B. Herman and J. Roberts

Nuclear Reactor Core Methods

June 4, 2012

Springer

Use the template dedic.tex *together with the Springer document class SVMono for monograph-type books or SVMult for contributed volumes to style a quotation or a dedication at the very beginning of your book in the Springer layout*

Foreword

Use the template *foreword.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your foreword in the Springer layout.

The foreword covers introductory remarks preceding the text of a book that are written by a *person other than the author or editor* of the book. If applicable, the foreword precedes the preface which is written by the author or editor of the book.

Place, month year *Firstname Surname*

Preface

Use the template *preface.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your preface in the Springer layout.

A preface is a book's preliminary statement, usually written by the *author or editor* of a work, which states its origin, scope, purpose, plan, and intended audience, and which sometimes includes afterthoughts and acknowledgments of assistance.

When written by a person other than the author, it is called a foreword. The preface or foreword is distinct from the introduction, which deals with the subject of the work.

Customarily *acknowledgments* are included as last part of the preface.

Place(s),	*Firstname Surname*
month year	*Firstname Surname*

Acknowledgements

Use the template *acknow.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) if you prefer to set your acknowledgement section as a separate chapter instead of including it as last part of your preface.

Contents

Part I Fundamentals

Part II Reactor Statics

Part III Reactor Dynamics

Acronyms

Use the template *acronym.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your list(s) of abbreviations or symbols in the Springer layout.

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced `description` environment.

PWR	Pressurized Water Reactor
BWR	Boiling Water Reactor
ANM	Analytic Nodal Method

Part I
Fundamentals

Lorem ipsum...

Chapter 1
Neutron Transport Equation

Abstract In this chapter, we introduce the *transport equation*, a linearized form of the Boltzmann equation suitable for describing neutral particles in nuclear systems. Our goal is to define several fundamental quantities needed to describe neutron populations in a reactor in terms of all phase space variables. In the next chapter, we will derive rather formally the *diffusion equation*, which is the foundation for much of the analysis remaining in this course.

1.1 A Look Back

At this point, we assume the reader has a solid understanding of several fundamentals, as covered in Chapters 1-4 in Lewis. This includes a basic familiarity with the various nuclear reactions of interest, such as scattering, fission and other resonance reactions, and decay modes, along with various neutron sources. Furthermore, the reader should understand microscopic and macroscopic cross-sections, their energy dependence, and where each is used. The basics of energy spectrum calculations should be familiar, as should be the path to multigroup cross-sections via condensation. Finally, the reader should have a basic appreciation for the various nuclear systems of interest and gross system features that fundamentally affect our treatment of the physics.

Our general goal now is to learn how to bring together these concepts to model nuclear reactors, and describing the population of neutrons in a reactor is our first step toward reaching this goal.

1.2 Transport Theory

Transport theory aims to describe mathematically the movement (*i.e* “transport”) of particles as they traverse a medium. For example, we might describe the transport of high energy gammas through a lead shield, or the movement of neutrons through uranium dioxide pellets. We might also describe the movement of particles in a dense gas as they navigate through a medium consisting of the gas itself.

In all cases, transport theory describes such processes in an *average* sense. For instance, we do not compute the individual trajectories of neutrons in a reactor via transport theory. That, instead, would require molecular dynamics, in which Newton's equation of force is solved for the many-bodied problem of all neutrons in the vicinity of interest (an essentially impossible problem), or perhaps Monte Carlo methods, where a sample of individual particles are tracked to approximate ensemble averages (a difficult, but tractable problem). Hence, the quantities we shall compute using the equations of transport theory or approximations thereof should be recognized as expected and not exact values.

1.3 Fundamental Quantities

We begin by defining several fundamental quantities, all of which apply to any neutral particle field. The most fundamental quantity of interest is the *phase space density*, knowledge of which we can use to define essentially all other relevant quantities:

$\bar{n}'''(\mathbf{r},\mathbf{v},t)d^3rd^3v \equiv$ expected number of particles in d^3r about $\mathbf{r}$ with velocity dv about $\mathbf{v}$ at time t.

Note, in almost any other text, the density is written simply as n, without bars or primes. We keep them for consistency with Lewis.

