Chapter 7

Nonlinear partial differential equations

7.1. Introduction

In this chapter we generalize the finite element method (Chapter 4, 6) to nonlinear, scalar-valued partial differential equations. This will be a relatively short chapter given that only minor modifications to the formulation are required to handle nonlinear problems; the finite element spaces remain the same. The main difference is the final FE system will be a *nonlinear* system of algebraic equations rather than a linear one.

7.2. Newton-Raphson method

Before we jump into the finite element formulation for nonlinear problems, we introduce the Newton-Raphson method for solving nonlinear systems of algebraic equations. This generalization of the familiar Newton's method for solving a single nonlinear equation in a single vairable will be needed to solve the final algebraic FE system that arises.

Let $\boldsymbol{R} : \mathbb{R}^m \to \mathbb{R}^m$ be a m-vector-valued function of m variables whose *root(s)* we wish to find, i.e., $\boldsymbol{x}^\star \in \mathbb{R}^m$ such that

$$\boldsymbol{R}(\boldsymbol{x}^\star) = \boldsymbol{0}. \tag{7.1}$$

Since there is no general procedure to solve nonlinear systems directly, i.e., like Gaussian elmination for linear systems, we turn to *iterative methods* whereby we produce a sequence $\{\boldsymbol{x}_k\}$ such that

$$\lim_{k\to\infty} \boldsymbol{R}(\boldsymbol{x}_k) = \boldsymbol{0}. \tag{7.2}$$

The idea behind the Newton-Raphson method is to *linearize* the function $\boldsymbol{R}$ about the current iterate $\boldsymbol{x}_k$

$$\boldsymbol{R}(\boldsymbol{x}) \approx \boldsymbol{R}_l(\boldsymbol{x}) := \boldsymbol{R}(\boldsymbol{x}_k) + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) \tag{7.3}$$

and define the next iterate in the sequence $\boldsymbol{x}_{k+1}$ to be the root of the linear approximation $\boldsymbol{R}_l(\boldsymbol{x})$, i.e.,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}(\boldsymbol{x}_k)\right]^{-1} \boldsymbol{R}(\boldsymbol{x}_k). \tag{7.4}$$

This is the Newton-Raphson nonlinear iteration for finding a root of the nonlinear vector-valued function $\boldsymbol{R}$.

Remark 7.1 (Initialization). The Newton-Raphson method only prescribes how to update $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$; it does not prescribe how to initialize the sequence, i.e., choose $\boldsymbol{x}_0$. It turns out that this is a very important consideration and somewhat of an "art".

Remark 7.2 (Convergence). If $\boldsymbol{x}_0$ is chosen to be "sufficiently close" to $\boldsymbol{x}^\star$, the Newton-Raphson method will converge quadratically, i.e.,

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\| \leqslant C \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|^2 , \tag{7.5}$$

where $C \in \mathbb{R}$ is a constant. This is extremely rapid convergence (Example 7.2), which is why the Newton-Raphson method is the most widely used nonlinear solver.

Remark 7.3 (Implementation). The implementation of the Newton-Raphson method is rather straightforward (Homework 5); however, notice that it relies on the Jacobian (matrix of partial derivatives) of the function $\boldsymbol{R}$. If the function $\boldsymbol{R}$ is very complicated or has many entries (m large), this can be somewhat cumbersome. Fortunately, many tools exist to facilitate derivative computations including automatic differentiation tools and symbolic mathematics packages (combined with code generation).

Remark 7.4 (Singularities). The Newton-Raphson iteration (7.4) requires the inverse of the Jacobian matrix, which assumes the Jacobian is nonsingular for all $\boldsymbol{x} \in \mathbb{R}^m$, or at the very least, nonsingular for all $\{\boldsymbol{x}_k\}$. If a point of singularity of the Jacobian (or it is severely ill-conditioned) is encountered, the algorithm will likely not converge.

