Department of Mathematics at University of California, San Diego

Computational Finance

Mathematical Background [**L**ecture 1]

Michael Holst

January 10, 2018

Contents

List of Algorithms

1 Review of Some Undergraduate Mathematics Tools

We will need to recall some basic notation and concepts from calculus and linear algebra so that we can understand how to formulate finance problems as optimization problems, and go on to develop and use computational algorithms to solve these problems. We will review linear algebra first before reviewing Calculus, since it will give us some natural notation for discussing functions of several variables.

To this end, in Section 1.1 we first review some very basic ideas from a typical first- or second-year undergraduate course on linear algebra, including concepts, notation, and the interpretation of matrix equations and eigenvalue problems. In Section 1.2, we then review some familiar material from a standard first-year undergraduate course on Calculus of functions of one and several variables. In Section 1.3, we also discuss a small amount of additional material from a typical second-year undergraduate course on ordinary and partial differential equations (ODE and PDE). We will need at least some basic familiarity with terminology and notation for ODE and PDE to understand how to use certain computational techniques in finance, such as numerical methods for Black-Scholes models. However, we do not assume that the reader has had any prior exposure to material on either ODE or PDE.

A good reference for the material in Section 1.2 on Calculus is any standard undergraduate textbook for a first-year Calculus course, such as [3]. At UCSD, most of our first-year undergraduates take a year-long Calculus sequence that goes well-beyond the material discussed in Section 1.2. A good reference for the Linear Algebra material in Section 1.1 are some of the later chapters in any standard undergraduate Calculus textbook such as [3], or any standard undergraduate textbook for a first- or second-year Linear Algebra course, such as [4]. A good reference for the ODE and PDE material below is any standard textbook for a second-year undergraduate course on differential equations, such as [1]. Numerical methods for PDE problems such as Black-Scholes is not a topic typically found in the references mentioned so far, but is still at the level of a second- or third year undergraduate course for science or engineering students. Again, we do not assume that the reader has had any prior exposure to material on either ODE or PDE; however, for the adventurous reader, the book [2] contains an overview of the numerical methods that we will study in the last several lectures of the course, as well as more advanced material on ODE and PDE.

1.1 Linear Algebra

We now review some basic ideas from a typical first- or second-year undergraduate course on linear algebra, including concepts, notation, and the interpretation of matrix equations and eigenvalue problems. Although we will need these concepts from linear algebra to formulate and understand algorithms for optimization problems and differential equation models in finance, nailing down some basic notation and some simple ideas from linear algebra will also make our discussion of Calculus in the next section a bit easier.

To get us started, recall that we use the symbol $\mathbb{R}$ to denote the set of real numbers. Given two real numbers $a \leq b$, we can define the closed interval $[a, b]$ and the open interval (a, b), as well as half-open/half-closed variants. It is standard to think of $[a, b]$ as forming a subset of $\mathbb{R}$, and then write $[a, b] \subset \mathbb{R}$, and similarly for the other various subsets we can form using a and b. What makes the set $\mathbb{R}$ so useful is not that it is a collection of useful numbers, but that it also comes with the structure of a *field*. A *field* is a set that has addition and multiplication defined on it, such that it is *closed* under those operations, meaning that you can generate only additional elements of the set through those operations. The two operations have the properties we are all familiar with in the case of addition and multiplication of real numbers (associativity, commutativity, existence of identity and inverse elements, and distributivity between the operations). The field $\mathbb{R}$ is actually an *ordered field*, in that the elements can be placed in a left-to-right ordering (the real line); the complex numbers $\mathbb{C}$ also form a field under complex addition and multiplication, but there is no natural ordering that we can place on $\mathbb{C}$.

1.1.1 Vectors, Matrices, Addition and Multiplication Operations

We use the symbol $\mathbb{R}^n$ to denote the set of n-vectors of real numbers, i.e., the set of column vectors of length $n \geq 1$, each component of which is a real number. Note that if $n = 1$, then we simply write $\mathbb{R}$ instead of $\mathbb{R}^1$. If x is a vector in the set $\mathbb{R}^n$, we use the set inclusion notation: $x \in \mathbb{R}^n$. (We can also consider vectors of complex numbers, which we would denote as $\mathbb{C}^n$; however, we will stick mostly to $\mathbb{R}^n$ throughout these notes.) We consider x to be a column vector of length n, and then denote as x^T, meaning the transpose of x, as a row-vector of length n:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad x^T = [x_1, x_2, \ldots, x_n],$$

where $x_k \in \mathbb{R}$ for $k = 1, \ldots, n$. Here are some specific examples:

$$x = \begin{bmatrix} 1 \\ 0 \\ -3 \\ 6.25 \end{bmatrix}, \qquad y = \begin{bmatrix} 0.1 \\ -1 \\ 5 \end{bmatrix}, \qquad y^T = [0.1, -1, 5].$$

Notice that we do not need to put an arrow on x to make it clear it is a vector; the type of object that x is will be clear from the context.

When we are working with an $m \times n$ matrix A of real numbers, with m the number of rows of A, and n the number of columns of A, we use the notation $A \in \mathbb{R}^{m \times n}$. We denote the entry of the matrix A located in the i-th row and j-th column as a_{ij}, or sometimes as A_{ij}.

Matrix-Matrix *addition* is well-defined for matrices having the same shape; we simply add their components together: If $A, B \in \mathbb{R}^{m \times n}$, then $C = A + B \in \mathbb{R}^{m \times n}$, and

$$C = A + B, \qquad c_{ij} = a_{ij} + b_{ij}.$$

Recall that if $A \in \mathbb{R}^{m \times n}$ and if $B \in \mathbb{R}^{n \times p}$, then matrix-matrix *multiplication* is well-defined, and the entries of the product matrix $C \in \mathbb{R}^{m \times p}$ are given by:

$$C = AB, \qquad c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

A special case of this is when $p = 1$, so that the matrix B consists of only a single column vector with length matching the number of columns of A. This is called *matrix-vector* multiplication, and the formula above for the entries of the resultant C, which will now consist of a single column vector with length matching the number of rows of A, reduces to:

$$C = AB, \qquad c_i = \sum_{k=1}^{n} a_{ik} b_k.$$

Recall that matrix-matrix multiplication is associative, but not commutative. In other words, although we (generally) cannot swap the order of two matrices in a product, so that $AB \neq BA$, we can change the order of evaluation in a product involving more than two matrices: $ABC = (AB)C = A(BC)$. Matrix multiplication is a *linear operation*, meaning that it distributes across linear combinations of arguments:

$$A(\alpha B + \beta D) = \alpha AB + \beta AD,$$

where $\alpha, \beta \in \mathbb{R}$, $A \in \mathbb{R}^{m \times n}$, and $B, D \in \mathbb{R}^{n \times p}$.

1.1.2 Vector Spaces, norms, inner-products

One of the central concepts in linear algebra is that of a *vector space*, and we make use of this concept throughout all of mathematics (including optimization and differential equations!) Recall that a vector space X can be though of as:

$$X = \langle V, \mathbb{K}, +, \cdot \rangle,$$

where V denotes the set of *vectors*, where $\mathbb{K}$ is the set of *scalars* (assumed to have the structure of a *field*), $+$ represents the addition operation for vectors, and $\cdot$ represents scalar-vector multiplication. The two operations are taken to have certain properties (commutativity, associativity, identity, inverse, compatibility, and distributivity), and the key idea of a vector space is that it is *closed* with respect to these operations. I.e., if we form a new vector z from given vectors $u, v \in V$, using a linear combination involving the $+$ and $\cdot$ operations, such as $w = \alpha u + \beta v$, with $\alpha, \beta \in \mathbb{K}$, then w must also be in V. We will not review the explicit vector space properties of the $+$ and $\cdot$ operations that we learn in linear algebra, and instead will appeal to the intuition we have from the "canonical" vector space that we have all worked with since our first-year mathematics courses:

$$V = \langle \mathbb{R}^n, \mathbb{R}, +, \cdot \rangle.$$

The vector space V is the set $\mathbb{R}^n$ of length n column vectors, together with the field $\mathbb{R}$ of real numbers, along with the usual operations of vector addition (component wise addition of n-vectors) and scalar-vector multiplication (scaling the components of an n-vector by a real number). For example, if $n = 2$, and if we have the two vectors $u = [1, 0]^T$ and $v = [2, 2]^T$, can take what is called a *linear combination* of them using e.g. $\alpha = 2$ and $\beta = 3$ to produce a third vector w:

$$w = \alpha u + \beta v = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}.$$

Some of the concepts we learn in linear algebra concerning vector spaces are the *span* of a given set of vectors (the set of all vectors that can be reached by some linear combination of a given set of vectors), and a *basis* for a vector space (the minimal spanning set that can produce any vector in the vector space). Also important was the idea of *linear independence* (a set of vectors which cannot be reached by a linear combination of other vectors in the same set), and *linear dependence* (a set of vectors which are not linear independent). A concept that arises naturally is that of a *subspace* S of a vector space V. The *subset* $S \subset V$ qualifies as a *subspace* of the vector space V if it is a self-contained vector space, meaning that it is closed with respect to the vector space operations it inherits from V; i.e., if $u, v \in S \subset V$, then any linear combination of u and v remain in S.