It is common to break the velocity into its scalar (speed) and vector (direction) components. The scalar component is recast in the energy variable via $E = mv^2/2$, and the direction vector is defined $\hat{\Omega} = \mathbf{v}/|\mathbf{v}|$. The phase space density can then be rewritten as

$\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)d^3rd\Omega dE \equiv$ expected number of particles in d^3r about $\mathbf{r}$ going in the directions $d\Omega$ about $\hat{\Omega}$ with energy dE about E at time t.

In this form, $\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)$ is often referred to as the *angular density*. We shall work exclusively with the angular density, since many expressions are most conveniently cast in terms of the energy E.

Figure 1.1 depicts a schematic of the phase space used in terms of the position $\mathbf{r}$ and direction $\hat{\Omega}$. The position vector is further broken down into the polar angle θ and azimuthal angle ϕ. The differential solid angle element $d\Omega$ is also shown, and can be expressed in terms of θ and ϕ via

$$d\Omega = \sin\theta d\theta d\phi \,.$$

In reactor physics, we are frequently interested in volumetric reaction rates (for computing power densities, radiation damage, and so forth). Eq. (3.10) of Lewis

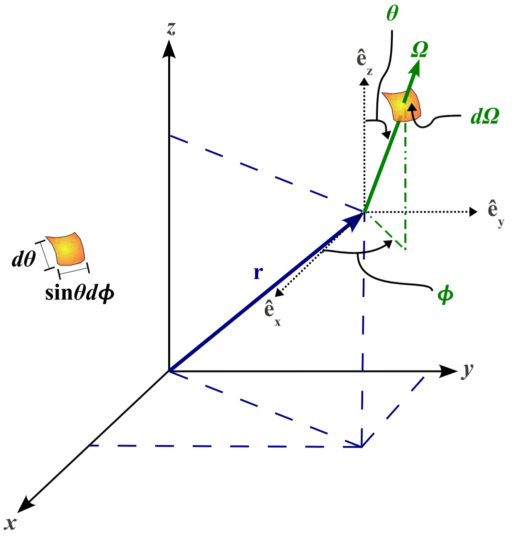


Fig. 1.1 Schematic of Phase Space.

defines the flux spectrum as

$$\varphi(E) = v(E)\bar{n}'''(E)\,,$$

where $v(E)$ is the speed at energy E. Generalizing to all of phase space, we define the *angular flux*

$$\psi(\mathbf{r},\hat{\Omega},E,t) = v(E)\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)\,, \tag{1.1}$$

and *scalar flux*

$$\varphi(\mathbf{r},E,t) = \int_{4\pi} d\Omega\,\psi(\mathbf{r},\hat{\Omega},E,t)\,. \tag{1.2}$$

A quantity closely related to the angular flux is the *angular current density*,

$$\mathbf{j}(\mathbf{r},\hat{\Omega},E,t) = \hat{\Omega}\,\psi(\mathbf{r},\hat{\Omega},E,t)\,. \tag{1.3}$$

Consider a surface S with outward normal $\hat{e}_S$, and let $\mathbf{S} = \hat{e}_S S$. Then we have

$\mathbf{j}(\mathbf{r},\hat{\Omega},E,t)\cdot\mathbf{S}d\Omega dE \equiv$ expected number of particles that cross an area S per second going in the directions $d\Omega$ about $\hat{\Omega}$ with energy dE about E at time t.

A *partial current density* can also be defined with respect to $\mathbf{S}$:

$J_{\pm}(\mathbf{r},E,t)SdE = \pm\int_{\hat{\Omega}\cdot\hat{e}_S \gtrless 0} d\Omega dE\mathbf{S}\cdot\mathbf{j}(\mathbf{r},\hat{\Omega},E,t) \equiv$ the rate at which particles in dE about E flow through S in the outward (+) or inward (-) direction at time t.

Similarly, the *current density* or just *current* is defined by integrating the angular current density over all directions, or

$$\mathbf{J}(\mathbf{r},E,t) = \int_{4\pi} d\Omega \mathbf{j}(\mathbf{r},\hat{\Omega},E,t)\,. \tag{1.4}$$

One way to think of the current is as the average direction of flow (the vector part) and the net number that flow (magnitude) in that direction at a particular point in phase space. More explicitly, we have

$\mathbf{J}(\mathbf{r},E,t)\cdot\mathbf{S}dE =$ net number of particles passing outward through S per second with energy dE about E at time t

