Chapter 9

Parabolic Problems

9.1 Introduction

The finite element method may be used to solve time-dependent problems as well as steady ones. This effort involves both parabolic and hyperbolic partial differential systems. Problems of parabolic type involve diffusion and dissipation while hyperbolic problems are characterized by conservation of energy and wave propagation. Simple one-dimensional heat conduction and wave propagation equations will serve as model problems of each type.

Example 9.1.1. The one-dimensional heat conduction equation

\[ u_t = p u_{xx}, \quad a < x < b, \quad t > 0, \]  

(9.1.1a)

where \( p \) is a positive constant called the diffusivity, is of parabolic type. Initial-boundary value problems consist of determining \( u(x,t) \) satisfying (9.1.1a) given the initial data

\[ u(x,0) = u^0(x), \quad a \leq x \leq b, \]  

(9.1.1b)

and appropriate boundary data, e.g.,

\[ p u_x(a,t) + \gamma_0 u(a,t) = \beta_0(t), \quad p u_x(b,t) + \gamma_1 u(b,t) = \beta_1(t). \]  

(9.1.1c)

As with elliptic problems, boundary conditions without the \( p u_x \) term are called Dirichlet conditions; those with \( \gamma_i = 0, \ i = 0,1 \), are Neumann conditions; and those with both terms present are called Robin conditions. The problem domain is open in the time direction \( t \); thus, unlike elliptic systems, this problem is evolutionary and computation continues in \( t \) for as long as there is interest in the solution.

Example 9.1.2. The one-dimensional wave equation

\[ u_{tt} = c^2 u_{xx}, \quad a < x < b, \quad t > 0, \]  

(9.1.2a)
where \( c \) is a constant called the wave speed, is a hyperbolic partial differential equation. Initial-boundary value problems consist of determining \( u(x, t) \) satisfying (9.1.2a) given the initial data

\[
  u(x, 0) = u^0(x), \quad u_t(x, 0) = u^0_t(x), \quad a \leq x \leq b, \tag{9.1.2b}
\]

and boundary data of the form (9.1.1c). Small transverse vibrations of a taut string satisfy the wave equation. In this case, \( u(x, t) \) is the transverse displacement of the string and \( c^2 = T/\rho, \) \( T \) being the applied tension and \( \rho \) being the density of the string.

We’ll study parabolic problems in this chapter and hyperbolic problems in the next. We shall see that there are two basic finite element approaches to solving time-dependent problems. The first, called the method of lines, uses finite elements in space and ordinary differential equations software in time. The second uses finite element methods in both space and time. We’ll examine the method of lines approach first and then tackle space-time finite element methods.

### 9.2 Semi-Discrete Galerkin Problems: The Method of Lines

Let us consider a parabolic problem of the form

\[
  u_t + \mathcal{L}[u] = f(x, y), \quad (x, y) \in \Omega, \quad t > 0, \tag{9.2.1a}
\]

where \( \mathcal{L} \) is a second-order elliptic operator. In two dimensions, \( u \) would be a function of \( x, y, \) and \( t \) and \( \mathcal{L}[u] \) could be the very familiar

\[
  \mathcal{L}[u] = -(pu_x)_x - (pu_y)_y + qu. \tag{9.2.1b}
\]

Appropriate initial and boundary conditions would also be needed, e.g.,

\[
  u(x, y, 0) = u^0(x, y), \quad (x, y) \in \Omega \cup \partial\Omega, \tag{9.2.1c}
\]

\[
  u(x, y, t) = \alpha(x, y, t), \quad (x, y) \in \partial\Omega_E, \tag{9.2.1d}
\]

\[
  pu_n + \gamma u = \beta, \quad (x, y) \in \partial\Omega_N. \tag{9.2.1e}
\]

We construct a Galerkin formulation of (9.2.1) in space in the usual manner; thus, we multiply (9.2.1a) by a suitable test function \( v \) and integrate the result over \( \Omega \) to obtain

\[
  (v, u_t) + (v, \mathcal{L}[u]) = (v, f),
\]
As usual, we apply the divergence theorem to the second-derivative terms in $L$ to reduce the continuity requirements on $u$. When $L$ has the form of (9.2.1b), the Galerkin problem consists of determining $u \in H^1_E \times \{t > 0\}$ such that

$$(v, u_t) + A(v, u) = (v, f) + <v, \beta - \gamma u>, \quad \forall v \in H^1_0, \quad t > 0. \quad (9.2.2a)$$

The $L^2$ inner product, strain energy, and boundary inner product are, as with elliptic problems,

$$(v, f) = \int_\Omega vf \, dx \, dy, \quad (9.2.2b)$$

$$A(v, u) = \int_\Omega [p(v_x u_x + v_y u_y) + vqu] \, dx \, dy, \quad (9.2.2c)$$

and

$$<v, pu_n> = \int_{\partial \Omega_N} vpu_n \, ds. \quad (9.2.2d)$$

The natural boundary condition (9.2.1e) has been used to replace $pu_n$ in the boundary inner product. Except for the presence of the $(v, u_t)$ term, the formulation appears to the same as for an elliptic problem.

Initial conditions for (9.2.2a) are usually determined by projection of the initial data (9.2.1c) either in $L^2$

$$(v, u) = (v, u^0), \quad \forall v \in H^1_0, \quad t = 0, \quad (9.2.3a)$$

or in strain energy

$$A(v, u) = A(v, u^0), \quad \forall v \in H^1_0, \quad t = 0. \quad (9.2.3b)$$

Example 9.2.1. We analyze the one-dimensional heat conduction problem

$$u_t = (pu_x)_x + f(x, t), \quad 0 < x < 1, \quad t > 0,$$

$$u(x, 0) = u^0(x), \quad 0 \leq x \leq 1,$$

$$u(0, t) = u(1, t) = 0, \quad t > 0,$$

thoroughly in the spirit that we did in Chapter 1 for a two-point boundary value problem.

A Galerkin form of this heat-conduction problem consists of determining $u \in H^1_0$ satisfying

$$(v, u_t) + A(v, u) = (v, f), \quad \forall v \in H^1_0, \quad t > 0,$$
Figure 9.2.1: Mesh for the finite element solution of Example 9.2.1.

\[(v, u) = (v, u^0), \quad \forall v \in H^1_0, \quad t = 0,\]

where

\[A(v, u) = \int_0^1 v u' \varphi d\alpha.\]

Boundary terms of the form (9.2.2d) disappear because \(v = 0\) at \(x = 0, 1\) with Dirichlet data.

We introduce a mesh on \(0 \leq x \leq 1\) as shown in Figure 9.2.1 and choose an approximation \(U\) of \(u\) in a finite-dimensional subspace \(S^N_0\) of \(H^1_0\) having the form

\[U(x, t) = \sum_{j=1}^{N-1} c_j(t) \phi_j(x).\]

Unlike steady problems, the coefficients \(c_j, j = 1, 2, \ldots, N-1\), depend on \(t\). The Galerkin finite element problem is to determine \(U \in S^N_0\) such that

\[(\phi_j, U_t) + A(\phi_j, U) = (\phi_j, f), \quad t > 0,\]

\[(\phi_j, U) = (\phi_j, u^0), \quad t = 0, \quad j = 1, 2, \ldots, N-1.\]

Let us choose a piecewise-linear polynomial basis

\[\phi_k(x) = \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}}, & \text{if } x_{k-1} < x \leq x_k \\ \frac{x-x_k}{x_{k+1}-x_k}, & \text{if } x_k < x \leq x_{k+1} \\ 0, & \text{otherwise} \end{cases}\]

This problem is very similar to the one-dimensional elliptic problem considered in Section 1.3, so we’ll skip several steps and also construct the discrete equations by vertices rather than by elements.
Since \( \phi_j \) has support on the two elements containing node \( j \) we have

\[
A(\phi_j, U) = \int_{x_{j-1}}^{x_j} \phi'_j p(x) dx + \int_{x_j}^{x_{j+1}} \phi'_j p(x) dx
\]

where \((\cdot)' = d(\cdot)/dx\). Substituting for \( \phi_j \) and \( U \)

\[
A(\phi_j, U) = \int_{x_{j-1}}^{x_j} \frac{1}{h_j} p(x) (c_j - c_{j-1}) dx + \int_{x_j}^{x_{j+1}} \frac{1}{h_{j+1}} p(x) \frac{(c_{j+1} - c_j)}{h_{j+1}} dx
\]

where

\( h_j = x_j - x_{j-1} \).

Using the midpoint rule to evaluate the integrals, we have

\[
A(\phi_j, U) \approx \frac{p_{j-1/2}}{h_j} (c_j - c_{j-1}) - \frac{p_{j+1/2}}{h_{j+1}} (c_{j+1} - c_j)
\]

where \( p_{j-1/2} = p(x_{j-1/2}) \).

Similarly,

\[
(\phi_j, U_t) = \int_{x_{j-1}}^{x_j} \phi_j U_t dx + \int_{x_j}^{x_{j+1}} \phi_j U_t dx
\]

or

\[
(\phi_j, U_t) = \int_{x_{j-1}}^{x_j} \phi_j (\dot{c}_{j-1} \phi_j + \dot{c}_j \phi_j) dx + \int_{x_j}^{x_{j+1}} \phi_j (\dot{c}_j \phi_j + \dot{c}_{j+1} \phi_j) dx
\]

where \((\cdot)' = d(\cdot)/dt\). Since the integrands are quadratic functions of \( x \) they may be integrated exactly using Simpson’s rule to yield

\[
(\phi_j, U_t) = \frac{h_j}{6} (\dot{c}_{j-1} + 2\dot{c}_j) + \frac{h_{j+1}}{6} (2\dot{c}_j + \dot{c}_{j+1}).
\]

Finally,

\[
(\phi_j, f) \approx \int_{x_{j-1}}^{x_j} \phi_j f(x) dx + \int_{x_j}^{x_{j+1}} \phi_j f(x) dx.
\]

Although integration of order one would do, we’ll, once again, use Simpson’s rule to obtain

\[
(\phi_j, f) \approx \frac{h_j}{6} (2f_{j-1/2} + f_j) + \frac{h_{j+1}}{6} (f_j + 2f_{j+1/2}).
\]

We could replace \( f_{j-1/2} \) by the average of \( f_{j-1} \) and \( f_j \) to obtain a similar formula to the one obtained for \( (\phi_j, U_t) \); thus,

\[
(\phi_j, f) \approx \frac{h_j}{6} (f_{j-1} + 2f_j) + \frac{h_{j+1}}{6} (2f_j + f_{j+1}).
\]

Combining these results yields the discrete finite element system

\[
\frac{h_j}{6} (\dot{c}_{j-1} + 2\dot{c}_j) + \frac{h_{j+1}}{6} (2\dot{c}_j + \dot{c}_{j+1}) + \frac{p_{j-1/2}}{h_j} (c_j - c_{j-1}) - \frac{p_{j+1/2}}{h_j + 1/2} (c_{j+1} - c_j)
\]
\[
\begin{align*}
&= \frac{h_j}{6} (f_{j-1} + 2f_j) + \frac{h_{j+1}}{6} (2f_j + f_{j+1}), \quad j = 1, 2, \ldots, N - 1.
\end{align*}
\]
(We have dropped the \( \approx \) and written the equation as an equality.)