Norms. The structure of a vector space is useful for understanding purely *algebraic properties* of sets of vectors: are they linearly independent, do they form a basis for a vector space, and so forth. However, there is no mechanism for discussing the "size" of a vector (other than its basic dimension n), and no way to describe whether one vector is "close" to another. For these types of questions, we need to add a function called a *norm*. A norm $\| \cdot \| \colon \mathbb{R}^n \to \mathbb{R}$ is a function that satisfies these three properties:

1) $\|u\| \geq 0, = 0$ iff (if and only if) $u = 0$.

2) $\|\alpha u\| = |\alpha| \|u\|, \quad \forall u \in \mathbb{R}^n, \forall \alpha \in \mathbb{R}$.

3) $\|u + v\| \leq \|u\| + \|v\|, \quad \forall u \in \mathbb{R}^n$.

The first property says that the only way the norm can be zero is if we plug in the special "zero" vector that every vector space contains. The second property says that we can pull scalar multiplication out of norms. The third property is the *triangle inequality*, which basically says the shortest distance between two points is the line connecting them. We saw several different norms in the study of linear algebra; the main ones were:

$$\|u\|_p = \left(\sum_{i=1}^{n} |u_i|^p \right)^{1/p}, \quad 1 \leq p < \infty, \qquad \|u\|_\infty = \max_{1 \leq i \leq n} |u_i|.$$

These are called the p-norms; the cases of $p = 1$, $p = 2$, and $p = \infty$ are particularly important:

$$\|u\|_2 = \left(\sum_{i=1}^{n} |u_i|^2 \right)^{1/2}, \qquad \|u\|_1 = \sum_{i=1}^{n} |u_i|, \qquad \|u\|_\infty = \max_{1 \leq i \leq n} |u_i|.$$

With these definitions, it is not difficult to show that for any p with $1 \leq p \leq \infty$, the corresponding norm satisfies the three properties above that every norm must satisfy. Here are some examples:

$$u = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \|u\|_1 = 1 + 2 = 3, \quad \|u\|_2 = (1^2 + 2^2)^{1/2} = \sqrt{5}, \quad \|u\|_\infty = \max\{1, 2\} = 2.$$

When we add one of these norms to the vector space formed from $\mathbb{R}^n$, then we have what is called a *Normed (Vector) Space*. One of the most useful properties of the norm is that it gives us the ability to measure the distance between two points (vectors) in the space:

$$d(u, v) = \|u - v\|. \tag{1.1}$$

The function $d \colon V \times V \to \mathbb{R}$ is known as a *metric*, and it satisfies its own three properties:

1) $d(u, v) \geq 0, = 0$ iff (if and only if) $u = v$.

2) $d(u, v) = d(v, u), \quad \forall u, v \in \mathbb{R}^n$.

3) $d(u, v) \leq d(u, w) + d(w, v), \quad \forall u, v, w \in \mathbb{R}^n$.

The first property says that the only way the distance between two points can be zero is if they are the same point. The second property says that the distance from a point u to a point v is the same as the distance from the point v to the point u. The third property is again the *triangle inequality*, which says the if we detour to a third point w on the way from a point u to a point v, the distance we travel is at least as great as the direct distance from u to v. As long as you start with a norm satisfying the three norm properties, the metric it "induces" via (1.1) will automatically satisfy the three metric properties. (This is a standard simple theorem in the theory of normed vector spaces.)

A metric adds to a vector space a structure called a *topology*; this is the ability to define open and closed sets in the space, convergence of sequences, continuity of maps, and so forth. In fact, if you start with simply a set of points rather than an entire vector space, and you add a metric to it, then you still get these features (open and closed sets, and so forth), independent of whether you also have a vector space structure. A set together with a metric on the set is called a *metric space*. Every normed space equipped with (1.1) as its metric can be viewed as a metric space.

While this is all very useful, one comparison between vectors still eludes us: how can we measure whether or not two vectors are pointing in the same direction?

Inner-Products. To answer this question, we need one more function: the *inner-product.* An inner-product on a vector space is two-argument function $(\cdot,\cdot)\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that satisfies these three properties:

1) $(u, u) \geq 0, = 0$ iff (if and only if) $u = 0$.
2) $(u, v) = (v, u), \quad \forall u, v \in \mathbb{R}^n$.
3) $(u, \alpha v + \beta w) = \alpha(u, v) + \beta(u, w), \quad \forall u, v, w \in \mathbb{R}^n, \quad \forall \alpha, \beta \in \mathbb{R}$.

The first property says that the inner-product of a vector with itself is never zero unless it is the special zero vector contained in every vector space. The second property says that the inner-product is symmetric, so it does not matter what order we take the arguments. The third property says that the inner-product is linear in its first argument; note that by the second property, it must be linear in its second argument as well.

The inner-product gives a vector space an additional structure called a *geometry.* It gives us the ability to measure the "angle" between vectors, and in particular it gives us the concept of *orthogonality*: We say two vectors u and v are *orthogonal* if they have zero inner-product, and write $u \perp v$:

$$(u, v) = 0.$$

The most useful inner-product for n-vectors is the Euclidean inner-product:

$$(u, v) = (u, v)_2 = \sum_{i=1}^{n} u_i v_i, \qquad u, v \in \mathbb{R}^n.$$

Here are some examples using this standard inner-product:

$$u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (u, v)_2 = 1 \cdot 0 + 0 \cdot 1 = 0, \quad (u, w)_2 = 1 \cdot 1 + 0 \cdot 1 = 1.$$

When we add an inner-product to a vector space, we call it an *Inner-Product Space.* One useful observation is that every inner-product automatically defines a norm:

$$\|u\| = (u, u)^{1/2}. \tag{1.2}$$

Similar to what we saw earlier when defining a metric using the norm, as long as we start with an inner-product satisfying the three inner-product properties, the norm it "induces" via (1.2) will automatically satisfy the three norm properties. (This is another standard theorem in the theory of normed vector spaces.) In particular, note that the Euclidean inner-product induces the 2-norm we encountered above:

$$\|u\|_2 = (u, u)_2^{1/2} = \left(\sum_{i=1}^{n} u_i^2 \right)^{1/2}.$$

In other words, every inner-product space is automatically a normed space. (Additionally, since we noted that every normed space is also a metric space, we have that every inner-product is automatically a metric space as well.)

Matrix Norms. Note that the idea of taking linear combinations of matrices is really just as natural as that of taking linear combinations of column vectors (which are just special cases of matrices). This leads naturally to the idea of thinking about the set of $m \times n$ matrices as forming its own vector space; it is clearly *closed* with respect to matrix-matrix addition, since we produce a matrix of the same dimension. This gives us access to one of the most useful mathematical tools for working with matrices: The *matrix norm*, and when added to the vector space of matrices it turns it into a normed vector space (of matrices). The most useful matrix norms are the ones *induced*, or *subordinate* to the p-norms of the vector spaces that are associated with the row and column dimensions of the matrix. In particular, if $A \in \mathbb{R}^{m\times n}$, we define the induced matrix norm of A as:

$$\|A\|_p = \sup_{x\in\mathbb{R}^n, x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}. \tag{1.3}$$

Note that by defining the matrix norm in this way, we get access to the following useful inequality:

$$\|Ax\|_p \leq \|A\|_p \|x\|_p, \qquad \forall x \in \mathbb{R}^n. \tag{1.4}$$

The reason that this holds is that since equality holds in (1.3) for the optimal choice of $x \in \mathbb{R}^n$ that maximizes the right-hand-side of (1.3), then any other choice $x \in \mathbb{R}^n$, $x \neq 0$, on the right-hand gives the following inequality:

$$\|A\|_p \geq \frac{\|Ax\|_p}{\|x\|_p}, \qquad \forall x \in \mathbb{R}^n, \quad x \neq 0.$$

Multiplying through by $\|x\|_p$ we obtain (1.4), which we note continues to hold even when $x = 0$.

Another matrix norm that is often useful, which is not an induced norm, is the *Frobenius* norm:

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2 \right)^{1/2}.$$

In both cases, the matrix norms can be shown to satisfy the three properties of a proper norm.

1.1.3 Matrix Equations

One of the main problems we study in linear algebra is the solution of matrix equations:

$$\text{Find } x \in \mathbb{R}^n \text{ such that } Ax = b, \text{ where } A \in \mathbb{R}^{m\times n}, \quad b \in \mathbb{R}^m.$$

There are three distinct cases with very different notions of solutions and techniques:

1) $m = n$: square (determined) systems,
2) $m > n$: over-determined systems,
3) $m < n$: under-determined systems.