From our definition of partial currents, the net current passing outward must also be $J_+ - J_-$, yielding the useful identity

$$\mathbf{J}(\mathbf{r},E,t)\cdot\hat{e}_s = J_+(\mathbf{r},E,t) - J_-(\mathbf{r},E,t)\,. \tag{1.5}$$

1.4 A General Transport Equation

Consider an arbitrary volume V with a surface S. Our goal is to represent the time rate of change of the neutron density $\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)$ within the volume. Neglecting external forces, the only factors affecting the density are collisions within the volume that change a particle's velocity, the streaming of particles through S into and out of the V, and any internal source of particles. This simple balance can be expressed mathematically as

$$\begin{aligned}\overbrace{\frac{\partial}{\partial t}\left(\int_V d^3r\,\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)\right)}^{\text{total rate of change of } \bar{n}''' \text{ in } V} &= -\overbrace{\int_S dS\hat{e}_S\cdot\mathbf{j}(\mathbf{r},\hat{\Omega},E,t)}^{\text{streaming rate}} + \overbrace{\int_V d^3r\left(\frac{\partial\bar{n}'''}{\partial t}\right)_{\text{coll}}}^{\text{collision rate}} \\ &\quad + \underbrace{\int_V d^3r\,S(\mathbf{r},\hat{\Omega},E,t)}_{\text{source emission rate}}\,,\end{aligned} \tag{1.6}$$

where S represents all sources inside the volume and $(\partial\bar{n}'''/\partial t)_{\text{coll}}$ is the time rate of change due to collisions, the specific form of which will be discussed below. Note the minus sign on the surface integral, often called the streaming term. Since the integral describes the net rate of neutrons going *out* of the surface, we negate it so that a positive net rate directed inward is a positive contribution to the total time rate of change of $\bar{n}'''$ in V.

Eq. 1.6 gives us a simple relation in terms of both volume and surface integrals. Our life is always easiest if we have the same integration on both sides. By the divergence (or Gauss') theorem, we can rewrite the streaming term

$$\int_S dS\hat{e}_S\cdot\mathbf{j}(\mathbf{r},\hat{\Omega},E,t)=\int_V d^3r\nabla\cdot\mathbf{j}(\mathbf{r},\hat{\Omega},E,t)\,. \tag{1.7}$$

Since ∇ acts on $\mathbf{r}$ and not $\hat{\Omega}$, we note $\nabla\cdot\mathbf{j}=\nabla\cdot(\hat{\Omega}\bar{n}''')=\hat{\Omega}\cdot\nabla\bar{n}'''+\overbrace{\bar{n}'''\nabla\cdot\hat{\Omega}}^{\to 0}=\hat{\Omega}\cdot\nabla\bar{n}'''$. Hence, the streaming term becomes

$$\int_V d^3r\nabla\cdot\mathbf{j}(\mathbf{r},\hat{\Omega},E,t)=\int_V d^3r\,\hat{\Omega}\cdot\nabla\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)\,. \tag{1.8}$$

For a constant volume, $(\partial/\partial t)\int_V d^3r\bar{n}'''=\int_V d^3(\partial\bar{n}'''/\partial t)$, and so our balance equation can be rewritten as

$$\begin{aligned}\overbrace{\left(\int_V d^3r\frac{\partial}{\partial t}\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)\right)}^{\text{total rate of change of } n \text{ in } V}=-\overbrace{\int_V d^3r\hat{\Omega}\cdot\nabla\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)}^{\text{streaming rate}}+\overbrace{\int_V d^3r\left(\frac{\partial\bar{n}'''}{\partial t}\right)_{\text{coll}}}^{\text{collision rate}}\\+\underbrace{\int_V d^3r\,S(\mathbf{r},\hat{\Omega},E,t)}_{\text{source emission rate}}\,.\end{aligned} \tag{1.9}$$

For an arbitrary volume V, the integrands of Eq 1.9 must vanish, yielding a general transport equation:

$$\frac{\partial}{\partial t}\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)=-\hat{\Omega}\cdot\nabla\bar{n}'''(\mathbf{r},\hat{\Omega},E,t)+\left(\frac{\partial n}{\partial t}\right)_{\text{coll}}+S(\mathbf{r},\hat{\Omega},E,t)\,. \tag{1.10}$$

