Appendix A. Basic Mathematical Concepts

A.1 Hilbert and Banach Spaces

(a) Hilbert Spaces

Let $X$ be a real vector space. An inner product on $X$ is a function $X \times X \to \mathbb{R}$, denoted by $(u, v)$, that satisfies the following properties:

(i) $(u, v) = (v, u)$ for all $u, v \in X$;
(ii) $(\alpha u + \beta v, w) = \alpha (u, w) + \beta (v, w)$ for all $\alpha, \beta \in \mathbb{R}$ and all $u, v, w \in X$;
(iii) $(u, u) \geq 0$ for all $u \in X$;
(iv) $(u, u) = 0$ implies $u = 0$.

Two elements $u, v \in X$ are said to be orthogonal in $X$ if $(u, v) = 0$. The inner product $(u, v)$ defines a norm on $X$ by the relation

$$\|u\| = (u, u)^{1/2} \quad \text{for all} \quad u \in X.$$  

The distance between two elements $u, v \in X$ is the positive number $\|u - v\|$. A Cauchy sequence in $X$ is a sequence $\{u_k \mid k = 0, 1, \ldots \}$ of elements of $X$ that satisfies the following property:

for each positive number $\varepsilon > 0$, there exists an integer $N = N(\varepsilon) > 0$ such that the distance $\|u_k - u_m\|$ between any two elements of the sequence is smaller than $\varepsilon$ provided both $k$ and $m$ are larger than $N(\varepsilon)$.

A sequence in $X$ is said to converge to an element $u \in X$ if the distance $\|u_k - u\|$ tends to 0 as $k$ tends to $\infty$.

A Hilbert space is a vector space equipped with an inner product for which all the Cauchy sequences are convergent.

Examples

(i) $\mathbb{R}^n$ endowed with the Euclidean product $(u, v) = \sum_{i=1}^{n} u_i v_i$ is a finite-dimensional Hilbert space.
(ii) If \([a, b] \subset \mathbb{R}\) is an interval, the space \(L^2(a, b)\) (see (A.6)) is an infinite-dimensional Hilbert space for the inner product
\[
(u, v) = \int_a^b u(x)v(x) \, dx .
\]

If \(X\) is a complex vector space, the inner product on \(X\) will be a complex-valued function. Then condition (i) has to be replaced by (i')
\[
(u, v) = \overline{(v, u)} \quad \text{for all } u, v \in X .
\]

(b) Banach Spaces

The concept of Banach space extends that of Hilbert space. Given a vector space \(X\), a norm on \(X\) is a function \(X \to \mathbb{R}\), denoted by \(\|u\|\), that satisfies the following properties:

- \(\|u + v\| \leq \|u\| + \|v\|\) for all \(u, v \in X\);
- \(\|\lambda u\| = |\lambda|\|u\|\) for all \(u \in X\), and all \(\lambda \in \mathbb{R}\);
- \(\|u\| \geq 0\) for all \(u \in X\);
- \(\|u\| = 0\) if and only if \(u = 0\).

A Banach space is a vector space equipped with a norm for which all the Cauchy sequences are convergent.

Examples

(i) \(\mathbb{R}^n\) endowed with the norm \(\|u\| = \left(\sum_{i=1}^n |u_i|^p\right)^{1/p}\) (with \(1 \leq p < +\infty\)) is a finite-dimensional Banach space.

(ii) If \([a, b] \subset \mathbb{R}\) is an interval and \(1 \leq p < +\infty\), the space \(L^p(a, b)\) (see again (A.6)) is an infinite-dimensional Banach space for the norm
\[
\|u\| = \left(\int_a^b |u(x)|^p \, dx\right)^{1/p} .
\]

(c) Dual Spaces

Let \(X\) be a Hilbert or a Banach space. A linear form \(F : X \to \mathbb{R}\) is said to be continuous if there exists a constant \(C > 0\) such that
\[
|F(u)| \leq C\|u\| \quad \text{for all } u \in X .
\]
The set of all the linear continuous forms on $X$ is a vector space. We can define a norm on this space by setting

$$\|F\|_{X'} = \sup_{\substack{u \in X \\ u \neq 0}} \frac{F(u)}{\|u\|}.$$ 

The vector space of all the linear continuous forms on $X$ is called the dual space of $X$ and is denoted by $X'$. Endowed with the previous norm, it is itself a Banach space.

The bilinear form from $X' \times X$ into $\mathbb{R}$ defined by

$$\langle F, u \rangle = F(u)$$

is called the duality pairing between $X$ and $X'$.

(d) The Riesz Representation Theorem

If $X$ is a Hilbert space, the dual space $X'$ can be canonically identified with $X$ (hence, it is a Hilbert space). In fact, the Riesz representation theorem states that for each linear continuous form $F$ on $X$, there exists a unique element $u \in X$ such that

$$\langle F, v \rangle = (u, v) \quad \text{for all } v \in X.$$

Moreover, $\|F\|_{X'} = \|u\|_X$. The Lax-Milgram Theorem (A.3) extends this result to the case in which $(u, v)$ is replaced by a non-symmetric bilinear form $a(u, v)$.

A.2 The Cauchy-Schwarz Inequality

Let $X$ be a Hilbert space, endowed with the inner product $(u, v)$ and the associated norm $\|u\|$ (see (A.1.a)). The Cauchy-Schwarz inequality states that

$$|\langle u, v \rangle| \leq \|u\| \|v\| \quad \text{for all } u, v \in X.$$

Of particular importance in the analysis of numerical methods for partial differential equations is the Cauchy-Schwarz inequality in the Lebesgue space $L^2(\Omega)$, where $\Omega$ is a domain in $\mathbb{R}^n$ (see (A.9.h)). The previous inequality becomes:

$$\left| \int_{\Omega} u(x)v(x) \, dx \right| \leq \left( \int_{\Omega} u^2(x) \, dx \right)^{1/2} \left( \int_{\Omega} v^2(x) \, dx \right)^{1/2}$$

for all functions $u, v \in L^2(\Omega)$. 
A.3 The Lax-Milgram Theorem

Let $V$ be a real Hilbert space (see (A.1.a)). Let $a : V \times V \to \mathbb{R}$ be a bilinear continuous form on $V$, i.e., $a$ satisfies

(i) $a(\lambda u + \mu v, w) = \lambda a(u, w) + \mu a(v, w)$ and $a(u, \lambda v + \mu w) = \lambda a(u, v) + \mu a(u, w)$ for all $u, v, w \in V$ and all $\lambda, \mu \in \mathbb{R}$;

(ii) there exists a constant $\beta > 0$ such that $|a(u, v)| \leq \beta \|u\|_V \|v\|_V$ for all $u, v \in V$.

Assume that the form $a$ is $V$-coercive, or $V$-elliptic, i.e.,

(iii) there exists a constant $\alpha > 0$ such that $a(u, u) \geq \alpha \|u\|_V^2$ for all $u \in V$.

Then for each form $F \in V'$ (the dual space of $V$, see (A.1.c)), there exists a unique solution $u \in V$ to the variational problem $a(u, v) = F(v)$ for all $v \in V$.

Moreover, the following inequality holds:

$$\|u\|_V \leq \frac{1}{\alpha} \|F\|_{V'}.$$ 

A.4 Dense Subspace of a Normed Space

Let $X$ be a Hilbert or a Banach space with norm $\|v\|$. Let $S \subset X$ be a subspace of $X$. $S$ is said to be dense in $X$ if for each element $v \in X$ there exists a sequence $\{v_n | n = 0, 1, \ldots\}$ of elements $v_n \in S$, such that $\|v - v_n\| \to 0$ as $n \to \infty$.

Thus, each element of $X$ can be approximated arbitrarily well by elements of $S$, in the distance induced by the norm of $X$.

For example, the subspace $C^0([a, b])$ of the continuous functions on a bounded, closed interval $[a, b]$ of the real line, is dense in $L^2(a, b)$, the space of the measurable square-integrable functions on $(a, b)$. Indeed, for each function $v \in L^2(a, b)$ and each $n > 0$, one can find a continuous function $v_n \in C^0([a, b])$ such that

$$\int_a^b |v(x) - v_n(x)|^2 \, dx \leq \frac{1}{n^2}.$$
A.5 The Spaces $C^m(\overline{\Omega})$, $m \geq 0$

Let $\Omega$ be an open subset of $\mathbb{R}^d$, with sufficiently smooth boundary. Let us denote by $\overline{\Omega}$ the closure of $\Omega$. For each multi-index $\alpha = (\alpha_1, \ldots, \alpha_d)$ of nonnegative integers, set $|\alpha| = \alpha_1 + \cdots + \alpha_d$ and $D^\alpha v = \partial^{|\alpha|} v / \partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}$.

We denote by $C^m(\overline{\Omega})$ the vector space of the functions $v : \overline{\Omega} \to \mathbb{R}$ such that for each multi-index $\alpha$ with $0 \leq |\alpha| \leq m$, $D^\alpha v$ exists and is continuous on $\overline{\Omega}$. Since a continuous function on a closed, bounded set is bounded there, one can set

$$\|v\|_{C^m(\overline{\Omega})} = \sup_{0 \leq |\alpha| \leq m} \sup_{x \in \overline{\Omega}} |D^\alpha v(x)| .$$

This is a norm for which $C^m(\overline{\Omega})$ is a Banach space (see (A.1.b)).

The space $C^\infty(\overline{\Omega})$ is the space of the infinitely differentiable functions on $\overline{\Omega}$. Thus, a function $v$ belongs to $C^\infty(\overline{\Omega})$ if and only if it belongs to $C^m(\overline{\Omega})$ for all $m > 0$.

A.6 The Spaces $L^p(\Omega)$, $1 \leq p \leq +\infty$

Let $\Omega$ denote a bounded, open domain in $\mathbb{R}^d$, for $d \geq 1$.

For $p < +\infty$, we denote by $L^p(\Omega)$ the space of the measurable functions $u : \Omega \to \mathbb{R}$ such that $\int_\Omega |u(x)|^p dx < +\infty$. It is a Banach space for the norm

$$\|u\|_{L^p(\Omega)} = \left( \int_\Omega |u(x)|^p dx \right)^{1/p} .$$

Let $L^\infty(\Omega)$ be the Banach space of the measurable functions $u : \Omega \to \mathbb{R}$ that are bounded outside a set of measure zero, equipped with the norm

$$\|u\|_{L^\infty(\Omega)} = \text{ess sup}_{x \in \Omega} |u(x)| .$$

The space $L^2(\Omega)$ is a Hilbert space for the inner product

$$(u, v) = \int_\Omega u(x)v(x) dx ,$$

which induces the norm

$$\|u\|_{L^2(\Omega)} = \left( \int_\Omega |u(x)|^2 dx \right)^{1/2} .$$

One can define spaces $L^p(\Omega)$ of complex functions in a straight-forward manner.
Appendix A. Basic Mathematical Concepts

A.7 Infinitely Differentiable Functions and Distributions

Let $\Omega$ be a bounded, open domain in $\mathbb{R}^d$, for $d = 1, 2$ or $3$. If $\alpha = (\alpha_1, \ldots, \alpha_d)$ is a multi-index of nonnegative integers, let us set

$$D^\alpha v = \frac{\partial^{\alpha_1+\cdots+\alpha_d}v}{\partial x_1^{\alpha_1}\cdots\partial x_d^{\alpha_d}}.$$

We denote by $\mathcal{D}(\Omega)$ the vector space of all the infinitely differentiable functions $\phi : \Omega \to \mathbb{R}$, for which there exists a closed set $K \subset \Omega$ such that $\phi \equiv 0$ outside $K$.

We say that a sequence of functions $\phi_n \in \mathcal{D}(\Omega)$ converges in $\mathcal{D}(\Omega)$ to a function $\phi \in \mathcal{D}(\Omega)$ as $n \to \infty$, if there exists a common closed set $K \subset \Omega$ such that all the $\phi_n$ vanish outside $K$, and $D^\alpha \phi_n \to D^\alpha \phi$ uniformly on $K$ as $n \to \infty$, for all multi-indices $\alpha$.

(a) Distributions

Let $T$ be a linear form on $\mathcal{D}(\Omega)$, i.e., a linear mapping $T : \mathcal{D}(\Omega) \to \mathbb{R}$. We shall denote the value of $T$ on the element $\phi \in \mathcal{D}(\Omega)$ by $\langle T, \phi \rangle$. $T$ is said to be continuous if for each sequence $\phi_n \in \mathcal{D}(\Omega)$ that converges in $\mathcal{D}(\Omega)$ to a function $\phi \in \mathcal{D}(\Omega)$ as $n \to \infty$, one has

$$\langle T, \phi_n \rangle \to \langle T, \phi \rangle \quad \text{as} \quad n \to \infty.$$

A distribution is a linear continuous form on $\mathcal{D}(\Omega)$. The set of all the distributions on $\Omega$ is a vector space denoted by $\mathcal{D}'(\Omega)$.

Examples

(i) Each integrable function $f \in L^1(\Omega)$ (see (A.6)) can be identified with the distribution $T_f$ defined by

$$\langle T_f, \phi \rangle = \int_{\Omega} f(x)\phi(x)dx \quad \text{for all} \quad \phi \in \mathcal{D}(\Omega).$$

(ii) Let $x_0 \in \Omega$. The linear form on $\mathcal{D}(\Omega)$,

$$\langle \delta_{x_0}, \phi \rangle = \phi(x_0) \quad \text{for all} \quad \phi \in \mathcal{D}(\Omega),$$

is a distribution, which is commonly (but improperly) called the “Dirac function”.

We notice that if $T_1$ and $T_2$ are two distributions, then they are “equal in the sense of distributions” if

$$\langle T_1, \phi \rangle = \langle T_2, \phi \rangle \quad \text{for all} \quad \phi \in \mathcal{D}(\Omega).$$
(b) Derivative of Distributions

Let $\alpha$ be a nonnegative multi-index and set $m = \alpha_1 + \cdots + \alpha_d$. For each distribution $T \in \mathcal{D}'(\Omega)$ let us consider the linear form on $\mathcal{D}(\Omega)$:

$$\langle D^\alpha T, \phi \rangle = (-1)^m (T, D^\alpha \phi) \quad \text{for all } \phi \in \mathcal{D}(\Omega).$$

This linear form is continuous on $\mathcal{D}(\Omega)$; hence, it is a distribution, which is called the $\alpha$-distributional derivative of $T$.

It follows that each integrable function $u \in L^1(\Omega)$ is infinitely differentiable in the sense of distributions, and the following Green’s formula holds:

$$\langle D^\alpha u, \phi \rangle = (-1)^m \int_{\Omega} u(x) D^\alpha \phi(x) \, dx \quad \text{for all } \phi \in \mathcal{D}(\Omega).$$

If $u$ is $m$-times continuously differentiable in $\Omega$, then the $\alpha$-distributional derivative of $u$ coincides with the classical derivative of index $\alpha$. In general, a distributional derivative of an integrable function can be an integrable function or merely a distribution. We say that the $\alpha$-distributional derivative of an integrable function $u \in L^1(\Omega)$ is an integrable function if there exists $g \in L^1(\Omega)$ such that

$$\langle D^\alpha u, \phi \rangle = \int_{\Omega} g(x) \phi(x) \, dx \quad \text{for all } \phi \in \mathcal{D}(\Omega).$$

**Examples**

(i) Consider the function $u(x) = \frac{1}{2} |x|$ in the interval $(-1, 1)$. Note that $u$ is not classically differentiable at the origin. The first derivative of $u$ in the distributional sense is represented by the step function

$$v(x) = \begin{cases} 
1/2 & \text{if } x > 0 \\
-1/2 & \text{if } x < 0 . 
\end{cases}$$

(ii) Consider the function $v$ now defined. Note that the classical derivative is zero at all the points $x \neq 0$. The first derivative of $v$ in the sense of distributions is the “Dirac function” $\delta_0$ at the origin. This distribution cannot be represented by an integrable function.

Functions having a certain number of distributional derivatives that can be represented by integrable functions play a fundamental role in the modern theory of partial differential equations. The spaces of these functions are named Sobolev spaces (see (A.8)).

(c) Periodic Distributions

Let $\Omega = (0, 2\pi)^d$, for $d = 1, 2$ or 3. We define the space $C^\infty_p(\Omega)$ as the vector space of the functions $u : \bar{\Omega} \to \mathbb{C}$ that have derivatives of any order
continuous in the closure $\overline{\Omega}$ of $\Omega$, and $2\pi$-periodic in each space direction. A sequence $\phi_n \in C^\infty_p(\overline{\Omega})$ converges in $C^\infty_p(\overline{\Omega})$ to a function $\phi \in C^\infty_p(\overline{\Omega})$ if $D^\alpha \phi_n \to D^\alpha \phi$ uniformly on $\overline{\Omega}$, as $n \to \infty$ for all nonnegative multi-indices $\alpha$.

A periodic distribution is a linear form $T : C^\infty_p(\overline{\Omega}) \to \mathbb{C}$ that is continuous, i.e., such that $\langle T, \phi_n \rangle \to \langle T, \phi \rangle$ as $n \to \infty$, whenever $\phi_n \to \phi$ in $C^\infty_p(\overline{\Omega})$.

The derivative of index $\alpha$ of a periodic distribution $T$ is the periodic distribution $D^\alpha T$ defined by

$$\langle D^\alpha T, \phi \rangle = (-1)^m \langle T, D^\alpha \phi \rangle \text{ for all } \phi \in C^\infty_p(\overline{\Omega})$$

(where $m = \alpha_1 + \cdots + \alpha_d$).

Note that each function in $\mathcal{D}(\Omega)$ also belongs to $C^\infty_p(\overline{\Omega})$. Thus, it is easily seen that each periodic distribution is indeed a distribution in classical sense.

### A.8 Sobolev Spaces and Sobolev Norms

We introduce hereafter some relevant Hilbert spaces, which occur in the numerical analysis of boundary-value problems. They are spaces of square-integrable functions, which possess a certain number of derivatives (in the sense of distributions) representable as square-integrable functions.

**(a) The Spaces $H^m(a, b)$ and $H^m(\Omega)$, $m \geq 0$**

Let $(a, b)$ be a bounded interval of the real line, and let $m \geq 0$ be an integer.

We define $H^m(a, b)$ to be the vector space of the functions $v \in L^2(a, b)$ such that all the distributional derivatives of $v$ of order up to $m$ can be represented by functions in $L^2(a, b)$. In short,

$$H^m(a, b) = \left\{ v \in L^2(a, b) : \text{ for } 0 \leq k \leq m, \frac{d^k v}{dx^k} \in L^2(a, b) \right\}.$$

$H^m(a, b)$ is endowed with the inner product

$$(u, v)_m = \sum_{k=0}^m \int_a^b \frac{d^k u}{dx^k}(x) \frac{d^k v}{dx^k}(x) dx$$

for which $H^m(a, b)$ is a Hilbert space. The associated norm is

$$\|v\|_{H^m(a, b)} = \left( \sum_{k=0}^m \left\| \frac{d^k v}{dx^k} \right\|_{L^2(a, b)}^2 \right)^{1/2}.$$
The Sobolev spaces $H^m(a, b)$ form a hierarchy of Hilbert spaces, in the sense that $\cdots \subset H^{m+1}(a, b) \subset H^m(a, b) \subset \cdots \subset H^0(a, b) = L^2(a, b)$, each inclusion being continuous. Clearly, if a function $u$ has $m$ classical continuous derivatives in $[a, b]$, then $u$ belongs to $H^m(a, b)$—in other words, $C^m([a, b]) \subset H^m(a, b)$ with continuous inclusion. Conversely, if $u$ belongs to $H^m(a, b)$ for $m \geq 1$, then $u$ has $m - 1$ classical continuous derivatives in $[a, b]$, i.e., $H^m(a, b) \subset C^{m-1}([a, b])$ with continuous inclusion. This is an example of the so-called “Sobolev imbedding theorems”. As a matter of fact, $H^m(a, b)$ can be equivalently defined as

$$H^m(a, b) = \left\{ v \in C^{m-1}([a, b]) : \frac{d}{dx} v^{(m-1)} \in L^2(a, b) \right\},$$

where the last derivative is in the sense of distributions.

Functions in $H^m(a, b)$ can be approximated arbitrarily well by infinitely differentiable functions in $[a, b]$, in the distance induced by the norm of $H^m(a, b)$. In other words,

$$C^\infty([a, b]) \text{ is dense in } H^m(a, b)$$

(see (A.4) for the definition of density of a subspace).