In an undergraduate course in linear algebra, we typically study mainly the first two cases. For over-determined systems, one has more equations than unknowns, and the notion of "solution" is reduced to finding an x that minimizes the residual $b - Ax$, rather than trying to force $Ax = b$. If one minimizes the 2-norm of the residual, $\|b - Ax\|_2$, then it can be shown that the minimizer satisfies the *normal equations*:

$$A^T Ax = A^T b,$$

where A^T is the transpose of A, i.e., $(A^T)_{ij} = A_{ji}$. Note the number of rows and columns in $A^T A$ is n, so this is now an $n \times n$ matrix system. If A has linearly independent columns, then it can be shown that $A^T A$ is *invertible*, so we have reduced the problem of solving the original over-determined system to that of solving a square system.

Solution of square systems, i.e., those where $m = n$, is possible when the target vector b on the right hand side of the equation $Ax = b$ is in the *range* of A, which is the space spanned by the column vectors of A. I.e., there is some linear combination of columns of A that can exactly recover b, and the solution x is nothing other than the linear combination coefficients:

$$b = \sum_{i=1}^{n} x_i a_i,$$

where a_i is the i-th column vector of A. If $A \in \mathbb{R}^{n\times n}$ has n linearly independent columns, then the columns form a basis for all of $\mathbb{R}^n$, in which case any choice of $b \in \mathbb{R}^n$ can be written as a linear combination of the columns of A. In this case the problem $Ax = b$ is always uniquely solvable for any b, and we say that A is *nonsingular* (or invertible), and write $x = A^{-1}b$. A standard test for invertibility of A is the determinant: $\det(A) \neq 0$ if and only if A is nonsingular. Here is a useful trick to remember for computing the determinant of a 2×2 matrix:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc.$$

Let us apply this to a specific case:

$$\det \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = 1 \cdot 4 - 2 \cdot 3 = -2.$$

The inverse matrix A^{-1} can be formed explicitly, but in practice one uses algorithms to produce the solution x without explicitly forming A^{-1}, such as Gaussian elimination, LU-factorization (which is really the same as Gaussian elimination), or iterative methods such as the Gauss-Seidel iteration or the Conjugate Gradient Method.

For a general (potentially non-square) matrix $A \in \mathbb{R}^{m\times n}$, matrix-vector multiplication tells us that $A \colon \mathbb{R}^n \to \mathbb{R}^m$. The vector spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ are the basic *domain* and *range* spaces of A, respectively. Solvability of the associated linear system $Ax = b$ can be understood through what are known as the *four fundamental subspaces* of A, which are four distinct subspaces of the basic domain and range spaces. These four fundamental subspaces are:

$$\begin{aligned}
\mathcal{R}(A) &= \{\ y \in \mathbb{R}^m \mid y = Ax \text{ for some } x \in \mathbb{R}^n\ \} \subset \mathbb{R}^m, \\
\mathcal{N}(A) &= \{\ x \in \mathbb{R}^n \mid Ax = 0 \in \mathbb{R}^m\ \} \subset \mathbb{R}^n, \\
\mathcal{R}(A^T) &= \{\ x \in \mathbb{R}^n \mid x = A^T y \text{ for some } y \in \mathbb{R}^m\ \} \subset \mathbb{R}^n, \\
\mathcal{N}(A^T) &= \{\ y \in \mathbb{R}^m \mid A^T y = 0 \in \mathbb{R}^n\ \} \subset \mathbb{R}^m.
\end{aligned}$$

The subspace $\mathcal{R}(A)$ is called the (true) *range* of A, and $\mathcal{N}(A)$ is called the *nullspace* of A, with similar terminology for $\mathcal{R}(A^T)$ and $\mathcal{N}(A^T)$. The *Fundamental Theorem of Linear Algebra (FTLA)* tells us that the four fundamental subspaces have the following relationships:

$$\mathcal{R}(A) = [\mathcal{N}(A^T)]^{\perp}, \qquad \mathcal{R}(A^T) = [\mathcal{N}(A)]^{\perp}.$$

These relationships tell us something about solvability of the linear system $Ax = b$. If $b \in \mathcal{R}(A)$, then there exists $x \in \mathbb{R}^n$ such that $b = Ax$; i.e., the system is solvable. The FTLA tells us that we can infer that $b \in \mathcal{R}(A)$ by confirming that $b \perp \mathcal{N}(A^T)$. The simplest statement of this principle is the following: $Ax = b$ is solvable if and only if $b^T y = 0$ for all y such that $A^T y = 0$.

1.1.4 Eigenvalue Problems

Another important problem we study in linear algebra is the solution of eigenvalue problems:

$$\text{Find } (x_i, \lambda_i) \in \mathbb{C}^n \times \mathbb{C} \text{ such that } Ax_i = \lambda_i x_i, \text{ where } A \in \mathbb{R}^{n\times n}, \quad x_i \neq 0. \tag{1.5}$$

In the case of eigenvalue problems, one is after the set of *eigenpairs* of a square matrix; every square $n \times n$ matrix has n such pairs, (x_i, λ_i), $i = 1, \ldots, n$. We call λ_i an *eigenvalue*, and x_i is its associated *eigenvector*. The set of all eigenvalues is called the *spectrum* of A, and is denoted

$$\sigma(A) = \{\lambda \in \mathbb{C} \mid Ax = \lambda x, \text{ for some } x \in \mathbb{C}^n\},$$

and the largest of the eigenvalues in magnitude is called the *spectral radius*, and is denoted

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| = \max_{1 \le i \le n} |\lambda_i|.$$

Two features of any eigenvector x_i are worth noting: first, they are required to be nonzero; and second, we can always normalize them (give them unit length), because we can multiply both sides of (1.5) by $\|x_i\|^{-1}$ and absorb this factor into the definition of the normalized eigenvector, $v_i = x_i/\|x_i\|$. There are many possibilities for the eigenpairs of a general square matrix; the eigenvalues might be complex, they might be repeated (several eigenvalues might be the same value) and the eigenvectors might be linearly dependent. The terms *algebraic* and *geometric multiplicity* are concerned with these possible degeneracies. In the case of rectangular matrices, there is a generalization we call *singular values* and (left and right) *singular vectors*, which (nearly) reduce to eigenpairs in the case of square matrices. A number of powerful algorithms have been developed over the last several decades for finding all (or a desired subset) of the eigenpairs (or more generally, the singular values and vectors) of general matrices.

However, many applications are blessed with *symmetric* matrices, which are square matrices that are equal to their own transpose:

$$A = A^T.$$

When this property holds, the eigenvalue problem becomes much simpler, due to two facts:

1) $\lambda_i \in \mathbb{R}, x_i \in \mathbb{R}^n, \quad i = 1, \ldots n.$
2) $x_i \perp x_j, \quad i \neq j.$

The first fact one says that all of the eigenvalues and eigenvectors of A are real, and the second fact says that the eigenvectors are all orthogonal. Since we have n orthogonal eigenvectors, they must span all of $\mathbb{R}^n$. In this case, if we form a square matrix V with columns made from the n eigenvectors of A, we can write the eigenvector-eigenvalue equations all at once as the following matrix system:

$$AV = V\Lambda, \qquad V = \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & & \vdots \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}.$$

Since V has n linearly independent columns (orthogonal vectors are obviously linearly independent), it is invertible, and by multiplying on either the left or right of the equation above we obtain two useful facts:

$$V^{-1}AV = \Lambda, \qquad A = V\Lambda V^{-1}. \tag{1.6}$$

The first equation in (1.6) states that A can be diagonalized by a *similarity transformation* involving V and V^{-1}. Since similarity transformations preserve eigenvalues, the diagonal matrix Λ must have the eigenvalues of A on its diagonal. The second equation in (1.6) states that A can be factored into three matrices, one of which holds the eigenvectors as columns, and the middle matrix contains all of the eigenvalues of A on its diagonal. This is referred to as the *spectral decomposition* of A.

Computing Eigenvalues. Over the last several decades, a number of computationally efficient algorithms have been developed for both the general eigenvalue problem, and also the special case of symmetric matrices. Many of these algorithms are built into MATLAB. For small matrices (meaning 2×2 or 3×3), it is convenient to compute eigenvalues by hand using the *characteristic polynomial.* We obtain this polynomial by rewriting the eigenvalue equation slightly, and using the fact that the determinant of a singular matrix is zero:

$$Ax = \lambda x, \; x \neq 0, \quad \Longleftrightarrow \quad [A - \lambda I]\, x = 0, \; x \neq 0, \quad \Longleftrightarrow \quad \det(A - \lambda I) = 0.$$

The equation on the left is the original eigenvalue-eigenvector equation. The equation in the middle is the same equation written with terms rearranged to emphasize the fact that an eigenvalue is basically a shift that you can make to a matrix that turns the result into a singular matrix; this must be the case, since the solution to the resulting system is nonzero, but the right hand side is zero. (Only singular matrices can give nonzero solutions to problems with zero right hand sides; this is one way to characterize a singular matrix.) Finally, the third equation on the right is just the restatement of what we recalled earlier, namely that singular matrices have zero determinant.