1.5 Neutron Transport

The total volumetric collision rate at a particular point in phase space and time is simply

$$R_{\text{coll}}(\mathbf{r},\hat{\Omega},E,t)=\psi(\mathbf{r},\hat{\Omega},E,t)\Sigma_t(\mathbf{r},E)\,. \tag{1.11}$$

Then $R_{\text{coll}}(\mathbf{r},\hat{\Omega},E,t)d^3rd\Omega dE$ is the rate at which neutrons beginning in the differential phase space volume $d^3rd\Omega dE$ centered about $(\mathbf{r},\hat{\Omega},E,t)$ are sent to *any* other region of phase space by *any* possible mechanism. The time rate of change due to collisions is thus

$$\left(\frac{\partial n}{\partial t}\right)_{\text{coll}}=-\psi(\mathbf{r},\hat{\Omega},E,t)\Sigma_t(\mathbf{r},E)\,. \tag{1.12}$$

The source term, S, is in general a combination of several contributions, including in-scatter, fission, and external sources. We treat these individually.

When a neutron at one energy and angle scatters, it must end up at a different energy and different angle (or else it hasn't scattered). Then the source contribution in one part of phase space due to scattering events in another is defined formally as

$$S_s(\mathbf{r},\hat{\Omega},E,t) = \int_0^\infty dE' \int_{4\pi} d\Omega' \Sigma_s(\mathbf{r},\hat{\Omega}\cdot\hat{\Omega}',E' \to E)\psi(\mathbf{r},\hat{\Omega}',E',t)\,. \tag{1.13}$$

The volumetric rate of fission neutron production in a system is defined as

$$s_f'''(\mathbf{r},t) = \int_0^\infty dE' \nu(E')\Sigma_f(\mathbf{r},E')\varphi(\mathbf{r},E',t)\,, \tag{1.14}$$

Recall that fission neutrons appear at a variety of energies, the probability of which is denoted $\chi(E)$. Further recall that fission neutrons are essentially emitted isotropically; that is, the outgoing neutrons exhibit no "memory" of the incident neutron. Therefore, the fission source can be written

$$S_f(\mathbf{r},\hat{\Omega},E,t) = \frac{\chi(E)}{4\pi}\int_0^\infty dE' \nu(E')\Sigma_f(\mathbf{r},E')\varphi(\mathbf{r},E',t)\,. \tag{1.15}$$

However, Eq. 1.15 is only partially accurate. Neutrons born from fission come in two distinct flavors: those emitted *promptly*, and those emitted after some *delay*. One distinguishes between the two through use of the average number of prompt neutrons emitted per fission, $\nu_p(E)$ and a prompt fission neutron energy spectrum, $\chi_p(E)$. We'll use a special treatment for delayed neutrons in a later chapter. Worth noting is that ν for delayed neutrons is less than $\nu_p/100$ and that the associated χ-spectrum is softer but more erratic.

For now, we define a *prompt fission source*

$$S_{fp}(\mathbf{r},\hat{\Omega},E,t) = \frac{\chi_p(E)}{4\pi}\int_0^\infty dE' \nu_p(E')\Sigma_f(\mathbf{r},E')\varphi(\mathbf{r},E',t)\,, \tag{1.16}$$

and denote the *delayed fission source* by $S_{fd}(\mathbf{r},\hat{\Omega},E,t)$ where applicable.

Finally, one can include external sources, represented as $S_e(\mathbf{r},\hat{\Omega},E,t)$, and substituting this along with Eqs. 1.13, 1.16, and the delayed fission source into Eq. 1.10, we arrive at the canonical *neutron transport equation*:

$$\begin{aligned}\frac{1}{v}\frac{\partial\psi}{\partial t} &+ \hat{\Omega}\cdot\nabla\psi + \Sigma_t(\mathbf{r},E)\psi(\mathbf{r},\hat{\Omega},E,t) = \\ &+ \int_0^\infty dE' \int_{4\pi} d\Omega' \Sigma_s(\mathbf{r},\hat{\Omega}\cdot\hat{\Omega}',E' \to E)\psi(\mathbf{r},\hat{\Omega}',E',t) \\ &+ \frac{\chi_p(E)}{4\pi}\int_0^\infty dE' \nu_p(E')\Sigma_f(\mathbf{r},E')\varphi(\mathbf{r},E',t) \\ &+ S_{fd}(\mathbf{r},\hat{\Omega},E,t) + S_e(\mathbf{r},\hat{\Omega},E,t)\,.\end{aligned} \tag{1.17}$$