Now, let $\Omega \subset \mathbb{R}^d$, for $d \geq 2$, be an open, bounded set with sufficiently smooth boundary. Given a multi-index $\alpha = (\alpha_1, \ldots, \alpha_d)$ of nonnegative integers, we set $|\alpha| = \alpha_1 + \cdots + \alpha_d$ and

$$D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

The previous definition of Sobolev spaces can be extended to higher space dimensions as follows. We define

$$H^m(\Omega) = \{ v \in L^2(\Omega) : \text{for each nonnegative multi-index } \alpha \text{ with } |\alpha| \leq m, \text{the distributional derivative } D^\alpha v \text{ belongs to } L^2(\Omega) \}.$$

This is a Hilbert space for the inner product

$$(u, v)_m = \sum_{|\alpha| \leq m} \int_\Omega D^\alpha u(x) D^\alpha v(x) dx,$$

which induces the norm

$$\|v\|_{H^m(\Omega)} = \left( \sum_{|\alpha| \leq m} \|D^\alpha v\|^2_{L^2(\Omega)} \right)^{1/2}.$$

Functions in $H^m(\Omega)$ for $m \geq 1$ need not have their derivatives of order $m - 1$ continuous in $\Omega$. However, for all $m > d/2$, the weaker Sobolev inclusion
$H^m(\Omega) \subset C^k(\Omega)$ (where $k = \lfloor m - d/2 \rfloor$ is the integer part of $m - d/2$) holds. On the other hand, as in the 1D case

$$C^\infty(\Omega) \text{ is dense in } H^m(\Omega).$$

Sobolev spaces $H^s(\Omega)$ of non-integer order $s$ can be defined, e.g., by “interpolation” (see Bergh and Löfstrom (1976)) between the two Sobolev spaces $H^m(\Omega)$ and $H^{m+1}(\Omega)$, with $m < s < m + 1$.

Sobolev spaces $H^s(\Gamma)$, where $\Gamma$ is a sufficiently smooth $(d-1)$-dimensional manifold in $\mathbb{R}^d$, can be defined as well.

(b) The Spaces $H^1_0(a, b)$ and $H^1_0(\Omega)$

Dirichlet conditions are among the simplest and most common boundary conditions to be associated with a differential operator. Therefore, the subspaces of the Sobolev spaces $H^m$ spanned by the functions satisfying homogeneous Dirichlet boundary conditions play a fundamental role.

Since the functions of $H^1(a, b)$ are continuous up to the boundary by the Sobolev imbedding theorem, it is meaningful to introduce the following subspace of $H^1(a, b)$:

$$H^1_0(a, b) = \{ v \in H^1(a, b) : v(a) = v(b) = 0 \}.$$

This is a Hilbert space for the same inner product of $H^1(a, b)$. It is often preferable to endow $H^1(a, b)$ with a different, although equivalent, inner product. This is defined as

$$[u, v] = \int_a^b \frac{du}{dx}(x) \frac{dv}{dx}(x) dx.$$

By the Poincaré inequality, it is indeed an inner product on $H^1_0(a, b)$. The associated norm, denoted by

$$\|v\|_{H^1_0(a, b)} = \left( \int_a^b \left| \frac{dv}{dx} \right|^2 dx \right)^{1/2},$$

is equivalent to the $H^1(a, b)$-norm, in the sense that there exists a constant $C > 0$ such that, for all $v \in H^1_0(a, b)$,

$$C\|v\|_{H^1(a, b)} \leq \|v\|_{H^1_0(a, b)} \leq \|v\|_{H^1(a, b)}.$$

Again, this follows from the Poincaré inequality (A.10).

The functions of $H^1_0(a, b)$ can be approximated arbitrarily well in the norm of this space not only by infinitely differentiable functions on $[a, b]$, but also by infinitely differentiable functions that vanish identically in a neighborhood
of $x = a$ and $x = b$. In other words

$$\mathcal{D}((a, b)) \text{ is dense in } H^1(a, b)$$

(see (A.4) and (A.7)).

We turn now to more space dimensions. If $\Omega$ is a bounded domain in $\mathbb{R}^d$ with sufficiently smooth boundary, the functions of $H^1(\Omega)$ need not be continuous on the closure of $\Omega$. Thus, their pointwise values on the boundary $\partial \Omega$ of $\Omega$ need not be defined. However, it is possible to extend the trace operator $v \mapsto v|_{\partial \Omega}$ (classically defined for functions $v \in C^0(\overline{\Omega})$) so as to be a linear continuous mapping between $H^1(\Omega)$ and $L^2(\partial \Omega)$, the space of the square-integrable functions on $\partial \Omega$. (More precisely, the image of the trace operator is a proper subspace of $L^2(\partial \Omega)$, indicated by $H^{1/2}(\partial \Omega)$.) With this in mind, it is meaningful to define $H^1_0(\Omega)$ as the subspace of $H^1(\Omega)$ of the functions whose trace at the boundary is zero. Precisely we set

$$H^1_0(\Omega) = \{ v \in H^1(\Omega) : v|_{\partial \Omega} = 0 \}.$$ 

This is a Hilbert space for the inner product of $H^1(\Omega)$, or for the inner product

$$[u, v] = \int_{\Omega} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x}.$$ 

The associated norm is denoted by

$$\|v\|_{H^1_0(\Omega)} = \left( \int_{\Omega} |\nabla v|^2 \, d\mathbf{x} \right)^{1/2}$$

and is equivalent to the $H^1(\Omega)$-norm, by the Poincaré inequality (A.10).

Concerning the approximation of the functions of $H^1_0(\Omega)$ by infinitely smooth functions, the following result holds:

$$\mathcal{D}(\Omega) \text{ is dense in } H^1_0(\Omega).$$

The dual spaces (see (A.1.c)) of the Hilbert spaces of type $H^1_0$ now defined are usually denoted by $H^{-1}$. Thus, $H^{-1}(a, b)$ is the dual space of $H^1_0(a, b)$, $H^{-1}(\Omega)$ is the dual space of $H^1_0(\Omega)$.

Finally let us mention that for $m \geq 2$, one can define the subspaces $H^m_0(a, b)$ of $H^m(a, b)$ (and similarly for $H^m_0(\Omega)$) of the functions of $H^m(a, b)$ whose derivatives of order up to $m - 1$ vanish on the boundary of the domain of definition. Again, these spaces are Hilbert spaces for the inner product of $H^m(a, b)$, or for an equivalent inner product that only involves the derivatives of order $m$.

(c) The Spaces $H^m_p(0, 2\pi)$ and $H^m_p(\Omega)$, $m \geq 0$

In the analysis of Fourier methods, the natural Sobolev spaces are those of periodic functions. In this framework, functions are complex valued, and their
derivatives are taken in the sense of the periodic distributions (see (A.7)). We set

\[ H^m_p(0, 2\pi) = \left\{ v \in L^2(0, 2\pi) : \text{for } 0 \leq k \leq m, \text{ the derivative } \frac{d^kv}{dx^k} \text{ in the sense of periodic distribution belongs to } L^2(0, 2\pi) \right\}. \]

\( H^m_p(0, 2\pi) \) is a Hilbert space for the inner product

\[ (u, v)_m = \sum_{k=0}^{m} \int_0^{2\pi} \frac{d^k u}{dx^k}(x) \frac{d^k v}{dx^k}(x) dx, \]

whose associated norm is

\[ \|v\|_{H^m_p(0, 2\pi)} = \left( \sum_{k=0}^{m} \left\| \frac{d^k v}{dx^k} \right\|_{L^2(0, 2\pi)}^2 \right)^{1/2}. \]

The space \( H^m_p(0, 2\pi) \) coincides with the space of the functions \( v : [0, 2\pi] \to \mathbb{C} \) that have \( m-1 \) continuously differentiable, 2\pi-periodic derivatives on \([0, 2\pi] \), and such that the periodic distributional derivative \((d/dx)^{(m-1)} v\) can be represented by a function of \( L^2(0, 2\pi) \). The space \( C^\infty_p([0, 2\pi]) \) is dense in \( H^m_p(0, 2\pi) \).

If \( \Omega = (0, 2\pi)^d \) for \( d = 2 \) or \( 3 \), we set

\[ H^m_p(\Omega) = \{ v \in L^2(\Omega) : \text{for each multi-index } \alpha \text{ with } |\alpha| \leq m, \text{ the derivative } D^\alpha v \text{ in the sense of periodic distributions belongs to } L^2(\Omega) \}. \]

This is a Hilbert space for the inner product

\[ (u, v)_m = \sum_{|\alpha|\leq m} \int_\Omega D^\alpha u(x) \overline{D^\alpha v(x)} dx, \]

with associated norm

\[ \|v\|_{H^m_p(\Omega)} = \left( \sum_{|\alpha|\leq m} \|D^\alpha v\|_{L^2(\Omega)}^2 \right)^{1/2}. \]

The space \( C^\infty(\Omega) \) is dense in \( H^m_p(\Omega) \). Note that since a periodic distribution is also a distribution, each space \( H^m_p(0, 2\pi) \) (resp. \( H^m_p(\Omega) \)) is a subspace of the space \( H^m(0, 2\pi) \) (resp. \( H^m(\Omega) \)).
A.9 The Sobolev Inequality

Let \((a, b) \subset \mathbb{R}\) be a bounded interval of the real line. For each function \(u \in H^1(a, b)\) (see (A.8)) the following inequality holds:

\[
\|u\|_{L^\infty(a,b)} \leq \left( \frac{1}{b-a} + 2 \right)^{1/2} \|u\|_{L^2(a,b)}^{1/2} \|u\|_{H^1(a,b)}^{1/2}.
\]

A.10 The Poincaré Inequality

Let \(v\) be a function of \(H^1(a, b)\) (see (A.8)). We know that \(v\) is continuous on \([a, b]\). Assume that at a point \(x_0 \in [a, b]\), \(v_0(x_0) = 0\). The Poincaré inequality states that there exists a constant \(C\) (depending upon the interval length \(b - a\)) such that

\[
\|v\|_{L^2(a,b)} \leq C \|v'\|_{L^2(a,b)},
\]

i.e., the \(L^2\)-norm of the function is bounded by the \(L^2\)-norm of the derivative. The Poincaré inequality applies to functions belonging to \(H^1_0(a, b)\), for which \(x_0 = a\) or \(b\), and also to functions of \(H^1(a, b)\) that have zero average on \((a, b)\), since necessarily such functions change sign in the domain.

In space dimension \(d \geq 2\), the functions to which the Poincaré inequality applies must vanish on a manifold of dimension \(d - 1\). Confining ourselves to the case of functions vanishing on the boundary \(\partial \Omega\) of the domain of definition \(\Omega\), one has

\[
\|v\|_{L^2(\Omega)} \leq C \|
abla v\|_{(L^2(\Omega))^d} \quad \text{for all } v \in H^1_0(\Omega).
\]

The same result holds if the domain \(\Omega\) is simply connected and \(v\) only vanishes on a portion of \(\partial \Omega\) of positive measure.

The smallest constant \(C = C_P\) for which (A.10.1) or (A.10.2) holds is termed the Poincaré constant of the domain. Often, this term refers to any constant \(C\) appearing in (A.10.1) or in (A.10.2).
Appendix B. Fast Fourier Transforms

Basics

The Fast Fourier Transform (FFT) is a recursive algorithm for evaluating the discrete Fourier transform and its inverse. The FFT is conventionally written for the evaluation of

$$\tilde{u}_k = \sum_{j=0}^{N-1} u_j e^{2\pi ijk/N}, \quad k = 0, 1, \ldots, N - 1,$$  \hspace{1cm} (B.1.a)

or

$$\tilde{u}_k = \sum_{j=0}^{N-1} u_j e^{-2\pi ijk/N}, \quad k = 0, 1, \ldots, N - 1,$$  \hspace{1cm} (B.1.b)

where $u_j, j = 0, 1, \ldots, N - 1$ are a set of complex data. The FFT quickly became a widely used tool in signal processing after its description by Cooley and Tukey (1965). (As noted later by Cooley, Lewis and Welch (1969), most essential components of the FFT date back to the 1920s.) The Cooley–Tukey algorithm enables the sums in (B.1) to be evaluated in $5N \log_2 N$ real operations (when $N$ is a power of 2), instead of the $8N^2$ real operations required by the straightforward sum. Moreover, calculation of (B.1) via the FFT incurs less error due to round-off than the direct summation method (Cooley et al. (1969)).

Many versions of the FFT are now in existence. The review by Temperton (1983) contains an especially clear description of a simple yet efficient one. It allows $N$ to be of the form

$$N = 2^p 3^q 4^r 5^s 6^t$$  \hspace{1cm} (B.2)

and has the operation count

$$N \left( 5p + 9\frac{1}{3}q + 8\frac{1}{2}r + 13\frac{3}{5}s + 13\frac{1}{3}t - 6 \right).$$  \hspace{1cm} (B.3)

No additional flexibility is gained by the inclusion of the factors 4 and 6. The algorithm is, however, more efficient when these factors are included. Not only is the operation count lower—for example, by 15% when $N = 64$—but,
due to the higher ratio of arithmetic operations to memory accesses, most Fortran compilers generate more efficient code for the larger factors. For the sake of simplicity, however, throughout this book we use \((5 \log_2 N - 6)N\) as the operation count for the complex FFT; moreover, the lower order term linear in \(N\) is usually omitted.

We should also mention the book by Brigham (1974), which is devoted entirely to the Fast Fourier Transform, and the FFTW package by Frigo and Johnson (2005), which received the 1999 Wilkinson Prize for Numerical Software. (The FFTW software is available at http://www.fftw.org/.)

**Use in Spectral Methods**

In applications of Fourier spectral methods, the sums that one must evaluate are

\[
\tilde{u}_k = \frac{1}{N} \sum_{j=0}^{N-1} u_j e^{-2\pi i jk/N}, \quad k = -\frac{N}{2}, -\frac{N}{2} + 1, \ldots, \frac{N}{2} - 1, \quad (B.4)
\]

and

\[
u_j = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k e^{2\pi i jk/N}, \quad j = 0, 1, \ldots, N - 1 \quad (B.5)
\]

(see (8.1.21) and (8.1.23)). From (B.4) it is apparent that, for integers \(p\) and \(k\),

\[
\tilde{u}_{k+pN} = \tilde{u}_k. \quad (B.6)
\]

When the array \((u_0, u_1, \ldots, u_{N-1})\) is fed into a standard FFT for evaluating (B.1.b) it returns, in effect, the array

\[
(N\tilde{u}_0, N\tilde{u}_1, \ldots, N\tilde{u}_{N/2-1}, N\tilde{u}_{-N/2}, N\tilde{u}_{-N/2+1}, \ldots, N\tilde{u}_{-1}).
\]

Conversely, when this array (without the factor \(N\)) is fed into the standard FFT for evaluating (B.1.a) (with the plus sign), the array \((u_0, u_1, \ldots, u_{N-1})\) is returned.

In most applications of spectral methods, the direct use of the complex FFT (B.1) is needlessly expensive. This is true, for example if the function \(u_j\) is real or if a cosine transform (for a Chebyshev spectral method) is desired. These issues have been addressed by Orszag (1971a, Appendix II) and by Brachet et al. (1983, Appendix C). A summary of some of the relevant transformations follows.

**Real Transforms**

The simplest case occurs when many real transforms are desired at once, as arises for multidimensional problems. They can be computed pairwise.
Suppose that \( u_j^1 \) and \( u_j^2 \), \( j = 0, 1, \ldots, N - 1 \), are two sets of real data. Then one can define
\[
v_j = u_j^1 + i u_j^2
\] (B.7)
and compute \( \tilde{v}_k \) according to (B.4) by the standard \( N \)-point complex FFT. Then the transforms \( \tilde{u}_k^1 \) and \( \tilde{u}_k^2 \) can be extracted according to
\[
\begin{align*}
\tilde{u}_k^1 &= \frac{1}{2} ( \tilde{v}_k + \overline{\tilde{v}}_{-k} ) , \quad k = 0, 1, \ldots, \frac{N}{2} - 1 . \\
\tilde{u}_k^2 &= -i \frac{1}{2} ( \tilde{v}_k - \overline{\tilde{v}}_{-k} ) , \quad k = 0, 1, \ldots, \frac{N}{2} - 1 .
\end{align*}
\] (B.8)
(The Fourier coefficients of real data for negative \( k \) are related to those for positive \( k \) by \( \tilde{u}_{-k} = \overline{\tilde{u}}_k \).) This process is readily reversed. In fact, if one is performing a Fourier interpolation derivative, one need not even bother with the separation (B.8) in Fourier space, since
\[
\left. \frac{d u_j^1}{dx} \right|_j + i \left. \frac{d u_j^2}{dx} \right|_j = \sum_{k=-N/2}^{N/2-1} i k \tilde{v}_k .
\] (B.9)

If only a single real transform is desired, then one may follow the prescription given by Orszag (1971a). Let \( M = N/2 \) and define
\[
v_j = u_{2j} + i u_{2j+1} , \quad j = 0, 1, \ldots, M - 1 . \quad (B.10)
\]
Then take an \( M \)-point transform of \( v_j \), set \( \tilde{v}_M = \overline{\tilde{v}}_0 \), and extract the desired coefficients via
\[
\tilde{u}_k = \frac{1}{2} ( \tilde{v}_k + \overline{\tilde{v}}_{M-k} ) - i \frac{e^{2\pi ik/M}}{2} ( \tilde{v}_k - \overline{\tilde{v}}_{M-k} ) , \quad k = 0, 1, \ldots, M - 1 . \quad (B.11)
\]

For both of these approaches the cost of a single, real-to-half-complex transform is essentially \((5/2)N \log_2 N\).

**Chebyshev Transforms**

The discrete Chebyshev transforms based on the Gauss–Lobatto points (8.3.15) are given by
\[
\tilde{u}_k = \frac{2}{N \overline{c}_k} \sum_{j=0}^{N} \frac{1}{\overline{c}_j} u_j \cos \frac{\pi j k}{N} , \quad k = 0, 1, \ldots, N , \quad (B.12)
\]
(see (8.2.24) and (8.3.17)) and
\[
u_j = \sum_{k=0}^{N} \tilde{u}_k \cos \frac{\pi j k}{N} , \quad j = 0, 1, \ldots, N \quad (B.13)
Appendix B. Fast Fourier Transforms

(see (8.2.25) and (8.3.19)). Suppose that the transform (B.12) is desired for two real sets of data $u^1_j$ and $u^2_j$. Then define the complex data $v_j$ by

$$v_j = \begin{cases} u^1_j + iu^2_j, & j = 0, 1, \ldots, N, \\ v_{2N-j}, & j = N + 1, N + 2, \ldots, 2N - 1, \end{cases} \quad (B.14)$$

and by periodicity (with period $2N$) for other integers $j$. Next, define $\tilde{v}_k$, $k = 0, 1, \ldots, N$, by (B.12) and define $\tilde{V}_k$, $k = 0, 1, \ldots, 2N - 1$, by (B.1.a) with $N$ replaced by $2N$. It is readily shown that

$$\tilde{V}_k = \frac{1}{N} \tilde{v}_k, \quad k = 0, 1, \ldots, N, \quad (B.15)$$

and that

$$\tilde{V}_k = \sum_{l=0}^{N-1} v_{2l} e^{2\pi i kl/N} + e^{\pi i k/N} \sum_{l=0}^{N-1} v_{2l+1} e^{2\pi i kl/N}. \quad (B.16)$$

Now, define $w_j$ by

$$w_j = v_{2j} + i(v_{2j+1} - v_{2j-1}), \quad j = 0, 1, \ldots, N - 1, \quad (B.17)$$

and compute $\tilde{w}_k$ according to the complex FFT (B.1.a). We have

$$\tilde{w}_k = \sum_{l=0}^{N-1} v_{2l} e^{2\pi i kl/N} + i(1 - e^{2\pi i k/N}) \sum_{l=0}^{N-1} v_{2l+1} e^{2\pi i kl/N}, \quad (B.18)$$

$$\tilde{w}_{N-k} = \sum_{l=0}^{N-1} v_{2l} e^{2\pi i kl/N} - i(1 - e^{2\pi i k/N}) \sum_{l=0}^{N-1} v_{2l+1} e^{2\pi i kl/N}.$$

Consequently,

$$\tilde{v}_0 = \frac{1}{N} \sum_{j=0}^{N} \frac{1}{\tilde{c}_j} v_j,$$

$$\tilde{v}_k = \frac{1}{N} \left[ \left( \frac{1}{2} + \frac{1}{4 \sin \frac{\pi k}{N}} \right) \tilde{w}_k + \left( \frac{1}{2} - \frac{1}{4 \sin \frac{\pi k}{N}} \right) \tilde{w}_{N-k} \right], \quad (B.19)$$

$$\tilde{v}_N = \frac{1}{N} \sum_{j=0}^{N} (-1)^j \frac{1}{\tilde{c}_j} v_j.$$

The desired real coefficients $\tilde{u}^1_k$ and $\tilde{u}^2_k$ are the real and imaginary parts, respectively, of the $\tilde{v}_k$. Thus, the discrete Chebyshev transform (B.12) can be computed in $\frac{5}{2} N \log_2 N + 4N$ real operations per transform, assuming that a large number of such transforms are computed. The inverse discrete
Chebyshev transform (B.13) can be evaluated with only minor modifications to the algorithm given by (B.14), (B.17) and (B.19).

