f = \mathbf{J}^T \mathbf{r}$ 。对其再求一次导得 Hessian：

$$\nabla^2 f = \mathbf{J}^T \mathbf{J} + \sum_{p=1}^{n_r} r_p \nabla^2 r_p$$

第一项 $\mathbf{J}^T \mathbf{J} \in \mathbb{R}^{n_\theta \times n_\theta}$ 只用一阶导数即可得到；第二项含每个残差的 Hessian，计算昂贵。当残差较小或问题接近线性时，第二项可忽略——这是高斯-牛顿的核心近似。

高斯-牛顿法：残差线性化

在 $\theta^{(n)}$ 处把残差作**仿射近似**（一阶 Taylor）

$$\mathbf{r}(\theta^{(n)} + \delta) \approx \mathbf{r}^{(n)} + \mathbf{J}^{(n)} \delta$$

代入目标函数，得到关于 δ 的**线性最小二乘子问题**

$$\min_{\delta} \frac{1}{2} \|\mathbf{r}^{(n)} + \mathbf{J}^{(n)} \delta\|_2^2$$

因此非线性最小二乘被分解为**一系列线性最小二乘问题**：每步用切平面代替非线性模型，做一次线性回归，更新后重复。

高斯-牛顿法：展开得正规方程

记 $\mathbf{r} = \mathbf{r}^{(n)}$, $\mathbf{J} = \mathbf{J}^{(n)}$, 把范数平方逐项展开 (标量 $\mathbf{r}^T \mathbf{J} \delta = \delta^T \mathbf{J}^T \mathbf{r}$):

$$\|\mathbf{r} + \mathbf{J}\delta\|_2^2 = (\mathbf{r} + \mathbf{J}\delta)^T (\mathbf{r} + \mathbf{J}\delta) = \mathbf{r}^T \mathbf{r} + 2(\mathbf{J}^T \mathbf{r})^T \delta + \delta^T (\mathbf{J}^T \mathbf{J}) \delta$$

对 δ 求梯度 (用预备规则) 并令其为零:

$$2\mathbf{J}^T \mathbf{r} + 2\mathbf{J}^T \mathbf{J} \delta = \mathbf{0} \quad \implies \quad \boxed{\mathbf{J}^T \mathbf{J} \delta = -\mathbf{J}^T \mathbf{r}}$$

这就是**正规方程**, $\mathbf{J}^T \mathbf{J}$ 为 $n_\theta \times n_\theta$, $\mathbf{J}^T \mathbf{r}$ 为 $n_\theta \times 1$ 。

高斯-牛顿法：迭代格式与数值求解

由正规方程得迭代格式

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{r}$$

与牛顿法对照，它正是 $\nabla^2 f \approx \mathbf{J}^T \mathbf{J}$ 的特例。

数值上不直接形成 $\mathbf{J}^T \mathbf{J}$ (其条件数为 $\kappa(\mathbf{J})^2$ ，病态被平方放大)，而对 $\mathbf{J}$ 作 **QR 分解** $\mathbf{J} = \mathbf{Q}\mathbf{R}$ ，则

$$\boldsymbol{\delta} = -\mathbf{R}^{-1} \mathbf{Q}^T \mathbf{r}$$

收敛性：零残差问题二次收敛；残差越大、非线性越强，被忽略的二阶项影响越大，收敛退化为线性甚至发散。

MATLAB: 高斯-牛顿法 (核心)

```
1  for it = 1:maxit
2      [r, J] = fun(theta);
3      f = 0.5*(r'*r);    g = J'*r;
4      if norm(g) < tol, break; end
5      d = -(J \ r);           % Linear LSQ step (QR), robust
6      % equivalently: d = -(J'*J) \ (J'*r);
7      t = 1;                 % backtracking line search
8      while 0.5*(fun(theta+t*d)'*fun(theta+t*d)) ...
9          > f + alpha*t*(g'*d) && t > 1e-12
10         t = beta*t;
11     end
12     theta = theta + t*d;
13 end
```