1.6 Assumptions for the Neutron Transport Equation

In writing down Eq. 1.17 as we have, a number of assumptions have been made explicitly or implicitly. These include:

1. The neutron density is large so that it makes sense to be computing for mean values (which is all transport equations can provide).
2. The neutrons are point particles, meaning that wave effects are insignificant.
3. Collisions are well-defined, two-body interactions that occur instantaneously. Delayed neutrons from fission, which will be covered in a later chapter, are a notable exception and deserve special treatment.
4. Between collisions, neutrons stream with a constant velocity.
5. Neutron-neutron interactions are negligible, which implies that while the population is large enough to make averages meaningful, it's not so large that neutron-neutron collisions become relevant.
6. The properties of the medium are assumed known and time-independent. Burnup in a nuclear reactor is an exception, and requires special treatment.
7. The medium is taken to be isotropic, meaning the cross-sections do not depend on the neutron direction of flight. Exceptions include crystalline materials, for which the cross-sections can change considerably depending on the direction of flight. For example, this is true for the graphite structures found in very high temperature reactor (VHTR) designs, and proper treatment of the angular dependence is important.

1.7 Further Reading

See Duderstadt and Martin [**?**], Bell and Glasstone [**?**], and Duderstadt and Hamilton [**?**].

Chapter 2
Neutron Diffusion Equation

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

2.1 Continuous Energy Diffusion Equation

This section will contain the derivation of the continuous form of the diffusion equation from the neutron transport equation.

Chapter 3
Multigroup Neutron Diffusion Equation

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

3.1 Derivation of Multigroup Diffusion Equation

This section will contain the derivation of the multigroup diffusion equation from the continuous energy diffusion equation

Chapter 4
Finite Difference Methods

4.1 Taylor Series

The finite difference method relies heavily on the mathematical concept of Taylor Series. If we take a function, $f(x)$, the independent variable x can be discretized into many points as shown in Figure _. If the value of the function is known at x_i, the value at x_{i+1} can be determined by a Taylor series expansion at x_i,

$$f\left(x_{i+1}\right) = f\left(x_i\right) + f'\left(x_i\right)h + \frac{f''\left(x_i\right)}{2!}h^2 + \frac{f^{(3)}\left(x_i\right)}{3!}h^3 + \cdots + \frac{f^{(n)}\left(x_i\right)}{2!}h^n + \cdots \quad (4.1)$$

In Eq. (4.1), $f^{(3)}$ represents the n-th derivative of the function and h is the spacing between points, $h = x_{i+1} - x_i$.

The expansion shown above is exact if the number of terms in the Taylor series expansion is taken to infinity. Of course, this is not practical for computational methods and therefore we truncate the series at a finite number of terms. The error present caused by the truncation is known as truncation error. Instead of representing the full Taylor expansion of a function, we will truncate the expression after a few number of terms and repesent the truncation error with $\mathscr{O}\left(h^n\right)$. In this representation of the truncation error, n represents the order of convergence. Order of convergence means that as the grid is refined by a factor of two for example, the truncation error will reduce on the order of 2^n. This does not imply that one method is better than the order, just merely a concept of convergence rate due to truncation effects. Linear convergence is when $n = 1$, quadratic when $n = 2$ and cubic when $n = 3$. For example, if we expand a function to second order, we would rewrite Eq. (4.1) this as

$$f\left(x_{i+1}\right) = f\left(x_i\right) + f'\left(x_i\right)h + \frac{f''\left(x_i\right)}{2!}h^2 + \mathscr{O}\left(h^3\right). \quad (4.2)$$

As we approximate differentials, we can keep track of this truncation error to determine order of convergence of our methods. This is one way to ensure that our discretization method and implementation of solution algorithms are correct.

4.2 Approximation of First Derivatives

There are many different approximations of differentials that can be constructed based on Taylor series. We will first consider the approximation of first order derivatives. The first approximation is a *first order forward difference* where we use information about a point just to the right, x_{i+1}, to infer the derivative at x_i. If we perform a Taylor expansion about point x_{i+1} to first order we get

$$f\left(x_{i+1}\right) = f\left(x_i\right) + f'\left(x_i\right)h + \mathscr{O}\left(h^2\right). \tag{4.3}$$

This equation can be solved for the derivative of the function at x_i

$$\boxed{f'_{for}\left(x_i\right) = \frac{f\left(x_{i+1}\right) - f\left(x_i\right)}{h} - \mathscr{O}\left(h\right)}, \tag{4.4}$$

where $f'_{for}\left(x_i\right)$ represents the first order forward difference approximation to the derivative at x_i.