Discrete sine transforms can be handled in a similar manner: (B.14) (with $v_{2N-j}$ replaced by $-v_{2N-j}$) and (B.17) are retained as is the central equation in (B.19) with the coefficient of $\tilde{w}_{N-k}$ having the opposite sign; the entire $\tilde{v}_k$ term is multiplied by $i$ and one sets $\tilde{v}_0 = \tilde{v}_N = 0$. Swarztrauber (1986) described how real cosine and sine transforms can be computed without the pre- and post-processing costs incurred by (B.17) and (B.19).

**Other Cosine Transforms**

In some applications, such as the use of a staggered grid in Navier–Stokes calculations (see Sect. 3.4) and in simulations of flows with special symmetries (Brachet et al. (1983)), discrete Chebyshev transforms with respect to the Gauss points (see (8.3.13) but with $N-1$ in place of $N$) are required. Consider

$$\tilde{u}_k = \frac{2}{N} \sum_{j=0}^{N-1} u_j \cos \left( \frac{(2j + 1)\pi k}{2N} \right) , \quad k = 0, 1, \ldots, N - 1 . \tag{B.20}$$

Brachet et al. (1983) have provided prescriptions for computing efficiently this and related sums. Put

$$v_j = \begin{cases} 
  u_{2j} , & j = 0, 1, \ldots, \frac{N}{2} - 1 , \\
  u_{2N-2j-1} , & j = \frac{N}{2}, \frac{N}{2} + 1, \ldots, N - 1 , 
\end{cases} \tag{B.21}$$

and compute $\tilde{v}_k$ according to (B.1.a). Then $\tilde{u}_k$ may be extracted via

$$\tilde{u}_k = \frac{1}{N} [e^{2\pi ik/2N} \tilde{v}_k + e^{-2\pi ik/2N} \tilde{v}_{N-k}] , \quad k = 0, 1, \ldots, N - 1 . \tag{B.22}$$

The corresponding inverse Chebyshev transform

$$u_j = \sum_{k=0}^{N-1} \tilde{u}_k \cos \frac{(2j + 1)\pi k}{2N} \tag{B.23}$$

can be evaluated by reversing these steps.

For some problems, the Chebyshev expansion may be over the interval $[0, 1]$ instead of $[-1, 1]$. Moreover, it may also be useful to use only the odd (or even) polynomials (Spalart (1984); see also Sect. 8.8.2). Spalart (1986, pers. comm.) explained how to employ the FFT for an expansion over $[0, 1]$ in terms of just the odd Chebyshev polynomials. The collocation points are

$$x_j = \cos \frac{(2j + 1)\pi}{2N} , \quad j = 0, 1, \ldots, N - 1 , \tag{B.24}$$
the series expansion is

\[ u^N(x) = \sum_{k=0}^{N-1} \tilde{u}_k T_{2k+1}(x) , \]  

(B.25)

and the discrete transforms are

\[ \tilde{u}_k = \frac{2}{N} \sum_{j=0}^{N-1} u_j \cos \left( \frac{(2k+1)(2j+1)\pi}{4N} \right) , \quad k = 0, 1, \ldots, N-1 , \]  

(B.26)

and

\[ u_j = \sum_{k=0}^{N-1} \tilde{u}_k \cos \left( \frac{(2k+1)(2j+1)\pi}{4N} \right) , \quad j = 0, 1, \ldots, N-1 . \]  

(B.27)

(In order for a half-interval Chebyshev expansion to be spectrally accurate, one needs \( u(x) \) and all of its derivatives to vanish at \( x = 0 \).) Spalart’s trick for evaluating (B.27) is to define

\[ \tilde{v}_k = \frac{\tilde{u}_k + \tilde{u}_{k-1}}{2 \cos \left( \frac{k\pi}{2N} \right)} , \quad k = 0, 1, \ldots, N , \]  

(B.28)

where \( \tilde{u}_{-1} = \tilde{u}_N = 0 \), to compute \( v_j \) according to (B.13), and then to extract \( u_j \) via

\[ u_j = \frac{\tilde{v}_j + \tilde{v}_{j+1}}{2 \cos \left( \frac{(2j+1)\pi}{4N} \right)} , \quad j = 0, 1, \ldots, N-1 . \]  

(B.29)

(Note however that this transform is not suitable for use with the Gauss–Lobatto points.)
Appendix C. Iterative Methods for Linear Systems

In this appendix, we review some of the most important iterative methods for the solution of a linear system of the form

\[ Lu = f . \]  

(C.0.1)

The discussion will be at a tutorial level. For an extensive presentation and a thorough analysis the reader may refer to Golub and Van Loan (1996), Saad (1996), Greenbaum (1997), Van der Vorst (2003), and to the ample literature cited therein.

C.1 A Gentle Approach to Iterative Methods

A particularly simple iterative scheme is the Richardson (1910) method. Given an initial guess \( v^0 \) to \( u \), subsequent approximations are obtained via

\[ v^{n+1} = v^n + \omega r^n , \]  

(C.1.1)

where \( \omega \) is a relaxation parameter and

\[ r^n = f - Lv^n \]  

(C.1.2)

is the residual associated with \( v^n \). The error obeys the relation

\[ (v^{n+1} - u) = G (v^n - u) , \]  

(C.1.3)

where the iteration matrix \( G \) of the Richardson scheme is given by

\[ G = I - \omega L . \]  

(C.1.4)

The iterative scheme is convergent if the spectral radius \( \rho \) of \( G \) is less than 1. In the case of the Richardson scheme this condition is equivalent to

\[ |1 - \omega \lambda| < 1 , \]  

(C.1.5)
for all the eigenvalues \( \lambda \) of \( L \). The simultaneous fulfillment of these inequalities is possible only if all the eigenvalues of \( L \) have nonzero real parts of constant sign. A particularly relevant case is that of a matrix with all real and strictly positive eigenvalues; symmetric and positive-definite matrices enjoy this property, but these are not necessary conditions. For example, the matrices generated by Chebyshev or Legendre collocation discretizations of second-order problems have all real and strictly positive eigenvalues. In such a situation, we have \( 0 < \lambda_{\min} \leq \lambda_{\max} \), where \( \lambda_{\min} \) and \( \lambda_{\max} \) are the extreme eigenvalues of \( L \). The convergence condition (C.1.5) is satisfied for \( 0 < \omega < \omega_{\max} \), where

\[
\omega_{\max} = \frac{2}{\lambda_{\max}}.
\] (C.1.6)

The best choice of \( \omega \) is that which minimizes \( \rho \). It is obtained from the relation

\[
(1 - \omega \lambda_{\max}) = -(1 - \omega \lambda_{\min}),
\] (C.1.7)

for then the largest values of \( 1 - \omega \lambda \) are equal in magnitude and have opposite sign (see, e.g., Quarteroni and Valli (1994)). The optimal relaxation parameter is thus

\[
\omega_{\text{opt}} = \frac{2}{\lambda_{\max} + \lambda_{\min}}.
\] (C.1.8)

It produces the spectral radius

\[
\rho = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}.
\] (C.1.9)

Note that the dependence upon the extreme eigenvalues enters only in the combination

\[
\mathcal{K} = \mathcal{K}(L) = \frac{\lambda_{\max}}{\lambda_{\min}}.
\] (C.1.10)

We shall call this ratio the \textit{iterative condition number} of \( L \) to distinguish it from the generic definition of condition number,

\[
\kappa_{\|\cdot\|}(L) = \|L\| \|L^{-1}\|,
\] (C.1.11)

and the \textit{spectral condition number}

\[
\kappa_2(L) = \left[ \frac{\lambda_{\max}}{\lambda_{\min}} \left( L^T L \right)^{1/2} \right].
\] (C.1.12)

In the case that \( L \) is symmetric and positive definite, this becomes

\[
\kappa_2(L) = \frac{\lambda_{\max}}{\lambda_{\min}} \left( L \right) = \mathcal{K}.
\] (C.1.13)
For some nonsymmetric discretization matrices that have real positive eigenvalues, such as those mentioned above, the spectral and the iterative condition numbers might differ. In terms of $\mathcal{K}$, (C.1.9) becomes

$$\rho = \frac{\mathcal{K} - 1}{\mathcal{K} + 1}.$$  \hfill (C.1.14)

Define the rate of convergence $\mathcal{R}$ to be

$$\mathcal{R} = -\log \rho,$$  \hfill (C.1.15)

and denote its reciprocal by $\mathcal{J}$. The latter quantity measures the number of iterations required to reduce the error by a factor of $e$. This immediately follows from the error bound

$$\|v^n - u\|_L \leq \rho^n \|v^0 - u\|_L,$$

which holds with $\|v\|_L = (v^T Lv)^{1/2}$. Clearly, the larger the convergence rate that a method has for a problem, the fewer iterations that are required to obtain a solution to a given accuracy. For the Richardson method described above, the number of iterations increases as

$$\mathcal{J} \approx \frac{1}{2} \mathcal{K}.$$  \hfill (C.1.16)

The basic Richardson method (C.1.1) can be improved and extended in several ways. The discussion thus far concerned only the stationary Richardson method. In a non-stationary Richardson method, the parameter $\omega$ in (C.1.1) is allowed to depend on $n$, i.e., to change in the course of iterations, in order to speed up the convergence.

For a static non-stationary Richardson (NSR) method one cycles through a fixed number $k$ of parameters. Using the minimax property of Chebyshev polynomials, the following expressions for the optimal parameters can be derived:

$$\omega_j = \frac{2/\lambda_{\min}}{(\mathcal{K} - 1) \cos \frac{(2j - 1)\pi}{2k} + (\mathcal{K} + 1)}, \quad j = 1, \ldots, k,$$  \hfill (C.1.17)

yielding the effective spectral radius

$$\rho = \frac{1}{T_k \left( \frac{\mathcal{K} + 1}{\mathcal{K} - 1} \right)^{1/k}}.$$  \hfill (C.1.18)

Both $\omega_j$ (for all $j$) and $\rho$ depend on $\mathcal{K}$. However, this approach suffers from the same limitation as the basic Richardson method—information must be available on the eigenvalues of $L$ in order to compute $\mathcal{K}$. 
A broad family of dynamic non-stationary Richardson methods are based on an optimality strategy that does not require the knowledge of the extreme eigenvalues. We address dynamic non-stationary Richardson methods in Sect. C.2.

The primary cause of the inefficiency of the Richardson method is that the convergence rate decreases as the iterative condition number increases; in spectral methods, the condition number typically increases with the approximation parameter $N$. This can be alleviated by preconditioning the problem, in effect solving

$$H^{-1}Lu = H^{-1}f$$

rather than (C.0.1). (This is called left preconditioning. Other options are available as well, such as right preconditioning and symmetric preconditioning; see (C.2.15) and (C.2.18), respectively.)

A preconditioned version of (C.1.1) is

$$H \left( v_{n+1} - v_n \right) = \omega r_n . \quad \text{(C.1.19)}$$

One obvious requirement for $H$ is that this equation can be solved inexpensively, i.e., in fewer operations than are required to evaluate $Lv^n$. The effective iteration matrix is now

$$G = I - \omega H^{-1}L . \quad \text{(C.1.20)}$$

The second requirement on the preconditioning matrix is that $H^{-1}$ be a good approximation to $L^{-1}$, i.e., that the new iterative condition number $\mathcal{K}(H^{-1}L)$ be much smaller than $\mathcal{K}(L)$. In such circumstances, the new spectral radius $\rho$ is much smaller than that of the non-preconditioned Richardson method.

This property can be rigorously justified whenever $L$ and $H$ are both symmetric and positive definite. Indeed, denoting by $H^{1/2}$ the square root of $H$, (C.1.19) can be written equivalently as

$$w_{n+1} = w_n + \omega(H^{-1/2}f - H^{-1/2}LH^{-1/2}w_n)$$

with $w^n = H^{1/2}v^n$, showing that (C.1.19) is nothing but a Richardson iteration applied to the symmetric and positive-definite matrix $H^{-1/2}LH^{-1/2}$. Since this matrix is similar to $H^{-1}L$, we have

$$\mathcal{K}(H^{-1/2}LH^{-1/2}) = \mathcal{K}(H^{-1}L).$$

The discussion so far has presumed that the eigenvalues of $H^{-1}L$ are confined to the interval $[\lambda_{\min}, \lambda_{\max}]$ on the positive real-axis. However, the Richardson iteration schemes can work on problems for which the eigenvalues are complex but have positive real parts. If we still use a real $\omega$, then it should obey the following restriction for convergence:

$$\omega < 2 \frac{Re(\lambda_i)}{|\lambda_i|^2} ,$$
for all eigenvalues $\lambda_i$ of $H^{-1}L$ (see, e.g., Quarteroni and Valli (1994), Sect. 2.4). One could also use a complex $\omega$, in which case the iterations can be performed entirely in real arithmetic according to

$$v^{n+1} = v^n + 2\text{Re}(\omega)H^{-1}r^n + |\omega|^2H^{-1}LH^{-1}r^n. \quad (C.1.21)$$

The value of the optimal parameter $\omega_{opt}$ is obtained by solving a minimax problem in complex arithmetic.

**C.2 Descent Methods for Symmetric Problems**

Unlike the stationary Richardson method discussed previously, descent methods have no parameters such as $\omega$ that require knowledge of the extreme eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ of the matrix $L$ or of $H^{-1}L$, where $H$ is a suitable preconditioner. The principle is to adjust the current guess $v^n$ via

$$H(v^{n+1} - v^n) = \alpha_n r^n, \quad (C.2.1)$$

where $r^n = f - Lv^n$ is the residual, and the scalar $\alpha_n$—the dynamic relaxation parameter—is chosen according to some optimality criterion, as described below. In this section we will assume that both $L$ and $H$ are symmetric and positive definite (but the reader should be aware that these iterative methods may work even if this condition is not satisfied).

The most natural option for defining $\alpha_n$ is to minimize the Euclidean norm of the new residual $r^{n+1}$; another option is to minimize the so-called $H$-norm of the new preconditioned residual $p^{n+1} = H^{-1}r^{n+1}$, i.e., the quantity $||p^{n+1}||_H = (H p^{n+1}, p^{n+1})^{1/2} = ||r^{n+1}||_{H^{-1}}$. Both options are referred to as preconditioned minimum residual Richardson (PMRR) methods and will be denoted by PMRR$_2$ and PMRR$_H$, respectively. An additional option is to minimize the $L$-norm of the new error $e^{n+1} = u - v^{n+1}$, i.e., the quantity $||e^{n+1}||_L = (L e^{n+1}, e^{n+1})^{1/2}$. This is referred to as a preconditioned steepest descent Richardson (PSDR) method.

The corresponding algorithms can be written compactly as follows:

**Preconditioned Richardson Methods**

Initialize

$$v^0, \quad r^0 = f - Lv^0, \quad H p^0 = r^0.$$ 

Iterate

$$\alpha_n \quad \text{defined according to one of the rows of table C.1},$$

$$v^{n+1} = v^n + \alpha_n p^n,$$

$$r^{n+1} = r^n - \alpha_n Lp^n, \quad (C.2.2)$$

$$H p^{n+1} = r^{n+1}.$$
Table C.1. The three different strategies for Richardson iterations (PMRR)

<table>
<thead>
<tr>
<th>Name of method</th>
<th>acceleration parameter</th>
<th>method minimizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMRR(_2)</td>
<td>(\alpha_n = \frac{(r^n, Lp^n)}{(Lp^n, Lp^n)})</td>
<td>(|r^{n+1}|)</td>
</tr>
<tr>
<td>PMRR(_H)</td>
<td>(\alpha_n = \frac{(p^n, Lp^n)}{(Lp^n, H^{-1}Lp^n)})</td>
<td>(|p^{n+1}|_H)</td>
</tr>
<tr>
<td>PSDR</td>
<td>(\alpha_n = \frac{(p^n, r^n)}{(p^n, Lp^n)})</td>
<td>(|e^{n+1}|_L)</td>
</tr>
</tbody>
</table>

Note that for non-preconditioned iterations, then \(H = I\) and \(p^n = r^n\) in Table C.1. (In particular, PMRR\(_2\) and PMRR\(_H\) coincide if \(P = I\).)

For PMRR\(_H\) iterations the following estimate holds for the preconditioned residual:

\[
\|p^n\|_H \leq \left( \frac{\mathcal{K} - 1}{\mathcal{K} + 1} \right)^n \|p^0\|_H ,
\]  

(C.2.3)

where \(\mathcal{K}\) still denotes the iterative condition number of \(H^{-1}L\), while for PSDR iterations we have

\[
\|e^n\|_L \leq \left( \frac{\mathcal{K} - 1}{\mathcal{K} + 1} \right)^n \|e^0\|_L
\]  

(C.2.4)

(see Quarteroni and Valli (1994), Sect. 2.4). Note that when \(H = I\) (no preconditioning), the PSDR method reduces to the classical steepest descent (or gradient) algorithm. Also note that, in both cases, the number of iterations required for convergence is proportional to

\[
\mathcal{J} \simeq \frac{1}{2} \mathcal{K}.
\]  

(C.2.5)

When the eigenvalues of the preconditioned matrix \(H^{-1}L\) are complex but with dominant real parts, a surrogate for \(\mathcal{K}\) that is still representative of the convergence behavior of the Richardson iterations is

\[
\mathcal{K}^* = \frac{\max_j |\lambda_j|}{\min_j |\lambda_j|} .
\]  

(C.2.6)

A substantial improvement in the convergence rate can be achieved by using conjugate direction methods in place of PMRR or PSDR. The two most common conjugate direction methods are known as the conjugate gradient method and the conjugate residual method. These methods were proposed by
Hestenes and Stiefel in 1952 as direct methods for solving symmetric, positive-definite linear systems. For such problems, the conjugate direction methods produce the exact answer (in the absence of round-off errors) in a finite number of steps. In the late 1960s and early 1970s these methods began to be considered seriously as iterative, rather than direct, solution schemes that can produce a very accurate result in a small number of iterations.

In a non-preconditioned conjugate direction method the update of the iterate is generalized from (C.2.1) to

\[ v^{n+1} = v^n + \alpha_n p^n. \]  

(C.2.7)

In the conjugate gradient version, the directions satisfy the orthogonality property

\[ (p^{n+1}, Lp^n) = 0. \]  

(C.2.8)

The scheme is initialized with an initial guess \( v^0 \). The initial direction vector is chosen to be \( p^0 = r^0 \), where \( r^0 \) is the initial residual. Subsequent iterations are made according to the following formulas:

**Conjugate Gradient (CG) Method**

\[ \alpha_n = \frac{(r^n, r^n)}{(p^n, Lp^n)}, \]

\[ v^{n+1} = v^n + \alpha_n p^n, \]

\[ r^{n+1} = r^n - \alpha_n Lp^n, \]  

(C.2.9)

\[ \beta_n = \frac{(r^{n+1}, r^{n+1})}{(r^n, r^n)}, \]

\[ p^{n+1} = r^{n+1} + \beta_n p^n. \]

In (C.2.9) the formula for the familiar scalar \( \alpha_n \) results from the requirement that \( v^{n+1} \) minimize the energy norm of the error, and the formula for the additional scalar \( \beta_n \) is an equivalent, yet numerically preferred, form of the expression \( \beta_n = -(r^{n+1}, Lp^n)/(p^n, Lp^n) \), which follows from the requirement (C.2.8).

The following orthogonality properties hold:

\[ (r^k, r^l) = 0, \quad (p^k, Lp^l) = 0 \quad \text{for} \quad k \neq l. \]  

(C.2.10)

The first of these implies that \( r^m = 0 \) for some \( m \leq nd \), where \( nd \) is the order of the matrix \( L \). This explains the claim that the exact solution is obtained in a finite number of iterations. However, the presence of rounding errors leads to some contamination of the residual and direction vectors. The second orthogonality relation shows that the CG method does far more than
the original requirement (C.2.8); indeed, we say that the directions \( \{p^k\} \) are \( L \)-conjugated.