This third equation involves a degree n polynomial, known as the *characteristic polynomial of* $[A - \lambda I]$, the roots of which are the eigenvalues of A. Let us look at a simple example:

$$A = \left[\begin{array}{cc} 2 & -1 \\ -1 & 2 \end{array} \right], \quad \det[A - \lambda I] = \det \left[\begin{array}{cc} 2-\lambda & -1 \\ -1 & 2-\lambda \end{array} \right] = (2-\lambda)(2-\lambda) - 1 = \lambda^2 - 4\lambda + 3.$$

This polynomial has two roots, which are the two eigenvalues of A. We can either use the quadratic formula to find the roots, or we can actually factor this simple case by inspection:

$$\det[A - \lambda I] = \lambda^2 - 4\lambda + 3 = (\lambda - 1)(\lambda - 3).$$

Therefore, the eigenvalues of this symmetric 2×2 matrix A are $\lambda_1 = 1$ and $\lambda_2 = 3$. We knew they had to be real numbers since A is symmetric, but we could not easily know their actual values without the *characteristic equation* (the equation we solve involving the characteristic polynomial).

1.1.5 Condition Number of a Matrix

When solving matrix equations $Ax = b$ using computer algorithms, we are immediately faced with the following question: If the problem data (the entries A and/or b) contain errors, either due to data acquisition or simply computer roundoff, what impact does this have on the solution x that the algorithm produces? Let us examine this question a bit, since the answer is relevant to all of the algorithms will develop for both optimization and differential equation models in finance.

To begin, let us assume we are faced with solving the system $Ax = b$, where A is a nonsingular $n \times n$ matrix. Let's imagine we have made an error in the right-hand side vector b, and have instead $b + \delta b$. The solution to the matrix system $Ax = b$ will then also contain an error, which we will call δx. The system we are solving can be viewed as having the form:

$$A(x + \delta x) = b + \delta b.$$

Let's isolate δx on one side of the equation, so we can get a handle on what it depends on, and how large it could be. To do so, let's first left-multiply both sides by A^{-1}, which gives:

$$x + \delta x = A^{-1}(b + \delta b) = A^{-1}b + A^{-1}\delta b = x + A^{-1}\delta b.$$

Subtracting x from both sides gives then:

$$\delta x = A^{-1}\delta b.$$

To get a bound on the size of δx, we can take norms (any p-norm, for example):

$$\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\|\|\delta b\|, \tag{1.7}$$

where we have made use of (1.4). This tells us that we can bound the size of the error δx due to δb by this inequality. Since x itself may have very small or very large norm, this inequality is not so useful as it stands. A better measure of the impact would be a bound on the *relative error*, which is $\|\delta x\|/\|x\|$. We can get such a bound if we can get a *lower bound* on $\|x\|$ in terms of the data. We can do this by starting with $Ax = b$ and take norms of both sides as follows:

$$b = Ax \quad \Longrightarrow \quad \|b\| = \|Ax\| \quad \Longrightarrow \quad \|b\| \leq \|A\|\|x\| \quad \Longrightarrow \quad \|x\| \geq \|A\|^{-1}\|b\|. \tag{1.8}$$

If we combine (1.7) and (1.8), then we have what we were after:

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A)\frac{\|\delta b\|}{\|b\|}, \tag{1.9}$$

where we have introduced the following symbol in the expression above:

$$\kappa(A) = \|A\|\|A^{-1}\|. \tag{1.10}$$

The quantity $\kappa(A)$ as defined in (1.10) is called the *condition number* of A. Note that if we use a different matrix norm, then we will get a different condition number; for this reason, you will often see the particular norm used indicated as a subscript in the notation for $\kappa(A)$, for example $\kappa_p(A)$ for any p-norm. The standard convention is to use the 2-norm when discussing condition numbers, i.e. $\kappa(A) = \kappa_2(A)$. One of the reasons for this is that when the matrix A is symmetric, the 2-norm has a special relationship to the eigenvalues of A, and consequently so does the condition number. In particular, if $A \in \mathbb{R}^{n\times n}$ and$A = A^T$, then:

$$\|A\|_2 = \rho(A), \qquad \kappa_2(A) = \frac{\max_{\lambda\in\sigma(A)}|\lambda|}{\min_{\lambda\in\sigma(A)}|\lambda|}.$$

As we have seen above, condition numbers tell us how *sensitive* a particular matrix system $Ax = b$ is to errors in the right-hand-side b. A similar analysis can be done for errors in the entries matrix A itself; one finds again that the condition number is the quantity that appears in relative error bounds. In addition, condition numbers will naturally arise as determining factors in the number of iterations required by computer algorithms to solve matrix systems as well as optimization problems.

1.2 Calculus

We now review some familiar material from a standard first-year undergraduate course on Calculus of functions of one and several variables. We will need to understand basic notation and concepts from both calculus and linear algebra to understand how to formulate finance problems as optimization problems, and then to go on to use computational algorithms to solve these problems. Having covered basic notation and some simple ideas from linear algebra in the previous section will make our discussion below a bit easier.

1.2.1 Univariate Functions

The calculus (of functions of a single real variable) is the study of the properties of functions $f(x)$ that take as input a number $x \in \mathbb{R}$, and return as output a number $f(x) \in \mathbb{R}$. We write this as $f\colon \mathbb{R} \to \mathbb{R}$; these are also called *univariate functions*, and x is called the *independent variable*. Some examples of such real-valued functions of a single real variable include:

$$f(x) = x^2 + 3x, \quad f(x) = 7, \quad f(x) = \sin(x).$$

The first two examples are polynomials (the second of which is the simplest type of polynomial, namely a constant), and the third is a trigonometric function. In the first year of calculus we learn about differentiating and integrating these types of functions, and then applying this technology to solving various problems arising in science and engineering.

Differentiation. Recall that the derivative of a function $f\colon \mathbb{R} \to \mathbb{R}$ is defined formally as follows:

$$f'(x) = \frac{df(x)}{dx} = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h},$$

where we use f' as a shorthand notation for df/dx when it is convenient and does not lead to ambiguity. However, we quickly learn in first-year calculus that there are various formulas we can use to compute a derivative of common functions such as *polynomials* and *trigonometric* functions, rather than working directly with the definition involving the "limit" above. The first such formula we typically learn to use is the following one for differentiating individual terms in a polynomial (referred to as a *monomial*):

$$\frac{d}{dx}x^n = nx^{n-1}.$$

Finally, we learn how to compute derivatives of more complex functions using techniques such as the product rule and the chain rule:

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x), \qquad g(f(x))' = \frac{dg}{df}\,\frac{df}{dx}.$$

Here is an example using both the product and chain rules:

$$(3(x+2)^2x^3)' = 3((x+2)^2)'x^3 + 3(x+2)^2(x^3)' = 6(x+2)x^3 + 9(x+2)^2x^2.$$

A typical example of applying the derivative to solve a problem is: Given the function $p(t) = 3t^2$ representing the position of a particle on a line at time t, compute it's velocity $v(t)$ along the line, which is defined as the instantaneous rate of change of $p(t)$ with respect to time t. To solve this problem, we differentiate $p(t) = 3t^2$ to obtain:

$$v(t) = p'(t) = \frac{dp(t)}{dt} = 6t.$$

Higher-order derivatives of a function $f\colon \mathbb{R} \to \mathbb{R}$ are defined recursively: $f''(x) = (f'(x))'$, and so forth. It is clearly a cumbersome notation to use prime symbols for, say, the "11-th" derivative of $f(x)$, so it is common to use the notation $f^{(k)}(x)$ to represent the k-th derivative of $f(x)$.

As important class of functions are the *continuous functions*. Although one can give a mathematically precise definition of a continuous function as one for which a small amount in its argument produces only a small change in its value, intuitively we can think of continuous functions as functions for which there are no gaps or jumps in their graphs. If f is a continuous function on all of $\mathbb{R}$, we use the notation $f \in C^0(\mathbb{R})$. If f is continuous on the subset $[a, b] \subset \mathbb{R}$, then we would write $f \in C^0([a,b])$. If it is important to draw a distinction between the domain and range of the function, such as when $f\colon [a,b] \to \mathbb{R}$, then we would denote a continuous function as $f \in C^0([a,b],\mathbb{R})$. Another important, related class of functions are those whose derivatives of order k are also continuous functions; we would denote such a function as $f \in C^k(\mathbb{R})$, or $f \in C^k([a,b])$, or simply $f \in C^k$ if the domain of the function is clear from the context. If we can differentiate f endlessly and end up with a continuous function, we denote this as $f \in C^\infty$. The simplest example would be the case of a function f which is continuously differentiable (it has one continuous derivative), which we would denote as $f \in C^1$.