LM 法：阻尼最小二乘

$J^T J$ 奇异或病态时高斯-牛顿不稳定。Levenberg-Marquardt 在子问题中加阻尼项：

$$\min_{\delta} \frac{1}{2} \|\mathbf{r} + \mathbf{J}\delta\|_2^2 + \frac{1}{2} \lambda \|\delta\|_2^2$$

同样令梯度为零，得**阻尼正规方程**

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \delta = -\mathbf{J}^T \mathbf{r}$$

它等价于一个**增广线性最小二乘**

$$\min_{\delta} \left\| \begin{pmatrix} \mathbf{J} \\ \sqrt{\lambda} \mathbf{I} \end{pmatrix} \delta + \begin{pmatrix} \mathbf{r} \\ \mathbf{0} \end{pmatrix} \right\|_2^2$$

对增广矩阵作 QR 分解可稳定求解。

阻尼项等价于给步长加信赖域约束

$$\min_{\delta} \frac{1}{2} \|\mathbf{r} + \mathbf{J}\delta\|_2^2 \quad \text{s.t.} \quad \|\delta\|_2 \leq \Delta$$

λ 是该约束的 Lagrange 乘子。两个极限：

$\lambda \rightarrow 0 : \delta \rightarrow \delta_{\text{GN}}$ (高斯-牛顿); $\lambda \rightarrow \infty : \delta \approx -\frac{1}{\lambda} \mathbf{J}^T \mathbf{r}$ (小步最速下降)

自适应：试探步使 f 下降则接受并 $\lambda \leftarrow \lambda/\gamma$ ；否则拒绝并 $\lambda \leftarrow \gamma\lambda$ ，重算方向。兼具稳健与快速，是非线性最小二乘的标准方法。

MATLAB: LM 法 (核心)

```
1 [r,J] = fun(theta); f = 0.5*(r'*r);
2 for it = 1:maxit
3     g = J'*r; if norm(g) < tol, break; end
4     A = J'*J; accepted = false;
5     for inner = 1:30
6         d = -(A + lambda*eye(nparam)) \ g; % damped normal eq.
7         r_try = fun(theta + d); f_try = 0.5*(r_try'*r_try);
8         if f_try < f % accept, less damping
9             theta = theta + d; [r,J] = fun(theta);
10            f = 0.5*(r'*r); lambda = max(lambda/gamma,1e-12);
11            accepted = true; break;
12        else % reject, more damping
13            lambda = lambda*gamma;
14        end
15    end
16    if ~accepted, break; end
17 end
```

四种方法的比较

	梯度下降	牛顿法	高斯-牛顿	LM
导数信息	一阶	二阶 $\mathbf{H}$	一阶 $\mathbf{J}$	一阶 $\mathbf{J}$
每步迭代	无	$\mathbf{H}\delta = -\nabla f$	$\mathbf{J}^T \mathbf{J} \delta = -\mathbf{J}^T \mathbf{r}$	阻尼正规方程
收敛速度	线性	二次	零残差二次	自适应
每步成本	低	$O(n_\theta^3)$	中	中
稳健性	强	弱 (需正定)	中	强
适用	通用	通用	最小二乘	最小二乘

模型修正的目标函数天然是最小二乘形式，残差通常不为零且存在病态，故工程实践多以 LM 为首选：远离最优时偏向最速下降以保稳健，接近最优时偏向高斯-牛顿以求快速收敛。

构造目标函数：从测量数据到残差函数

优化算法需要给定参数 θ ，返回残差 $\mathbf{r}(\theta)$ （及雅可比 $\mathbf{J}$ ）的函数。构造分三步：

1. **测量数据**：把实测频率 ω^{EXP} 、振型 ϕ^{EXP} 、测点编号等作为常量随算法迭代不变；
2. **残差函数** `resid_vec(theta, data)`：内部解特征问题得 $\omega_i(\theta), \phi_i(\theta)$ ，按定义堆叠成 $\mathbf{r}$ ；目标即 $f = \frac{1}{2} \|\mathbf{r}\|^2$ ；
3. **适配优化器接口**：算法要求单参数函数 $\mathbf{r}(\theta)$ ，故用匿名函数把 `data` 捕获进闭包。