If \( p \) is constant and the mesh spacing \( h \) is uniform, we obtain

\[
\frac{h}{6} (\dot{c}_{j-1} + 4\dot{c}_j + \dot{c}_{j+1}) - \frac{p}{h} (c_{j-1} - 2c_j + c_{j+1}) = \frac{h}{6} (f_{j-1} + 4f_j + f_{j+1}),
\]
\[j = 1, 2, \ldots, N - 1.\]

The discrete systems may be written in matrix form and, for simplicity, we’ll do so for the constant coefficient, uniform mesh case to obtain

\[
M \dot{c} + K c = l \tag{9.2.4a}
\]

where

\[
M = \frac{h}{6} \begin{bmatrix}
4 & 1 & & & \\
1 & 4 & 1 & & \\
& \ddots & \ddots & \ddots & \\
& & 1 & 4 & 1 \\
& & & 1 & 4
\end{bmatrix}, \tag{9.2.4b}
\]

\[
K = \frac{p}{h} \begin{bmatrix}
2 & -1 & & & \\
-1 & 2 & -1 & & \\
& \ddots & \ddots & \ddots & \\
& & -1 & 2 & -1 \\
& & & -1 & 2
\end{bmatrix}, \tag{9.2.4c}
\]

\[
l = \frac{h}{6} \begin{bmatrix}
f_0 + 4f_1 + f_2 \\
f_1 + 4f_2 + f_3 \\
\vdots \\
f_{N-2} + 4f_{N-1} + f_N
\end{bmatrix}, \tag{9.2.4d}
\]

\[
c = [c_1, c_2, \ldots, c_{N-1}]^T. \tag{9.2.4e}
\]

The matrices \( M, K, \) and \( l \) are the global mass matrix, the global stiffness matrix, and the global load vector. Actually, \( M \) has little to do with mass and should more correctly be called a global dissipation matrix; however, we’ll stay with our prior terminology. In practical problems, element-by-element assembly should be used to construct global matrices and vectors and not the nodal approach used here.

The discrete finite element system (9.2.4) is an implicit system of ordinary differential equations for \( \dot{c} \). The mass matrix \( M \) can be “lumped” by a variety of tricks to yield an
explicit ordinary differential system. One such trick is to approximate \( (\phi_j, U_t) \) by using the right-rectangular rule on each element to obtain

\[
(\phi_j, U_t) = \int_{x_{j-1}}^{x_j} \phi_j(\dot{c}_j \phi_{j-1} + \ddot{c}_j \phi_j) dx + \int_{x_j}^{x_{j+1}} \phi_j(\ddot{c}_j \phi_j + \dddot{c}_j \phi_{j+1}) dx \approx h c_j.
\]

The resulting finite element system would be

\[
h I c + K c = l.
\]

Recall (cf. Section 6.3), that a one-point quadrature rule is satisfactory for the convergence of a piecewise-linear polynomial finite element solution.

With the initial data determined by \( L^2 \) projection onto \( S_E^N \), we have

\[
(\phi_j, U(\cdot, 0)) = (\phi_j, u^0), \quad j = 1, 2, \ldots, N - 1.
\]

Numerical integration will typically be needed to evaluate \( (\phi_j, u^0) \) and we’ll approximate it in the manner used for the loading term \( (\phi_j, f) \). Thus, with uniform spacing, we have

\[
M c(0) = u^0 = \frac{h}{6} \begin{bmatrix}
  u_0^0 + 4u_1^0 + u_2^0 & \\
  u_1^0 + 4u_2^0 + u_3^0 & \\
  \vdots & \\
  u_{N-2}^0 + 4u_{N-1}^0 + u_N^0 & 
\end{bmatrix}.
\] (9.2.4f)

If the initial data is consistent with the trivial Dirichlet boundary data, i.e., if \( u^0 \in H^1_0 \) then the above system reduces to

\[
c_j(0) = u^0(x_j), \quad j = 1, 2, 3, \ldots, N - 1.
\]

Had we solved the wave equation (9.1.2) instead of the heat equation (9.1.1) using a piecewise-linear finite element basis, we would have found the discrete system

\[
M \ddot{c} + K c = 0
\] (9.2.5)

with \( p \) in (9.2.4c) replaced by \( c^2 \).

The resulting initial value problems (IVPs) for the ordinary differential equations (ODEs) (9.2.4a) or (9.2.5) typically have to be integrated numerically. There are several excellent software packages for solving IVPs for ODEs. When such ODE software is used with a finite element or finite difference spatial discretization, the resulting procedure is called the method of lines. Thus, the nodes of the finite elements appear to be “lines” in the time direction and, as shown in Figure 9.2.2 for a one-dimensional problem, the temporal integration proceeds along these lines.
Using the ODE software, solutions are calculated in a series of time steps \((0, t_1], (t_1, t_2], \ldots\). Methods fall into two types. Those that only require knowledge of the solution at time \(t_n\) in order to obtain a solution at time \(t_{n+1}\) are called \textit{one-step methods}. Correspondingly, methods that require information about the solution at \(t_n\) and several times prior to \(t_n\) are called \textit{multistep methods}. Excellent texts on the subject are available [2, 6, 7, 8]. One-step methods are Runge-Kutta methods while the common multistep methods are Adams or backward difference methods. Software based on these methods automatically adjusts the time steps and may also automatically vary the order of accuracy of a class of methods in order to satisfy a prescribed local error tolerance, minimize computational cost, and maintain numerical efficiency.

The choice of a one-step or multistep method will depend on several factors. Generally, Runge-Kutta methods are preferred when time integration is simple relative to the spatial solution. Multistep methods become more efficient for complex nonlinear problems. Implicit Runge-Kutta methods may be efficient for problems with high-frequency oscillations. The ODEs that arise from the finite element discretization of parabolic problems are “stiff” [2, 8] so backward difference methods are the preferred multistep methods.

Most ODE software [2, 7, 8] addresses first-order IVPs of the explicit form

\[
\dot{y}(t) = f(t, y(t)), \quad y(0) = y^0. \tag{9.2.6}
\]

Second-order systems such as (9.2.5) would have to be written as a first-order system by, \textit{e.g.}, letting

\[
d = \dot{c}
\]
9.2. Semi-Discrete Galerkin Problems

and, hence, obtaining

\[
\begin{bmatrix}
\dot{c} \\
\text{M} \dot{d}
\end{bmatrix} = \begin{bmatrix}
d \\
-Kc
\end{bmatrix}.
\]

Unfortunately, systems having the form of (9.2.4a) or the one above are implicit and would require inverting or lumping \( \text{M} \) in order to put them into the standard explicit form (9.2.6). Inverting \( \text{M} \) is not terribly difficult when \( \text{M} \) is constant or independent of \( t \); however, it would be inefficient for nonlinear problems and impossible when \( \text{M} \) is singular. The latter case can occur when, \( e.g. \), a heat conduction and a potential problem are solved simultaneously.

Codes for differential-algebraic equations (DAEs) directly address the solution of implicit systems of the form

\[
f(t, y(t), \dot{y}(t)) = 0, \quad y(0) = y^0.
\]  

(9.2.7)

One of the best of these is the code DASSL written by Petzold [3]. DASSL uses variable-step, variable-order backward difference methods to solve problems without needing \( \text{M}^{-1} \) to exist.

Let us illustrate these concepts by applying some simple one-step schemes to problems having the forms (9.2.1) or (9.2.4). However, implementation of these simple methods is only justified in certain special circumstances. In most cases, it is far better to use existing ODE software in a method of lines framework.

For simplicity, we’ll assume that all boundary data is homogeneous so that the boundary inner product in (9.2.2a) vanishes. Selecting a finite-dimensional space \( S_0^N \subset H_0^1 \), we then determine \( U \) as the solution of

\[
(V, U_t) + A(V, U) = (V, f), \quad \forall v \in S_0^N.
\]  

(9.2.8)

Evaluation leads to ODEs having the form of (9.2.4a) regardless of whether or not the system is one-dimensional or the coefficients are constant. The actual matrices \( \text{M} \) and \( \text{K} \) and load vector \( \text{l} \) will, of course, differ from those of Example 9.2.1 in these cases. The systems (9.2.4a) or (9.2.8) are called semi-discrete Galerkin equations because time has not yet been discretized.

We discretize time into a sequence of time slices \( (t_n, t_{n+1}] \) of duration \( \Delta t \) with \( t_n = n\Delta t, n = 0, 1, \ldots \). For this discussion, no generality is lost by considering uniform time steps. Let:

- \( u(x, t_n) \) be the exact solution of the Galerkin problem (9.2.2a) at \( t = t_n \).
- \( U(x, t_n) \) be the exact solution of the semi-discrete Galerkin problem (9.2.8) at \( t = t_n \).
- \( U^n(x) \) be the approximation of \( U(x, t_n) \) obtained by ODE software.
\begin{itemize}
  \item \(c_j(t_n)\) be the Galerkin coefficient at \(t = t_n\); thus, for a one-dimensional problem
  \[
  U(x, t_n) = \sum_{j=1}^{N-1} c_j(t_n) \phi_j(x).
  \]

  For a Lagrangian basis, \(c_j(t_n) = U(x_j, t_n)\).

  \item \(c^n_j\) be the approximation of \(c_j(t_n)\) obtained by ODE software. For a one-dimensional problem
  \[
  U^n(x) = \sum_{j=1}^{N-1} c^n_j \phi_j(x).
  \]

  We suppose that all solutions are known at time \(t_n\) and that we seek to determine them at time \(t_{n+1}\). The simplest numerical scheme for doing this is the forward Euler method where (9.2.8) is evaluated at time \(t_n\) and
  \[
  U_t(x, t_n) \approx \frac{U^{n+1}(x) - U^n(x)}{\Delta t}. \tag{9.2.9}
  \]

  A simple Taylor’s series argument reveals that the local discretization error of such an approximation is \(O(\Delta t)\). Substituting (9.2.9) into (9.2.8) yields
  \[
  (V, \frac{U^{n+1} - U^n}{\Delta t}) + A(V, U^n) = (V, f^n), \quad \forall v \in S_0^N. \tag{9.2.10a}
  \]

  Evaluation of the inner products leads to
  \[
  M \frac{c^{n+1} - c^n}{\Delta t} + K^n c^n = f^n. \tag{9.2.10b}
  \]

  We have allowed the stiffness matrix and load vector to be functions of time. The mass matrix would always be independent of time for differential equations having the explicit form of (9.2.1a) as long as the spatial finite element mesh does not vary with time. The ODEs (9.2.10a,b) are implicit unless \(M\) is lumped. If lumping were used and, e.g., \(M \approx hI\) then \(c^{n+1}\) would be determined as
  \[
  c^{n+1} = c^n + \frac{\Delta t}{h} [I^n - K^n c^n].
  \]

  Assuming that \(c^n\) is known, we can determine \(c^{n+1}\) by inverting \(M\).