Integration. Recall that the integral of a function $f\colon \mathbb{R} \to \mathbb{R}$ can also be defined formally (as the limit of what are called Riemann sums). Again, we quickly learn a number of useful formulas that allow us to compute definite (integrals with limits of integration) and indefinite (integrals without limits) of commonly occurring functions such as polynomials and trigonometric functions. For example, in the case of monomials we have:

$$\int x^n \, dx = \frac{x^{n+1}}{n+1} + C, \qquad \int_a^b x^n \, dx = \left.\frac{x^{n+1}}{n+1}\right|_a^b = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}.$$

The front and back cover of a standard calculus book is typically filled with formulas for integrating more complex, but quite special functions. Unfortunately, in the case of integration, we do not have access to similar general techniques like the chain rule, and we instead often have to resort to approximating integrals numerically (a process referred to as *quadrature*). The integral has many applications, including its use as the anti-derivative, the area under a curve, and so forth. A typical application of the integral to solve a problem is: Find the area under the curve $f(x) = x^2$ in the interval $[0,1]$. To solve this problem, we integrate $f(x) = x^2$ over the interval $[0,1]$:

$$\int_0^1 x^2 \, dx = \left.\frac{1}{2}x\right|_0^1 = \frac{1}{2}.$$

1.2.2 Multivariate Functions

The calculus of (vector-valued) functions of a several variables is the natural extension of the single-variable calculus to the study of the properties of functions $F(x)$ that take as input a vector $x \in \mathbb{R}^n$, and return as output a vector $F(x) \in \mathbb{R}^n$. We write this as $F\colon \mathbb{R}^n \to \mathbb{R}^m$. This includes of course the case that $m = n = 1$, which is then just the calculus of functions of a single real variable as discussed above. The next simplest case is $m = 1$, $n \geq 1$, which are real-valued functions of n variables, $f\colon \mathbb{R}^n \to \mathbb{R}$; we denote again simply as $f(x)$, but now we have $x \in \mathbb{R}^n$. The general case of $F\colon \mathbb{R}^n \to \mathbb{R}^m$ are m-vector-valued functions of n variables, and we can think of F as a column vector consisting of m real-valued component functions:

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_m(x) \end{bmatrix}, \qquad \text{where} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Here is a specific example: Define $F\colon \mathbb{R}^2 \to \mathbb{R}^3$ as

$$F(x) = \begin{bmatrix} x_1^2 - 2x_2 \\ 2x_2^3 - x_1 \\ x_1 + x_2^2 \end{bmatrix}, \qquad \text{in which case} \qquad F(x)^T = \left[x_1^2 - 2x_2,\ 2x_2^3 - x_1,\ x_1 + x_2^2\right].$$

Differentiation. In the case of multivariate functions, we can consider differentiation with respect to any one of the independent variables by holding all of the other variables as fixed; we refer to this process as *partial differentiation*. For example, of $F\colon \mathbb{R}^n \to \mathbb{R}^m$, then the *partial derivative* of one of the component functions $F_i(x)$ of $F(x)$ with respect to a particular x_j is written as $\partial F_i(x)/\partial x_j$. The collection of all of the partial derivatives of $F(x)$ with respect to all of the independent variables x_k can be arranged into an $m \times n$ matrix referred to the Jacobian matrix of $F(x)$:

$$F'(x) = \begin{bmatrix} \frac{\partial F_1(x)}{\partial x_1} & \frac{\partial F_1(x)}{\partial x_2} & \cdots & \frac{\partial F_1(x)}{\partial x_n} \\ \frac{\partial F_2(x)}{\partial x_1} & \frac{\partial F_2(x)}{\partial x_2} & \cdots & \frac{\partial F_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m(x)}{\partial x_1} & \frac{\partial F_m(x)}{\partial x_2} & \cdots & \frac{\partial F_m(x)}{\partial x_n} \end{bmatrix},$$

Here is an example: We again consider our example from above with $F\colon \mathbb{R}^2 \to \mathbb{R}^3$, so that $x = [x_1, x_2]^T$, and

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ F_3(x) \end{bmatrix} = \begin{bmatrix} x_1^2 - 2x_2 \\ 2x_2^3 - x_1 \\ x_1 + x_2^2 \end{bmatrix}, \qquad F'(x) = \begin{bmatrix} \frac{\partial F_1(x)}{\partial x_1} & \frac{\partial F_1(x)}{\partial x_2} \\ \frac{\partial F_2(x)}{\partial x_1} & \frac{\partial F_2(x)}{\partial x_2} \\ \frac{\partial F_3(x)}{\partial x_1} & \frac{\partial F_3(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 & -2 \\ -1 & 6x_2 \\ 1 & 2x_2 \end{bmatrix}.$$

The Jacobian matrix $F'(x)$ above is an example of an $m \times n$ (rows by columns) matrix of *functions*; if we "fix" the independent variable x, then this becomes a matrix of *real numbers*, which is the familiar object from linear algebra. Similarly, $F(x)$ is an n-vector of *functions*, and if we "fix" the independent variable x, then this becomes an n-vector of *real numbers*. For example, if we fix $x = [x_1, x_2]^T = [0, 1]^T$, then $F(x)$ above becomes the following $m \times 1 = 3 \times 1$ column vector of real numbers, and $F'(x)$ becomes an $m \times n = 3 \times 2$ matrix of real numbers:

$$F(x) = \begin{bmatrix} -2 \\ 2 \\ 1 \end{bmatrix}, \qquad F'(x) = \begin{bmatrix} 0 & -2 \\ -1 & 6 \\ 1 & 2 \end{bmatrix}.$$

A special case that has its own terminology is the case of $m = 1$ and $n \geq 1$; i.e., $f\colon \mathbb{R}^n \to \mathbb{R}$. In this case, the Jacobian matrix reduces to a single row (the first row in the Jacobian matrix above), and we continue to use the notation $f'(x)$ to denote this row vector. However, the column vector obtained by taking its transpose has an important role itself, including in optimization, and it is given the special notation $\nabla f(x)$ and referred to as the *gradient vector*:

$$f'(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1}, & \frac{\partial f(x)}{\partial x_2}, & \ldots, & \frac{\partial f(x)}{\partial x_n} \end{bmatrix}, \qquad \nabla f(x) = f'(x)^T = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

The case of $f\colon \mathbb{R}^n \to \mathbb{R}$ arises as the general type of *objective function* that we will run into when we look at optimization problems below. In this case, the second derivative of f will also play a major role. Just as there are n possible first partial derivatives of f, there are n^2 possible second derivatives of f. If we arrange them in the natural way as the components of an $n \times n$ matrix, we obtain the *Hessian matrix* of $f(x)$ at x:

$$f''(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

Note that if $f \in C^2$, meaning that f has two derivatives which are continuous functions, then the order of differentiation does not matter, and all entries paired across the diagonal of the Hessian are the same:

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}, \qquad i = 1, \ldots, n, \quad j = 1, \ldots, n.$$

This means as long as $f \in C^2$, the Hessian will be a *symmetric matrix*: $f''(x) = f''(x)^T$, where the superscript "T" denotes matrix transpose.

Here is an example: Define $f\colon \mathbb{R}^4 \to \mathbb{R}$ as follows, and we then compute f', ∇f, and $f''(x)$:

$$f(x) = x_1^3 - x_2 x_3^2 + x_4, \qquad f'(x) = \begin{bmatrix} 3x_1^2, & -x_3^2, & -2x_2x_3, & 1 \end{bmatrix}, \qquad \nabla f(x) = \begin{bmatrix} 3x_1^2 \\ -x_3^2 \\ -2x_2x_3 \\ 1 \end{bmatrix},$$

$$f''(x) = \begin{bmatrix} 6x_1 & 0 & 0 & 0 \\ 0 & 0 & -2x_3 & 0 \\ 0 & -2x_3 & -2x_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Integration. For the case of a vector-valued function of a single real variable, $F\colon \mathbb{R} \to \mathbb{R}^m$, we can define integration component-wise:

$$\int_a^b F(t)\, dt = \begin{bmatrix} \int_a^b F_1(t)\, dt \\ \int_a^b F_2(t)\, dt \\ \vdots \\ \int_a^b F_m(t)\, dt \end{bmatrix}.$$

This is the main type of integration problem we will run into other than the case of $m = n = 1$.

1.2.3 Taylor Series and Remainder

One of the key tools we learn to use in the first year of Calculus is the idea of *Taylor series* or *Taylor expansion* of a function $f \in C^\infty(\mathbb{R})$ *about the point* $x \in \mathbb{R}$:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2!} + \cdots + \frac{f^{(n)}(x)h^n}{n!} + \cdots. \tag{1.11}$$

What is remarkable about this series is that it says that if we know only information about the function f at the point x, but the information we know includes all possible derivatives of f at that point, then in fact we also know what value f takes at a nearby point $x + h$.