The favorable convergence properties of this method are reflected by the estimate for the energy error (which improves the one in (C.2.4)):

\[
\|e^n\|_L \leq 2 \left( \frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^n \|e^0\|_L . \tag{C.2.11}
\]

The number of iterations required for convergence is therefore proportional to

\[
J = \frac{1}{2} \sqrt{K} . \tag{C.2.12}
\]

This is a decided improvement over the result (C.2.5). Of course, the CG method is more costly per iteration, both in CPU time and storage.

The \textit{conjugate residual method} is similar, but now the orthogonality property is

\[
(Lp^{n+1}, Lp^n) = 0 , \tag{C.2.13}
\]

and the requirement on \( v^{n+1} \) is that it minimize the Euclidean norm of the residual.

Let us now include a symmetric preconditioning, denoted as usual by \( H \), in these descent methods. It is tempting to write (C.0.1) as either

\[
\tilde{L}u = \tilde{f} \quad \text{with} \quad \tilde{L} = H^{-1}L \quad \text{and} \quad \tilde{f} = H^{-1}f \tag{C.2.14}
\]

or

\[
\tilde{L}\tilde{u} = f, \quad \text{where} \quad \tilde{L} = LH^{-1} \quad \text{and} \quad \tilde{u} = Hu , \tag{C.2.15}
\]

and then apply the preceding formulas to either (C.2.14) or (C.2.15). However, \( \tilde{L} \) is not necessarily symmetric and positive definite (unless \( L \) and \( H^{-1} \) commute). We can, however, choose \( Q \) such that

\[
H = QQ^T , \tag{C.2.16}
\]

and use

\[
\tilde{L}\tilde{u} = \tilde{f} , \tag{C.2.17}
\]

with

\[
\tilde{L} = Q^{-1}LQ^{-T} , \quad \tilde{f} = Q^{-1}f , \quad \tilde{u} = Q^Tu . \tag{C.2.18}
\]

We also use

\[
\tilde{v} = Q^Tv , \quad \tilde{p} = Q^Tp , \quad \tilde{r} = Q^{-1}r . \tag{C.2.19}
\]

This ensures that the matrix \( \tilde{L} \) is symmetric and positive definite. After inserting (C.2.18) into the preceding schemes and then manipulating the expressions into computationally convenient forms, we arrive at the following:
Preconditioned Conjugate Gradient (PCG) Method
Initialize
\[ v^0, \quad r^0 = f - Lv^0, \quad Hz^0 = r^0, \quad p^0 = z^0. \]
Iterate
\[ \alpha_n = \frac{(r^n, z^n)}{(p^n, Lp^n)}, \]
\[ v^{n+1} = v^n + \alpha_n p^n, \]
\[ r^{n+1} = r^n - \alpha_n Lp^n, \]
\[ Hz^{n+1} = r^{n+1}, \]
\[ \beta_n = \frac{(r^{n+1}, z^{n+1})}{(r^n, z^n)}, \]
\[ p^{n+1} = z^{n+1} + \beta_n p^n. \]  
\[ (C.2.20) \]

Preconditioned Conjugate Residual (PCR) Method
Initialize
\[ v^0, \quad r^0 = f - Lv^0, \quad Hz^0 = r^0, \quad p^0 = z^0. \]
Iterate
\[ \alpha_n = \frac{(r^n, Lp^n)}{(Lp^n, Lp^n)}, \]
\[ v^{n+1} = v^n + \alpha_n p^n, \]
\[ r^{n+1} = r^n - \alpha_n Lp^n, \]
\[ Hz^{n+1} = r^{n+1}, \]
\[ \beta_n = \frac{(Lz^{n+1}, Lp^n)}{(Lp^n, Lp^n)}, \]
\[ p^{n+1} = z^{n+1} + \beta_n p^n. \]
\[ Lp^{n+1} = Lz^{n+1} + \beta_n Lp^n. \]  
\[ (C.2.21) \]

The preconditioned conjugate gradient method minimizes the \( L \)-norm of the error; thus, the associated error satisfies (C.2.11). However, now the relevant condition number is that of \( Q^{-1}LQ^{-T} \) (which coincides with that of \( H^{-1}L \)) rather than that of \( L \).

For the CG and CR methods, their orthogonality properties are lost when applied to nonsymmetric problems. In this case they are more properly called the truncated conjugate gradient (TCG) and truncated conjugate residual (TCR) methods. Their preconditioned versions are abbreviated as the PTCG and PTCR methods, and they are given by (C.2.20) and (C.2.21), respectively.

Although the descent methods described in this section may work for nonsymmetric problems, the methods in the following section are generally preferable for the general case.
C.3 Krylov Methods for Nonsymmetric Problems

The subject of iterative schemes for nonsymmetric problems has received much attention since the 1980s. The descent methods that we discuss in this subsection are but a small subset of the schemes that have been proposed.

Since the matrix \( L \) is not symmetric, we can use either one of the transformations (C.2.14)–(C.2.15) or (C.2.16)–(C.2.19). The preconditioned matrix \( \tilde{L} \) determines the performance of Krylov methods.

When the Richardson method (C.1.1) is applied to the solution of the linear system (C.0.1), the residual, \( r^n = f - Lv^n \), at the \( n \)-th iteration can be related to the initial residual as

\[
r^n = \prod_{j=0}^{n-1} (I - \omega_j L)r^0 = p_n(L)r^0 ,
\]

where \( \omega_j \) is the relaxation parameter at the \( j \)-th step, while \( p_n(L) \) indicates a polynomial in \( L \) of degree \( n \).

Let us introduce the space

\[
K_m(L; w) = \text{span}\{w, Lw, \ldots, L^{m-1}w\} , \quad m \geq 1 ,
\]

called the Krylov space of order \( m \) associated with the matrix \( L \) and the vector \( w \). Then, \( r^n \in K_{n+1}(L; r^0) \). From (C.1.1) we obtain

\[
v^n = v^0 + \sum_{j=1}^{n-1} \omega_j r^j ;
\]

thus,

\[
v^n - v^0 \in K_n(L; r^0)
\]

and

\[
v^n - v^0 = p_{n-1}(L)r^0 .
\]

More generally, methods can be devised in such a way that

\[
v^n - v^0 = q_{n-1}(L)r^0 ,
\]

where \( q_{n-1} \) is a polynomial chosen so that \( v^n \) represents the “best” approximation of the solution, \( u \), in \( K_n = v^0 + K_n(L; r^0) \). Any such method is called a Krylov method.

For any fixed \( m \geq 1 \), an orthonormal basis \( \{w_i\} \) for \( K_m(L; w) \) can be computed using the so-called Arnoldi algorithm. Setting \( w_1 = v/\|w\| \), we apply the Gram-Schmidt procedure: for \( k \geq 1 \),

\[
g_{i k} = w_i^T Lw_k , \quad i = 1, \ldots, k ,
\]

\[
z_k = Lw_k - \sum_{i=1}^{k} g_{i k} w_i ,
\]

\[
g_{k+1,k} = \|z_k\| .
\]
Should $z_k = 0$ the process terminates, and we say that a breakdown of the algorithm has occurred. Otherwise, we set

$$w_{k+1} = \frac{z_k}{\|z_k\|},$$

(C.3.7)

and the algorithm continues, incrementing $k$ by 1.

If the algorithm terminates at the step $m$, then $\{w_1, \ldots, w_m\}$ forms a basis for $K_m(L; v)$. In such a case, denoting by $W_m \in \mathbb{R}^{n \times m}$ the matrix whose columns are the vectors $w_i$, we obtain

$$W_m^T L W_m = G_m, \quad W_{m+1}^T L W_m = \hat{G}_m,$$

(C.3.8)

where $\hat{G}_m \in \mathbb{R}^{(m+1) \times m}$ is an upper-Hessenberg matrix whose entries are the $g_{ij}$, while $G_m \in \mathbb{R}^{m \times m}$ is the restriction of $\hat{G}_m$ to the first $m$ rows and $m$ columns. In our application, the Krylov space will be invariably constructed for $v = r^0$.

This algorithm for generating an orthonormal basis for a Krylov space of any order is the foundation for solving the linear system (C.0.1) by a Krylov method. The most natural approach would be to search for $v^n$ as the vector that minimizes the error $\|v^n - u\|$ in $\tilde{K}_n$. However, since $u$ is unknown, this method would not work in practice. Two alternative strategies that are workable are

1. Compute $v^n$ by enforcing that the residual $r^n$ be orthogonal to any vector in $K_n(L; r^0)$, i.e.,

$$v^T (f - L v^n) = 0 \quad \forall v \in K_n(L; r^0).$$

(C.3.9)

This leads to the so-called full orthogonalization method (FOM).

2. Compute $v^n \in \tilde{K}_n$ by minimizing the norm of the residual $r^n$, i.e.,

$$\|f - L v^n\| = \min_{v \in \tilde{K}_n} \|f - L v\|,$$

(C.3.10)

which yields the generalized minimum residual method (GMRES).

Note that

$$v^n = v^0 + W_n q^n,$$

(C.3.11)

where $q^n$ has to be chosen according to the selected optimality criterion ((C.3.9) or (C.3.10)).

Then,

$$r^n = r^0 - LW_n q^n,$$

since $r^0 = w_1 \|r_0\|$. From (C.3.8) it follows that

$$r^n = W_{n+1} (\|r^0\| e_1 - \hat{G}_n q^n),$$

(C.3.12)

where $e_1$ is the first unit vector of the canonical basis of $\mathbb{R}^{n+1}$. Thus, in the GMRES method, the solution at step $n$ is computed through (C.3.11) where $q^n$ minimizes $\| (\|r^0\| e_1 - \hat{G}_n q) \|$ with respect to $q$. 

(C.3.13)
Note that the matrix $W_{n+1}$ appearing in (C.3.12) does not change the value of $\|r^0\|$ since it is an orthogonal matrix.

Clearly, the GMRES method will be the more effective the smaller the number of iterations, particularly since at each step one has to solve a least-squares problem (C.3.13). The GMRES method in exact arithmetic enjoys the so-called finite-termination property, i.e., it terminates after at most $nd$ iterations, where again $nd$ denotes the order of the matrix $L$. Premature stops are due to a breakdown in the Arnoldi orthonormalization algorithm. This breakdown occurs only if the computed solution $v^n$ coincides with the exact solution $u$ for some $n < nd$. However, unless acceptable convergence is reached after just a few iterations, the GMRES method requires prohibitive computational costs for the orthogonalization and excessive storage for the retention of the Krylov subspace bases.

A popular variant consists of restarting GMRES after each $m$ iteration steps. This algorithm is referred to as GMRES($m$); the nonrestarted version is sometimes called full GMRES. As pointed out in van der Vorst (2003), there is no simple rule to determine a suitable value of $m$; in fact, the speed of convergence of GMRES($m$) may vary drastically for nearby values of $m$. In some cases, a superlinear convergence behavior of the full GMRES iterations is observed.

The convergence analysis of GMRES is not trivial, and we report just some of the more elementary results here. If $L$ is positive definite, i.e., its symmetric part $L_S$ has positive eigenvalues, then the $n$-th residual decreases according to the following bound:

$$\|r^n\| \leq \sin^n(\beta)\|r^0\|,$$  \hspace{1cm} (C.3.14)

where $\cos(\beta) = \lambda_{\text{min}}(L_S)/\|L\|$ with $\beta \in [0, \pi/2)$. As usual, $\| \cdot \|$ denotes the Euclidean vector or matrix norm. Moreover, GMRES($m$) converges for all $m \geq 1$. In order to obtain a bound on the residual at a step $n \geq 1$, let us assume that the matrix $L$ is diagonalizable:

$$L = T \Lambda T^{-1},$$

where $\Lambda$ is the diagonal matrix of eigenvalues, $\{\lambda_j\}_{j=1,...,nd}$, and $T = (\omega^1, \ldots, \omega^{nd})$ is the matrix whose columns are the right eigenvectors of $L$. Under these assumptions, the residual norm after $n$ steps of GMRES satisfies

$$\|r^n\| \leq \kappa_2(T)\delta\|r^0\|,$$

where $\kappa_2(T) = \|T\|_2\|T^{-1}\|_2$ is the condition number of $T$ defined in (C.1.12), and

$$\delta = \min_{p \in \mathbb{P}_n, p(0)=1} \max_{1 \leq i \leq nd} |p(\lambda_i)|.$$

Moreover, suppose that the initial residual is dominated by $m$ eigenvectors, i.e., $r^0 = \sum_{j=1}^m \alpha_j \omega^j + e$, with $\|e\|$ small in comparison to $\|\sum_{j=1}^m \alpha_j \omega^j\|$. 

and assume that if some complex \( \omega^j \) appears in the previous sum, then its conjugate \( \overline{\omega^j} \) appears as well. Then

\[
\|r^n\| \leq \kappa_2(T) c_n \|e\|,
\]

\[
c_n = \max_{p>n} \prod_{j=1}^{n} \left| \frac{\lambda_p - \lambda_j}{\lambda_j} \right| .
\]

Very often, \( c_n \) is of order one; hence, \( n \) steps of GMRES reduce the residual norm to the order of \( \|e\| \) provided that \( \kappa_2(T) \) is not too large.

In general, as highlighted from the previous estimate, the eigenvalue information alone is not enough, and information on the eigensystem is also needed. If the eigensystem is orthogonal, as for normal matrices, then \( \kappa_2(T) = 1 \), and the eigenvalues are descriptive for convergence. Otherwise, upper bounds for \( \|r^n\| \) can be provided in terms of both spectral and pseudospectral information, as well as the so-called field of values of \( L \):

\[
\mathcal{F}(L) = \{ v^H L v : \|v\| = 1 \},
\]

where the superscript \( H \) denotes the Hermitian transpose. If \( 0 \notin \mathcal{F}(L) \), then the estimate (C.3.14) can be improved by replacing \( \lambda_{\min}(L_S) \) with \( \text{dist}(0, \mathcal{F}(L)) \).

An extensive discussion of convergence of GMRES and GMRES\( (m) \) can be found in Saad (1996) and van der Vorst (2003).

The GMRES method can of course be implemented for a preconditioned system. We provide here an implementation of the preconditioned GMRES method with a left preconditioner \( H \).

**Preconditioned GMRES (PGMRES) Method**

Initialize

\[
v^0, Hr^0 = f - Lv^0, \beta = \|r^0\|, v^1 = r^0/\beta.
\]

Iterate

\[
\text{For } j = 1, \ldots, n \text{ Do}
\]

\[
\text{Compute } Hw^j = Lv^j
\]

\[
\text{For } i = 1, \ldots, j \text{ Do}
\]

\[
g_{ij} = (v^i)^T w^j
\]

\[
w^j = w^j - g_{ij} v^i
\]

\[
\text{End Do}
\]

\[
g_{j+1,j} = \|w^j\|
\]

\[
(\text{if } g_{j+1,j} = 0 \text{ set } n = j \text{ and Goto (1))}
\]

\[
v^{j+1} = w^j / g_{j+1,j}
\]

\[
\text{End Do}
\]

\[
W_n = [v^1, \ldots, v^n], \quad \hat{G}_n = \{g_{ij}\}, 1 \leq j \leq n, 1 \leq i \leq j + 1;
\]

(1) Compute \( q^n \), the minimizer of \( \|\beta e_1 - \hat{G}_n q\| \)

Set \( v^n = v^0 + W_n q^n \)
More generally, as proposed by Saad (1996), a variable preconditioner $H_n$ can be used at the $n$-th iteration, yielding the so-called flexible GMRES method. The use of a variable preconditioner is especially interesting in those situations where the preconditioner is not explicitly given, but implicitly defined, for instance, as an approximate Jacobian in a Newton iteration or by a few steps of an inner iteration process. Another meaningful case is the one of domain-decomposition preconditioners (of either Schwarz or Schur type) where the preconditioning step involves one or several substeps of local solves in the subdomains (see Chap. 6).

Several considerations for the practical implementation of GMRES, its relation with FOM, how to restart GMRES, and the Householder version of GMRES can be found in Saad (1996).

A different approach to iterative methods for nonsymmetric matrices consists of generalizing the conjugate gradient method through a specific characterization of the properties satisfied by the residual.

The property that the residual vectors $\mathbf{r}^n$ generated by the CG method satisfy a three-term recurrence is lost when $L$ is not symmetric. The bi-conjugate gradient (Bi-CG) method introduced by Fletcher in 1976 constructs a residual $\mathbf{r}^k$ orthogonal to another row of vectors $\tilde{\mathbf{r}}^0, \tilde{\mathbf{r}}^1, \ldots, \tilde{\mathbf{r}}^{n-1}$, and, vice versa, $\tilde{\mathbf{r}}^n$ is orthogonal with respect to $\mathbf{r}^0, \mathbf{r}^1, \ldots, \mathbf{r}^{n-1}$. This method enjoys the finite-termination property, but there is no minimization property as in CG or GMRES for the intermediate steps. When this method converges, both $\{\mathbf{r}^n\}$ and $\{\tilde{\mathbf{r}}^n\}$ converge towards zero but only the convergence of the $\{\tilde{\mathbf{r}}^n\}$ is exploited. Based on this observation, Sonneveld in 1989 proposed a modification called the conjugate gradient-squared (CGS) method that focuses more strongly on the $\{\mathbf{r}^n\}$ vectors. CGS generates residual vectors $\mathbf{r}^n$ given by

$$\mathbf{r}^n = p_n^2(L)\mathbf{r}^0,$$

where $p_n(L)$ is that $n$-th degree polynomial in $L$ for which $p_n(L)\mathbf{r}^0$ is equal to the residual at the $n$-th step obtained by means of the Bi-CG method.

In the Bi-CGStab method, introduced by van der Vorst (1992), instead of simply squaring the Bi-CG polynomial, as in CGS, the more general form

$$\mathbf{r}^n = q_n(L)p_n(L)\mathbf{r}^0,$$  \hspace{1cm} (C.3.16)

is used, where now $q_n(x) = \prod_{i=1}^n (1 - \omega_i x)$, and $\omega_i$ are suitable constants chosen in such a way that $\|\mathbf{r}^n\|$ is minimized with respect to $\omega_i$.

The preconditioned algorithm can be described as follows:

Preconditioned Bi-CGStab (PBi-CGStab) Method

Initialize

$$\mathbf{v}^0, \quad \mathbf{r}^0 = \mathbf{f} - L\mathbf{v}^0, \quad \text{choose } \tilde{\mathbf{r}}^0 \text{ s.t. } (\tilde{\mathbf{r}}^0, \mathbf{r}^0) \neq 0, \ (\text{e.g., } \tilde{\mathbf{r}}^0 = \mathbf{r}^0)$$
Iterate

\[
\rho_{n-1} = (r^{n-1}, \tilde{r}^0)
\]

if \(\rho_{n-1} = 0\)
then the method fails
end if

if \(n = 1\)
then \(p^n = r^{n-1}\)
else \(\beta_{n-1} = (\rho_{n-1}/\rho_{n-2})(\alpha_{n-1}/\omega_{n-1})\)
\(p^n = r^{n-1} + \beta_{n-1}(p^{n-1} - \omega_{n-1}w^{n-1})\)
end if

\[H\hat{p} = p^n\]
\[w^n = L\hat{p}\]
\[\alpha_n = \rho_{n-1}/(w^n, \tilde{r}^0)\]

if \(\|s\|\) small enough
then \(v^n = v^{n-1} + \alpha_n\hat{p}; \text{ quit}\)
end if

\[H\hat{s} = s\]
\[t = L\hat{s}\]
\[\omega_n = (t, s)/(t, t)\]
\[v^n = v^{n-1} + \alpha_n\hat{p} + \omega_n\hat{s}\]
if \(v^n\) is accurate enough
then quit
end if
\[r^n = s - \omega_n t\]

For continuation it is necessary that \(\omega_n \neq 0\).

For an unfavorable choice of \(\tilde{r}^0, \rho_n\) or \((w^n, \tilde{r}^0)\) can be 0 or very small. In this case one has to restart, e.g., with \(\tilde{r}^0\) and \(\tilde{v}^0\) given by the last available values of \(r^n\) and \(v^n\). In exact arithmetic, Bi-CGStab is also a finite termination method (i.e., \(v^n = u\) for some \(n \leq nd\)). Its theoretical convergence properties are similar to those of CGS; however, it converges more smoothly, i.e., the oscillations of the residuals (with \(n\) of Bi-CGStab are in general less pronounced than those of CGS.

It is clear from the previous algorithm description that a weakness of Bi-CGStab is that a breakdown occurs if an \(\omega_n\) is equal to zero (but also a very small \(\omega_n\) may be troublesome).