  Using the backward Euler method, we evaluate (9.2.8) at \(t_{n+1}\) and use the approximation (9.2.9) to obtain
  \[
  (V, \frac{U^{n+1} - U^n}{\Delta t}) + A(V, U^{n+1}) = (V, f^{n+1}), \quad \forall v \in S_0^N. \tag{9.2.11a}
  \]
\end{itemize}
9.2. Semi-Discrete Galerkin Problems

and

\[ M \frac{c^{n+1} - c^n}{\Delta t} + K^{n+1} c^{n+1} = l^{n+1}. \]  

(9.2.11b)

The backward Euler method is implicit regardless of whether or not lumping is used. Computation of \( c^{n+1} \) requires inversion of

\[ \frac{1}{\Delta t} M + K^{n+1}. \]

The most popular of these simple schemes uses a weighted average of the forward and backward Euler methods with weights of \( 1 - \theta \) and \( \theta \), respectively. Thus,

\[ (V, \frac{U^{n+1} - U^n}{\Delta t}) + (1 - \theta) A(V, U^n) + \theta A(V, U^{n+1}) = (1 - \theta)(V, f^n) + \theta(V, f^{n+1}), \]

\[ \forall V \in S_0^N. \]  

(9.2.12a)

and

\[ M \frac{c^{n+1} - c^n}{\Delta t} + (1 - \theta) K^n c^n + \theta K^{n+1} c^{n+1} = (1 - \theta) l^n + \theta l^{n+1}. \]  

(9.2.12b)

The forward and backward Euler methods are recovered by setting \( \theta = 0 \) and \( 1 \), respectively.

Let us regroup terms involving \( c^n \) and \( c^{n+1} \) in (9.2.12b) to obtain

\[ [M + \theta \Delta t K^{n+1}] c^{n+1} = [M - (1 - \theta) \Delta t K^n] c^n + \Delta t [(1 - \theta) l^n + \theta l^{n+1}]. \]

(9.2.12c)

Thus, determination of \( c^{n+1} \) requires inversion of

\[ M + \theta \Delta t K^{n+1}. \]

In one dimension, this system would typically be tridiagonal as with Example 9.2.1. In higher dimensions it would be sparse. Thus, explicit inversion would never be performed. We would just solve the sparse system (9.2.12c) for \( c^{n+1} \).

Taylor’s series calculations reveal that the global discretization error is

\[ \|c(t_n) - c^n\| = O(\Delta t) \]

for almost all choices of \( \theta \in [0, 1] \) [6]. The special choice \( \theta = 1/2 \) yields the Crank-Nicolson method which has a discretization error

\[ \|c(t_n) - c^n\| = O(\Delta t^2). \]

The foregoing discussion involved one-step methods. Multistep methods are also used to solve time-dependent finite element problems and we’ll describe them for an ODE in
the implicit form (9.2.7). The popular backward difference formulas (BDFs) approximate $y(t)$ in (9.2.7) by a $k$th degree polynomial $Y(t)$ that interpolates $y$ at the $k+1$ times $t_{n+1-i}$, $i = 0, 1, \ldots, k$. The derivative $\dot{y}$ is approximated by $\dot{Y}$. The Newton backward difference form of the interpolating is most frequently used to represent $Y$ [2, 3], but since we’re more familiar with Lagrangian interpolation we’ll write

$$y(t) \approx Y(t) = \sum_{i=0}^{k} y^{n+1-i} N_i(t), \quad t \in (t_{n+1-k}, t_{n+1}],$$

(9.2.13a)

where

$$N_i(t) = \prod_{j=0, j \neq i}^{k} \frac{t - t_{n+1-j}}{t_{n+1-i} - t_{n+1-j}}.$$  

(9.2.13b)

The basis (9.2.13b) is represented by the usual Lagrangian shape functions (cf. Section 2.4), so $N_i(t_{n+1-j}) = \delta_{ij}$.

Assuming $y^{n+1-i}$, $i = 1, 2, \ldots, k$, to be known, the unknown $y^{n+1}$ is determined by collocation at $t_{n+1}$. Thus, using (9.2.7)

$$f(t_{n+1}, Y(t_{n+1}), \dot{Y}(t_{n+1})) = 0.$$  

(9.2.14)

**Example 9.2.2.** The simplest BDF formula is obtained by setting $k = 1$ in (9.2.13) to obtain

$$Y(t) = y^{n+1} N_0(t) + y^n N_1(t),$$

$$N_0(t) = \frac{t - t_n}{t_{n+1} - t_n}, \quad N_1(t) = \frac{t - t_{n+1}}{t_n - t_{n+1}}.$$  

Differentiating $Y(t)$

$$\dot{Y}(t) = \frac{y^{n+1} - y^n}{t_{n+1} - t_n};$$

thus, the numerical method (9.2.13) is the backward Euler method

$$f(t_{n+1}, y^{n+1}, \frac{y^{n+1} - y^n}{t_{n+1} - t_n}) = 0.$$  

**Example 9.2.3.** The second-order BDF follows by setting $k = 2$ in (9.2.13) to get

$$Y(t) = y^{n+1} N_0(t) + y^n N_1(t) + y^{n-1} N_2(t)$$

$$N_0(t) = \frac{(t - t_n)(t - t_{n-1})}{2\Delta t^2}, \quad N_1(t) = \frac{(t - t_{n+1})(t - t_{n-1})}{-\Delta t^2},$$

$$N_2(t) = \frac{(t - t_{n+1})(t - t_n)}{2\Delta t^2},$$  

where time steps are of duration $\Delta t$. 


9.3. Finite Element Methods in Time

Differentiating and setting $t = t_{n+1}$

\[
\hat{N}_0(t_{n+1}) = \frac{3}{2\Delta t}, \quad \hat{N}_1(t_{n+1}) = -\frac{2}{\Delta t}, \quad \hat{N}_2(t_{n+1}) = \frac{1}{2\Delta t}.
\]

Thus,

\[
\dot{Y}(t_{n+1}) = \frac{3y_{n+1} - 4y^n + y^{n-1}}{2\Delta t}
\]

and the second-order BDF is

\[
f(t_{n+1}, y_{n+1}, \frac{3y_{n+1} - 4y^n + y^{n-1}}{2\Delta t}) = 0.
\]

Applying this method to (9.2.4a) yields

\[
M \frac{3c_{n+1} - 4c^n + c_{n-1}}{2\Delta t} + K_{n+1}c_{n+1} = l_{n+1}.
\]

Thus, computation of $c_{n+1}$ requires inversion of

\[
\frac{M}{2\Delta t} + K.
\]

Backward difference formulas through order six are available [2, 3, 6, 7, 8].

9.3 Finite Element Methods in Time

It is, of course, possible to use the finite element method in time. This can be done on space-time triangular or quadrilateral elements for problems in one space dimension; on hexahedra, tetrahedra, and prisms in two space dimensions; and on four-dimensional parallelepipeds and prisms in three space dimensions. However, for simplicity, we’ll focus on the time aspects of the space-time finite element method by assuming that the spatial discretization has already been performed. Thus, we’ll consider an ODE system in the form (9.2.4a) and construct a Galerkin problem in time by multiplying it by a test function $w \in L^2$ and integrating on $(t_n, t_{n+1})$ to obtain

\[
(w, M\dot{c})_n + (w, Kc)_n = (w, l)_n, \quad \forall w \in L^2(t_n, t_{n+1}), \quad (9.3.1a)
\]

where the $L^2$ inner product in time is

\[
(w, c)_n = \int_{t_n}^{t_{n+1}} w^T c \ dt. \quad (9.3.1b)
\]

Only first derivatives are involved in (9.2.4a); thus, neither the trial space for $c$ nor the test space for $w$ have to be continuous. For our initial method, let us assume that $c(t)$ is continuous at $t_n$. By assumption, $c(t_n)$ is known in this case and, hence, $w(t_n) = 0$. 
Example 9.3.1. Let us examine the method that results when \( c(t) \) and \( w(t) \) are linear on \((t_n, t_{n+1})\). We represent \( c(t) \) in the manner used for a spatial basis as

\[
c(\tau) \approx c^n N_n(\tau) + c^{n+1} N_{n+1}(\tau)
\]

where

\[
N_n(\tau) = \frac{1 - \tau}{2}, \quad N_{n+1}(\tau) = \frac{1 + \tau}{2}
\]

are hat functions in time and

\[
\tau = \frac{2t - t_n - t_{n+1}}{\Delta t}
\]

defines the canonical element in time. The test function

\[
w = N_{n+1}(\tau)[1, 1, \ldots, 1]^T
\]

vanishes at \( t_n \) (\( \tau = -1 \)) and is linear on \((t_n, t_{n+1})\).

Transforming the integrals in (9.3.1a) to \((-1, 1)\) using (9.3.2c) and using (9.3.2a,b,d) yields

\[
(w, M\dot{c})_n = \frac{\Delta t}{2} \int_{-1}^{1} \frac{1 + \tau}{2} M \frac{c^{n+1} - c^n}{\Delta t} d\tau,
\]

\[
(w, Kc)_n = \frac{\Delta t}{2} \int_{-1}^{1} \frac{1 + \tau}{2} K[c^n 1 - \tau + c^{n+1}/2] d\tau.
\]

(Again, we have written equality instead of \( \approx \) for simplicity.) Assuming that \( M \) and \( K \) are independent of time, we have

\[
(w, M\dot{c})_n = M \frac{c^{n+1} - c^n}{2},
\]

\[
(w, Kc)_n = \frac{\Delta t}{6} K(c^n + 2c^{n+1}).
\]

Substituting these into (9.3.1a)

\[
M \frac{c^{n+1} - c^n}{2} + \frac{\Delta t}{6} K(c^n + 2c^{n+1}) = \frac{\Delta t}{2} \int_{-1}^{1} \frac{1 + \tau}{2} 1(\tau) d\tau
\]

or, if \( I \) is approximated like \( c \),

\[
M \frac{c^{n+1} - c^n}{2} + \frac{\Delta t}{6} K(c^n + 2c^{n+1}) = \frac{\Delta t}{6}(I^n + 2I^{n+1}).
\]

Regrouping terms

\[
[M + \frac{2}{3} \Delta t K]c^{n+1} = [M - \frac{1}{3} \Delta t K]c^n + \frac{1}{3} \Delta t[I^n + 2I^{n+1}],
\]
we see that the piecewise-linear Galerkin method in time is a weighted average scheme (9.2.12c) with $\theta = 2/3$. Thus, at least to this low order, there is not much difference between finite difference and finite element methods. Other similarities appear in Problem 1 at the end of this section.

Low-order schemes such as (9.2.12) are popular in finite element packages. Our preference is for BDF or implicit Runge-Kutta software that control accuracy through automatic time step and order variation. Implicit Runge-Kutta methods may be derived as finite element methods by using the Galerkin method (9.3.1) with higher-order trial and test functions. Of the many possibilities, we’ll examine a class of methods where the trial function $c(t)$ is discontinuous.

**Example 9.3.2.** Suppose that $c(t)$ is a polynomial on $[t_n, t_{n+1}]$ with jump discontinuities at $t_n$, $n \geq 0$. When we need to distinguish left and right limits, we’ll use the notation

\[ c^n_- = \lim_{\epsilon \to 0^-} c(t_n - \epsilon), \quad c^n_+ = \lim_{\epsilon \to 0^+} c(t_n + \epsilon). \quad (9.3.4a) \]

With jumps at $t_n$, we’ll have to be more precise about the temporal inner product (9.3.1b) and we’ll define

\[ (u, v)_{n-} = \lim_{\epsilon \to 0^-} \int_{t_n - \epsilon}^{t_{n+1} - \epsilon} uv dt, \quad (u, v)_{n+} = \lim_{\epsilon \to 0^+} \int_{t_n + \epsilon}^{t_{n+1} - \epsilon} uv dt. \quad (9.3.4b) \]

The inner product $(u, v)_{n-}$ may be affected by discontinuities in functions at $t_n$, but $(u, v)_{n+}$ only involves integrals of smooth functions. In particular:

- $(u, v)_{n-} = (u, v)_{n+}$ when $u(t)$ and $v(t)$ are either continuous or have jump discontinuities at $t_n$;
- $(u, v)_{n-}$ exists and $(u, v)_{n+} = 0$ when either $u$ or $v$ are proportional to the delta function $\delta(t - t_n)$; and
- $(u, v)_{n-}$ doesn’t exist while $(v, u)_{n+} = 0$ when both $u$ and $v$ are proportional to $\delta(t - t_n)$.