Notationally, it is common to write these types of expansions using what is known as "big-O" notation. The idea is to make the observation that all of the terms in the series have h raised to a power that matches the position of the term in the series; i.e., the first term has h^0, the second has h^1, the third has h^2, and so forth. We typically use the series when h is small (we will see why below when we examine the *Taylor Remainder*). If we are thinking of h as small (less than one, typically much smaller), then clearly h^2 will be even smaller, and so forth for higher powers. What this means is that for h sufficiently small, the terms with lower powers will dominate in magnitude all of the terms appearing later in the series with higher powers. The "big-O" notation allows us to reflect this with a single term; using this notation for the Taylor expansion, and choosing (arbitrarily) to keep explicitly only the first two terms, the series would read:

$$f(x+h) = f(x) + f'(x)h + O(h^2).$$

This tells us that we are looking at a series involving f, and the remaining terms each have powers of h^2 or higher multiplying each term.

For fixed $x \in \mathbb{R}$, the sum of terms on the right in the Taylor series (1.11) is an example of a *countably infinite series* of real numbers, or simply *infinite series*, meaning that it is a sum of an infinite number of terms (each of which evaluates to a real number), each of which can be put in one-to-one correspondence (or labeled) with the integers. Such series then have the form:

$$S = \sum_{i=1}^{\infty} a_i. \tag{1.12}$$

One of the main topics in Calculus is to study properties of infinite series, and in particular, to know when they sum to a finite quantity (i.e, when S is finite). Such series are called *summable* or *convergent*, and in Calculus we develop various tests (such as the ratio test, the root test, and so forth) that we apply to a typical term in the sum to guarantee that the series is convergent.

In the case of this very specific series above, namely the Taylor series (1.11), if we freeze the point x at which we are evaluating the series, then it becomes precisely a series of the form (1.12). If $f \in C^\infty$, then clearly each term in the series is well-defined. However, since there are infinitely many terms, the series may in fact not be convergent (sum to a finite real number). However, if the series is convergent for this fixed x, then we say f is *analytic* at x. If the series is convergent for any $x \in [a,b]$, then we say f is analytic on $[a,b]$, and if it is convergent for any $x \in \mathbb{R}$, then we say f is analytic on $\mathbb{R}$, or simply that f is analytic.

One of the most important theorems concerning Taylor series is known as the *Taylor Remainder Theorem*. This theorem says that we can actually capture all of the infinite number of remaining terms in the series with a single term as follows:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2!} + \cdots + \frac{f^{(n)}(x)h^n}{n!} + R_n,$$

where the *Taylor Remainder* R_n has the expression:

$$R_n = \frac{f^{(n+1)}(\xi(x))h^{n+1}}{(n+1)!}.$$

This remainder term looks like the next term in the series, except that it is evaluated at a special point $\xi(x)$ between x and $x+h$. It is generally difficult to determine what $\xi(x)$ is, but the theorem establishes that such a point always exists. Note that we can choose to *truncate* a Taylor expansion at any order k, and then use R_k to represent the remaining terms. For example, below we will use the following 3-term expansion in order to derive Newton's method for solving nonlinear equations:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(\xi(x))h^2}{2!}. \tag{1.13}$$

Note that the size of the Remainder term gives us a very useful piece of information: it tells us what *error* we would be making if we choose to ignore that remaining terms and use only the finite number of terms appearing before the remainder as a way of approximating $f(x+h)$. This type of approximation of f, and characterizing the error made via the Taylor remainder, forms the foundation for much of *numerical analysis* and *computational science*. We will come back to this topic at some length when we study numerical methods for Black-Scholes models later in the course.

Let us consider now the more general case of multivariate functions of the form $F\colon \mathbb{R}^n \to \mathbb{R}^m$. Is it possible to Taylor expand this type of function? The answer is yes, and this idea forms the basis for nearly all of the most effective techniques and algorithms for solving nonlinear equations and optimization problems arising in finance, science, engineering, and other areas. Let us begin by again assuming that F is sufficient smooth so that we can at least write down all of the terms in the series; it will be sufficient for us to assume that all of the partial derivatives of F of all orders exist, similar to the one-dimensional case. The series then takes the form:

$$F(x+h) = F(x) + F'(x)h + O(\|h\|^2).$$

This looks very much like (1.13), but now each term in the series is an m-vector. The quantity $F'(x)$ appearing in the second term is the $m \times n$ Jacobian matrix of F at x, and it multiplies the n-vector h to produce the m-vector $F'(x)h$ which then is added to the m-vector $F(x)$. The remainder term is written using "Big-O" notation to avoid writing down the second derivative of F; this is an $m \times n \times n$ collection of all second partial derivatives which can not be written naturally in matrix form. Nevertheless, the remaining terms in the series mimic the one-dimensional case, and each term has increasing "powers" of h. (To explicitly write the terms, we would need to introduce the concept of *tensors*, or *multilinear forms*.) The series converges under appropriate assumptions as in the one-dimensional case, and we can use the Taylor Remainder to characterize the error of truncating the series, just as in the one-dimensional case.

We focused earlier on the special case where $m = 1$, due to the fact that functions of the form $f\colon \mathbb{R}^n \to \mathbb{R}$ play a central role in optimization. In this case, we can write one additional term in the Taylor expansion without resorting to using tensor notation:

$$f(x+h) = f(x) + f'(x)h + h^T f''(x)h + O(\|h\|^3).$$

Now the series is again simply a sum of real numbers, but the terms are themselves formed from products of vectors and matrices. The second term is formed from the inner-product of the *row* n-vector $f'(x)$ with the n-vector h, giving a real number that is added to the first term (the real number $f(x)$). To this is added the third term, which is formed by multiplying the $n \times n$ Hessian matrix against the n-vector h, giving an n-vector, and then taking the inner-product of this n-vector with the *row* n-vector h^T, yielding finally a real number. The remainder term would involve the third derivative of f, which involves $n \times n \times n$ partial derivatives, again requiring tensor notation to explicitly write. It would be reduced to a real number by multiplication against h three times (arranged appropriately), and so forth for higher-order terms in the series. We will use this expansion later to develop methods for Quadratic Programming and methods for more general nonlinear optimization problems.

1.2.4 Newton's Method

To solve both optimization problems and PDE problems in finance, we are going to need to be able to solve nonlinear equations of the form:

$$\text{Find } x \in \mathbb{R}^n \text{ such that } F(x) = 0,$$

where $F\colon \mathbb{R}^n \to \mathbb{R}^n$. Let us assume we have some initial guess at the solution, call it $x^0 \in \mathbb{R}^n$. Unless we are very lucky, the only thing we know for sure is that $F(x^0) \neq 0$. However, in the previous section we found that we have access to Taylor expansion; let us Taylor expand F about the point x^0:

$$F(x^0 + h) = F(x^0) + F'(x^0)h + O(\|h\|^2).$$

Ideally, if we could determine h such that $x = x^0 + h$, giving us $F(x) = F(x^0 + h) = 0$, then our problem would be solved. I.e., we just need to set the series above to zero and solve for h:

$$0 = F(x^0 + h) = F(x^0) + F'(x^0)h + O(\|h\|^2).$$

However, this is an infinite series in h, and looks harder to solve that our original problem. What if we truncate the series at the first non-trivial appearance of h in the series, and try to use this as an approximation to finding h? This would give us:

$$0 = F(x^0) + F'(x^0)h.$$

Let us rearrange this a bit:

$$F'(x^0)h = -F(x^0).$$

This is a well-defined square matrix system; the known information is the right hand side n-vector of real numbers (which we get by evaluating F at x^0), and the $n \times n$ matrix on the left that comes to us by evaluating the Jacobian matrix $F'(x^0)$ at the point x^0. We would just need to solve this matrix system for the n-vector h. We then compute the corrected solution:

$$x^1 = x^0 + h.$$

Since we truncated the series, we generally will not recover the correct h that will give us $x^1 = x$, where x solves $F(x) = 0$. However, we are presumably better off than we were with x^0. This turns out to be the case; in fact, this technique, discovered nearly at the birth of Calculus and known as *Newton's method*, is so effective that to this day it forms the basis of the best algorithms for solving nonlinear equations of this type, and also optimization problems (which lead to nonlinear equations).

If we formulate Newton's method as a complete algorithm for solving a given nonlinear equation $F(x) = 0$, where $F\colon \mathbb{R}^n \to \mathbb{R}^n$, it becomes Algorithm 1. The algorithm is not bullet-proof; for example, if $F'(x^k)$ becomes singular as a matrix for some x^k that appears in the iteration, then we cannot complete the first step to solve for h. Even if we can complete that step, the overall algorithm may not converge at all; Newton's method has very nice *local convergence* properties (converges when x^0 is close to x), but its *global convergence* properties (when x^0 is arbitrarily far from x) are not great. However, for a broad class of problems one can show that $F'(x^k)$ will always be invertible. Moreover, one can use techniques known as *damping* and *backtracking* to improve the global convergence properties of Newton's method.