Example 7.1: Single nonlinear equation

In the special case where we seek the root of a nonlinear function $f : \mathbb{R} \to \mathbb{R}$, the Newton-Raphson method reduces to the well-known Newton iteration

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \tag{7.6}$$

where $\{x_k\}$ is the sequence produced by the iteration that is initialized by x_0.

Example 7.2: Nonlinear system in $\mathbb{R}^2$

Consider the nonlinear function $\boldsymbol{R} : \mathbb{R}^2 \to \mathbb{R}^2$ and its Jacobian $\boldsymbol{J} : \mathbb{R}^2 \to M_{2,2}(\mathbb{R})$

$$\boldsymbol{R}(\boldsymbol{x}) = \begin{bmatrix} x_1 - 4x_1^2 - x_1 x_2 \\ 2x_2 - x_2^2 - 3x_1 x_2 \end{bmatrix}, \qquad \boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} 1 - 8x_1 - x_2 & -x_1 \\ -3x_2 & 2 - 2x_2 - 3x_1 \end{bmatrix}. \tag{7.7}$$

For this system, the Newton-Raphson method generates the sequence $\{\boldsymbol{x}_k\}$ in Table 7.1 that rapidly converges to the root $\boldsymbol{x}^\star = (0, 2)$.

Table 7.1: Sequence of iterates and convergence history of Newton-Raphson method applied to the system in Example 7.2, where $\Delta\boldsymbol{x}_k = \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_\infty$ and $m_k = \log(\|\boldsymbol{R}(\boldsymbol{x}_{k+1})\|_\infty)/\log(\|\boldsymbol{R}(\boldsymbol{x}_k)\|_\infty)$.

iteration (k)	$\boldsymbol{x}_k$	$\|\boldsymbol{R}(\boldsymbol{x}_k)\|_\infty$	$\|\Delta\boldsymbol{x}_k\|$	m_k
0	(1.00000000e+00, -1.00000000e+00)	2.000000e+00	-	-
1	(3.33333333e-01, 1.00000000e+00)	4.444444e-01	2.000000e+00	-1.169925e+00
2	(6.66666667e-02, 1.80000000e+00)	7.111111e-02	8.000000e-01	3.259851e+00
3	(3.92156863e-03, 1.98823529e+00)	3.936947e-03	1.882353e-01	2.094695e+00
4	(1.52590219e-05, 1.99995422e+00)	1.525925e-05	1.171893e-02	2.002822e+00
5	(2.32830644e-10, 2.00000000e+00)	2.328306e-10	4.577637e-05	2.000006e+00
6	(5.42101086e-20, 2.00000000e+00)	3.252607e-19	6.984919e-10	1.919220e+00

7.3. Nonlinear model problem

Let $\Omega \subset \mathbb{R}^d$ (open) and consider the Poisson equation with a nonlinear source term: find $u \in H^2(\Omega)$ such that

$$\begin{aligned} -\Delta u + u^2 &= f && \text{in } \Omega \\ u &= g && \text{on } \partial\Omega_D \\ \nabla u \cdot n &= h && \text{on } \partial\Omega_N, \end{aligned} \tag{7.8}$$

where the boundary of the domain $\partial\Omega$ is partitioned into $\partial\Omega_D$ (essential/Dirichlet condition applied) and $\partial\Omega_N$ (natural/Neumann condition applied), i.e., $\partial\Omega = \overline{\partial\Omega_D \cup \partial\Omega_N}$. For convenience, we convert this equation to indicial notation: $-u_{,ii} + u^2 = f$ in Ω, $u = g$ on $\partial\Omega_D$, $u_{,i}n_i = h$ on $\partial\Omega_N$.