MATLAB: lsqnonlin 的使用

```
1 function [theta, resnorm, output] = solve_with_lsqnonlin(theta0, data)
2 % data: measured modes etc.; theta0: initial guess
3
4 opts = optimoptions('lsqnonlin', ...
5     'Algorithm','levenberg-marquardt', ... % choose LM
6     'SpecifyObjectiveGradient',true, ...   % objfun returns J too
7     'FunctionTolerance',1e-12, ...         % stop: ||r||^2 change
8     'StepTolerance',1e-12, ...             % stop: step size
9     'MaxIterations',100, 'Display','off');
10
11 % [],[] = no lower/upper bounds (LM does not accept bounds)
12 [theta,resnorm,~,~,output] = ...
13     lsqnonlin(@objfun, theta0(:), [], [], opts);
14
15 % nested function: sees `data` from the parent workspace
16 function [r, J] = objfun(th)
17     [r, J] = residual_jacobian(th, data); % r and Jacobian
18 end
19 end
```

lsqnonlin：嵌套函数与调用流程

objfun 是**嵌套函数**（写在父函数内部），它能直接看到父函数工作区里的 data，因而仍是“单参数”接口 objfun(th)，却隐含携带了测量数据——与前面匿名闭包等价。

调用流程：

1. lsqnonlin 给定当前 θ ，调用 $[r,J]=\text{objfun}(\text{th})$ ；
2. objfun 转交 residual_jacobian，解特征问题算出残差 r 与雅可比 J ；
3. 因 SpecifyObjectiveGradient=true，lsqnonlin 用返回的 J 构造 LM 步，更新 θ ，回到第 1 步；
4. 满足容差后返回 theta、resnorm = $\|r\|^2$ 、迭代信息 output。

关键点：把“数据”与“待优化变量”分离——数据靠闭包/嵌套携带，变量是唯一的函数输入。

数值算例：四层剪切框架

四层剪切框架，各层质量 $m_s = 5 \text{ kg}$ ，名义层间刚度 $k_s^{\text{nom}} = 2000 \text{ N/m}$ 。设真实相对修正量 $\theta_{\text{true}} = (-0.20, 0.05, -0.35, 0.10)$ ，即第 1、3 层刚度分别下降 20%、35%（模拟损伤）。初始模型 $\theta^{(0)} = \mathbf{0}$ 。

用真实参数求解特征问题得前 3 阶模态，对频率与振型分别叠加 0.2% 与 1% 随机噪声，归一化后作为实测数据。残差由频率相对误差与振型差堆叠，实测与数值模态按频率升序对应。

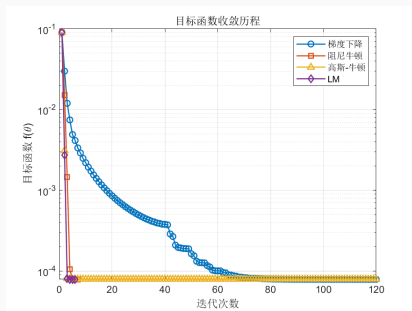
驱动脚本 (节选)

```
1 ndof=4; mass=5*ones(ndof,1); k_nom=2000*ones(ndof,1);
2 theta_true=[-0.20;0.05;-0.35;0.10]; theta0=zeros(ndof,1);
3 nmode=3; meas=(1:ndof)';
4 [M_true,K_true]=shear_frame_KM(theta_true,k_nom,mass);
5 [omega_true,Phi_true]=modal_solve(K_true,M_true,ndof);
6 omega_exp = omega_true(1:nmode).*(1+0.002*randn(nmode,1));
7 Phi_exp    = Phi_true(meas,1:nmode)+0.01*randn(ndof,nmode);
8 % ... normalize, build data struct ...
9 fun=@(th) residual_jacobian(th,data);
10 [theta_gn,hist_gn]=gauss_newton(fun,theta0,opt);
11 [theta_lm,hist_lm]=levenberg_marquardt(fun,theta0,optLM);
12 [theta_ls,resnorm]=solve_with_lsqnonlin(theta0,data);
```