As a final remark, note that in the one-dimensional case, where $f\colon \mathbb{R} \to \mathbb{R}$, Algorithm 1 can be written a bit more more simply; this is due to the fact that $F'(x)$ is no longer a matrix, but simply the real number $f'(x)$. Solving for h is then just division by $f'(x)$, as long as $f'(x) \neq 0$. In other words, Step 2) in Algorithm 1 becomes simply:

$$h = -\frac{f(x^k)}{f'(x^k)}.$$

Algorithm 1 Newton's Method

```
• Pick an initial guess $x^0 \in \mathbb{R}^n$.
• For $k = 0, 1, 2, 3, \ldots$ do:
    1) $F'(x^k)h = -F(x^k)$.        (Solve for $h$)
    2) $x^{k+1} = x^k + h$.          (Update $x$)
• End For
```

1.2.5 Convergence Rates of Sequences in $\mathbb{R}^n$

One of the central problems we run into when building computational algorithms in finance and other application areas concerns the behavior of sequences that our algorithms will generate; we view these sequences as having the form:

$$\{x^k\}_{k=1}^{\infty}, \qquad x^k \in \mathbb{R}^n.$$

An algorithm we design (such as Newton's method that we derived above) is designed to generate a (countably) infinite sequence of vectors $x^k \in \mathbb{R}^n$, and we need to make sure that we design the algorithm so that this sequence converges to the solution of whatever problem we are trying to solve. What our algorithm does is generate an infinite sequence $\{x^k\}_{k=1}^{\infty}$, and our hope is that it *converges* to x (which we denote: $\lim_{k\to\infty} x^k = x$), such that $F(x) = 0$. In a practical setting, one hopes to stop the algorithm very early in order to make the algorithm useful; this will be possible only if the algorithm *converges rapidly.* The fact that Newton's method has this property is why it forms the basis of many (most) modern algorithms for solving nonlinear equations, including those that arise as part of solving optimization problems.

Therefore, a practical concern for us when building algorithms is not only that they converge (produce convergent sequences), but that the convergence rate is reasonably fast. Otherwise, the algorithm may not be efficient enough to be useful. The *rate of convergence* of such sequences can be characterized by the following inequality:

$$\|x - x^{k+1}\| \le C_k \|x - x^k\|^p.$$

If $p = 1$ and $0 \le C_k < 1$, we say that the convergence rate is *linear.* If $p = 2$ and $0 \le C_k < 1$, we say that the convergence rate is *quadratic.* If $p = 1$ and $\lim_{k\to\infty} C_k = 0$, we say that the convergence rate is *super-linear.* One can define other similar types of convergence rates, but these three are the most important rates that arise in standard algorithms for nonlinear equations and optimization problems.

1.3 Differential Equations

We now discuss a small amount of additional material from a typical second- or third-year undergraduate course on ordinary and partial differential equations (ODE and PDE). We will need at least some basic familiarity with terminology and notation for ODE and PDE to understand how to use certain computational techniques in finance, such as numerical methods for Black-Scholes models. (We do not assume that the reader has had any previous exposure to either ODEs or PDEs.)

1.3.1 Ordinary Differential Equations

Mathematical modeling in finance, as in areas of science and engineering, often involves the use of ordinary and/or partial differential equations. An *ordinary differential equation (ODE)* is an equation that involves an unknown function, in which derivatives of the unknown function appear. The goal is to "solve" the ODE to determine the unknown function. Here are some examples of ODEs for determining an unknown function $y\colon \mathbb{R} \to \mathbb{R}$:

$$y' = 2x^2 + yx + 4, \tag{1.14}$$
$$y'' = 5y' + 2y - 2, \tag{1.15}$$
$$y' = y^3 - 3y^2. \tag{1.16}$$

The first equation (1.14) is an example of a *first-order linear* ODE; the unknown appears *linearly* throughout the equation, and the highest-order derivative that appears is the first derivative. The second equation (1.15) is an example of a *second-order* linear ODE; the unknown again appears *linearly* throughout the equation, and the highest-order derivative that appears is the second derivative. The third equation (1.16) is an example of a first-order *nonlinear* ODE; the unknown now appears *nonlinearly* in the equation, and the highest-order derivative that appears is the first derivative. The first equation is an example of a *non-autonomous* ODE, because the independent variable x appears explicitly in the ODE. On the other hand, the second and third equations are examples of *autonomous* ODEs, because the independent variable x does not appear explicitly in either equation.

When attempting to solve an ODE, one finds that the ODE itself is in fact insufficient information for determining the solution; one needs additional *side conditions* to make the ODE a *well-posed problem* (we discuss the requirements of a well-posed mathematical problem later in this section). The appropriate side conditions depend on what the ODE is modeling; for example, if the independent variable is *time*, and the ODE is first-order, then the appropriate side condition is an *initial condition* for y, namely its value at the initial time $t = 0$:

$$y' = f(t, y), \quad t \in (0, T], \tag{1.17}$$
$$y(0) = y_0. \tag{1.18}$$

The two equations (1.17) and (1.18) together are referred to as an *initial value problem in ordinary differential equations*, or *IVP in ODE* for short. If the independent variable is *space*, which typically goes along with a second-order ODE, then the appropriate side condition is called a *boundary condition*, and together with the ODE this defines a *boundary value problem in ordinary differential equations*, or *BVP in ODE* for short:

$$y'' = f(x, y, y'), \quad x \in (a, b), \tag{1.19}$$
$$y(a) = y_a, \tag{1.20}$$
$$y(b) = y_b. \tag{1.21}$$

The first equation (1.19) is the ODE, and the remaining two equations (1.20) and (1.21) are the boundary conditions. The IVP and BVP problems are "complete" problem specifications, in that they have the potential to be *well-posed mathematical problems*, which we define below.

1.3.2 Partial Differential Equations

In our discussion of calculus, we first considered real-value functions of a single real variable, $f\colon \mathbb{R} \to \mathbb{R}$, and then considered functions of several variables, $F\colon \mathbb{R}^n \to \mathbb{R}^m$. ODEs are equations for the first type of function, whereas PDEs are equations for the second type of function. Here, we will be mainly interested in PDE problems involving functions of the form $f\colon \mathbb{R}^n \to \mathbb{R}$, in other words, real-value functions of several independent variables. These are the same types of functions we will also run into as objective functions in optimization. PDE problems that arise in mathematical modeling can be generally thought of as being one of three types: elliptic, parabolic, and hyperbolic. The terminology (elliptic, parabolic, hyperbolic) comes from a technical condition that is used to separate PDEs into these three types; the condition is related to how we classify conic sections in Calculus. There are two main motivations for making the distinction between the three types:

1) Each type represents a mathematical model of three very different physical phenomena: equilibrium, diffusion, and wave propagation.
2) Each type needs a different set of side conditions in order to "complete" each problem; this is critical so that we can produce a well-posed mathematical problem.

We briefly describe these three types of PDE below.

Elliptic Equations. Elliptic equations represent models of stationary (time-independent) phenomena, such as electrostatics, elastostatics, and so forth. The BVP in ODE we discussed above is the simplest case, and can be viewed as a one-dimensional elliptic equation. The simplest and most common elliptic equation is known as the *Poisson Equation*:

$$-\Delta u = f, \quad x \in \Omega \subset \mathbb{R}^d, \tag{1.22}$$
$$u = 0, \quad x \text{ on } \partial\Omega. \tag{1.23}$$

Here, $\Omega \subset \mathbb{R}^d$, known as the *spatial domain*, is the set over which the independent variable $x \in \mathbb{R}^d$ is allowed to range. (For example, Ω might be the sphere of radius one centered at the origin, known as the unit sphere, or it might be a cube.) The set $\partial\Omega$ is the boundary of Ω (e.g. the surface of the unit sphere, or the boundary of a cube). The unknown function u is a real-value function of the d variables $x = [x_1, \ldots, x_d]^T$, so can be viewed as a function of the form $u\colon \Omega \subset \mathbb{R}^d \to \mathbb{R}$. The function f appearing on the right in (1.22) is a *forcing function*, and is a function of the same form as u, so that $f\colon \Omega \subset \mathbb{R}^d \to \mathbb{R}$. The forcing function represents the particular mathematical model, and drives the behavior of the solution u. The first equation (1.22) is the *PDE*, and the second equation (1.23) is the *boundary condition*. The final symbol appearing in (1.22) that we need to define is the *Laplacean operator* Δ; it is just the multi-dimensional analogue of the second derivative appearing in our BVP in ODE (1.19) above:

$$\Delta u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}. \tag{1.24}$$

If there is only one independent variable, so that $\Omega = (a, b) \subset \mathbb{R}$, then the Laplacean is reduced to simply the second derivative in that one variable:

$$\Delta u = \frac{d^2 u}{dx^2} = u'', \tag{1.25}$$

and the Poisson equation (1.22)–(1.23) reduces to the following BVP in ODE:

$$u'' = f(x), \quad x \in (a, b), \tag{1.26}$$
$$u(a) = y_a, \tag{1.27}$$
$$u(b) = y_b. \tag{1.28}$$