Another non-ideal property is that the \(q_n\) polynomial in (C.3.16) has only real roots by construction, whereas optimal reduction polynomials for matrices with complex eigenvalues may also have complex roots. These considerations have led to the introduction of a variant, called Bi-CGStab(2), in which \(q_n\) is constructed as a product of quadratic factors. For its derivation and analysis, the reader is referred, e.g., to van der Vorst (2003).
Unfortunately, for a general nonsymmetric matrix, Krylov methods are not guaranteed to converge, but neither are any other known iterative methods. As noted earlier, GMRES\((m)\) does have a convergence guarantee if \(L_S\) has positive eigenvalues.
Appendix D. Time Discretizations

In this appendix we will make some general comments about time-discretizations, survey classical methods for ODEs and their stability regions, and highlight some low-storage time-discretization formulas that have been widely used in conjunction with spectral methods. Some standard references from the extensive literature on numerical methods for ODEs are the books by Gear (1971), Lambert (1991), Shampine (1994), Hairer, Norsett and Wanner (1993), Hairer and Wanner (1996) and Butcher (2003).

D.1 Notation and Stability Definitions

The typical evolution equation can be written

\[
\frac{\partial u}{\partial t} = f(u, t) , \quad t > 0 ,
\]

\[u(0) = 0 ,\]  \tag{D.1.1}

where the (generally) nonlinear operator \(f\) contains the spatial part of the PDE. The unknown \(u\) can be either a scalar function or a vector function (the latter case occurs, e.g., for the Euler or Navier–Stokes equations in fluid dynamics).

After space discretization (by one of the several methods of spectral type considered in this book), for all times \(t > 0\) the exact solution \(u(t)\) is replaced by a function that is a polynomial \(u^N(t)\) for single-domain classical spectral methods or a piecewise (mapped) polynomial function \(u_\delta(t)\) for multidomain spectral methods. In both cases, this function is represented according to a chosen basis; let \(u(t)\) denote the vector of the unknown coefficients in this representation. Then, the spatial discretization can be written in the algebraic form

\[
\frac{du}{dt} = f(u, t) , \quad t > 0 ,
\]

\[u(0) = u_0 ,\]  \tag{D.1.2}

where \(f\) is the vector-valued function governing the semi-discrete problem. For Galerkin-type methods (such as, e.g., SEM, MEM, SDGM and their
\(\frac{\text{d}u}{\text{d}t} = -L u + b\), \(t > 0\), \(u(0) = u_0\),

where \(L\) is the matrix representing the spatial discretization by the chosen spectral method. This is also called a method-of-lines approach or a continuous-in-time discretization.

A corresponding, representative scalar model problem is

\[
\frac{\text{d}u}{\text{d}t} = \lambda u,
\]

where \(\lambda\) is a complex number, which for (D.1.2) is “representative” of the partial derivative of \(f\) with respect to \(u\) (in the scalar case) or of the eigenvalues of the Jacobian matrix \((\partial f_i/\partial u_j)\) in the vector case, and which for (D.1.3) is representative of the eigenvalues of \(-L\).

In most applications of spectral methods to partial differential equations, the spatial discretization is spectral but the temporal discretization uses conventional finite differences. In describing the time-discretizations, we denote the time-step by \(\Delta t\), the \(n\)-th time-level by \(t_n = n\Delta t\), the approximate solution at time-step \(n\) by \(u^n\), and set \(f^n = f(u^n, t^n)\).

If the spatial discretization is presumed fixed, then we use the term stability in its ODE context. The time-discretization is said to be \textit{stable} (sometimes called \textit{zero-stable}) if there exist positive constants \(\sigma, \varepsilon\) and \(C(T)\), independent of \(\Delta t\), such that, for all \(T > 0\) (perhaps limited by a maximal \(T_{\text{max}}\) depending on the problem) and for all \(0 \leq \Delta t < \sigma\),

\[
\|u^n - v^n\| \leq C(T)\|u^0 - v^0\| \quad \text{for } 0 \leq t_n \leq T
\]

provided that \(\|u^0 - v^0\| < \varepsilon\), where \(\|u^n\|\) is some spatial norm of \(u^n\). The constant \(C(T)\) is permitted to grow with \(T\). Here, \(v^n\) is the solution obtained by the same numerical method corresponding to a (perturbed) initial data \(v^0\).

For many problems involving integration over long time intervals, a method which admits the temporal growth allowed by the estimate (D.1.5) is undesirable. As one example, take a problem of the form (D.1.2) for which \((\partial f_i/\partial u_j)(w, t)\) is negative for all \(w\) and \(t\), or more generally, for which \(f\) satisfies the \textit{right Lipschitz condition}: there exists \(\mu < 0\) such that

\[
\langle f(u, t) - f(v, t), u - v \rangle \leq \mu\|u - v\|^2 \quad \text{for all } u, v, t ,
\]
where $\langle \cdot, \cdot \rangle$ is a suitable scalar product and $\| \cdot \|$ its associated norm. In these cases,
\[
\| u(t) - v(t) \| \leq e^{\mu t} \| u(0) - v(0) \|.
\]
(Such problems are referred to as dissipative Cauchy problems in the ODE literature.) The ODEs resulting from spectral spatial discretizations of the heat equation or the time-dependent Stokes equations for incompressible flows (with homogeneous boundary data and zero source term) fall into this category. In this case one desires that the time-discretization be \textit{asymptotically stable}, i.e., that instead of (D.1.5) it satisfy the stronger requirement
\[
\| u^n - v^n \| \to 0 \quad \text{as} \quad t_n \to +\infty,
\]
(D.1.7)

As another example for which the above notion of stability is too weak, consider ODEs resulting from the spatial discretization of linear, spatially periodic, purely hyperbolic systems. For these problems, asymptotic stability for the time-discretization is undesirable since the exact solution is undamped in time. Instead, we rather desire a time-discretization which is \textit{temporally stable}, for which we merely require that
\[
\| u^n \| \leq \| u^0 \| \quad \text{for all} \quad n \geq 1.
\]
(D.1.8)

The notion of \textit{weak instability} is sometimes used in a loose sense for schemes which admit solutions to periodic hyperbolic problems which grow with time, but for which the growth rate decreases with $\Delta t$. For example, the constant $C(T)$ in (D.1.5) might have the form
\[
C(T) = e^{\alpha(\Delta t)^p T},
\]
where $\alpha > 0$ and $p$ is a positive integer. For such weakly unstable schemes, the longer the time interval of interest, i.e., the larger is $T$, the smaller must $\Delta t$ be chosen to keep the spurious growth of the solution within acceptable bounds.

Another notion that is relevant to periodic, hyperbolic problems is that of \textit{reversible} (or \textit{symmetric}) time-discretizations. These are schemes for which the solution may be marched forward from $t^n$ to $t^{n+1}$ and then backwards to $t^n$ with the starting solution at $t^n$ recovered exactly (except for round-off errors).

Two final definitions are in order for our subsequent discussion. The \textit{absolute stability region} (often referred to just as the \textit{stability region}), say $\mathcal{A}$, of a numerical method is customarily defined for the scalar model problem (D.1.4) to be the set of all complex numbers $\alpha = \lambda \Delta t$ such that any sequence $\{u^n\}$ generated by the method with such $\lambda$ and $\Delta t$ satisfies $\|u^n\| \leq C$ as $t_n \to \infty$, for a suitable constant $C$. Furthermore, a method is called $A$-\textit{stable} if the region of absolute stability includes the region $\text{Re}(\lambda \Delta t) < 0$. We warn the reader that in some books the absolute stability region is defined as the set of all $\lambda \Delta t$ such that $\|u^n\| \to 0$ as $t_n \to \infty$. This new region, say
$A^0$, would not necessarily coincide with $A$. In general, if $A^0$ is non-empty, $A$ is its closure. However, there are cases for which $A^0$ is empty (e.g., the midpoint or leap-frog method) and $A$ is not ($A = \{z = \alpha i, -1 \leq \alpha \leq 1\}$ for the midpoint method). Finally, we note that zero-stable methods are those for which $A$ contains the origin $z = 0$ of the complex plane.

As noted by Reddy and Trefethen (1990, 1992), having the eigenvalues scaled by the time-step $\Delta t$ falling within the absolute stability region of the ODE method is not always sufficient for stability of the computation. They present a stability criterion utilizing the so-called $\epsilon$-pseudospectra. However, as discussed by Trefethen (2000, Chap. 10), in almost all cases the “rule-of-thumb” condition involving the standard eigenvalues is acceptable.

On the other hand, we may be interested in the behavior of the computed solution as both the spatial and temporal discretizations are refined. We now define stability by an estimate of the form (D.1.5) where $C$ is independent of $\Delta t$, $\epsilon$ and the spatial discretization parameter $N$ or $\delta$, the norm is independent of $N$ or $\delta$, but $\sigma$ will in general be a function of $N$ or $\delta$. The functional dependence of $\sigma$ upon $N$ or $\delta$ which is necessary to obtain an estimate of the form (D.1.5) is termed the stability limit of the numerical method. If $\sigma$ is in fact independent of $N$ or $\delta$, then the method is called unconditionally stable. Clearly, a necessary condition for the fully discrete problem to be stable is that the semi-discrete problem be spatially stable. Likewise, a temporal stability limit for the fully discrete scheme for a hyperbolic system is the functional dependence of $\sigma$ upon $N$ or $\delta$ which is necessary to obtain an estimate of the form (D.1.8).

### D.2 Standard ODE Methods

In this section we furnish as a convenience the basic formulas and diagrams for the absolute stability regions for those time-discretizations of (D.1.2) that are most commonly used in conjunction with spectral discretizations in space. Among the factors which influence the choice of a time-discretization are the accuracy, stability, storage requirements, and work demands of the methods. The storage and work requirements of a method can be deduced in a straightforward manner from the definition of the method and the nature of the PDE. The accuracy of a method follows from a truncation error analysis, and the stability for a given problem is intimately connected with the spectrum of the spatial discretization. In this section we will describe some of the standard methods for ODEs and relate their stability regions to the spectra of the advection and diffusion operators. Bear in mind that in many problems different time-discretizations are used for different spatial terms in the equation. The illustrations of the spectra of the spectral differentiation, mass and stiffness matrices furnished in CHQZ2 and throughout this book combined with the stability diagrams in this section suffice for general conclusions to be drawn
on appropriate choices of time-discretization methods and time-step limits for temporal stability.

For the reader’s convenience, Table D.1 provides the numerical values of the intersections of the absolute stability regions with the negative real axis and the positive imaginary axis for all methods discussed in this section.

### D.2.1 Leap Frog Method

The *leap frog* (LF) method (also called *midpoint* method) is a second-order, two-step scheme given by

\[ u^{n+1} = u^{n-1} + 2\Delta t f^n. \]  

(D.2.1)

This produces solutions of constant norm for the model problem provided that \( \lambda \Delta t \) is on the imaginary axis and that \( |\lambda \Delta t| \leq 1 \) (see Table D.1). Thus, leap frog is a suitable explicit scheme for problems with purely imaginary eigenvalues. It also is a reversible, or symmetric, method. However, since it is only well-behaved on a segment in the complex \( \lambda \Delta t \)-plane for the model problem, extra care is needed in practical situations.

The most obvious application is to periodic advection problems, for the eigenvalues of the Fourier approximation to \( d/dx \) are imaginary. The difficulty with the leap frog method is that the solution is subject to a temporal oscillation with period \( 2\Delta t \). This arises from the extraneous (spurious) solution to the temporal difference equations. The oscillations can be controlled by every so often averaging the solution at two consecutive time-levels.

Leap frog is quite inappropriate for problems whose spatial eigenvalues have nonzero real parts. This certainly includes the approximation of diffusion operators. Leap frog is also not viable for advection operators with nonperiodic boundary conditions, since the discrete spectra of Chebyshev and Legendre approximations to the standard advection operator have appreciable real parts.

### D.2.2 Adams–Bashforth Methods

This is a class of explicit multistep methods which includes the simple *forward Euler* (FE) method

\[ u^{n+1} = u^n + \Delta t f^n, \]  

(D.2.2)

the popular *second-order Adams–Bashforth* (AB2) method

\[ u^{n+1} = u^n + \frac{1}{2} \Delta t \left[ 3f^n - f^{n-1} \right], \]  

(D.2.3)

the still more accurate *third-order Adams–Bashforth* (AB3) method

\[ u^{n+1} = u^n + \frac{1}{12} \Delta t \left[ 23f^n - 16f^{n-1} + 5f^{n-2} \right], \]  

(D.2.4)
and the fourth-order Adams–Bashforth (AB4) method

$$u^{n+1} = u^n + \frac{1}{24} \Delta t \left[ 55f^n - 59f^{n-1} + 37f^{n-2} - 9f^{n-3} \right].$$ (D.2.5)

These methods are not reversible.

The stability regions $\mathcal{A}$ of these methods are shown in Fig. D.1 (left) and the stability boundaries along the axes are given in Table D.1. Note that the size of the stability region decreases as the order of the method increases. Note also that except for the origin, no portion of the imaginary axis is included in the stability regions of the first and second-order methods, whereas the third- and fourth-order versions do have some portion of the imaginary axis included in their stability regions. Nevertheless, the AB2 method is weakly unstable, i.e., for a periodic, hyperbolic problem the acceptable $\Delta t$ decreases at $T$ increases.

As is evident from Fig. D.1 (left), higher order AB methods are temporally stable for Fourier approximations to periodic advection problems. Let the upper limit of the absolute stability region along the imaginary axis be denoted by $c$. Then the temporal stability limit is

$$\frac{N}{2} \Delta t \leq c, \quad \text{or} \quad \Delta t \leq \frac{c}{\pi} \Delta x.$$ (D.2.6)

The limit on $\Delta t$ is smaller by a factor of $\pi$ than the corresponding limit for a second-order finite-difference approximation in space. The Fourier spectral approximation is more accurate in space because it represents the high-frequency components much more accurately than the finite-difference method. The artificial damping of the high-frequency components which is produced by finite-difference methods enables the stability restriction on the time-step to be relaxed.

Chebyshev and Legendre approximations to advection problems appear to be temporally stable under all Adams–Bashforth methods for sufficiently small $\Delta t$; precisely, for $\Delta t \leq CN^{-2}$ for a suitable constant $C$. (For simplicity, this and the subsequent stability limits refer to a single-domain discretization. For multidomain methods, the limits on $\Delta t$ should also scale with the size of the subdomains, in a way that depends on the specific spatial discretization method that is being used). Since the spatial eigenvalues all have negative real parts, the failure of the AB2 method to include the imaginary axis in its absolute stability region does not preclude temporal stability.

The temporal stability limits for Adams–Bashforth methods for Fourier, Chebyshev and Legendre approximations to diffusion equations are easy to deduce since their spatial eigenvalues (i.e., the eigenvalues of the matrix $-L$) are real, negative and limited in modulus as indicated, e.g., in CHQZ2, Chap. 4. Combining this information with the stability bounds along the negative real axis as provided in Table D.1, one gets that $\Delta t$ should be limited by a constant times $N^{-2}$ for Fourier approximations, by a constant times $N^{-4}$ for Chebyshev or Legendre collocation approximations, and by a constant times $N^{-3}$ for Legendre G-NI approximations.
D.2.3 Adams–Moulton Methods

A related set of implicit multistep methods are the Adams–Moulton methods. They include the *backward Euler* (BE) method

\[ u^{n+1} = u^n + \Delta t f^{n+1}, \]  

(D.2.7)

the *Crank–Nicolson* (CN) method

\[ u^{n+1} = u^n + \frac{1}{2} \Delta t [f^{n+1} + f^n], \]  

(D.2.8)

the *third-order Adams–Moulton* (AM3) method

\[ u^{n+1} = u^n + \frac{1}{12} \Delta t [5f^{n+1} + 8f^n - f^{n-1}], \]  

(D.2.9)

and the *fourth-order Adams–Moulton* (AM4) method

\[ u^{n+1} = u^n + \frac{1}{24} \Delta t [9f^{n+1} + 19f^n - 5f^{n-1} + f^{n-2}]. \]  

(D.2.10)

Forward Euler (FE) (see D.2.2), backward Euler (BE) and Crank–Nicolson (CN) methods are special cases of \( \theta \)-methods, defined as

\[ u^{n+1} = u^n + \Delta t [\theta f^{n+1} + (1 - \theta) f^n], \]  

(D.2.11)

for \( 0 \leq \theta \leq 1 \). Precisely, they correspond to the choice \( \theta = 0 \) (FE), \( \theta = 1 \) (BE) and \( \theta = 1/2 \) (CN). All \( \theta \)-methods except for FE are implicit. All \( \theta \)-methods are first-order accurate, except for CN, which is second-order. For each \( \theta < \frac{1}{2} \), the absolute stability region is the circle in the left half-plane \( \text{Re}(\lambda \Delta t) \leq 0 \) with center \( z = (2\theta - 1)^{-1} \) and radius \( r = (1 - 2\theta)^{-1} \). The stability region of the CN method coincides with the half-plane \( \text{Re}(\lambda \Delta t) \leq 0 \). For each \( \theta > \frac{1}{2} \), the absolute stability region is the exterior of the open circle in the right half-plane \( \text{Re}(\alpha) > 0 \) with center \( z = (2\theta - 1)^{-1} \) and radius \( r = (2\theta - 1)^{-1} \). Thus, all \( \theta \)-methods for \( \frac{1}{2} \leq \theta \leq 1 \) are A-stable.

The absolute stability regions of the third- and fourth-order Adams–Moulton methods are displayed in Fig. D.1 (right) and the stability boundaries along the axes are given in Table D.1. In comparison with the explicit Adams–Bashforth method of the same order, an Adams–Moulton method has a smaller truncation error (by factors of five and nine for second and third-order versions), a larger stability region, and requires one fewer levels of storage. However, it does require the solution of an implicit set of equations. The CN method is reversible; the others are not.

The CN method is commonly used for diffusion problems. In Navier–Stokes calculations, it is frequently applied to the viscous and pressure gradient components. Although CN is absolutely stable for the former and temporally stable for the latter, it has the disadvantage that it damps high-frequency components very weakly, whereas in reality these components decay very rapidly.
Fig. D.1. Absolute stability regions of Adams–Bashforth (left) and Adams–Moulton (right) methods.

Fig. D.2. Absolute stability regions of backwards-difference formulas (left) and Runge–Kutta methods (right). The BDF methods are absolutely stable on the exteriors (and boundaries) of the regions enclosed by the curves, whereas the RK methods are absolutely stable on the interiors (and boundaries) of the regions enclosed by the curves.