Suppose, for example, that $v(t)$ is continuous at $t_n$ and $u(t) = \delta(t - t_n)$. Then

\[ (u, v)_{n-} = \lim_{\epsilon \to 0} \int_{t_n - \epsilon}^{t_{n+1} - \epsilon} \delta(t - t_n)v(t) dt = v(t_n). \]

The delta function can be approximated by a smooth function that depends on $\epsilon$ as was done in Section 3.2 to help explain this result.

Let us assume that $w(t)$ is continuous and write $c(t)$ in the form

\[ c(t) = c^n_- + [\tilde{c}(t) - c^n_-]H(t - t_n) \quad (9.3.5a) \]
where
\[ H(t) = \begin{cases} 1, & \text{if } t > 0 \\ 0, & \text{otherwise} \end{cases} \] (9.3.5b)
is the Heaviside function and \( \bar{c} \) is a polynomial in \( t \).

Differentiating
\[
\bar{c}(t) = [\bar{c}(t) - c^{n-}]\delta(t - t_n) + \bar{c}(t)H(t - t_n). \tag{9.3.5c}
\]

With the interpretation that inner products in (9.3.1) are of type (9.3.4), assume that \( w(t) \) is continuous and use (9.3.5) in (9.3.1a) to obtain
\[
w^T(t_n)M(t_n)(c^{n+} - c^{n-}) + (w, M\bar{c})_{n+} + (w, K\bar{c})_{n+} = (w, l)_{n+}, \quad \forall w \in H^1. \tag{9.3.6}
\]

The simplest discontinuous Galerkin method uses a piecewise constant \((p = 0)\) basis in time. Such approximations are obtained from (9.3.5a) by selecting
\[
\bar{c}(t) = c^{n+} = c^{(n+1)-}.
\]

Testing against the constant function
\[
w(t) = [1, 1, \ldots, 1]^T
\]
and assuming that \( M \) and \( K \) are independent of \( t \), (9.3.6) becomes
\[
M(c^{(n+1)-} - c^{n-}) + Kc^{(n+1)-} \Delta t = \int_{t_n}^{t_{n+1}} l(t) dt.
\]
The result is almost the same as the backward Euler formula (9.2.11b) except that the load vector \( l \) is averaged over the time step instead of being evaluated at \( t_{n+1} \).

With a linear \((p = 1)\) approximation for \( \bar{c}(t) \), we have
\[
\bar{c}(t) = c^{n+}N_n(t) + c^{(n+1)-}N_{n+1}(t)
\]
where \( N_{n+i}, i = 0, 1 \), are given by (9.3.2b). Selecting the basis for the test space as
\[
w_i(t) = N_{n+i}(t)[1, 1, \ldots, 1]^T, \quad i = 0, 1,
\]
assuming that \( M \) and \( K \) are independent of \( t \), and substituting the above approximations into (9.3.6), we obtain
\[
M(c^{n+} - c^{n-}) + \frac{1}{2}M(c^{(n+1)-} - c^{n+}) + \frac{\Delta t}{6}K(2c^{n+} + c^{(n+1)-}) = \int_{t_n}^{t_{n+1}} N_n l(t) dt
\]
and
\[
\frac{1}{2} M(\mathbf{c}^{n+1} - \mathbf{c}^n) + \frac{\Delta t}{6} K(2\mathbf{c}^{n+} + 2\mathbf{c}^{(n+1)^-}) = \int_{t_n}^{t_{n+1}} N_{n+1} \mathbf{l}(t) \, dt.
\]
Simplifying the expressions and assuming that \( \mathbf{l}(t) \) can be approximated by a linear function on \((t_n, t_{n+1})\) yields the system
\[
M(\frac{\mathbf{c}^{n+} + \mathbf{c}^{(n+1)^-}}{2} - \mathbf{c}^n) + \frac{\Delta t}{6} K(2\mathbf{c}^{n+} + \mathbf{c}^{(n+1)^-}) = \frac{\Delta t}{6}(2\mathbf{l}^n + \mathbf{l}^{(n+1)^-}),
\]
\[
M\frac{\mathbf{c}^{(n+1)^-} - \mathbf{c}^{n+}}{2} + \frac{\Delta t}{6} K(\mathbf{c}^{n+} + 2\mathbf{c}^{(n+1)^-}) = \frac{\Delta t}{6}(\mathbf{l}^n + 2\mathbf{l}^{(n+1)^-}).
\]

This pair of equations must be solved simultaneously for the two unknown solution vectors \( \mathbf{c}^{n+} \) and \( \mathbf{c}^{(n+1)^-} \). This is an implicit Runge-Kutta method.

**Problems**

1. Consider the Galerkin method in time with a continuous basis as represented by (9.3.1). Assume that the solution \( \mathbf{c}(t) \) is approximated by the linear function (9.3.2a-c) on \((t_n, t_{n+1})\) as in Example 9.3.1, but do not assume that the test space \( \mathbf{w}(t) \) is linear in time.

1.1. Specifying
\[
\mathbf{w}(\tau) = \omega(\tau)[1, 1, \ldots, 1]^T
\]
and assuming that \( M \) and \( K \) are independent of \( t \), show that (9.3.1a) is the weighted average scheme
\[
[M + \theta \Delta t K]\mathbf{c}^{n+1} = [M - (1 - \theta) \Delta t K]\mathbf{c}^n + \Delta t [(1 - \theta) \mathbf{l}^n + \theta \mathbf{l}^{n+1}]
\]
with
\[
\theta = \frac{\int_{-1}^{1} \omega(\tau) N_{n+1}^1(\tau) d\tau}{\int_{-1}^{1} \omega(\tau) d\tau}.
\]
When different trial and test spaces are used, the Galerkin method is called a *Petrov-Galerkin method*.

1.2. The entire effect of the test function \( \omega(t) \) is isolated in the weighting factor \( \theta \). Furthermore, no integration by parts was performed, so that \( \omega(t) \) need not be continuous. Show that the choices of \( \omega(t) \) listed in Table 9.3.1 correspond to the cited methods.

2. The discontinuous Galerkin method may be derived by simultaneously discretizing a partial differential system in space and time on \( \Omega \times (t - n, t_{(n+1)^-}) \). This form may have advantages when solving problems with rapid dynamics since the mesh may be either moved or regenerated without concern for maintaining continuity.
between time steps. Using (9.2.2a) as a model spatial finite element formulation, assume that test functions \(v(x,y,t)\) are continuous but that trial functions \(u(x,y,t)\) have jump discontinuities at \(t_n\). Assume Dirichlet boundary data and show that the space-time discontinuous Galerkin form of the problem is

\[
(v, u_t)_{ST} + (v(\cdot, t_n), u(\cdot, t_{n+}) - u(\cdot, t_{n-})) + A_{ST}(v, u) = (v, f)_{ST},
\forall v \in H^1_0(\Omega \times (t_{n+}, t_{n+1}-)),
\]

where

\[
(v, u)_{ST} = \int_{t_{n+}}^{t_{n+1}} \int_{\Omega} v u dx dy dt
\]

and

\[
A_{ST}(v, u) = (v_x, pu_x)_{ST} + (v_y, pu_y)_{ST} + (v, qu)_{ST}.
\]

In this form, the finite element problem is solved on the three-dimensional strips \(\Omega \times (t_{n-}, t_{n+1}-), n = 0, 1, \ldots\).

### 9.4 Convergence and Stability

In this section, we will study some theoretical properties of the discrete methods that were introduced in Sections 9.2 and 9.3. Every finite difference or finite element scheme for time integration should have three properties:

1. **Consistency**: the discrete system should be a good approximation of the differential equation.
2. **Convergence**: the solution of the discrete system should be a good approximation of the solution of the differential equation.
3. **Stability**: the solution of the discrete system should not be sensitive to small perturbations in the data.
Somewhat because they are open ended, finite difference or finite element approximations in time can be sensitive to small errors, e.g., introduced by round off. Let us illustrate the phenomena for the weighted average scheme (9.2.12c)

\[
[M + \theta \Delta t K] c^{n+1} = [M - (1 - \theta) \Delta t K] c^n + \Delta t [(1 - \theta) I^n + \theta I^{n+1}]. \quad (9.4.1)
\]

We have assumed, for simplicity, that \(K\) and \(M\) are independent of time.

Sensitivity to small perturbations implies a lack of stability as expressed by the following definition.

**Definition 9.4.1.** A finite difference scheme is *stable* if a perturbation of size \(\|\delta\|\) introduced at time \(t_n\) remains bounded for subsequent times \(t \leq T\) and all time steps \(\Delta t \leq \Delta t_0\).

We may assume, without loss of generality, that the perturbation is introduced at time \(t = 0\). Indeed, it is common to neglect perturbations in the coefficients and confine the analysis to perturbations in the initial data. Thus, in using Definition 9.4.1, we consider the solution of the related problem

\[
[M + \theta \Delta t K] \tilde{c}_n^{n+1} = [M - (1 - \theta) \Delta t K] \tilde{c}_n^n + \Delta t [(1 - \theta) I^n + \theta I^{n+1}],
\]

\[
\tilde{c}_0 = c^0 + \delta.
\]

Subtracting (9.4.1) from the perturbed system

\[
[M + \theta \Delta t K] \delta^{n+1} = [M - (1 - \theta) \Delta t K] \delta^n, \quad \delta^0 = \delta, \quad (9.4.2a)
\]

where

\[
\delta^n = \tilde{c}_n^n - c^n. \quad (9.4.2b)
\]

Thus, for linear problems, it suffices to apply Definition 9.4.1 to a homogeneous version of the difference scheme having the perturbation as its initial condition. With these restrictions, we may define stability in a more explicit form.

**Definition 9.4.2.** A linear difference scheme is *stable* if there exists a constant \(C > 0\) which is independent of \(\Delta t\) and such that

\[
\|\delta^n\| < C\|\delta^0\| \quad (9.4.3)
\]

as \(n \to \infty, \Delta t \to 0, t \leq T\).
Both Definitions 9.4.1 and 9.4.2 permit the initial perturbation to grow, but only by a bounded amount. Restricting the growth to finite times \( t < T \) ensures that the definitions apply when the solution of the difference scheme \( c^n \to \infty \) as \( n \to \infty \). When applying Definition 9.4.2, we may visualize a series of computations performed to time \( T \) with an increasing number of time steps \( M \) of shorter-and-shorter duration \( \Delta t \) such that \( T = M\Delta t \). As \( \Delta t \) is decreased, the perturbations \( \delta^n, n = 1, 2, \ldots, M \), should settle down and eventually not grow to more than \( C \) times the initial perturbation.

Solutions of continuous systems are often stable in the sense that \( c(t) \) is bounded for all \( t \geq 0 \). In this case, we need a stronger definition of stability for the discrete system.

**Definition 9.4.3.** The linear difference scheme (9.4.1) is **absolutely stable** if

\[
\|\delta^n\| < \|\delta^0\|. \tag{9.4.4}
\]

Thus, perturbations are not permitted to grow at all.