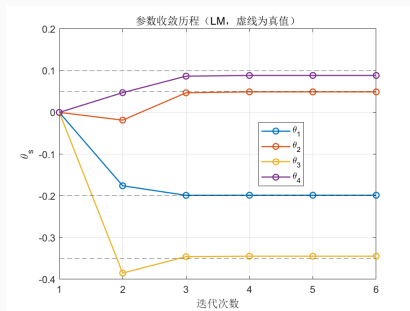
刚度修正系数 θ						固有频率 (Hz)			
层	真实	初始	GN	LM	lsqnonlin	阶	实测	修正前	修正后
1	-0.200	0.000	-0.199	-0.199	-0.199	1	1.013	1.106	1.016
2	0.050	0.000	0.049	0.049	0.049	2	2.845	3.183	2.842
3	-0.350	0.000	-0.345	-0.345	-0.345	3	4.908	4.877	4.902
4	0.100	0.000	0.089	0.089	0.089				

各算法收敛到一致结果并逼近真值，残余偏差来自人为噪声。修正后前三阶频率误差由约 10% 降至 0.5% 以内，第 1、3 层刚度下降被准确识别。

收敛与振型对比



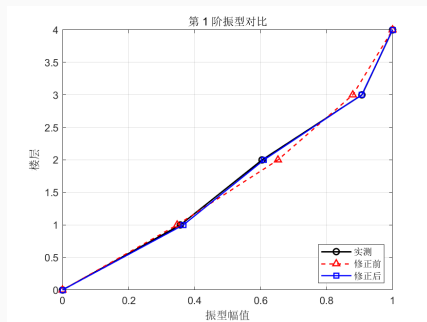
目标函数收敛历程



参数收敛 (LM, 虚线为真值)

牛顿法与 LM 在最优解附近收敛最快，梯度下降最慢，与理论一致。

第 1 阶振型对比



修正后振型与实测高度吻合，模型成功反映了结构的实际刚度分布。

小结与思考题

频域模型修正先把频率与振型的差异写成最小二乘目标函数（必要时加正则化），再用迭代优化算法求解。四种算法要点：

- 梯度下降只用一阶信息，稳健但线性收敛，接近最优时缓慢；
- 牛顿法用二阶信息，二次收敛，但需组装并分解 Hessian 且要求正定；
- 高斯-牛顿利用最小二乘结构，把问题化为一系列线性最小二乘，以 $J^T J$ 近似 Hessian，避免二阶导数；
- LM 以阻尼项在高斯-牛顿与最速下降间自适应过渡，最为稳健实用。

实践中可直接调用 `lsqnonlin`，只需提供残差与雅可比。

1. 由 $f = \frac{1}{2} \|\mathbf{r}\|^2$ 推导 $\nabla f = \mathbf{J}^T \mathbf{r}$ 与 $\nabla^2 f = \mathbf{J}^T \mathbf{J} + \sum_p r_p \nabla^2 r_p$, 说明高斯-牛顿近似何时精确。
2. 解释为何高斯-牛顿每步等价于一个线性最小二乘问题, 并比较正规方程法与 QR 分解法的数值稳定性。
3. 写出 LM 子问题的增广最小二乘形式, 分析 $\lambda \rightarrow 0$ 与 $\lambda \rightarrow \infty$ 两种极限。
4. 目标函数病态时, 正则化项如何改变正规方程? 与 LM 阻尼项有何异同?

完整可运行代码见 examples/ 目录。

谢谢

Q & A