The Adams–Moulton methods of third and higher order are only conditionally stable for advection and diffusion problems. The stability limits implied by Fig. D.1 indicate that the stability limit of a high-order Adams–Moulton method is roughly ten times as large for a diffusion problem as the stability limit of the corresponding Adams–Bashforth method. In addition, AM3 and AM4 are weakly unstable for Fourier approximations to advection problems, since the origin is the only part of the imaginary axis which is included in their absolute stability regions.
D.2 Standard ODE Methods

Table D.1. Intersections of absolute stability regions with the negative real axis (left) and with the positive imaginary axis (right)

<table>
<thead>
<tr>
<th>Method</th>
<th>$A \cap \mathbb{R}^-$</th>
<th>$A \cap i\mathbb{R}^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leap frog (midpoint)</td>
<td>${0}$</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Forward Euler</td>
<td>$[-2, 0]$</td>
<td>${0}$</td>
</tr>
<tr>
<td>Crank–Nicolson</td>
<td>$(-\infty, 0]$</td>
<td>[0, $+\infty$)</td>
</tr>
<tr>
<td>Backward Euler</td>
<td>$(-\infty, 0]$</td>
<td>[0, $+\infty$)</td>
</tr>
<tr>
<td>$\theta$-method, $\theta &lt; 1/2$</td>
<td>$[2/(2\theta - 1), 0]$</td>
<td>${0}$</td>
</tr>
<tr>
<td>$\theta$-method, $\theta \geq 1/2$</td>
<td>$(-\infty, 0]$</td>
<td>[0, $+\infty$)</td>
</tr>
<tr>
<td>AB2</td>
<td>$(-1, 0]$</td>
<td>${0}$</td>
</tr>
<tr>
<td>AB3</td>
<td>$[-6/11, 0]$</td>
<td>[0, 0.723]</td>
</tr>
<tr>
<td>AB4</td>
<td>$[-3/10, 0]$</td>
<td>[0, 0.43]</td>
</tr>
<tr>
<td>AM3</td>
<td>$[-6, 0]$</td>
<td>${0}$</td>
</tr>
<tr>
<td>AM4</td>
<td>$[-3, 0]$</td>
<td>${0}$</td>
</tr>
<tr>
<td>BDF2</td>
<td>$(-\infty, 0]$</td>
<td>[0, $+\infty$)</td>
</tr>
<tr>
<td>BDF3</td>
<td>$(-\infty, 0]$</td>
<td>[0, 1.94]</td>
</tr>
<tr>
<td>BDF4</td>
<td>$(-\infty, 0]$</td>
<td>[0, 4.71]</td>
</tr>
<tr>
<td>RK2</td>
<td>$[-2, 0]$</td>
<td>${0}$</td>
</tr>
<tr>
<td>RK3</td>
<td>$[-2.51, 0]$</td>
<td>[0, 1.73]</td>
</tr>
<tr>
<td>RK4</td>
<td>$[-2.79, 0]$</td>
<td>[0, 2.83]</td>
</tr>
</tbody>
</table>

D.2.4 Backwards-Difference Formulas

Another class of implicit time discretizations is based upon backwards-difference formulas. These include the first-order backwards-difference scheme (BDF1), which is identical to backward Euler, the second-order backwards-difference scheme (BDF2)

$$u^{n+1} = \frac{1}{3}[4u^n - u^{n-1}] + \frac{2}{3} \Delta tf^{n+1}, \quad (D.2.12)$$

the third-order backwards-difference scheme (BDF3)

$$u^{n+1} = \frac{1}{11}[18u^n - 9u^{n-1} + 2u^{n-2}] + \frac{6}{11} \Delta tf^{n+1}, \quad (D.2.13)$$

and the fourth-order backwards-difference scheme (BDF4)

$$u^{n+1} = \frac{1}{25}[48u^n - 36u^{n-1} + 16u^{n-2} - 3u^{n-3}] + \frac{12}{25} \Delta tf^{n+1}. \quad (D.2.14)$$

The absolute stability regions of these methods are displayed in Fig. D.2 (left), and the stability boundaries along the axes are given in Table D.1. The stability regions are much larger than those of the corresponding AM methods.
D.2.5 Runge–Kutta Methods

Runge–Kutta methods are single-step, but multistage, time-discretizations. The modified Euler version of a second-order Runge–Kutta (RK2) method can be written

$$u^{n+1} = u^n + \frac{1}{2} \Delta t [f(u^n, t^n) + f(u^n + \Delta t f(u^n, t^n), t^n + \Delta t)].$$ \hfill (D.2.15)

A popular third-order Runge–Kutta (RK3) method is

$$k_1 = f(u^n, t^n),
\quad k_2 = f(u^n + \frac{1}{2} \Delta t k_1, t^n + \frac{1}{2} \Delta t),
\quad k_3 = f(u^n + \frac{3}{4} \Delta t k_2, t^n + \frac{3}{4} \Delta t),$$

$$u^{n+1} = u^n + \frac{1}{6} \Delta t [2k_1 + 3k_2 + 4k_3].$$ \hfill (D.2.16)

The classical fourth-order Runge–Kutta (RK4) method is

$$k_1 = f(u^n, t^n),
\quad k_2 = f(u^n + \frac{1}{2} \Delta t k_1, t^n + \frac{1}{2} \Delta t),
\quad k_3 = f(u^n + \frac{1}{2} \Delta t k_2, t^n + \frac{1}{2} \Delta t),
\quad k_4 = f(u^n + \Delta t k_3, t^n + \Delta t),$$

$$u^{n+1} = u^n + \frac{1}{6} \Delta t [k_1 + 2k_2 + 2k_3 + 4k_4].$$ \hfill (D.2.17)

All Runge–Kutta methods of a given order have the same stability properties. The absolute stability region are given in Fig. D.2 (right), and the stability boundaries along the axes are given in Table D.1. Note that the stability region expands as the order increases. Note also that RK2 methods are afflicted with the same weak instability as the AB2 scheme. When storage is not an issue, then the classical RK4 method is commonly used. Otherwise, the low-storage versions of third and fourth-order methods, such as those described in Sect. D.3, have been preferred.

In the event that $f$ contains no explicit dependence upon $t$, the following formulation, due to Jameson et al. (1981), applies:

Set

$$u = u^n$$

For $k = s, 1, -1$

$$u \leftarrow u^n + \frac{1}{k} \Delta t f(u)$$ \hfill (D.2.18)

End For

$$u^{n+1} = u.$$
It yields a Runge–Kutta method of order $s$ (for linear problems) and requires at most three levels of storage.

### D.3 Low-Storage Schemes

When high-order discretization schemes such as spectral methods are employed in space, the primary contributor to the error in the fully discrete approximation is usually the temporal discretization error unless the time-discretization itself is at least third order or the time-step is very small. When computations are constrained by memory limitations, a premium is placed on minimizing storage demands. This has made special low-storage Runge–Kutta methods very attractive for large-scale problems. Several popular low-storage Runge–Kutta methods are available that permit third-order or fourth-order temporal accuracy to be obtained with only two levels of storage. Such economies are not available for multistep methods.

We shall note here some of the low-storage Runge–Kutta methods that have been widely used for large-scale spectral computations. The description shall be given for the ODE

$$\frac{du}{dt} = g(u, t) + l(u, t), \quad (D.3.1)$$

where $g(u, t)$ is treated with a low-storage Runge–Kutta method and $l(u, t)$ is treated implicitly with the Crank–Nicolson method. Such mixed explicit/implicit time-discretizations are very common for incompressible Navier–Stokes computations, for which $g(u, t)$ represents (nonlinear) advection and $l(u, t)$ (linear) diffusion.

The general representation of a low-storage Runge–Kutta/Crank–Nicolson method requiring only 2 levels of storage (for $u$ and $h$) is

$$h = 0$$

$$u = u^n$$

For $k = 1$ to $K$

$$t^k = t^n + \alpha_k \Delta t$$

$$t^{k+1} = t^n + \alpha_{k+1} \Delta t$$

$$h \leftarrow g(u, t^k) + \beta_k h$$

$$\mu = \frac{1}{2} \Delta t (\alpha_{k+1} - \alpha_k)$$

$$v - \mu l(v, t^{k+1}) = u + \gamma_k \Delta t h + \mu l(u, t^k)$$

$$u \leftarrow v$$

End For

$$u^{n+1} = u$$
Table D.2. Coefficients of low-storage Runge–Kutta/Crank–Nicolson schemes

<table>
<thead>
<tr>
<th></th>
<th>Williamson 3rd-order</th>
<th>Carpenter-Kennedy 4th-order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>1/3</td>
<td>0.1496590219993</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>3/4</td>
<td>0.3704009573644</td>
</tr>
<tr>
<td>$\alpha_4$</td>
<td>1</td>
<td>0.6222557631345</td>
</tr>
<tr>
<td>$\alpha_5$</td>
<td>–</td>
<td>0.9582821306748</td>
</tr>
<tr>
<td>$\alpha_6$</td>
<td>–</td>
<td>1</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>–5/9</td>
<td>–0.4178904745</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>–153/128</td>
<td>–1.192151694643</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>–</td>
<td>–1.697784692471</td>
</tr>
<tr>
<td>$\beta_5$</td>
<td>–</td>
<td>–1.514183444257</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>1/3</td>
<td>0.1496590219993</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>15/16</td>
<td>0.3792103129999</td>
</tr>
<tr>
<td>$\gamma_3$</td>
<td>8/15</td>
<td>0.8229550293869</td>
</tr>
<tr>
<td>$\gamma_4$</td>
<td>–</td>
<td>0.6994504559488</td>
</tr>
<tr>
<td>$\gamma_5$</td>
<td>–</td>
<td>0.1530572479681</td>
</tr>
</tbody>
</table>

(note that the penultimate instruction in the loop indicates that $\mathbf{v}$ is the solution of the implicit equation on the left-hand side).

Table D.2 lists the values of these parameters for one third-order scheme, due to Williamson (1980), and one fourth-order scheme from Carpenter and Kennedy (1994). The stability limits (on the imaginary axis) for these schemes are 1.73 for the third-order scheme and 3.34 for the fourth-order scheme. Both of these have been widely used for the time-discretization in applications of spectral methods. Both references contain a family of low-storage methods. Another low-storage family popular in the spectral methods community originated with A. Wray (unpublished), and was extended by Spalart et al. (1993).
Appendix E. Supplementary Material

E.1 Numerical Solution
of the Generalized Eigenvalue Problem

The generalized eigenvalue problems produced by linear stability methods (regardless of their spatial discretizations) typically have the form

\[ Aq = \omega Bq \]  \hspace{1cm} (E.1.1)

for temporal stability and

\[ A_0q + \alpha A_1q + \alpha^2 A_2q = 0 \]  \hspace{1cm} (E.1.2)

for spatial stability in formulations using second-order equations, such as (1.5.5)–(1.5.8) for incompressible flow and (1.5.19) for compressible flow. (When higher-order equations, such as the Orr–Sommerfeld equation (1.5.10) are used, then clearly (E.1.2) contains additional terms with higher powers of \( \alpha \).)

Consider first the temporal eigenvalue problem. In some applications, the complete spectrum of (E.1.1) is desired. The LAPACK package (Anderson et al. (1999)) contains numerous routines for finding all the eigenvalues and some or all of the eigenvectors of such generalized eigenvalue problems. One must be aware that the computed spectrum often contains spurious eigenvalues due entirely to the numerical discretization of the problem. (See Fig. 2.12 and the related discussion.) Frequently, these spurious eigenvalues have the largest absolute values. Care is needed in selecting the eigenvalue(s) of physical interest.

In other cases, one is interested in only a few, or perhaps just one, eigenvalue of (E.1.1). In many cases, a good guess for \( \omega \) is available. This occurs, for example, when computing a neutral curve—the locus \((\alpha, \text{Re})\) for which \( \text{Im}\{\omega}\) = 0. A simple approach to computing just a single eigenvalue is inverse Rayleigh iteration. Suppose that \( \omega_0 \) is an approximate value of \( \omega \), and that \( x^n \) and \( y^n \) are current approximations to the eigenvectors of (E.1.1)
and to the adjoint problem, respectively. These approximations are updated via
\[
(A - \omega_0 B)x^{n+1} = Bx^n, \tag{E.1.3}
\]
\[
(A - \omega_0 B)^*y^{n+1} = B^*y^n.
\]
The eigenvalue $\omega$ is then approximated by
\[
\omega \approx \omega^{n+1} = \frac{(y^{n+1})^H A x^{n+1}}{(y^{n+1})^H B x^{n+1}}, \tag{E.1.4}
\]
and, of course, $x^{n+1}$ and $y^{n+1}$ are improved approximations to the right and left eigenvectors corresponding to $\omega$. (The superscript $H$ denotes the Hermitian transpose.)

A more sophisticated approach, which is both more efficient and returns a set of eigenvalues, uses the Arnoldi algorithm (Saad (1980)), which is a Krylov subspace method. Equation (E.1.1) is rewritten as
\[
Cq = \mu q, \tag{E.1.5}
\]
with
\[
C = (A - \omega_0 B)^{-1}B, \quad \mu = \frac{1}{\omega - \omega_0}. \tag{E.1.6}
\]
The Arnoldi iteration starts with an initial guess $q_1$ of unit $L^2$-norm and then computes $m - 1$ additional vectors by
\[
\hat{q}_{j+1} = Cq_j - \sum_{i=1}^j h_{ij} q_i, \tag{E.1.7}
\]
\[
h_{j+1,j} = \|\hat{q}_{j+1}\|_2, \quad q_{j+1} = \hat{q}_{j+1}/h_{j+1,j},
\]
for $j = 1, \ldots, m - 1$, where $h_{ij} = (q_i)^H C q_j$ (and so $q_{j+1}$ is $C$-orthogonal to all earlier $q_j$ with respect to the symmetric part of $C$). Then the $QR$ algorithm (Wilkinson (1965)) applied to the Hessenberg matrix with elements $h_{ij}$ yields approximations to the eigenvalues of $C$ with largest absolute values. Note that the $LU$ decomposition of $(A - \omega_0 B)$ is only computed once. For Chebyshev collocation discretizations, transform methods may be used to reduce the cost of computing the $Bq_j$ contribution to the $Cq_j$ term in the iteration. See Nayar and Ortega (1993) for more details in the context of fluid dynamic stability problems. The ARPACK package (Lehoucq et al. (1998)) contains routines for various Arnoldi iterative methods that are applicable to finding some of the eigenvalues and eigenvectors of such generalized eigenvalue problems.

For spatial eigenvalue problems, the companion matrix approach (Bridges and Morris (1984)) is widely used. For the particular case of (E.1.2) the
E.2 Tau Correction for the Kleiser–Schumann Method

The tau terms \( \tilde{\tau}_{x,m} \) and \( \tilde{\tau}_{y,m} \) vanish for \( 0 \leq m \leq N - 2 \). The application of the discrete divergence to (E.2.1) and (E.2.3) yields

\[
\nu \tilde{d}_m^{(2)} - \lambda \tilde{d}_m - \tilde{p}_m^{(2)} + k^2 \tilde{p}_m = -\tilde{r}_m - (ik \tilde{\tau}_{x,m} + \tilde{\tau}_{y,m}^{(1)}), \quad m = 0, \ldots, N ,
\]
where

\[
\tilde{r}_m = ik \tilde{\tau}_{x,m} + \tilde{\tau}_{y,m}^{(1)}, \quad m = 0, \ldots, N .
\]

But, (E.2.5) is equivalent to

\[
\tilde{d}_m = 0 \quad m = 0, \ldots, N - 2, \quad \tilde{d}(\pm 1) = 0 .
\]
Hence, from (E.2.6) the discrete $A$-problem is
\[
\hat{p}^{(2)}_m - k^2 \hat{p}_m = \hat{r}_m + \tilde{\sigma}^{(1)}_m, \quad m = 0, \ldots, N - 2, \quad \hat{v}'(\pm 1) = 0
\]
\[
\nu \hat{v}^{(2)}_m - \lambda \hat{v}_m - \tilde{\sigma}^{(1)}_m = -\hat{r}_{y,m} - \tilde{\sigma}_m, \quad m = 0, \ldots, N, \quad \hat{v}(\pm 1) = 0,
\]
and the discrete $B$-problem is
\[
\hat{p}^{(2)}_m - k^2 \hat{p}_m = \tilde{r}_m + \tilde{\sigma}^{(1)}_m, \quad m = 0, \ldots, N - 2, \quad \dot{\hat{p}}(\pm 1) = \dot{\hat{p}}_\pm,
\]
\[
\nu \hat{v}^{(2)}_m - \lambda \hat{v}_m - \tilde{\sigma}^{(1)}_m = -\hat{r}_{y,m} - \tilde{\sigma}_m, \quad m = 0, \ldots, N, \quad \dot{\hat{v}}(\pm 1) = 0,
\]
where we use $\tilde{\sigma}_m = \tilde{r}_{y,m}$. If not for the “tau correction” embodied by the $\tilde{\sigma}_n$ and $\tilde{\sigma}^{(1)}_n$ terms, the influence-matrix solution procedure would be a straightforward application of the Helmholtz equation techniques discussed in CHQZ2, Sect. 4.1.2. Kleiser and Schumann (1980) describe how to solve the discrete $B$-problem. Define the $\overline{B}_1$-problem by
\[
\hat{p}^{(2)}_m - k^2 \hat{p}_m = \hat{r}_m, \quad m = 0, \ldots, N - 2, \quad \dot{\hat{p}}(\pm 1) = \dot{\hat{p}}_b \pm,
\]
\[
\nu \hat{v}^{(2)}_m - \lambda \hat{v}_m - \tilde{\sigma}^{(1)}_m = -\hat{r}_{y,m} - \tilde{\sigma}_m, \quad m = 0, \ldots, N - 2, \quad \dot{\hat{v}}(\pm 1) = 0,
\]
and the $\overline{B}_0$-problem by
\[
\hat{p}^{(2)}_m - k^2 \hat{p}_m = \frac{2}{m'} \hat{p}_m', \quad m = 0, \ldots, N - 2, \quad \dot{\hat{p}}(\pm 1) = 0,
\]
\[
\nu \hat{v}^{(2)}_m - \lambda \hat{v}_m = \tilde{\sigma}^{(1)}_m, \quad m = 0, \ldots, N - 2, \quad \dot{\hat{v}}(\pm 1) = 0,
\]
where
\[
m' = \begin{cases} N - 1, & m \text{ even}, \\ N, & m \text{ odd} \end{cases}
\]
(assuming $N$ is even). Furthermore, define $\tilde{\sigma}_m$ and $\tilde{\sigma}_0$ for $m = N - 1$, $N$ as the tau terms that must be added to the $v$-momentum equations in (E.2.11) and (E.2.12), respectively, for them to hold for $m = N - 1$, $N$. One can show that
\[
\tilde{\sigma}_m = \tilde{\sigma}_1 m / (1 - \tilde{\sigma}_0 m), \quad m = N - 1, N,
\]
and that
\[
\hat{p}_m = \hat{p}_1 m + \tilde{\sigma}_m' \hat{p}_0 m, \quad m = 0, \ldots, N.
\]
\[
\tilde{\sigma}_m = \tilde{\sigma}_1 m + \tilde{\sigma}_0 (m + 1)' \tilde{v}_0 m, \quad m = 0, \ldots, N.
\]
The solution to the original $B$-problem is achieved by

1. Solving (E.2.12) for $\tilde{p}_0$, $\tilde{v}_0$, and evaluating $\tilde{\sigma}_{0,m}$ for $m = N - 1$, $N$ from the $v$-momentum equation of (E.2.12).
2. Solving (E.2.11) for $\tilde{p}_1$, $\tilde{v}_1$, and evaluating $\tilde{\sigma}_{1,m}$ for $m = N - 1$, $N$ from the $v$-momentum equation of (E.2.11).
3. Determining $\tilde{\sigma}_m$ from (E.2.14) and $\tilde{\sigma}_{1,m}^{(1)}$ from the standard recurrence relation (8.3.23).
4. Determining $\tilde{\sigma}$ from (E.2.15).

Step (1) is redundant for the second $B$-problem in the influence-matrix calculation. Hence, for each wavenumber $k$ the tau solution to the $A$-problem can be found at the cost of five complex Helmholtz equation solutions. (This can be reduced to four if one wishes to store $\tilde{p}_0$ and $\tilde{v}_0$.) To this cost must be added the cost of solving for $\tilde{u}$ from either (E.2.1) and (E.2.2) or (E.2.5). The cost is negligible in the latter case.

If the tau correction is simply ignored, then the computed solution will not satisfy all of (E.2.1)–(E.2.5). If (E.2.1)–(E.2.2) are used to determine $\tilde{u}$, then the solution will have a nonzero divergence in the interior. If (E.2.5) is used instead, then the momentum equation will not be satisfied, and the numerical experience is that catastrophic numerical instability occurs (Kleiser (1986)).

### E.3 The Piola Transform

Let $\hat{\Omega}$ and $\Omega$ be two domains in $\mathbb{R}^d$, $d \geq 2$, with coordinates $\hat{x} \in \hat{\Omega}$ and $x \in \Omega$, respectively. Let $F : \hat{\Omega} \to \Omega$ be a differentiable and invertible mapping with differentiable inverse $F^{-1} : \Omega \to \hat{\Omega}$. Let $DF(\hat{x})$ denote the Jacobian matrix of $F$ at the point $x \in \Omega$, and let $J(\hat{x}) = |DF(\hat{x})| \neq 0$ denote its determinant.

Given a vector field $\hat{\mathbf{v}}$ in $\hat{\Omega}$, we define the vector field $\mathbf{v}$ in $\Omega$ by

$$\mathbf{v} = J^{-1}DF\hat{\mathbf{v}},$$

(E.3.1)

i.e.,

$$\mathbf{v}(x) = J^{-1}(\hat{x})DF(\hat{x})\hat{\mathbf{v}}(\hat{x}), \quad \text{if} \quad \hat{x} = F^{-1}(x).$$

Equivalently, we can express $\hat{\mathbf{v}}$ in terms of $\mathbf{v}$ as

$$\hat{\mathbf{v}} = JDF^{-1}\mathbf{v}.$$  

(E.3.2)

The mappings $\hat{\mathbf{v}} \to \mathbf{v}$ and $\mathbf{v} \to \hat{\mathbf{v}}$ are the direct and inverse Piola transforms associated with $F$. They have the remarkable property that the divergences of the (supposed smooth) vector fields $\hat{\mathbf{v}}$ and $\mathbf{v}$ are related by the formula

$$\nabla_x \cdot \mathbf{v} = J^{-1}\nabla_{\hat{x}} \cdot \hat{\mathbf{v}},$$

(E.3.3)
which, in particular, proves that $\hat{v}$ is divergence-free if and only if $v$ is. An equivalent integral formulation of (E.3.3) is

$$\int_{\Omega} \nabla_x \cdot v \varphi \, dx = \int_{\hat{\Omega}} \nabla_{\hat{x}} \cdot \hat{v} \hat{\varphi} \, d\hat{x} \quad \text{if} \quad \hat{\varphi} = \varphi \circ F ,$$  

(E.3.4)

which holds for all integrable scalar fields $\varphi$ in $\Omega$. This implies the conservation property

$$\int_{\omega} \nabla_x \cdot v \, dx = \int_{F^{-1}(\omega)} \nabla_{\hat{x}} \cdot \hat{v} \, d\hat{x}$$  

(E.3.5)

for all open domains $\omega$ contained in $\Omega$, as can be seen by taking as $\varphi$ in (E.3.4) the characteristic function of $\omega$, i.e., the function which has the value 1 if $x \in \omega$ and 0 otherwise.