The opposite approximation is to consider a point to the left, x_{i-1}, to infer the derivative at x_i, *known as the first order backward difference*. Here, we take a Taylor expansion to the left,

$$f\left(x_{i-1}\right) = f\left(x_i\right) - f'\left(x_i\right)h + \mathscr{O}\left(h^2\right). \tag{4.5}$$

Solving for the derivative we can arrive at

$$\boxed{f'_{bac}\left(x_i\right) = \frac{f\left(x_i\right) - f\left(x_{i-1}\right)}{h} + \mathscr{O}\left(h\right).} \tag{4.6}$$

Comparing Eqs. (4.4) and (4.6) we see that the formulation looks the same in that it is always the right point minus the left point in the numerator of the fraction. The only difference is the sign in the truncation error is reversed. Therefore, we can expect that one of these approximations will under-predict the true answer and the other one will over-predict. Again both of these methods are first order methods.

The last simple approximation of a first derivative is a *second-order central difference*. In this method we look at both left and right points. We can Taylor expand each of these to second order to get

$$f\left(x_{i+1}\right) = f\left(x_i\right) + f'\left(x_i\right)h + \frac{f''\left(x_i\right)}{2!}h^2 + \mathscr{O}\left(h^3\right) \tag{4.7}$$

$$f\left(x_{i-1}\right) = f\left(x_i\right) - f'\left(x_i\right)h + \frac{f''\left(x_i\right)}{2!}h^2 - \mathscr{O}\left(h^3\right). \tag{4.8}$$

Subtracting the x_{i-1} equation from the x_{i+1}, we are left with

$$f\left(x_{i+1}\right) - f\left(x_{i-1}\right) = 2f'\left(x_i\right)h + \mathscr{O}\left(h^3\right). \tag{4.9}$$

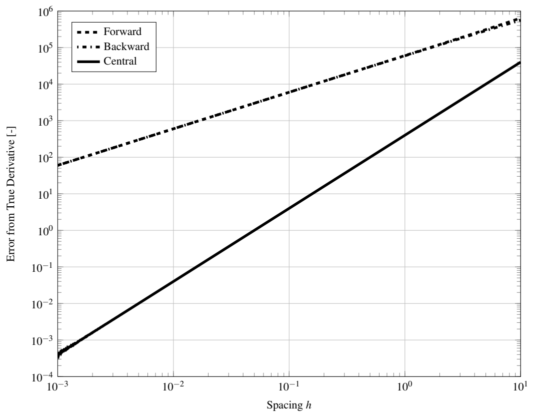


Fig. 4.1 Convergence rate of forward, backward and central difference approximations. The slope of the error as a function of mesh spacing is an estimate of the order of convergence of an approximation scheme.

Solving for the derivative at x_i we arrive at the second order central difference approximation

$$\boxed{f'_{cen}\left(x_i\right) = \frac{f\left(x_{i+1}\right) - f\left(x_{i-1}\right)}{2h} - \mathscr{O}\left(h^2\right).} \tag{4.10}$$

From the resulting expression, this approximation method does not depend on the value of the function at x_i and that the scheme is second order convergent.

Example - Order of Convergence First Derivative

As a simple example, we can approximate the derivative of the function, $f\left(x\right) = x^4$, at $x = 100$ with each of the above approximations. We can choose an array of spacing values between x_i and x_{i+1} and x_{i-1} and x_i. For each spacing value we compute the estimate of the derivative using the three approximations above. To characterize the error of each we find the absolute difference between the approximation and the true value of the derivative at $x = 100$. To infer the order of convergence, we can graph the errors as a function of spacing on a log-log scale. These convergence plots are shown in Fig. 4.1.