Stability analyses of linear constant coefficient difference equations such as (9.4.2) involve assuming a perturbation of the form

\[
\delta^n = (\lambda)^n r. \tag{9.4.5}
\]

Substituting into (9.4.2a) yields

\[
[M + \theta \Delta t K](\lambda)^{n+1}r = [M - (1 - \theta) \Delta t K](\lambda)^n r.
\]

Assuming that \( \lambda \neq 0 \) and \( M + \theta \Delta t K \) is not singular, we see that \( \lambda \) is an eigenvalue and \( r \) is an eigenvector of

\[
[M + \theta \Delta t K]^{-1}[M - (1 - \theta) \Delta t K]r_k = \lambda_k r_k, \quad k = 1, 2, \ldots, N. \tag{9.4.6}
\]

Thus, \( \delta^n \) will have the form (9.4.5) with \( \lambda = \lambda_k \) and \( r = r_k \) when the initial perturbation \( \delta^0 = r_k \). More generally, the solution of (9.4.2a) is the linear combination

\[
\delta^n = \sum_{k=1}^{N} \delta^0_k (\lambda_k)^n r_k \tag{9.4.7a}
\]

when the initial perturbation has the form

\[
\delta^0 = \sum_{k=1}^{N} \delta^0_k r_k. \tag{9.4.7b}
\]

Using (9.4.7a), we see that (9.4.2) will be absolutely stable when

\[
|\lambda_k| \leq 1, \quad k = 1, 2, \ldots, N. \tag{9.4.8}
\]
The eigenvalues and eigenvectors of many tridiagonal matrices are known. Thus, the analysis is often straightforward for one-dimensional problems. Analyses of two- and three-dimensional problems are more difficult; however, eigenvalue-eigenvector pairs are known for simple problems on simple regions.

Example 9.4.1. Consider the eigenvalue problem (9.4.6) and rearrange terms to get

$$[M + \theta \Delta t K] \lambda_k r_k = [M - (1 - \theta) \Delta t K] r_k$$

or

$$(\lambda_k - 1) Mr_k = -[\lambda_k \theta + (1 - \theta)] \Delta t Kr_k$$

or

$$-Kr_k = \mu_k Mr_k$$

where

$$\mu_k = \frac{\lambda_k - 1}{[\lambda_k \theta + (1 - \theta)] \Delta t}$$

Thus, $\mu_k$ is an eigenvalue and $r_k$ is an eigenvector of $-M^{-1}K$.

Let us suppose that $M$ and $K$ correspond to the mass and stiffness matrices of the one-dimensional heat conduction problem of Example 9.2.1. Then, using (9.2.4b,c), we have

$$\frac{p}{h} \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & \\ \\ & & & & -1 \\ \end{bmatrix} \begin{bmatrix} r_{k1} \\ r_{k2} \\ \vdots \\ r_{k,N-1} \end{bmatrix} = \frac{\mu_k h}{6} \begin{bmatrix} 4 & 1 & 1 & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & \\ & & & & 1 \\ \end{bmatrix} \begin{bmatrix} r_{k1} \\ r_{k2} \\ \vdots \\ r_{k,N-1} \end{bmatrix}.$$ 

The diffusivity $p$ and mesh spacing $h$ have been assumed constant. Also, with Dirichlet boundary conditions, the dimension of this system is $N - 1$ rather than $N$.

It is difficult to see in the above form, but writing this eigenvalue-eigenvector problem in component form

$$\frac{p}{h} (r_{j-1} - 2r_j + r_{j+1}) = \frac{\mu_k h}{6} (r_{j-1} + 4r_j + r_{j+1}), \quad j = 1, 2, \ldots, N - 1,$$

we may infer that the components of the eigenvector are

$$r_{kj} = \sin \frac{k\pi j}{N}, \quad j = 1, 2, \ldots, N - 1.$$ 

This guess of $r_k$ may be justified by the similarity of the discrete eigenvalue problem to a continuous one; however, we will not attempt to do this. Assuming it to be correct, we substitute $r_{kj}$ into the eigenvalue problem to find

$$\frac{p}{h} \left( \sin \frac{k\pi (j - 1)}{N} - 2 \sin \frac{k\pi j}{N} + \sin \frac{k\pi (j + 1)}{N} \right)$$

$$= \frac{\mu_k h}{6} \left( \sin \frac{k\pi (j - 1)}{N} + 4 \sin \frac{k\pi j}{N} + \sin \frac{k\pi (j + 1)}{N} \right).$$
Parabolic Problems

\[ = \frac{\mu_k h}{6} \left( \sin \frac{k\pi (j - 1)}{N} + 4 \sin \frac{k\pi j}{N} + \sin \frac{k\pi (j + 1)}{N} \right), \quad j = 1, 2, \ldots, N - 1. \]

But

\[ \sin \frac{k\pi (j - 1)}{N} + \sin \frac{k\pi (j + 1)}{N} = 2 \sin \frac{k\pi j}{N} \cos \frac{k\pi}{N} \]

and

\[ \frac{p}{h} (\cos \frac{k\pi}{N} - 1) \sin \frac{k\pi j}{N} = \frac{\mu_k h}{6} (\cos \frac{k\pi}{N} + 2) \sin \frac{k\pi j}{N}. \]

Hence,

\[ \mu_k = \left( \frac{6p}{h^2} \right) \left( \frac{\cos k\pi / N - 1}{\cos k\pi / N + 2} \right). \]

With \( \cos k\pi / N \) ranging on \([-1, 1]\), we see that \(-12p/h^2 \leq \mu_k \leq 0\). Determining \( \lambda_k \) in terms of \( \mu_k \)

\[ \lambda_k = \frac{1 + \mu_k (1 - \theta) \Delta t}{1 - \mu_k \theta \Delta t} = 1 + \frac{\mu_k \Delta t}{1 - \mu_k \theta \Delta t}. \]

We would like \( |\lambda_k| \leq 1 \) for absolute stability. With \( \mu_k \leq 0 \), we see that the requirement that \( \lambda_k \leq 1 \) is automatically satisfied. Demanding the \( \lambda_k \geq -1 \) yields

\[ |\mu_k| \Delta t (1 - 2\theta) \leq 2. \]

If \( \theta \geq 1/2 \) then \( 1 - 2\theta \leq 0 \) and the above inequality is satisfied for all choices of \( \mu_k \) and \( \Delta t \). Methods of this class are unconditionally absolutely stable. When \( \theta < 1/2 \), we have to satisfy the condition

\[ \frac{p \Delta t}{h^2} \leq \frac{1}{6(1 - 2\theta)}. \]

If we view this last relation as a restriction of the time step \( \Delta t \), we see that the forward Euler method (\( \theta = 0 \)) has the smallest time step. Since all other methods listed in Table 9.3.1 are unconditionally stable, there would be little value in using the forward Euler method without lumping the mass matrix. With lumping, the stability restriction of the forward Euler method actually improves slightly to \( p \Delta t / h^2 \leq 1/2 \).

Let us now turn to a more general examination of stability and convergence. Let’s again focus on our model problem: determine \( u \in H_0^1 \) satisfying

\[ (v, u_t) + A(v, u) = (v, f), \quad \forall v \in H_0^1, \quad t > 0, \quad (9.4.9a) \]

\[ (v, u) = (v, u^0), \quad \forall v \in H_0^1, \quad t = 0. \quad (9.4.9b) \]

The semi-discrete approximation consists of determining \( U \in S_0^N \subset H_0^1 \) such that

\[ (V, U_t) + A(V, U) = (V, f), \quad \forall V \in S_0^N, \quad t > 0, \quad (9.4.10a) \]
(V, U) = (V, u^0), \quad \forall V \in S_0^N, \quad t = 0. \quad (9.4.10b)

Trivial Dirichlet boundary data, again, simplifies the analysis.

Our first result establishes the absolute stability of the finite element solution of the semi-discrete problem (9.4.10) in the $L^2$ norm.

**Theorem 9.4.1.** Let $\delta \in S_0^N$ satisfy

\begin{align}
(V, \delta_t) + A(V, \delta) &= 0, \quad \forall V \in S_0^N, \quad t > 0, \quad (9.4.11a) \\
(V, \delta) &= (V, \delta^0), \quad \forall V \in S_0^N, \quad t = 0. \quad (9.4.11b)
\end{align}

Then

$$\|\delta(\cdot, \cdot, t)\|_0 \leq \|\delta^0\|_0, \quad t > 0. \quad (9.4.11c)$$

**Remark 1.** With $\delta(x, t)$ being the difference between two solutions of (9.4.10a) satisfying initial conditions that differ by $\delta^0(x)$, the loading $(V, f)$ vanishes upon subtraction (as with (9.4.2)).

**Proof.** Replace $V$ in (9.4.11a) by $\delta$ to obtain

$$(\delta, \delta_t) + A(\delta, \delta) = 0,$$

or

$$\frac{1}{2} \frac{d}{dt} \|\delta\|_0^2 + A(\delta, \delta) = 0.$$

Integrating

$$\|\delta(\cdot, \cdot, t)\|_0^2 = \|\delta(\cdot, \cdot, 0)\|_0^2 - 2 \int_0^t A(\delta, \delta) \, dr.$$

The result (9.4.11c) follows by using the initial data (9.4.11b) and the non-negativity of $A(\delta, \delta)$.

We’ve discussed stability at some length, so now let us turn to the concept of convergence. Convergence analyses for semi-discrete Galerkin approximations parallels the lines of those for elliptic systems. Let us, as an example, establish convergence for piecewise-linear solutions of (9.4.10) to solutions of (9.4.9).

**Theorem 9.4.2.** Let $S_0^N$ consist of continuous piecewise-linear polynomials on a family of uniform meshes $\Delta_h$ characterized by their maximum element size $h$. Then there exists a constant $C > 0$ such that

$$\max_{t \in [0, T]} \|u - U\|_0 \leq C (1 + \log \frac{T}{h^2}) h^2 \max_{t \in [0, T]} \|u\|_2. \quad (9.4.12)$$
Proof. Create the auxiliary problem: determine $W \in S_0^N$ such that

$$-(V, W(\cdot, \cdot, \tau)) + A(V, W(\cdot, \cdot, \tau)) = 0, \quad \forall V \in S_0^N, \quad \tau \in (0, t), \quad (9.4.13a)$$

$$W(x, y, t) = E(x, y, t) = U(x, y, t) - \bar{U}(x, y, t), \quad (9.4.13b)$$

where $\bar{U} \in S_0^N$ satisfies

$$A(V, u(\cdot, \cdot, \tau) - \bar{U}(\cdot, \cdot, \tau)) = 0, \quad \forall V \in S_0^N, \quad \tau \in (0, T). \quad (9.4.13c)$$

We see that $W$ satisfies a terminal value problem on $0 \leq \tau \leq t$ and that $\bar{U}$ satisfies an elliptic problem with $\tau$ as a parameter.