Parabolic Equations. Parabolic equations represent models of diffusion (time-dependent) phenomena such as heat flow in a body. Parabolic and elliptic equations are very closely related; one can view an elliptic equation is the "limit" of an associated parabolic equation where the solution has reached *equilibrium*, and is no longer changing. The simplest and most common parabolic equation is known as the *Heat Equation*:

$$\begin{aligned} u_t - \Delta u &= f, && (t,x) \in (0,T) \times \Omega, && (1.29)\\ u(t,x) &= 0, && x \text{ on } \partial\Omega, && (1.30)\\ u(0,x) &= u_0(x), && x \in \Omega. && (1.31) \end{aligned}$$

As in the elliptic case, the spatial domain is $\Omega \subset \mathbb{R}^d$, but now we also have a *time domain*, which is $(0,T) \subset \mathbb{R}$. The unknown function u is still a real-value function, but it is now a function of both the d spatial variables $x = [x_1, \ldots, x_d]^T$ and the single time variable t:

$$u \colon [0,T] \times \Omega \to \mathbb{R}.$$

The first equation (1.29) is the *PDE*, the second equation (1.30) is a *boundary condition* analogous to what we had for the elliptic equation and our BVP in ODE, and the third equation (1.31) is an *initial condition* analogous to what we had for our IVP in ODE. In this sense, the parabolic equation combines the IVP and BVP problems we encountered for ODEs into a single problem.

Of the three types of equations, parabolic equations arguably play the most important role in finance; this is due to the fact that Black-Scholes models for standard options lead to parabolic equations.

Hyperbolic Equations. While parabolic equations will be our main focus later in the course due to the importance of Black-Scholes models, toward the end of the course we will examine Black-Scholes models for exotic options. This will give rise to more complex partial differential equations, including both hyperbolic equations and also potentially nonlinear equations. While elliptic and parabolic equations are closely connected as described above, hyperbolic equations, which model wave propagation phenomena such as electromagnetic radiation and gas dynamics, are fundamentally different from both elliptic and parabolic equations. They also need side conditions (again, a combination of both initial and boundary conditions) to "complete" the problem specification. The simplest and most common hyperbolic equation is known as the *Wave Equation*:

$$\begin{aligned} u_{tt} - \Delta u &= f, && (t,x) \in (0,T) \times \Omega, && (1.32)\\ u(t,x) &= u_d(t,x), && x \text{ on } \partial\Omega, && (1.33)\\ u(0,x) &= u_0(x), && x \in \Omega. && (1.34)\\ u_t(0,x) &= u_1(x), && x \in \Omega. && (1.35) \end{aligned}$$

One now needs two initial conditions (due to the appearance of two time derivatives in the PDE), and the boundary condition must satisfy certain compatibility conditions that do not appear in the parabolic case. We will discuss hyperbolic equations in a bit more detail when we discuss Black-Scholes models for exotic options.

1.3.3 Well-Posed Mathematical Problems and Their Solution

In our discussions of matrix equations and differential equations (both ODE and PDE), our interest was in "solving" the equations. What does that actually mean, and when is this actually possible? We certainly do not want to try to solve a problem that is not solvable, so we would prefer to know this fact before we expend the effort to try to do so. What if a problem has more than one solution? Is one of the solutions the "good" one and the other one the "bad" one? If we manage to find one of the solutions, how do we know which solution we have: the good one or the bad one?

Another troubling possibility is that any solution we come up with might be highly sensitive to error in the data, so that we never actually get the solution at all, at least in the sense of an exact solution. Even worse, using the solution in any "predictive" way, such as predicting whether a bridge will be able to sustain a certain weight, becomes potentially dangerous. The use of computers as part of the solution process makes this scenario unavoidable; computers do not actually work with sets like $\mathbb{R}$ and $\mathbb{R}^n$, but rather only with the subsets of these sets that are representable as floating point numbers of the particular precision that the computer hardware supports. This means there is *always* error in the problem data, even if it is very small.

How do we manage these various difficulties?

To begin, let us characterize the type of mathematical problem that we would ideally like to be faced with, which we is referred to as a *Well-Posed (Mathematical) Problem.* A Well-Posed Problem has the following three features:

1) Existence: The problem has a solution.
2) Uniqueness: The solution is unique.
3) Continuous Dependence: The solution depends continuously on the problem data.

The first property is that the solution exists; the second property is that there is only one such solution. The third property says that if the data has error, then the solution should depend in a continuous way on that error.

As an example, let us consider the case of a matrix system $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is nonsingular. Since A is nonsingular, we know that the linear system $Ax = b$ has a unique solution $x = A^{-1}b$, so that both the first and the second properties hold. That the third condition holds will follow from the simple inequality that we derived earlier when we discussed condition numbers:

$$\|x\| = \|A^{-1}b\| \leq \|A^{-1}\|\|b\|. \tag{1.36}$$

The argument that x is a continuous function of b is as follows. Let $\epsilon > 0$ be given. Choose $\delta = \epsilon/\|A^{-1}\|$. Let $x = A^{-1}b$ and $\tilde{x} = A^{-1}\tilde{b}$. If $\|b - \tilde{b}\| < \delta$, then

$$\|x - \tilde{x}\| = \|A^{-1}b - A^{-1}\tilde{b}\| = \|A^{-1}(b - \tilde{b})\| \leq \|A^{-1}\|\|b - \tilde{b}\| < \delta\|A^{-1}\| = \epsilon.$$

In other words, a sufficiently small change in b will produce a small change in x.

Note that the third condition, namely *continuous dependence of the solution on the problem data*, allows for very complex solutions. For example, in the case of matrix equations, we derived the following inequality earlier involving the condition number:

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A)\frac{\|\delta b\|}{\|b\|}. \tag{1.37}$$

Even if A is nonsingular, the condition number $\kappa(A) = \|A\|\|A^{-1}\|$ can be quite large. In this case, even though we showed above that the inequality (1.36) implies that x depends continuously on b, it still allows the relative change in x to be potentially quite large even for small change in b. For this reason, matrices A with large condition numbers are called *ill-conditioned matrices*, and the corresponding linear systems involving such A are called *ill-conditioned systems.* They are inherently sensitive to error in the data vector b (as well as in the entries of A itself), and it is generally very difficult to produce accurate solutions to ill-conditioned problems. This phenomena of extreme sensitivity to changes in the problem data can potentially occur in every type of mathematical model, and it represents a fundamental limit to how useful the model can be in predicting the behavior of the system it is modeling. The subject of *chaotic dynamical systems*, for example, concerns well-posed ODE (and PDE) systems which are nevertheless highly sensitive to errors in the side conditions such as initial data. Unfortunately, our best mathematical models of local and regional weather behavior can be shown to be chaotic dynamical systems of PDE, which is why atmospheric scientists are unable to predict the weather for any duration beyond the span of hours or a few days.

We make a few final comments about the *nature of the solution* to a problem. Sometimes it is sufficient to simply *know* that a problem as at least one solution, even if we do not have a way of actually writing that solution down as some mathematical expression (called "closed form"). In some cases, we would also want to know whether or not the solution is unique, again even if we cannot write the solution down. In different areas of mathematics, research mathematicians focus on problems of this type, and prove theorems about existence of solutions, and possibly theorems about uniqueness of solutions (or lack thereof), and then they move on to the next problem, leaving it to others to actually produce the solution if it is needed.

Unfortunately, for most of the problems arising in mathematical modeling, we can rarely write down the solution in closed form, even if it is a well-posed problem so that we know that a unique solution exists. This is true for nearly every nonlinear equation one encounters in ODE and PDE models, as well as in any type of non-trivial optimization problem. How does we proceed? Well, knowing that we have a well-posed problem to begin with is actually the best place to begin: If the problem is not known to be well-posed, then we had better check with a mathematician who works on such problems to confirm that it is. (If we can not establish that it is well-posed, we need to look for a better mathematical model!)

Once we have a well-posed mathematical model, the most effective approach in using the model that has been developed over the last half-century has been to design *approximation methods* and *numerical algorithms*, and along the way develop corresponding *approximation theory* and *algorithm convergence theory*, so that we know exactly what kind of errors we make in solving an approximate version of the problem, and so that we also know for what situations and how rapidly our algorithms will produce this approximation. *This is essentially the content of much of this course on computational finance.*

References

[1] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems.* John Wiley & Sons, Inc., New York, NY, fourth edition, 1977.

[2] I. Stakgold and M. Holst. *Green's Functions and Boundary Value Problems.* John Wiley & Sons, Inc., New York, NY, third edition, 2011.

[3] J. Stewart. *Calculus.* Cengage Learning, La Jolla, 7th edition, 2012.

[4] G. Strang. *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, San Diego, CA, 3rd edition, 1988.