To derive the weak formulation, we follow the standard procedure and setup the weighted residual equation by multiplying by a test function $w \in H^2(\Omega)$ and integrate over the domain

$$\int_\Omega w\left[(-u_{,ii}) + u^2 - f\right] dV = 0.$$

We apply integration-by-parts to the (use the identity $(wu_{,i})_{,i} = w_{,i}u_{,i} + wu_{,ii}$ and apply the divergence theorem; see (2.48)) yields

$$\int_\Omega w(-u_{,ii})\, dV = \int_\Omega (w_{,i}u_{,i} - (wu_{,i})_{,i})\, dV = \int_\Omega w_{,i}u_{,i}\, dV - \int_{\partial\Omega} wu_{,i}n_i\, dS = 0.$$

We see that u is the primary variable and $u_{,i}n_i$ is the secondary variable. Next, we choose $w(x) = 0$ for $x \in \partial\Omega_D$ because the primary variable is specified on $\partial\Omega_D$. This causes the integral over the entire boundary to become an integral over only $\partial\Omega_N$ because of the additive property of integration

$$\int_{\partial\Omega} wu_{,i}n_i\, dS = \int_{\partial\Omega_D} wu_{,i}n_i\, dS + \int_{\partial\Omega_N} wu_{,i}n_i\, dS = \int_{\partial\Omega_N} wu_{,i}n_i\, dS,$$

where the last equality used $w = 0$ on $\partial\Omega_D$. Finally, we substitute the natural boundary condition $u_{,i}n_i = h$ on $\partial\Omega_N$ to yield the weak formulation

$$\int_\Omega \left[w_{,i}u_{,i} + w(u^2 - f)\right] dV - \int_{\partial\Omega_N} w\, h\, dS = 0, \tag{7.9}$$

which can be formulated for $u, w \in H^1(\Omega)$ (only one weak derivative required). Notice that this cannot be written as a bilinear form because the dependence on u is nonlinear (quadratic in this case). Instead we abstract the weak form as: find $u \in \mathcal{V}$ such that

$$r(w, u) = 0 \tag{7.10}$$

for all $w \in \mathcal{V}^0$, where the finite element spaces are

$$\mathcal{V} := \{f \in H^1(\Omega) \mid f|_{\partial\Omega_D} = g\}, \qquad \mathcal{V}^0 := \{f \in H^1(\Omega) \mid f|_{\partial\Omega_D} = 0\}, \tag{7.11}$$

and $r : \mathcal{V}^0 \times \mathcal{V} \to \mathbb{R}$ is the functional

$$r(w, u) = \int_\Omega \left[w_{,i}u_{,i} + w(u^2 - f)\right] dV - \int_{\partial\Omega_N} w\, h\, dS. \tag{7.12}$$

In general, $r(w, u)$ is linear in its first argument, but not its second. This functional is the point of departure for this chapter; in the remainder, we will derive the finite element method in terms of a general functional that is linear in its first argument and nonlinear in its second argument.

7.4. Finite element method

The finite element method introduces a finite-dimensional linear subspace $\mathcal{V}_h^0 \subset \mathcal{V}^0$, which leads to the variational Galerkin formulation:

$$\text{find } u_h \in \mathcal{V}_h \text{ such that } r(w_h, u_h) = 0 \text{ for all } w_h \in \mathcal{V}_h^0, \tag{7.13}$$

where $\mathcal{V}_h := \varphi + \mathcal{V}_h^0$ for any $\varphi \in \mathcal{V}$. being a linear subspace of $\mathcal{V}^0$.

Following the approach in Chapter 4 and 6, we consider a mesh of Ω with elements $\{\Omega_e\}_{e=1}^{N_{\text{el}}}$ and nodes $\mathcal{N} = \{\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{N_{\text{nd}}}\}$, which we use to define a collection of finite elements $(\Omega_e, \mathcal{Y}_e, \mathcal{N}_e)$ (mapped from master element) and the finite element subspace $\mathcal{V}_h^0 \subset \mathcal{V}^0$

$$\mathcal{V}_h^0 := \left\{f \in H^1(\Omega) \;\middle|\; f|_{\Omega_e} \in \mathcal{Y}(\Omega_e)\ \forall \Omega_e \in \mathcal{E}_h,\ f|_{\partial\Omega_D} = 0\right\}. \tag{7.14}$$