Consider the identity

$$\frac{d}{d\tau} (W, E) = (W_\tau, E) + (W, E_\tau).$$

Integrate and use (9.4.13b)

$$\|E(\cdot, \cdot, t)\|_0^2 = (W, E(\cdot, \cdot, 0)) + \int_0^t [(W_\tau, E) + (W, E_\tau)] d\tau.$$  

Use (9.4.13a) with $V$ replaced by $E$

$$\|E(\cdot, \cdot, t)\|_0^2 = (W, E(\cdot, \cdot, 0)) + \int_0^t [A(W, E) + (W, E_\tau)] d\tau. \quad (9.4.14)$$

Setting $v$ in (9.4.9) and $V$ in (9.4.10) to $W$ and subtracting yields

$$(W, u_\tau - U_\tau) + A(W, u - U) = 0, \quad \tau > 0,$$

$$(W, u - U)(0) = 0, \quad \tau = 0.$$  

Add these results to (9.4.14) and use (9.4.13b) to obtain

$$\|E(\cdot, \cdot, t)\|_0^2 = (W, \theta(\cdot, \cdot, 0)) + \int_0^t [A(W, \theta) + (W, \theta_\tau)] d\tau,$$

where

$$\theta = u - \bar{U}.$$  

The first term in the integrand vanishes by virtue of (9.4.13c). The second term is integrated by parts to obtain

$$\|E(\cdot, \cdot, t)\|_0^2 = (W, \theta(\cdot, \cdot, t)) - \int_0^t (W_\tau, \theta) d\tau. \quad (9.4.15a)$$
This result can be simplified slightly by use of Cauchy's inequality \(|\langle W, V \rangle| \leq \|W\|_0 \|V\|_0\) to obtain
\[
\|E(\cdot, \cdot, t)\|_0^2 \leq \|W(\cdot, \cdot, t)\|_0 \|\theta(\cdot, \cdot, t)\|_0 + \int_0^t \|W_\tau\|_0 \|\theta\|_0 d\tau.
\] (9.4.15b)

Introduce a basis on \(S_0^N\) and write \(W\) in the standard form
\[
W(x, y, \tau) = \sum_{j=0}^N c_j(\tau) \phi_j(x, y). \tag{9.4.16}
\]

Substituting (9.4.16) into (9.4.13a) and following the steps introduced in Section 9.2, we are led to
\[
-M\dot{c} + Kc = 0, \tag{9.4.17a}
\]
where
\[
M_{ij} = (\phi_i, \phi_j), \tag{9.4.17b}
\]
\[
K_{ij} = A(\phi_i, \phi_j), \quad i, j = 1, 2, \ldots, N. \tag{9.4.17c}
\]

Assuming that the stiffness matrix \(K\) is independent of \(\tau\), (9.4.17a) may be solved exactly to show that (cf. Lemmas 9.4.1 and 9.4.2 which follow)
\[
\|W(\cdot, \cdot, \tau)\|_0 \leq \|E(\cdot, \cdot, t)\|_0, \quad 0 < \tau \leq t, \tag{9.4.18a}
\]
\[
\int_0^t \|W_\tau\|_0 d\tau \leq C(1 + |\log t|) \|E(\cdot, \cdot, t)\|_0. \tag{9.4.18b}
\]

Equation (9.4.18a) is used in conjunction with (9.4.15b) to obtain
\[
\|E(\cdot, \cdot, t)\|_0^2 \leq (\|E(\cdot, \cdot, t)\|_0 + \int_0^t \|W_\tau\|_0 d\tau) \max_{\tau \in [0, t]} \|\theta(\cdot, \cdot, \tau)\|_0.
\]

Now, using (9.4.18b)
\[
\|E(\cdot, \cdot, t)\|_0 \leq C(1 + |\log t|) \max_{\tau \in [0, t]} \|\theta(\cdot, \cdot, \tau)\|_0. \tag{9.4.19}
\]

Writing
\[u - U = u - \bar{U} + \bar{U} - U = \theta - E\]
and taking an \(L^2\) norm
\[
\|u - U\|_0 \leq \|\theta\|_0 + \|E\|_0.
\]
Using (9.4.19)

\[ \|u - U\|_0 \leq C(1 + |\log \frac{t}{h^2}|) \max_{\tau \in [0,t]} \|\theta(\cdot, \cdot, \tau)\|_0. \]  

(9.4.20a)

Finally, since \( \theta \) satisfies the elliptic problem (9.4.13c), we can use Theorem 7.2.4 to write

\[ \|\theta(\cdot, \cdot, \tau)\|_0 \leq Ch^2 \|u(\cdot, \cdot, \tau)\|_2. \]  

(9.4.20b)

Combining (9.4.20a) and (9.4.20b) yields the desired result (9.4.12).

The two results that were used without proof within Theorem 9.4.2 are stated as Lemmas.

**Lemma 9.4.1.** Under the conditions of Theorem 9.4.2, there exists a constant \( C > 0 \) such that

\[ A(V, V) \leq \frac{C}{h^2} \|V\|_0^2, \quad \forall V \in S_0^N. \]  

(9.4.21)

**Proof.** The result can be inferred from Example 9.2.1; however, a more formal proof is given by Johnson [9], Chapter 7.

Instead of establishing (9.4.18b), we'll examine a slightly more general situation. Let \( c \) be the solution of

\[ M\dot{c} + Kc = 0, \quad t > 0, \quad c(0) = c^0. \]  

(9.4.22)

The mass and stiffness matrices \( M \) and \( K \) are positive definite, so we can diagonalize (9.4.22). In particular, let \( \Lambda \) be a diagonal matrix containing the eigenvalues of \( M^{-1}K \) and \( R \) be a matrix whose columns are the eigenvectors of the same matrix, i.e.,

\[ M^{-1}KR = RA. \]  

(9.4.23a)

Further let

\[ d(t) = R^{-1}c(t). \]  

(9.4.23b)

Then (9.4.22) can be written in the diagonal form

\[ \dot{d} + \Lambda d = 0 \]  

(9.4.24a)

by multiplying it by \((MR)^{-1}\) and using (9.4.23a,b). The initial conditions generally remain coupled through (9.4.23a,b), i.e.,

\[ d(0) = d^0 = R^{-1}c^0. \]  

(9.4.24b)

With these preliminaries, we state the desired result.
Lemma 9.4.2. If $d(t)$ is the solution of (9.4.24) then
\[ |\dot{d}| + |A d| \leq \frac{C|d^0|}{t}, \quad t > 0, \tag{9.4.25a} \]
where $|d| = \sqrt{d^T d}$. If, in addition,
\[ \max_{\xi \neq 0} \frac{|A \xi|}{|\xi|} \leq \frac{C}{h^2} \tag{9.4.25b} \]
then
\[ \int_0^T (|\dot{d}| + |A d|) dt \leq C (1 + |\log T h^2|)|d^0|. \tag{9.4.25c} \]
Proof. cf. Problem 1. \hfill \square

Problems

1. Prove Lemma 9.4.2.

9.5 Convection-Diffusion Systems

Problems involving convection and diffusion arise in fluid flow and heat transfer. Let us consider the model problem
\[ u_t + \omega \cdot \nabla u = \nabla \cdot (p \nabla u) \tag{9.5.1a} \]
where $\omega = [\omega_1, \omega_2]^T$ is a velocity vector. Written in scalar form, (9.5.1a) is
\[ u_t + \omega_1 u_x + \omega_2 u_y = (pu_x)_x + (pu_y)_y. \tag{9.5.1b} \]
The vorticity transport equation of fluid mechanics has the form of (9.5.1). In this case, $u$ would represent the vorticity of a two-dimensional flow.

If the magnitude of $\omega$ is small relative to the magnitude of the diffusivity $p(x, y)$, then the standard methods that we have been studying work fine. This, however, is not the case in many applications and, as indicated by the following example, standard finite element methods can produce spurious results.

Example 9.5.1 [1]. Consider the steady, one-dimensional, convection-diffusion equation
\[ -\epsilon u'' + u' = 0, \quad 0 < x < 1, \tag{9.5.2a} \]
with Dirichlet boundary conditions
\[ u(0) = 1, \quad u(1) = 2. \tag{9.5.2b} \]
The exact solution of this problem is

\[ u(x) = 1 + \frac{e^{-(1-x)/\epsilon} - e^{-1/\epsilon}}{1 - e^{-1/\epsilon}}. \]  

(9.5.2c)

If \( 0 < \epsilon \ll 1 \) then, as shown by the solid line in Figure 9.5.1, the solution features a boundary layer near \( x = 1 \). At points removed from an \( O(\epsilon) \) neighborhood of \( x = 1 \), the solution is smooth with \( u \approx 1 \). Within the boundary layer, the solution rises sharply from its unit value to \( u = 2 \) at \( x = 1 \).

![Figure 9.5.1: Solutions of (9.5.2) with \( \epsilon = 10^{-3} \). The exact solution is shown as a solid line. Piecewise-linear Galerkin solutions with 10- and 11-element meshes are shown as dashed and dashed-dotted lines, respectively [1].](image)

The term \( \epsilon u'' \) is diffusive while the term \( u' \) is convective. With a small diffusivity \( \epsilon \), convection dominates diffusion outside of the narrow \( O(\epsilon) \) boundary layer. Within this layer, diffusion cannot be neglected and is on an equal footing with convection. This simple problem will illustrate many of the difficulties that arise when finite element methods are applied to convection-diffusion problems while avoiding the algebraic and geometric complexities of more realistic problems.

Let us divide \([0, 1]\) into \( N \) elements of width \( h = 1/N \). Since the solution is slowly varying over most of the domain, we would like to choose \( h \) to be significantly larger than
the boundary layer thickness. This could introduce large errors within the boundary layer which we assume can be reduced by local mesh refinement. This strategy is preferable to the alternative of using a fine mesh everywhere when the solution is only varying rapidly within the boundary layer.

Using a piecewise-linear basis, we write the finite element solution as

\[ U(x) = \sum_{j=0}^{N} c_j \phi_j(x), \quad c_0 = 1, \quad c_N = 2, \] (9.5.3a)

where

\[ \phi_k(x) = \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}}, & \text{if } x_{k-1} < x \leq x_k \\ \frac{x_{k+1}-x_k}{x_{k+1}-x_k}, & \text{if } x_k < x \leq x_{k+1} \\ 0, & \text{otherwise} \end{cases} \] (9.5.3b)

The coefficients \( c_0 \) and \( c_N \) are constrained so that \( U(x) \) satisfies the essential boundary conditions (9.5.2b).

The Galerkin problem for (9.5.2) consists of determining \( U(x) \in S_0^N \) such that

\[ \epsilon(\phi_i', U') + (\phi_i, U') = 0, \quad i = 1, 2, \ldots, N - 1. \] (9.5.4a)

Since this problem is similar to Example 9.2.1, we’ll omit the development and just write the inner products

\[ (\phi_i', U') = \frac{\epsilon}{h} (c_{i-1} - 2c_i + c_{i+1}), \] (9.5.4b)

\[ (\phi_i, U') = \frac{c_{i+1} - c_{i-1}}{2}. \] (9.5.4c)

Thus, the discrete finite element system is

\[ (1 - \frac{h}{2\epsilon})c_{i+1} - 2c_i + (1 + \frac{h}{2\epsilon})c_{i-1} = 0, \quad i = 1, 2, \ldots, N - 1. \] (9.5.4d)

The solution of this second-order, constant-coefficient difference equation is

\[ c_i = 1 + \frac{1 - \beta^i}{1 - \beta^N}, \quad i = 0, 1, \ldots, N, \] (9.5.4e)

\[ \beta = \frac{1 + h/2\epsilon}{1 - h/2\epsilon}. \] (9.5.4f)

The quantity \( h/2\epsilon \) is called the cell Peclet or cell Reynolds number. If \( h/2\epsilon \ll 1 \), then

\[ \beta = 1 + \frac{h}{\epsilon} + O\left(\frac{h}{\epsilon}\right)^2 = e^{h/\epsilon} + O\left(\left(\frac{h}{\epsilon}\right)^2\right). \]
which is the correct solution. However, if $h/2\varepsilon \gg 1$, then $\beta \approx -1$ and

$$c_i \approx \begin{cases} 1, & \text{if } i \text{ is even} \\ 2, & \text{if } i \text{ is odd} \end{cases}$$

when $N$ is odd and

$$c_i \approx \begin{cases} (N + i)/N, & \text{if } i \text{ is even} \\ O(1/\varepsilon), & \text{if } i \text{ is odd} \end{cases}$$

when $N$ is even. These two obviously incorrect solutions are shown with the correct results in Figure 9.5.1.