Identities (E.3.3) and (E.3.4) can be proven as follows. Assume that $\varphi$ is continuously differentiable in $\Omega$ and vanishing on $\partial \Omega$. Then, by the divergence theorem and the chain rule applied to $\nabla_x \varphi$, one has

$$\int_{\Omega} \nabla_x \cdot v \varphi \, dx = -\int_{\Omega} v \cdot \nabla_x \varphi \, dx = -\int_{\hat{\Omega}} J^{-1} \hat{v}^T (DF)^T (DF^{-1})^T \nabla_{\hat{x}} \hat{\varphi} J \, d\hat{x}$$

$$= -\int_{\hat{\Omega}} \hat{v} \cdot \nabla_{\hat{x}} \hat{\varphi} \, d\hat{x} = \int_{\hat{\Omega}} \nabla_{\hat{x}} \cdot \hat{v} \, d\hat{x} = \int_{\Omega} (\nabla_x \cdot \hat{v}) \varphi J^{-1} \, dx .$$

Formula (E.3.3) follows directly since $\varphi$ can be chosen to be nonzero at any interior point. The validity of (E.3.4) for any integrable $\varphi$ (not necessarily vanishing at the boundary) can be established by a density argument.

In order to implement the inverse Piola transform (E.3.2), one needs the inverse matrix $DF^{-1}$. Let us detail its computation in 3D. Denote by

$$a_j = \frac{\partial x}{\partial \hat{x}_j} , \quad j = 1, 2, 3 ,$$  

(E.3.6)

the column vectors of the matrix $DF$. Then, it is easily seen that the rows $a^i = \nabla_x \hat{x}_i, \ i = 1, 2, 3$, of the matrix $DF^{-1}$ satisfy the relations

$$a^i = J^{-1} (a_j \times a_k)^T , \quad (i, j, k) \text{ cyclic} ,$$  

(E.3.7)

where the Jacobian determinant itself can be computed in the form

$$J = a_i \cdot (a_j \times a_k) , \quad (i, j, k) \text{ cyclic} .$$  

(E.3.8)

The entries of $a^i$ can be equivalently written as

$$J (a^i)_n = J (DF^{-1})_{i,n} = -\hat{e}_i \cdot \nabla_{\hat{x}} \times (x_l \nabla_{\hat{x}} x_m) , \quad (n, m, l) \text{ cyclic} ,$$  

(E.3.9)

where $\hat{e}_i$ are the vectors of the canonical basis. Note that the inverse Jacobian only appears in (E.3.2) in the combination $J (DF^{-1})$. 

In many situations, only the values of $F$ at a set of interpolation nodes in $\hat{\Omega}$ (such as the LGL or the LG nodes) are used to implement the transformation numerically. Then, as shown in detail by Kopriva (2006), the discrete version of (E.3.9), rather than of (E.3.7), should be used in order to guarantee the conservation property at the discrete level. Precisely, denoting by

$$D_{\hat{x}_j} = \frac{\partial}{\partial \hat{x}_j} \hat{I}_N$$

the interpolation partial derivative with respect to $\hat{x}_j$ (where $\hat{I}_N$ is the interpolation operator at the selected set of nodes), and by $\hat{\nabla}_x = \nabla_x \hat{I}_N$ the interpolation gradient operator, one replaces (E.3.6) and (E.3.9) by

$$\tilde{a}_j = D_{\hat{x}_j} x, \quad j = 1, 2, 3,$$  \hspace{1cm} (E.3.10)

and

$$(\tilde{J} \tilde{a})^i_n = (\tilde{J} \tilde{D} \tilde{F}^{-1})^i_{t,n} = -\hat{e}_i \cdot \hat{\nabla}_x \times (x_l \hat{\nabla}_x x_m), \quad (n, m, l) \text{ cyclic },$$  \hspace{1cm} (E.3.11)

respectively. In this way, if

$$\eta_t + \nabla_x \cdot \mathcal{F} = 0$$

is a conservation law in $\Omega$, and if one sets $\tilde{\eta} = \tilde{J} \eta$ and $\tilde{\mathcal{F}} = \tilde{J} \tilde{D} \tilde{F}^{-1} \mathcal{F}$, one gets the conservation law in $\hat{\Omega}$:

$$\tilde{\eta}_t + \nabla_{\hat{x}} \cdot \tilde{\mathcal{F}} = 0 .$$

Finally, in the context of enforcing the discrete conservation laws in integral form, as used for example in discontinuous Galerkin spectral methods, there are additional subtleties with exactly how the integral form of the flux term is written. There are some forms of the integral form of the flux term that are not integrated exactly with the Gauss–Lobatto formula, whereas they are integrated exactly with the Gauss formula. This makes the Gauss formula more generally recommendable for ensuring conservation in the discrete equations. See Kopriva (2006) for the details.
References


European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics, ed. by R. Decuyper and G. Dibelius, Technologisch Institut, Antwerpen, Belgium, pp. 99–108


W. Blake (1975): ‘A Statistical Description of Pressure and Velocity Fields at the Trailing Edge of a Flat Strut.’ David Taylor Naval Ship Research and Development Center Report 4241


J. le R. d’Alembert (1752): Essai d’une nouvelle theorie de la resistance des fluides (Opuscules Mathematiques)


T.L. Holst, W.F. Ballhaus (1979): Fast, conservative schemes for the full potential equation applied to transonic flows. AIAA J. 17(2), 145–152


L. Kleiser (1986): private communication


References

L.M. Mack (1984): ‘Boundary-Layer Linear Stability Theory’. In Special Course on Stability and Transition to Turbulence, AGARD-R-709


References


J.S. Przemieniecki (1985): Theory of Matrix Structural Analysis (Dover, New York)


H. Rouse, S. Ince (1963): History of Hydraulics (Dover, New York)
tional Conference on Numerical Methods in Fluid Dynamics, ed. by Soubbarameyer, J. Boujot (Springer-Verlag, Heidelberg)


W. Sutherland (1893): The viscosity of gases and molecular force. Philosophical Magazine 5(36), 507–531
G.I. Taylor, J.W. Maccoll (1933): The air pressure on a cone moving at high speeds. Proc. R. Soc. A 139, 278


B. Wohlmuth (2001): *Discretization Techniques and Iterative Solvers Based on Domain Decomposition*, Lectures Notes in Computational Science and Engineering **17** (Springer-Verlag, Heidelberg)


Index

$L^p$ Spaces, 493

adaptive refinement, 263, 267
- $N$-refinement, 267, 310
- $h$-refinement, 267, 310
adjoint operator, 306, 319
advection equation
- G-NI method, 301
- Galerkin method, 301
- generalized collocation method, 303, 318
- in multiple dimensions, 317
- penalty method, 303
- SDGM, see SDGM
- SEM-NI, 275
- skew-symmetric form, 264, 318
advection-diffusion equation
- in multiple dimensions, 317
- penalty method, 303
- SDGM, see SDGM
- SEM-NI, 275
- skew-symmetric form, 264, 318
advection-diffusion-reaction equation
- SEM formulation, 239, 250
- SEM-NI, 273
algebraic factorization methods, 133, 352, 415, 425
- Chorin–Temam, 426
- pressure correction, 427
- splitting error, 426
- Yosida, 426–428
algebraic fractional-step methods, see algebraic factorization methods
algebraic splitting methods, see algebraic factorization methods
aliasing error, 440
- channel flow turbulence, 121
- compressible turbulence, 213
- for Chebyshev expansions, 454
- for Fourier expansions, 441
- for momentum equation forms, 120
- isotropic turbulence, 120
Arnoldi orthonormalization algorithm, 520

basis
- boundary adapted, 241, 458
- modal, 458
- multidimensional
  - boundary-adapted, 463
  - Fourier, 463
  - Fourier-Chebyshev, 463
  - non-tensor-product, 468
  - tensor-product, 463
- nodal, 458
basis functions
- bubble, see function
- edge, 464, 467
- face, 464
- internal, 458
- multidimensional bubble, 464
- vertex, 458, 464, 467
- warped tensor-product, 465
bilinear form, 492
- coercive, 492
boundary-layer equations
- Blasius equation, 42
- collocation method
  - compressible, 48
  - incompressible, 42
  - compressible, 19
- Falkner-Skan equation, 42
- Görtler variables, 43
- incompressible, 27
boundary-layer flow
- asymptotic boundary conditions, 141
- compressible algorithms, 221, 223
- explicit time-step limitations, 222
- incompressible algorithms, 140
boundary-layer thickness, 20
broken norm, 297

Cauchy sequence, 489
CCM, 189, 347, 351
bi-characteristics, 195, 347
collocation methods, 193
dissipativity condition, 198
Euler equations, 205
in spectral shock fitting, 230
multidimensional linear system, 195
one-D linear system, 192
reflection matrices, 193
weak enforcement, 197
channel flow, 29, 121
  compressible algorithm, 223
coupled algorithms, 123, 124, 127
divergence-free bases algorithms, 131
  numerical example, 144
  splitting algorithms, 133, 135
  staggered grid, 133
characteristic compatibility method,
  see CCM
characteristic Lagrange polynomial
  barycentric form, 450
Chebyshev, 454
  Fourier, 439
  Legendre, 450
characteristic variables, 304
Chebyshev
  characteristic Lagrange polynomial, 454
collocation points, see nodes
derivative recursion relation, 454
interpolation derivative matrix, 455
nodes
  Gauss, 453
  Gauss–Lobatto, 453
  Gauss–Radau, 453
  normalization factors, 453
  polynomial recursion relation, 452
  polynomials, 451, 452
  quadrature rule
  Gauss, 453
  Gauss–Lobatto, 453
  Gauss–Radau, 453
series
  continuous, 453
  transform matrix, 453
  weight
  Gauss, 453
  Gauss–Lobatto, 453
  Gauss–Radau, 453
collapsed coordinates, 465
collocation method, 481
compact differences
  boundary stencil, 190
  fourth-order method
  first derivative, 190
  sixth-order method
  second derivative, 111
  condition number, 252, 510
  iterative, 510
  spectral, 252, 510
conservation laws
  scalar, 313
  systems in 1D, 315
  systems, in multiple dimensions, 319
consistent Poisson operator, 430
convergence
  exponential, 48, 437, 438
  infinite-order, 48, 437
Darcy equations, 93, 409
de-aliasing, 99
  2/3-rule, 102
  3/2-rule, 100
direct numerical simulation (DNS), 13
discontinuous Galerkin methods, see SDGM
discrete delta-function
  see characteristic Lagrange polynomial, 435
displacement thickness, 20
dissipation function, 8
distribution, 494
  derivative, 494
  periodic, 496, 500
  derivative, 496
domain decomposition methods, 237, 363
  analysis, 383
domain decomposition preconditioners, 361
  for stiffness matrix, 398
  optimal, 361
  scalable, 361
  Schur
  balanced Neumann–Neumann, 433
  FETI, 433
  wire basket, 433
Schwarz, 433
duality pairing, 491

eigenfunctions
  – Laplace operator
    – – SEM-NI, 275
energy
  – internal, 3
  – total, 4
enthalpy, 3, 4
  – total, 4
entropy, 4
equation of state
  – calorically perfect gas, 6
  – general, 4
  – polytropic, 6
  – thermally perfect gas, 5
error estimator
  – efficient, 262
  – reliable, 262
error indicator
  – global, 262
  – local, 262
Euler equations, 17
  – artificial boundary conditions, 207
  – collocation methods, 345
    – in 1D, 332
    – conservation form, 204
    – eigenvalues, 204
    – eigenvectors, 205
    – nonconservative form, 204
    – in multiple dimensions, 319
  – linearized, locally 1D CCM, 205
  – multidomain collocation method
    – on nonstaggered grid, 346
    – on staggered grid, 348
  – multidomain shock fitting, 350
  – SDGM, see SDGM
  – shock capturing, 226, 233, 337
  – shock fitting, 226, 337
  – shock tracking, 226, 337
  – single-domain collocation methods, 218
extrapolation formulas, 413

Fast Fourier Transform
  – for discrete Chebyshev series
    – at Gauss points, 507
    – at Gauss–Lobatto points, 454, 505
    – for odd polynomials, 507
  – for discrete Fourier series, 439
    – for complex data, 503
    – for real data, 504
  – – operation count, 503
Favre average, 15
Fekete points, 378, 433
FEM–G-NI equivalence, 254
FEM–SEM-NI equivalence, 381
FEM-SEM equivalence, 254, 380, 394
field of values, 521
filter, 209, 444
  – Cesáro, 444
  – derivative filtering, 210, 233
  – exponential, 209, 445
  – Lanczos, 444
  – raised cosine, eighth-order, 444
  – raised cosine, second-order, 444
  – solution filtering, 208, 217, 233, 234
  – Vandeven, 209, 445
finite differences
  – fourth-order, 111
  – second-order, 111
finite-elements
  – hp-version, 237, 240
  – p-version, 240
  – preconditioning, 253
fluid
  – classical, 8
  – Newtonian, 8
  – Stokesian, 8
formulation
  – integral, 478
  – strong, 478
  – weak, 478
Fourier
  – characteristic Lagrange polynomial, 439
  – collocation points, see nodes
  – cosine transform, 435
  – expansion coefficients
    – continuous, 435
    – discrete, 438
  – inner product
    – discrete, 439
  – interpolating polynomial, 439
  – interpolation derivative, 442
  – interpolation derivative matrix, 442
  – norm
    – discrete, 439
  – projection derivative, 441
  – series
    – continuous, 436
    – discrete, 439
  – sine transform, 435
  – transform
    – continuous, 435
discrete, 439
fractional-step methods, 95, 409
  – Chorin–Temam, 95, 409
  – consistent boundary condition, 414
  – incompressible channel flow, 133
  – incremental pressure correction, 412
  – KIO scheme, 414, 430
  – pressure correction, 410, 414
  – rotational form of boundary conditions, 412
  – splitting error, 427
free-shear-layer flow
  – compressible algorithms, 223, 224
  – incompressible algorithms, 142
Friedrichs system, 303
fringe method, 161
function
  – bubble, 458, 467
  – Dirac, 495
Galerkin method, 482
Galerkin with numerical integration (G-NI) method, 484
gas
  – calorically perfect, 5
  – ideal, 5
  – thermally perfect, 5
generalized collocation method, 244, 251
generalized Galerkin method, 260
Gibbs phenomenon, 443
GLS method, 281
harmonic extension, 386, 387
heat flux, 9
Helmholtz
  – decomposition principle, 409, 413
  – equations, 407
  – operator, 416
Hermite
  – functions, 462
  – polynomial recursion relation, 462
  – polynomials, 461
  – Rodriguez formula, 462
homogeneous turbulence
  – compressible, 211
  – eddy shocklets, 214
  – fluctuating Mach number, 214
  – incompressible
    – re-gridding, 106
    – Rogallo transformation, 106
    – hyperbolic systems
      – boundary conditions, see CCM
      – G-NI stability analysis, 200
      – SDGM, see SDGM
      – spectral accuracy, 199
inequality
  – Cauchy-Schwarz, 491
  – strengthened, 384
  – Poincaré, 501
  – Sobolev, 501
inexact block factorization, 425
inf-sup condition, 168, 169, 178, 179, 280, 283, 285, 419, 421
inf-sup constant, 170, 173–176, 181
influence matrix, 94, 126, 127, 130, 139–141, 153, 183, 389
inner product, 489
  – periodic Sobolev space, 500
  – Sobolev space, 496, 497
interior penalty method, 281, 331
interpolant
  – polynomial, 449
  – trigonometric, 439
interpolation divergence operator, 89
interpolation gradient operator, 89
isotropic turbulence
  – compressible, 214
  – incompressible
    – de-aliasing, 99
    – Fourier collocation method, 103
    – Fourier Galerkin method, 98
    – Fourier pseudospectral method, 103
    – large-eddy simulation, 108
    – Taylor-Green vortex numerical example, 110
iterative algorithms
  – bi-conjugate gradient (Bi-CG), 522
  – bi-conjugate gradient stabilized (Bi-CGStab), 522
  – preconditioned (PBi-CGStab), 522
    – conjugate direction, 514
    – conjugate gradient (CG), 515
      – preconditioned (PCG), 517
      – conjugate gradient-squared (CGS), 522
    – conjugate residual (CR), 516
      – preconditioned (PCR), 517
      – generalized minimum residual (GMRES), 519
    – flexible, 522
    – full, 520
    – preconditioned (PGMRES), 521
with restart (GMRES(m)), 520
- Krylov
- full orthogonalization method, 519
- minimum residual Richardson (MRR)
- preconditioned (PMRR), 513
- Newton–Krylov–Schwarz, 333
- preconditioned steepest descent Richardson (PSDR), 513
- Richardson
- non-stationary (NSR), 511
- preconditioned, 512
- stationary, 511
- Richardson method, 509
- truncated conjugate gradient (TCG), 517
- preconditioned (PTCG), 517
- truncated conjugate residual (TCR), 517
- preconditioned (PTCR), 517
iterative methods
- iteration matrix, 509, 512
- optimal relaxation parameter, 510
- spectral radius, 509
iterative substructuring, 362
iterative substructuring methods, 402
- Dirichlet–Neumann, 402, 404
- Neumann–Neumann, 403, 405
- Robin, 404
Jacobi
- polynomial recursion relation, 445
- polynomials, 445
- Rodriguez formula, 445
- weight, 445
Kolmogorov length scale, 103
Laguerre
- functions, 461
- polynomial recursion relation, 461
- polynomials, 460
- Rodriguez formula, 461
large-eddy simulation (LES)
- compressible flow equations, 15
- compressible isotropic turbulence, 217
- cross stress, 23
- dynamic model, 24
- incompressible flow equations, 22
- Leonard stress, 23
- nonlinear Galerkin method, 25
- residual stress, 15
- Reynolds stress, 23
- Smagorinsky model, 23, 109
- subgrid-scale stress, 15
- variational multiscale method, 25, 109
Legendre
- characteristic Lagrange polynomial, 450
- collocation points, see nodes
- derivative recursion relation, 457
- interpolation derivative, 451
- interpolation derivative matrix, 458
- nodes
  - Gauss, see LG
  - Gauss–Lobatto, see LGL
  - Gauss–Radau, 457
- normalization factors, 457
- polynomial recursion relation, 456
- polynomials, 455
- projection derivative, 451
- quadrature rule
  - Gauss, 457
  - Gauss–Lobatto, 457
  - Gauss–Radau, 457
- series
  - continuous, 457
  - weight
    - Gauss, 457
    - Gauss–Lobatto, 457
    - Gauss–Radau, 457
LG, 457
LGL, 241, 457
- interpolant
  - polynomial, 244
linear stability
- dispersion relation, 30
- spatial, 30
- supersonic modes, 67
- temporal, 30
linear stability equations
- compressible, 34
  - nonstaggered-grid collocation method, 65
  - staggered-grid collocation method, 64
- global stability analysis, 71
- incompressible, 30
  - collocation method, 54
  - divergence-free Galerkin method, 62
  - G-NI method, 55
  - Galerkin method, 54
  - integral tau method, 62
  - spatial stability methods, 64
spurious eigenvalues, 59
– tau method, 52
– Orr–Sommerfeld equation, 31
– transient growth, 72
– vertical vorticity equation, 31
liquid, 2

Mach number, 12, 17
mapping
– affine, 469
– Fourier clustering, 470
– Gordon–Hall, 245, 475
– isoparametric, 477
– Kosloff-Tal-Ezer, 470
– polynomial clustering, 470
– real line, 473
– semi-infinite intervals, 471
– uniformly regular, 272
master element, see reference domain
matrix
– negative part, 304
– positive part, 304
– skew-symmetric part, 419
– symmetric part, 419
MEM, 238, 289
– algebraic aspects, 294
– error analysis, 296
– formulation, 290
– matching conditions, 291, 294
– Navier–Stokes equations inf-sup constant, 300
– saddle-point formulation, 295
– Schur complement system, 296
MEM-NI, 293
monotone scheme, 313
mortars, 292
– in SDGM, 321
– in staggered-grid collocation method, 348
– mortar conditions, 292
– mortar element method, see MEM
– mortar function, 292
multigrid method, 433
– for SEM, 432