Let $\{\Phi_1, \ldots, \Phi_{\bar{N}_{\text{dof}}}\}$ be a basis for $\mathcal{V}_h^0$, then any $u_h \in \mathcal{V}_h$ can be written as

$$u_h = \varphi + \sum_{I=1}^{\bar{N}_{\text{dof}}} \hat{u}_I^{\text{u}} \Phi_I. \tag{7.15}$$

In finite-dimensional setting, the variational problem (7.13) is equivalent to

$$\text{find } u_h \in \mathcal{V}_h \text{ such that } r(\Phi_I, u_h) = 0 \text{ for } I = 1, \ldots, \bar{N}_{\text{dof}} \tag{7.16}$$

by Proposition 3.1, which can be compactly written as

$$\hat{\boldsymbol{R}}^{\text{u}}(\hat{\boldsymbol{u}}^{\text{u}}) = \boldsymbol{0}, \tag{7.17}$$

where we defined $\hat{\boldsymbol{R}}^{\text{u}} : \mathbb{R}^{\bar{N}_{\text{dof}}} \to \mathbb{R}^{\bar{N}_{\text{dof}}}$ as

$$\hat{R}_I^{\text{u}}(\hat{\boldsymbol{u}}^{\text{u}}) = r(\Phi_I, \varphi + \sum_{J=1}^{\bar{N}_{\text{dof}}} \hat{u}_J^{\text{u}} \Phi_J). \tag{7.18}$$

We cannot simplify this further because the function r is nonlinear in its second entry. The system in (7.17) can be solved for the vector of unknown coefficients using the Newton-Raphson method (Section 7.2). The Jacobian of the nonlinear finite element residual in (7.18), required to use the Newton-Raphson method, is the matrix-valued function $\hat{\boldsymbol{J}}^{\text{uu}} : \mathbb{R}^{\bar{N}_{\text{dof}}} \to M_{\bar{N}_{\text{dof}}, \bar{N}_{\text{dof}}}(\mathbb{R})$ defined as

$$\hat{J}_{IJ}^{\text{uu}}(\hat{\boldsymbol{u}}^{\text{u}}) := \frac{\partial \hat{R}_I^{\text{u}}}{\partial \hat{u}_J^{\text{u}}}(\hat{\boldsymbol{u}}^{\text{u}}) = \frac{\partial r}{\partial u}(\Phi_I, \varphi + \sum_{K=1}^{\bar{N}_{\text{dof}}} \hat{u}_K^{\text{u}} \Phi_K) \Phi_J \tag{7.19}$$

for $I, J = 1, \ldots, \bar{N}_{\text{dof}}$.

7.5. Assembly of finite element system

Following the construction in Section 6.7, we relate our local and global spaces to derive the assembly procedure in the nonlinear setting. First, observe that the semi-linear form can be written as a summation of element contributions

$$r(w, u) = \sum_{e=1}^{N_{\text{el}}} r_e(w, u) = \sum_{e=1}^{N_{\text{el}}} r_e(w|_{\Omega_e}, u|_{\Omega_e}), \tag{7.20}$$

where $r_e : \mathcal{V}_h^0 \times \mathcal{V}_h \to \mathbb{R}$ is the element contribution of Ω_e. The semi-linear form for the model problem in (7.9) is clearly

$$r_e(w, u) = \int_{\Omega_e} \left[w_{,i} u_{,i} + w(u^2 - f) \right] dV - \int_{\partial\Omega_e \cap \partial\Omega_N} w\, h\, dS. \tag{7.21}$$

Then, following the development in Section 6.6, we introduce the global nodal shape functions $\{\Psi_1, \ldots, \Psi_{N_{\text{nd}}}\}$ that satisfy (6.162) and the nodal basis functions $\{\phi_1^e, \ldots, \phi_{N_{\text{nd}}^{\text{el}}}^e\}$ of the local function space $\mathcal{Y}_e$ associated with the nodes $\mathcal{N}_e$. The shape functions are related to the element basis functions by (6.168) and the global nodal basis functions $\{\Phi_1, \ldots, \Phi_{\bar{N}_{\text{dof}}}\}$ by (6.170), where $\mathcal{I}_c$ is the collection of node numbers on which an essential boundary condition is prescribed and $\mathcal{I}_u$ are the remaining node numbers. Finally, we take the affine offset to be

$$\varphi = \sum_{I \in \mathcal{I}_c} g(\hat{\boldsymbol{x}}_I) \Psi_I. \tag{7.22}$$