Let us try to remedy the situation. For simplicity, we’ll stick with an ordinary differential equation and consider a two-point boundary value problem of the form

$$L[u] = -\varepsilon u'' + \omega u' + qu = f, \quad 0 < x < 1, \quad (9.5.5a)$$

$$u(0) = u(1) = 0. \quad (9.5.5b)$$

Let us assume that $u, v \in H^1_0$ with $u'$ and $v'$ being continuous except, possibly, at $x = \xi \in (0, 1)$. Multiplying (9.5.5a) by $v$ and integrating the second derivative terms by parts yields

$$(v, L[u]) = A(v, u) + [\varepsilon u'v]_{x=\xi} \quad (9.5.6a)$$

where

$$A(v, u) = \varepsilon (v', u') + (v, \omega u') + (v, qu), \quad (9.5.6b)$$

$$[Q]_{x=\xi} = \lim_{\delta \to 0} [Q(\xi + \delta) - Q(\xi - \delta)]. \quad (9.5.6c)$$

We must be careful because the “strain energy” $A(v, u)$ is not an inner product since $A(u, u)$ need not be positive definite. We’ll use the inner product notation here for convenience.

Integrating the first two terms of (9.5.6b) by parts

$$(v, L[u]) = (L^*[v], u) - [\varepsilon (v'u - u'v) + \omega vu]_{x=\xi}$$

or, since $u$ and $v$ are continuous

$$(v, L[u]) = (L^*[v], u) - [\varepsilon (v'u - u'v)]_{x=\xi} \quad (9.5.7a)$$

The differential equation

$$L^*[v] = -\varepsilon v'' - (\omega v)' +qv. \quad (9.5.7b)$$

with the boundary conditions $v(0) = v(1) = 0$ is called the adjoint problem and the operator $L^*[\cdot]$ is called the adjoint operator.
**Definition 9.5.1.** A Green’s function $G(\xi, x)$ for the operator $\mathcal{L} \cdot$ is the continuous function that satisfies

\[
\mathcal{L}^*[G(\xi, x)] = -\epsilon G_{xx} - (\omega G)_x + qG = 0, \quad x \in (0, \xi) \cup (\xi, 1), \quad (9.5.8a)
\]

\[
G(\xi, 0) = G(\xi, 1) = 0 \quad (9.5.8b)
\]

\[
[G_x(\xi, x)]_{x=\xi} = -\frac{1}{\epsilon}. \quad (9.5.8c)
\]

Evaluating (9.5.7a) with $v(x) = G(\xi, x)$ while using (9.5.5a, 9.5.8) and assuming that $u'(x) \in H^1(0, 1)$ gives the familiar relationship

\[
u(\xi) = (\mathcal{L}u, G(\xi, \cdot)) = \int_0^1 G(\xi, x)f(x)dx. \quad (9.5.9a)
\]

A more useful expression for our present purposes is obtained by combining (9.5.7a) and (9.5.6a) with $v(x) = G(\xi, x)$ to obtain

\[
u(\xi) = A(u, G(\xi, \cdot)). \quad (9.5.9b)
\]

As usual, Galerkin and finite element Galerkin problems for (9.5.5a) would consist of determining $u \in H^1_0$ or $U \in S_0^N \subset H^1_0$ such that

\[
A(v, u) = (v, f), \quad \forall v \in H^1_0, \quad (9.5.10a)
\]

and

\[
A(V, U) = (V, f), \quad \forall v \in S_0^N. \quad (9.5.10b)
\]

Selecting $v = V$ in (9.5.10a) and subtracting (9.5.10b) yields

\[
A(V, e) = 0, \quad \forall v \in S_0^N, \quad (9.5.10c)
\]

where

\[
e(x) = u(x) - U(x). \quad (9.5.10d)
\]

Equation (9.5.9b) did not rely on the continuity of $u'(x)$; hence, it also holds when $u$ is replaced by either $U$ or $e$. Replacing $u$ by $e$ in (9.5.9b) yields

\[
e(\xi) = A(e, G(\xi, \cdot)). \quad (9.5.11a)
\]
Subtracting (9.5.10c)

\[ e(\xi) = A(\epsilon, G(\xi, \cdot) - V). \] 
(9.5.11b)

Assuming that \( A(v, u) \) is continuous in \( H^1 \), we have

\[ |e(\xi)| \leq C\|\epsilon\|_1\|G(\xi, \cdot) - V\|_1. \] 
(9.5.11c)

Expressions (9.5.11b,c) relate the local error at a point \( \xi \) to the global error. Equation (9.5.11c) also explains superconvergence. From Theorem 7.2.3 we know that \( \|e\|_1 = O(h^p) \) when \( S^N \) consists of piecewise polynomials of degree \( p \) and \( u \in H^{p+1} \). The test function \( V \) is also an element of \( S^N \); however, \( G(\xi, x) \) cannot be approximated to the same precision as \( u \) because it may be less smooth. To elaborate further, consider

\[ \|G(\xi, \cdot) - V\|_1^2 = \sum_{j=1}^{N} \|G(\xi, \cdot) - V\|_{1,j}^2 \]

where

\[ \|u\|_{1,j}^2 = \int_{x_{j-1}}^{x_j} [(u')^2 + u^2] \text{dx}. \]

If \( \xi \in (x_{k-1}, x_k), k = 1, 2, \ldots, N, \) then the discontinuity in \( G_x(\xi, x) \) occurs on some interval and \( G(\xi, x) \) cannot be approximated to high order by \( V \). If, on the other hand, \( \xi = x_k, k = 0, 1, \ldots, N, \) then the discontinuity in \( G_x(\xi, x) \) is confined to the mesh and \( G(\xi, x) \) is smooth on every subinterval. Thus, in this case, the Green’s function can be approximated to \( O(h^p) \) by the test function \( V \) and, using (9.5.11c), we have

\[ u(x_k) = Ch^{2p}, \quad k = 0, 1, \ldots, N. \] 
(9.5.12)

The solution at the vertices is converging to a much higher order than it is globally.

Equation (9.5.11c) suggests that there are two ways of minimizing the pointwise error. The first is to have \( U \) be a good approximation of \( u \) and the second is to have \( V \) be a good approximation of \( G(\xi, x) \). If the problem is not singularly perturbed, then the two conditions are the same. However, when \( \epsilon \ll 1 \), the behavior of the Green’s function is hardly polynomial. Let us consider two simple examples.

**Example 9.5.2 [5]**. Consider (9.5.5) in the case when \( \omega(x) > 0, x \in [0, 1] \). Balancing the first two terms in (9.5.5a) implies that there is a boundary layer near \( x = 1 \); thus, at points other than the right endpoint, the small second derivative terms in (9.5.5) may be neglected and the solution is approximately

\[ \omega u'_R + qu_R = f, \quad 0 < x < 1, \quad u_R(0) = 0, \]
where $u_R$ is called the reduced solution. Near $x = 1$ the reduced solution must be corrected by a boundary layer that brings it from its limiting value of $u_R(1)$ to zero. Thus, for $0 < \varepsilon \ll 1$, the solution of (9.5.5) is approximately

$$u(x) \sim u_R(x) - u_R(1)e^{-(1-x)/\varepsilon}.$$

Similarly, the Green’s function (9.5.8) has boundary layers at $x = 0$ and $x = \xi^-$. At points other than these, the second derivative terms in (9.5.8a) may be neglected and the Green’s function satisfies the reduced problem

$$-(\omega G_R)' + qG_R = 0, \quad x \in (0, \xi) \cup (\xi, 1), \quad G_R(\xi, x) \in C(0, 1), \quad G_R(\xi, 1) = 0.$$

Boundary layer jumps correct the reduced solution at $x = 0$ and $x = \xi$ and determine an asymptotic approximation of $G(\xi, x)$ as

$$G(\xi, x) \sim c(\xi) \begin{cases} G_R(\xi, x) - G_R(\xi, 0)e^{-\omega(0)x/\varepsilon}, & \text{if } x \leq \xi \\ e^{-(x-\xi)\omega(\xi)/\varepsilon}, & \text{if } x > \xi \end{cases}.$$

The function $c(\xi)$ is given in Flaherty and Mathon [5].

Knowing the Green’s function, we can construct test functions that approximate it accurately. To be specific, let us write it as

$$G(\xi, x) = \sum_{j=1}^{N} G(\xi, x_j)\psi_j(x) \quad (9.5.13)$$

where $\psi_j(x)$, $j = 0, 1, \ldots, N$, is a basis. Let us consider (9.5.5) and (9.5.8) with $\omega > 0$, $x \in [0, 1]$. Approximating the Green’s function for arbitrary $\xi$ is difficult, so we’ll restrict $\xi$ to $x_k$, $k = 0, 1, \ldots, N$, and establish the goal of minimizing the pointwise error of the solution. Mapping each subinterval to a canonical element, the basis $\psi_j(x)$, $x \in (x_{j-1}, x_{j+1})$ is

$$\psi_j(x) = \hat{\psi}\left(\frac{x - x_j}{h}\right) \quad (9.5.14a)$$

where

$$\hat{\psi}(s) = \begin{cases} \frac{1-e^{-\rho(1+s)}}{1-e^{-\rho}}, & \text{if } -1 \leq s < 0 \\ \frac{e^{-\rho(1+s)} - e^{-\rho}}{1-e^{-\rho}}, & \text{if } 0 \leq s < 1 \\ 0, & \text{otherwise} \end{cases} \quad (9.5.14b)$$

where

$$\rho = \frac{h\tilde{\omega}}{\varepsilon} \quad (9.5.14c)$$
Figure 9.5.2: Canonical basis element $\hat{\psi}(s)$ for $\rho = 0$, 10, and 100 (increasing steepness).

is the cell *Peclet number*. The value of $\bar{\omega}$ will remain undefined for the moment. The canonical basis element $\hat{\psi}(s)$ is illustrated in Figure 9.5.2. As $\rho \to 0$ the basis (9.5.14b) becomes the usual piecewise-linear hat function

$$
\hat{\psi}(s) = \frac{1}{2} \begin{cases} 
1 + s, & \text{if } -1 \leq s < 0 \\
1 - s, & \text{if } 0 \leq s < 1 \\
0, & \text{otherwise}
\end{cases}
$$

As $\rho \to \infty$, (9.5.14b) becomes the piecewise-constant function

$$
\hat{\psi}(s) = \begin{cases} 
1, & \text{if } -1 < s \leq 0 \\
0, & \text{otherwise}
\end{cases}
$$

The limits of this function are nonuniform at $s = -1, 0$.