Navier–Stokes
– methods, 84
Navier–Stokes equations
– compressible, 9
– continuity, 10
– entropy, 10
– incompressible, 22, 83
– buffer domain outflow boundary conditions, 161
– conservation properties, 86
– convection form, 86, 111
– divergence form, 86, 111
– fringe-region method, 159
– outflow boundary conditions, 159
– rotation form, 86, 111
– skew-symmetric form, 87, 111
– inflow & outflow boundary conditions, 11
– internal energy, 10
– momentum conservation, 10
– nondimensional, 12
– pressure, 10
– rotational form, 412
– SDGM, see SDGM
– SEM, 278
– analysis, 285
– SEM approximation, 282, 352
– SEM-NI, 282
– analysis, 285
– flow around a cylinder, 287
– lid driven cavity, 286
– streamfunction-vorticity equations, 26
– temperature, 10
– time-discretization, 407
– total energy conservation, 10
– vorticity-velocity equations, 27

Neumann expansion, 426
Neumann-to-Dirichlet map, 394
nonlinear stability equations
– finite-amplitude states
– spectral methods, 76
– secondary instability theory (SIT)
– equations, 77
– spectral methods, 78
norm, 489, 490
– of a dual space, 491
– periodic Sobolev space, 500
– Sobolev, 496, 497
numerical flux, 306, 313, 319, 320
– Bassi and Rebay, 336
– Engquist–Osher, 314
– Godunov, 314
– Lax–Friedrichs, 314, 316, 336
– Lax-Friedrichs, 332
– Roe, 314, 316, 332
– Rusanov, 314
– upwind, 315

orthogonal polynomials
– expansion coefficients
– continuous, 447
Index 593

-- discrete, 449
-- inner product
-- -- continuous, 447
-- -- discrete, 449
-- interpolating polynomial, 449
-- norm
-- -- continuous, 447
-- -- discrete, 449
-- transform
-- -- continuous, 448
-- -- discrete, 449
-- truncated series, 448
orthogonal projection
-- Fourier
-- -- continuous, 437
-- orthogonal polynomials
-- -- continuous, 448
-- -- discrete, 449
overlap
-- generous, 364, 378
-- minimum, 364, 376
-- small, 364

parabolized stability equations (PSE), 36
-- collocation methods, 69, 81
parallel flow, 28
-- approximation, 33
parent element, 245
partition
-- geometrically conforming, 238, 246, 320
-- geometrically nonconforming, 238, 321
-- hybrid, 279
-- shape regular, 272, 285
partition of unity, 273
patching collocation method, 341, 402
-- for advection equation, 342
-- for compressible flows, 343
-- for Poisson equation, 341
-- Schur complement
-- -- condition number, 393
-- shock fitting, 350
phase diagram, 2
physical space, 449
Piola transform, 280, 285, 541
pipe flow algorithms, 131, 139, 140
Poiseuille flow, see channel flow
Poisson equation
-- for pressure, 407, 414
Poisson problem
-- multidomain formulation, 340
-- transmission conditions, 340
polynomials
-- Chebyshev, 452
-- Gegenbauer, 446
-- Hermite, 461
-- Jacobi, 445
-- Laguerre, 460
-- Legendre, 455
-- ultraspherical, 446
potential equation
-- compressible, 18
Prandtl number, 9, 12
preconditioner
-- LU-symmetric Gauss–Seidel, 334
pressure, 3
pressure boundary layer, 411
pressure mass matrix, 420, 421
pressure mode
-- pseudo-spurious, 151, 154, 157, 158
-- spurious, 134, 150, 152, 154, 157, 166, 167
pressure operator, 179
pressure Schur complement matrix, 417, 419
-- approximate, 427
-- -- finite-element preconditioner, 431
-- -- preconditioners, 430
-- -- Schwarz preconditioner, 431
-- multigrid solver
-- -- Schwarz preconditioner, 432
-- preconditioners
-- -- Cahouet and Chabard, 421
-- -- Elman et al., 421
-- -- Fischer and Rønquist, 421
-- -- pressure mass matrix, 420
-- -- Schwarz, 425
pressure Schur complement system, 419, 422
pseudospectra, 74
quadrature rule
-- tensorized, 463
Rayleigh–Bénard flow algorithms, 139
reconstruction algorithms, 233, 235
reference domain, 237, 266
residual
-- element, 261, 265
-- interface, 266
Reynolds number, 12
-- based on wall shear velocity, 127
Riemann problem, 311, 313, 316, 318, 320, 321, 323

parabolized stability equations (PSE), 36
parallel flow, 28
partition
-- geometrically conforming, 238, 246, 320
-- geometrically nonconforming, 238, 321
-- hybrid, 279
-- shape regular, 272, 285
partition of unity, 273
patching collocation method, 341, 402
-- for advection equation, 342
-- for compressible flows, 343
-- for Poisson equation, 341
-- Schur complement
-- -- condition number, 393
-- shock fitting, 350
phase diagram, 2
physical space, 449
Piola transform, 280, 285, 541
pipe flow algorithms, 131, 139, 140
Poiseuille flow, see channel flow
Poisson equation
-- for pressure, 407, 414
Poisson problem
-- multidomain formulation, 340
-- transmission conditions, 340
polynomials
-- Chebyshev, 452
-- Gegenbauer, 446
-- Hermite, 461
-- Jacobi, 445
-- Laguerre, 460
-- Legendre, 455
-- ultraspherical, 446
potential equation
-- compressible, 18
Prandtl number, 9, 12
preconditioner
-- LU-symmetric Gauss–Seidel, 334
pressure, 3
pressure boundary layer, 411
pressure mass matrix, 420, 421
pressure mode
-- pseudo-spurious, 151, 154, 157, 158
-- spurious, 134, 150, 152, 154, 157, 166, 167
pressure operator, 179
pressure Schur complement matrix, 417, 419
-- approximate, 427
-- -- finite-element preconditioner, 431
-- -- preconditioners, 430
-- -- Schwarz preconditioner, 431
-- multigrid solver
-- -- Schwarz preconditioner, 432
-- preconditioners
-- -- Cahouet and Chabard, 421
-- -- Elman et al., 421
-- -- Fischer and Rønquist, 421
-- -- pressure mass matrix, 420
-- -- Schwarz, 425
pressure Schur complement system, 419, 422
pseudospectra, 74
quadrature rule
-- tensorized, 463
Rayleigh–Bénard flow algorithms, 139
reconstruction algorithms, 233, 235
reference domain, 237, 266
residual
-- element, 261, 265
-- interface, 266
Reynolds number, 12
-- based on wall shear velocity, 127
Riemann problem, 311, 313, 316, 318, 320, 321, 323

parabolized stability equations (PSE), 36
parallel flow, 28
partition
-- geometrically conforming, 238, 246, 320
-- geometrically nonconforming, 238, 321
-- hybrid, 279
-- shape regular, 272, 285
partition of unity, 273
patching collocation method, 341, 402
-- for advection equation, 342
-- for compressible flows, 343
-- for Poisson equation, 341
-- Schur complement
-- -- condition number, 393
-- shock fitting, 350
phase diagram, 2
physical space, 449
Piola transform, 280, 285, 541
pipe flow algorithms, 131, 139, 140
Poiseuille flow, see channel flow
Poisson equation
-- for pressure, 407, 414
Poisson problem
-- multidomain formulation, 340
-- transmission conditions, 340
polynomials
-- Chebyshev, 452
-- Gegenbauer, 446
-- Hermite, 461
-- Jacobi, 445
-- Laguerre, 460
-- Legendre, 455
-- ultraspherical, 446
potential equation
-- compressible, 18
Prandtl number, 9, 12
preconditioner
-- LU-symmetric Gauss–Seidel, 334
pressure, 3
pressure boundary layer, 411
pressure mass matrix, 420, 421
pressure mode
-- pseudo-spurious, 151, 154, 157, 158
-- spurious, 134, 150, 152, 154, 157, 166, 167
pressure operator, 179
pressure Schur complement matrix, 417, 419
-- approximate, 427
-- -- finite-element preconditioner, 431
-- -- preconditioners, 430
-- -- Schwarz preconditioner, 431
-- multigrid solver
-- -- Schwarz preconditioner, 432
-- preconditioners
-- -- Cahouet and Chabard, 421
-- -- Elman et al., 421
-- -- Fischer and Rønquist, 421
-- -- pressure mass matrix, 420
-- -- Schwarz, 425
pressure Schur complement system, 419, 422
pseudospectra, 74
quadrature rule
-- tensorized, 463
Rayleigh–Bénard flow algorithms, 139
reconstruction algorithms, 233, 235
reference domain, 237, 266
residual
-- element, 261, 265
-- interface, 266
Reynolds number, 12
-- based on wall shear velocity, 127
Riemann problem, 311, 313, 316, 318, 320, 321, 323
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>saddle-point problem</td>
<td>163, 169, 279, 282, 415, 433</td>
</tr>
<tr>
<td>Schur complement</td>
<td></td>
</tr>
<tr>
<td>- iterative methods</td>
<td>385</td>
</tr>
<tr>
<td>- matrix</td>
<td>385, 387, 389, 399</td>
</tr>
<tr>
<td>- condition number</td>
<td>393</td>
</tr>
<tr>
<td>- many subdomains</td>
<td>390, 392</td>
</tr>
<tr>
<td>- two subdomains</td>
<td>387</td>
</tr>
<tr>
<td>- system</td>
<td>385, 389</td>
</tr>
<tr>
<td>Schur complement preconditioners</td>
<td>393, 401</td>
</tr>
<tr>
<td>- balancing Neumann–Neumann</td>
<td>396</td>
</tr>
<tr>
<td>- condition number</td>
<td>397</td>
</tr>
<tr>
<td>- Dirichlet–Neumann</td>
<td>394, 401</td>
</tr>
<tr>
<td>- FETI</td>
<td>397</td>
</tr>
<tr>
<td>- FETI-DP</td>
<td>398</td>
</tr>
<tr>
<td>- Neumann–Neumann</td>
<td>395, 396</td>
</tr>
<tr>
<td>- condition number</td>
<td>396</td>
</tr>
<tr>
<td>- wire-basket</td>
<td>398</td>
</tr>
<tr>
<td>Schur preconditioners</td>
<td>361</td>
</tr>
<tr>
<td>Schwarz alternating method</td>
<td>364</td>
</tr>
<tr>
<td>- additive</td>
<td>365</td>
</tr>
<tr>
<td>- Richardson equivalence</td>
<td>367</td>
</tr>
<tr>
<td>- analysis</td>
<td>381</td>
</tr>
<tr>
<td>- for finite elements</td>
<td>367</td>
</tr>
<tr>
<td>- multiplicative</td>
<td>365</td>
</tr>
<tr>
<td>- convergence</td>
<td>366</td>
</tr>
<tr>
<td>- Richardson equivalence</td>
<td>367</td>
</tr>
<tr>
<td>Schwarz operator</td>
<td>384</td>
</tr>
<tr>
<td>- condition number</td>
<td>384</td>
</tr>
<tr>
<td>Schwarz preconditioners</td>
<td>361, 370, 432</td>
</tr>
<tr>
<td>- analysis</td>
<td>383</td>
</tr>
<tr>
<td>- coarse problem</td>
<td>371</td>
</tr>
<tr>
<td>- condition number</td>
<td>372</td>
</tr>
<tr>
<td>- for high-order methods</td>
<td>373</td>
</tr>
<tr>
<td>- coarse problem</td>
<td>374, 376</td>
</tr>
<tr>
<td>- condition number</td>
<td>374, 378</td>
</tr>
<tr>
<td>SDGM, 238, 300, 352</td>
<td></td>
</tr>
<tr>
<td>- acoustics</td>
<td>353</td>
</tr>
<tr>
<td>- advection equation</td>
<td>301</td>
</tr>
<tr>
<td>- in 1D, time-dependent</td>
<td>308</td>
</tr>
<tr>
<td>- in multiple dimensions</td>
<td>317</td>
</tr>
<tr>
<td>- analysis</td>
<td>326</td>
</tr>
<tr>
<td>- convergence</td>
<td>328</td>
</tr>
<tr>
<td>- dispersion error</td>
<td>329</td>
</tr>
<tr>
<td>- dissipation error</td>
<td>330</td>
</tr>
<tr>
<td>- stability</td>
<td>327</td>
</tr>
<tr>
<td>- conservation laws</td>
<td></td>
</tr>
<tr>
<td>- scalar</td>
<td>313</td>
</tr>
<tr>
<td>- systems in 1D</td>
<td>315</td>
</tr>
<tr>
<td>- diffusion equation</td>
<td>323</td>
</tr>
<tr>
<td>- dispersion properties</td>
<td>312</td>
</tr>
<tr>
<td>- dissipation properties</td>
<td>312</td>
</tr>
<tr>
<td>- electromagnetics</td>
<td>322</td>
</tr>
<tr>
<td>- Euler equations</td>
<td>332, 334</td>
</tr>
<tr>
<td>- in multiple dimensions</td>
<td>320</td>
</tr>
<tr>
<td>- hyperbolic systems</td>
<td></td>
</tr>
<tr>
<td>- in 1D</td>
<td>303</td>
</tr>
<tr>
<td>- in 1D, time-dependent</td>
<td>310</td>
</tr>
<tr>
<td>- in multiple dimensions, time-dependent</td>
<td>319</td>
</tr>
<tr>
<td>- mortars</td>
<td>321</td>
</tr>
<tr>
<td>- Navier–Stokes equations</td>
<td>335</td>
</tr>
<tr>
<td>- numerical flux</td>
<td>306</td>
</tr>
<tr>
<td>- Riemann problem</td>
<td>322</td>
</tr>
<tr>
<td>- SDGM level set method</td>
<td>338</td>
</tr>
<tr>
<td>- SDGM-NI, see SDGM</td>
<td></td>
</tr>
<tr>
<td>- SEM, 238</td>
<td></td>
</tr>
<tr>
<td>- algebraic system</td>
<td>359</td>
</tr>
<tr>
<td>- beam intercepting device demo</td>
<td>354</td>
</tr>
<tr>
<td>- GLS stabilization</td>
<td>288</td>
</tr>
<tr>
<td>- in 1D</td>
<td>239</td>
</tr>
<tr>
<td>- a posteriori error bounds</td>
<td>269</td>
</tr>
<tr>
<td>- a posteriori analysis</td>
<td>261</td>
</tr>
<tr>
<td>- a priori analysis</td>
<td>258</td>
</tr>
<tr>
<td>- basis functions</td>
<td>241, 242</td>
</tr>
<tr>
<td>- in multiple dimensions</td>
<td>245</td>
</tr>
<tr>
<td>- a posteriori analysis</td>
<td>265</td>
</tr>
<tr>
<td>- a priori analysis</td>
<td>264</td>
</tr>
<tr>
<td>- function spaces</td>
<td>245, 247</td>
</tr>
<tr>
<td>- preconditioning</td>
<td>360</td>
</tr>
<tr>
<td>- Roman Colosseum demo</td>
<td>356</td>
</tr>
<tr>
<td>- Stokes equations</td>
<td></td>
</tr>
<tr>
<td>- analysis</td>
<td>283</td>
</tr>
<tr>
<td>- SEM-NI, 238, 352</td>
<td></td>
</tr>
<tr>
<td>- algebraic aspects</td>
<td>252</td>
</tr>
<tr>
<td>- arteriovenous graft demo</td>
<td>352</td>
</tr>
<tr>
<td>- condition number</td>
<td>253</td>
</tr>
<tr>
<td>- in 1D</td>
<td>243</td>
</tr>
<tr>
<td>- a priori analysis</td>
<td>260</td>
</tr>
<tr>
<td>- matrix</td>
<td>243</td>
</tr>
<tr>
<td>- in multiple dimensions</td>
<td>251</td>
</tr>
<tr>
<td>- error estimate</td>
<td>258, 268</td>
</tr>
<tr>
<td>- preconditioning</td>
<td>253</td>
</tr>
<tr>
<td>- Schur complement</td>
<td></td>
</tr>
<tr>
<td>- condition number</td>
<td>392, 393</td>
</tr>
<tr>
<td>- Schwarz preconditioners</td>
<td>374</td>
</tr>
<tr>
<td>- Stokes equations</td>
<td></td>
</tr>
<tr>
<td>- analysis</td>
<td>283</td>
</tr>
<tr>
<td>shock capturing, see Euler equations</td>
<td></td>
</tr>
<tr>
<td>shock fitting, see Euler equations</td>
<td></td>
</tr>
<tr>
<td>shock tracking, see Euler equations</td>
<td></td>
</tr>
<tr>
<td>simplicial domains</td>
<td></td>
</tr>
</tbody>
</table>
– 2D basis functions, 465
– 3D basis functions, 466
singularity detection, 233, 235
smoothing, see filter
Sobolev seminorm
– s-to-N, 268
space
– Banach, 490
– finite-dimensional, 490
– infinite-dimensional, 490
– dual, 491, 499
– Hilbert, 489
– finite-dimensional, 489
– infinite-dimensional, 490
– normed
– dense subspace, 492
– Sobolev, 497
– periodic, 499
specific heat
– constant pressure, 4
– constant volume, 4
specific volume, 3
spectral accuracy, 48, 437
spectral discontinuous Galerkin
methods, see SDGM
spectral element methods, see SEM
– with numerical integration, see SEM-NI
speed of sound, 5
splitting methods, see fractional step
methods
spurious pressure modes, 281
stabilization
– by bubble functions, 281, 282
– by filtering, 282
staggered grid
– Euler equations, 219
Steklov–Poincaré
– operator, 385, 386
– problem, 385
– Richardson iterations, 404
Stokes equations
– SEM-NI approximation, 421
Stokes equations, 94, 279
– algebraic form, 415, 421
– compressible, 20
– generalized, 407
– MEM approximation, 300
– SEM, 278, 281
– SEM approximation, 421
– SEM-NI, 281
– Kovasznay solution, 286
Stokes matrix
– LDU factorization, 417
– artificial compressibility, 418
– preconditioners, 416
– block diagonal, 417
– block triangular, 417
– finite elements, 417
– IISS, 419
– SIMPLE, 418
– SIMPLEX, 418
– SIMPLER, 418
Stokes problem, 164, 168
Stokes system
– block-diagonal preconditioner, 424
– block-triangular preconditioners, 424
– global Schur complement, 422
– block-diagonal preconditioners, 423
– block-triangular preconditioners, 423
– Neumann–Neumann preconditioner, 423
– wire-basket preconditioners, 423
– Schwarz preconditioners, 424
– subdomain Schur complement, 423
Sturm–Liouville problem
– Chebyshev polynomials, 451
– Jacobi polynomials, 445
– Legendre polynomials, 455
subsonic, 17, 204
supersonic, 17, 204
SUPG method, 281
tau correction, 127, 130, 185, 539
tau method, 479
Taylor–Couette flow algorithms, 139
temperature, 3
tensor
– strain-rate, 8
– viscous stress, 8
theorem
– Lax-Milgram, 492
– Riesz, 491
– Sobolev imbedding, 497, 498
thermal conductivity, 9
time-discretization
– formulas
– Adams–Bashforth (AB) methods, 529
– Adams–Moulton (AM) methods, 531
– backward-difference (BDF) methods, 412, 413, 427, 429, 533
– backward-Euler (BE) method, 531
– Crank–Nicolson (CN) method, 531
-- forward-Euler (FE) method, 529
-- leap-frog (LF) method, 529
-- low-storage Runge–Kutta/Crank–Nicolson method, 535
-- Runge–Kutta (RK) methods, 534
-- integrating-factor technique, 96, 99
-- reversible, 527
-- right Lipschitz condition, 526
-- stability
-- absolute stability region, 527
-- A-stable, 527
-- asymptotic stability, 527
-- stability limit, 528
-- stability region, 527
-- stable, 526
-- temporal stability, 527
-- unconditionally stable, 528
-- weak instability, 527
-- zero-stable, 526
-- stability limit table, 533
-- stability region plots, 532
-- symmetric, 527
trace operator, 499

transform space, 449
trigonometric, see Fourier
truncation error, 441
turbulence models
-- closure problem, 14
-- detached-eddy simulation (LES), 13
-- large-eddy simulation (LES), 13
-- Reynolds-averaged Navier–Stokes (RANS), 13

viscosity, 8
-- artificial, 233, 234
-- bulk, 8
-- eddy, 23
-- first coefficient, 8
-- hyperviscosity, 210, 215
-- kinematic, 22
-- second coefficient, 8
-- shear, 8
-- spectral vanishing, 210, 234
-- Sutherland’s formula, 9
vorticity, 25