From these relationships, we introduce the complete (without static condensation) finite element residual and expand in terms of element contributions as

$$\begin{aligned} \hat{R}_I(\hat{\boldsymbol{u}}) := r(\Psi_I, u_h) = \sum_{e=1}^{N_{\text{el}}} r_e(\Psi_I|_{\Omega_e}, u_h|_{\Omega_e}) &= \sum_{e=1}^{N_{\text{el}}} \sum_{i=1}^{N_{\text{nd}}^{\text{el}}} \delta_{I \Theta_{ie}} r_e \left(\phi_i^e, \sum_{k=1}^{N_{\text{nd}}^{\text{el}}} \hat{u}_k^e \phi_k^e \right) \\ &= \sum_{e=1}^{N_{\text{el}}} \sum_{i=1}^{N_{\text{nd}}^{\text{el}}} \delta_{I \Theta_{ie}} r_e \left(\phi_i^e, \sum_{k=1}^{N_{\text{nd}}^{\text{el}}} \hat{u}_{\Theta_{ke}} \phi_k^e \right), \end{aligned} \tag{7.23}$$

where we used (6.168) and the expansion of u_h in the element basis

$$u_h|_{\Omega_e} = \sum_{j=1}^{N_{\text{nd}}^{\text{el}}} \hat{u}_j^e \phi_j^e \tag{7.24}$$

and the relationship between the local and global degrees of freedom $\hat{u}_j^e = \hat{u}_{\Theta_{je}}$. The corresponding Jacobian matrix is

$$\frac{\partial \hat{R}_I}{\partial \hat{u}_J}(\hat{\boldsymbol{u}}) = \sum_{e=1}^{N_{\text{el}}} \sum_{i=1}^{N_{\text{nd}}^{\text{el}}} \sum_{j=1}^{N_{\text{nd}}^{\text{el}}} \frac{\partial r_e}{\partial u_e} \left(\phi_i^e, \sum_{k=1}^{N_{\text{nd}}^{\text{el}}} \hat{u}_{\Theta_{ke}} \phi_k^e \right) \delta_{I\Theta_{ie}} \delta_{J\Theta_{je}} \phi_j^e. \tag{7.25}$$

These terms can be related to the residual and Jacobian in the finite element system in (7.18) as

$$\hat{R}_I^{\text{u}}(\hat{\boldsymbol{u}}^{\text{u}}; \hat{\boldsymbol{u}}^{\text{c}}) = \hat{R}_{\mathcal{I}_u(I)}(\hat{\boldsymbol{u}}), \qquad \frac{\partial \hat{R}_I^{\text{u}}}{\partial \hat{u}_J^{\text{u}}} = \frac{\partial \hat{R}_{\mathcal{I}_u(I)}}{\partial \hat{u}_{\mathcal{I}_u(J)}}(\hat{\boldsymbol{u}}), \tag{7.26}$$

where the partition of $\hat{\boldsymbol{u}}$ into $\hat{\boldsymbol{u}}^{\text{u}}$ and $\hat{\boldsymbol{u}}^{\text{c}}$ is given in (6.175); this is static condensation in the nonlinear setting.

7.6. Summary

In this chapter, we extended the finite element method to handle nonlinear problems and introduced the Newton-Raphson method for solving systems of nonlinear equations. The finite element method for nonlinear PDEs is very similar to the construction for linear PDEs; however, we cannot assume the weak formulation is bilinear, rather it is only linear in its first argument. As a result, the discrete finite element system is a nonlinear system of equations that we can solve using the Newton-Raphson method.