We’re now in a position to apply the Petrov-Galerkin method with $U \in S_0^N$ and $V \in \dot{S}_0^N$ to (9.5.5). The trial space $S^N$ will consist of piecewise linear functions and, for the moment, the test space will remain arbitrary except for the assumptions

$$
\psi_j(x) \in H^1[0,1], \quad \psi_j(x_k) = \delta_{jk}, \quad \int_{-1}^{1} \hat{\psi}(s) ds = 1, \quad j, k = 1, 2, \ldots, N - 1.
$$

(9.5.15)
9.5. Convection-Diffusion Systems

The Petrov-Galerkin system for (9.5.5) is

\[ \epsilon(\psi_i', U') + (\psi_i', \omega U') + (\psi_i, qU) = (\psi_i, f), \quad i = 1, 2, \ldots, N - 1. \]  

(9.5.16)

Let us use node-by-node evaluation of the inner products in (9.5.16). For simplicity, we’ll assume that the mesh is uniform with spacing \( h \) and that \( \omega \) and \( q \) are constant. Then

\[ \epsilon(\psi_i', U') = \frac{\epsilon}{h} \int_{-1}^{1} \hat{\psi}'(s) \hat{U}'(s) ds \]

where \( \hat{U}(s) \) is the mapping of \( U(x) \) onto the canonical element \(-1 \leq s \leq 1\). With a piecewise linear basis for \( \hat{U} \) and the properties noted in (9.5.15) for \( \psi_j \), we find

\[ \epsilon(\psi_i', U') = -\frac{\epsilon}{h} \delta^2 c_i. \]  

(9.5.17a)

We’ve introduced the central difference operator

\[ \delta c_i = c_{i+1/2} - c_{i-1/2} \]  

(9.5.17b)

for convenience. Thus,

\[ \delta^2 c_i = \delta(\delta c_i) = c_{i+1} - 2c_i + c_{i-1}. \]  

(9.5.17c)

Considering the convective term,

\[ \omega(\psi_i, U') = \omega \int_{-1}^{1} \hat{\psi}(s) \hat{U}'(s) ds = \omega(\mu \delta - \gamma \delta^2/2)c_i \]  

(9.5.18a)

where \( \mu \) is the averaging operator

\[ \mu c_i = (c_{i+1/2} + c_{i-1/2})/2. \]  

(9.5.18b)

Thus,

\[ \mu \delta c_i = \mu(\delta c_i) = (c_{i+1} - c_{i-1})/2. \]  

(9.5.18c)

Additionally,

\[ \gamma = -\int_{0}^{1} [\hat{\psi}(s) - \hat{\psi}(-s)] ds \]  

(9.5.18d)

Similarly

\[ q(\psi_i, U) = q h \int_{-1}^{1} \hat{\psi}(s) \hat{U}(s) ds = qh(1 - \beta \mu \delta + \alpha \delta^2/2)c_i \]  

(9.5.19a)
where

\[ \alpha = \int_{-1}^{1} |s| \hat{\psi}(s) ds, \quad (9.5.19b) \]

\[ \beta = -\int_{-1}^{1} s \hat{\psi}(s) ds. \quad (9.5.19c) \]

Finally, if \( f(x) \) is approximated by a piecewise-linear polynomial, we have

\[ (\psi_i, f) \approx h(1 - \beta \mu \delta + \alpha \delta^2 / 2) f_i \quad (9.5.20) \]

where \( f_i = f(x_i) \).

Substituting (9.5.17a), (9.5.18a), (9.5.19a), and (9.5.20) into (9.5.16) gives a difference equation for \( c_k \), \( k = 1, 2, \ldots, N - 1 \). Rather than facing the algebraic complexity, let us continue with the simpler problem of Example 9.5.1.

**Example 9.5.3.** Consider the boundary value problem (9.5.2). Thus, \( q = f(x) = 0 \) in (9.5.17-9.5.20) and we have

\[ \epsilon \psi''_i + \omega \psi'_i = -\frac{\epsilon}{h} \delta^2 c_i + \omega (\mu \delta - \gamma \delta^2 / 2) c_i, \quad i = 1, 2, \ldots, N - 1, \quad (9.5.21a) \]

or, using (9.5.14c), (9.5.17c), and (9.5.18c)

\[ -\frac{1}{2}(\gamma + \frac{2}{\rho})(c_{i+1} - 2c_i + c_{i-1}) + \frac{c_{i+1} - c_{i-1}}{2} = 0, \quad i = 1, 2, \ldots, N - 1. \quad (9.5.21b) \]

This is to be solved with the boundary conditions

\[ c_0 = 1, \quad c_N = 2. \quad (9.5.21c) \]

The exact solution of this second-order constant-coefficient difference equation is

\[ c_i = 1 + \frac{1 - \zeta^i}{1 - \zeta^N}, \quad i = 0, 1, \ldots, N. \quad (9.5.22a) \]

where

\[ \zeta = \frac{\gamma + 2 / \rho + 1}{\gamma + 2 / \rho - 1}. \quad (9.5.22b) \]

In order to avoid the spurious oscillations found in Example 9.5.1, we’ll insist that \( \zeta > 0 \). Using (9.5.22b), we see that this requires

\[ \gamma > \text{sgn} \rho - \frac{2}{\rho}. \quad (9.5.22c) \]

Some specific choices of \( \gamma \) follow:
1. *Galerkin’s method*, $\gamma = 0$. In this case,

$$\hat{\psi}(s) = \hat{\phi}(s) = \frac{1 - |s|}{2}.$$  

Using (9.5.22), this method is oscillation free when

$$\frac{2}{|\rho|} > 1.$$  

From (9.5.14c), this requires $h < 2|\epsilon/\omega|$. For small values of $|\epsilon/\omega|$, this would be too restrictive.

2. *Il’in’s scheme*. In this case, $\hat{\psi}(s)$ is given by (9.5.14b) and

$$\gamma = \coth \frac{\rho}{2} - \frac{2}{\rho}.$$  

This scheme gives the exact solution at element vertices for all values of $\rho$. Either this result or the use of (9.5.22c) indicates that the solution will be oscillation free for all values of $\rho$. This choice of $\gamma$ is shown with the function $1 - 2/\rho$ in Figure 9.5.3.

3. *Upwind differencing*, $\gamma = \text{sgn}\rho$. When $\rho > 0$, the shape function $\hat{\psi}(s)$ is the piecewise constant function

$$\hat{\psi}(s) = \begin{cases} 
1, & \text{if } -1 < s \leq 0 \\
0, & \text{otherwise}
\end{cases}.$$  

This function is discontinuous; however, finite element solutions still converge. With $\gamma = 1$, (9.5.22b) becomes

$$\zeta = \frac{2(1 + 1/\rho)}{2/\rho}.$$  

In the limit as $\rho \to \infty$, we have $\zeta \approx \rho$, thus, using (9.5.22a)

$$c_i \sim 1 - \rho^{-[N-i]}, \quad i = 0, 1, \ldots, N, \quad \rho \gg 1.$$  

This result is a good asymptotic approximation of the true solution.

Examining (9.5.21) as a finite difference equation, we see that positive values of $\gamma$ can be regarded as adding dissipation to the system.

This approach can also be used for variable-coefficient problems and for nonuniform mesh spacing. The cell Peclet number would depend on the local value of $\omega$ and the mesh spacing in this case and could be selected as

$$\rho_j = \frac{h_j \bar{u}_j}{\epsilon} \quad (9.5.23)$$
where \( h_j = x_j - x_{j-1} \) and \( \bar{\omega}_j \) is a characteristic value of \( \omega(x) \) when \( x \in [x_{j-1}, x_j) \), e.g., \( \bar{\omega}_j = \mu \omega_{j+1/2} \). Upwind differencing is too diffusive for many applications. Il’in’s scheme offers advantages, but it is difficult to extend to problems other than (9.5.5).

The Petrov-Galerkin technique has also been applied to transient problems of the form (9.5.1); however, the results of applying Il’in’s scheme to transient problems have more diffusion than when it is applied to steady problems.

**Example 9.5.4** [4]. Consider Burgers’s equation

\[
\epsilon u_{xx} - uu_x = 0, \quad 0 < x < 1,
\]

with the Dirichlet boundary conditions selected so that the exact solution is

\[
u(x) = \tanh \frac{1-x}{\epsilon}.\]

Burgers’s equation is often used as a test problem because it is a nonlinear problem with a known exact solution that has a behavior found in more complex problems. Flaherty [4] solved problems with \( h/\epsilon = 6, 500 \) and \( N = 20 \) using upwind differencing and Il’in’s scheme (the Petrov-Galerkin method with the exponential weighting given by (9.5.14b)).
### Table 9.5.1: Maximum pointwise errors for the solution of Example 9.5.4 using upwind differencing (γ = sgnρ) and exponential weighting (γ = coth ρ/2 − 2/ρ) [4].

<table>
<thead>
<tr>
<th>h/ε</th>
<th>Maximum Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upwind</td>
</tr>
<tr>
<td>6</td>
<td>0.124</td>
</tr>
<tr>
<td>500</td>
<td>0.00200</td>
</tr>
</tbody>
</table>

The cell Peclet number (9.5.23) used

\[
\bar{\omega}_j = \begin{cases} 
U(x_j), & \text{if } \mu U_{j-1/2} < 0 \\
\mu U(x_{j-1/2}), & \text{if } \mu U_{j-1/2} = 0 \\
U(x_{j-1}), & \text{if } \mu U_{j-1/2} > 0
\end{cases}
\]

The nonlinear solution is obtained by iteration with the values of \( U(x) \) evaluated at the beginning of an iterative step.

The results for the pointwise error

\[
|\epsilon|_\infty = \max_{0 \leq j \leq N} |u(x_j) - U(x_j)|
\]

are shown in Table 9.5.1. The value of \( h/\epsilon = 6 \) is approximately where the greatest difference between upwind differencing (γ = sgnρ) and exponential weighting (γ = coth ρ/2 − 2/ρ) exists. Differences between the two methods decrease for larger and smaller values of \( h/\epsilon \).

The solution of convection-diffusion problems is still an active research area and much more work is needed. This is especially the case in two and three dimensions. Those interested in additional material may consult Roos et al. [10].

#### Problems

1. Consider (9.5.5) when \( \omega(x) \equiv, q(x) > 0, x \in [0, 1] \) [5].

   1.1. Show that the solution of (9.5.5) is asymptotically given by

   \[
u(x) \approx \frac{f(x)}{q(x)} - u_R(0) e^{-x\sqrt{q(0)/\epsilon}} - u_R(1) e^{-(1-x)\sqrt{q(1)/\epsilon}}.
   \]

   Thus, the solution has \( O(\sqrt{\epsilon}) \) boundary layers at both \( x = 0 \) and \( x = 1 \).

   1.2. In a similar manner, show that the Green’s function is asymptotically given by

   \[
   G(\xi, x) \sim \frac{1}{2|[\epsilon^2 q(x)q(\xi)]^{1/4}|} \begin{cases} 
   e^{-(\xi-x)\sqrt{q(\xi)/\epsilon}}, & \text{if } x \leq \xi \\
   e^{-(x-\xi)\sqrt{q(\xi)/\epsilon}}, & \text{if } x > \xi
   \end{cases}
   \]

   The Green’s function is exponentially small away from \( x = \xi \), where it has two boundary layers. The Green’s function is also unbounded as \( O(\epsilon^{-1/2}) \) at \( x = \xi \) as \( \epsilon \to 0 \).
Bibliography