There are two distinct convergence trends present in Fig. 4.1. The curve with a slope of 1 on the log-log scale represents forward and backward finite difference approximations. This shows that these methods have linear convergence consistent with the truncation error. For the central difference approximation we predicted that it would have quadratic convergence. We can see from the plot that the magnitude of the slope is 2 on the log-log scale. MATLAB code to solve generate this plot is included below.

```
% Simple Approximation of First Order Derivative

% Function
f = @(x) x.^4;
```

```
% Analytical Derivative of Function
df_exact = @(x) 4*x.^3;

% Create a log space of h (width between x values)
h = logspace(-3,1,1000);

% Perform approximation with forward finite difference
df_for = (f(100+h) - f(100))./h;

% Perform approximation with backward finite difference
df_bac = (f(100) - f(100-h))./h;

% Perform approximation with central finite difference
df_cen = (f(100+h) - f(100-h))./(2*h);

% Compute Error
err_for = abs(df_for - df_exact(100));
err_bac = abs(df_bac - df_exact(100));
err_cen = abs(df_cen - df_exact(100));

% Plot Results
loglog(h,err_for,'k--','LineWidth',2);
hold on
loglog(h,err_bac,'k-.','LineWidth',2);
loglog(h,err_cen,'k-','LineWidth',2);
grid
grid minor
xlabel('x spacing [-]','LineWidth',2);
ylabel('Error from true derivative [-]');
legend('Forward','Backward','Central','Location','NorthWest');
```

4.3 Approximation of Second Derivatives

In nuclear reactor physics applications, we also need approximations for second derivatives. The only difference in these approximations is that more points to the left or right of x_i need to be included. For the *first order forward difference* approximation, we write two equations to second order. One equation representating a Taylor expansion to x_{i+1} and the other to x_{i+2},

$$f\left(x_{i+1}\right)=f\left(x_i\right)+f'\left(x_i\right)h+\frac{f''\left(x_i\right)}{2!}h^2+\mathscr{O}\left(h^3\right) \tag{4.11}$$

$$f\left(x_{i+2}\right)=f\left(x_i\right)+f'\left(x_i\right)(2h)+\frac{f''\left(x_i\right)}{2!}(2h)^2+\mathscr{O}\left(h^3\right). \tag{4.12}$$

Since we are approximating the second derivative, the first derivative needs to be canceled out. To cancel this term out, we multiply Eq. (4.11) by a 2 and subtract it from Eq. (4.12). The resulting expression is

$$f\left(x_{i+2}\right)-2f\left(x_{i+1}\right)=-f\left(x_{i}\right)+f''\left(x_{i}\right)h^{2}+\mathcal{O}\left(h^{3}\right). \tag{4.13}$$

The approximation of the second derivative for a first-order forward difference is therefore

$$\boxed{f''_{for}\left(x_{i}\right)=\frac{f\left(x_{i+2}\right)-2f\left(x_{i+1}\right)+f\left(x_{i}\right)}{h^{2}}-\mathcal{O}\left(h\right).} \tag{4.14}$$

For the *first-order backward finite difference* approximation of the second derivative we Taylor expand the function at x_{i-1} and x_{i-2}

$$f\left(x_{i-1}\right)=f\left(x_{i}\right)-f'\left(x_{i}\right)h+\frac{f''\left(x_{i}\right)}{2!}h^{2}-\mathcal{O}\left(h^{3}\right) \tag{4.15}$$

$$f\left(x_{i-2}\right)=f\left(x_{i}\right)-f'\left(x_{i}\right)\left(2h\right)+\frac{f''\left(x_{i}\right)}{2!}\left(2h\right)^{2}-\mathcal{O}\left(h^{3}\right). \tag{4.16}$$

Similar to the forward finite difference case, we must eliminate the first derivative term by multiplying Eq. (4.15) by 2 and subtract from Eq. (4.16). This results in the following expression:

$$f\left(x_{i-2}\right)-2f\left(x_{i-1}\right)=-f\left(x_{i}\right)+f''\left(x_{i}\right)h^{2}-\mathcal{O}\left(h^{3}\right). \tag{4.17}$$

The approximation of the second derivative for a first-order backward difference is therefore

$$\boxed{f''_{bac}\left(x_{i}\right)=\frac{f\left(x_{i}\right)-2f\left(x_{i-1}\right)+f\left(x_{i-2}\right)}{h^{2}}+\mathcal{O}\left(h\right).} \tag{4.18}$$

Lastly, the *second-order central difference* approximation to the second derivative can be derived by performing a Taylor expansion at x_{i-1} and x_{i+1} to fourth-order,

$$f\left(x_{i+1}\right)=f\left(x_{i}\right)+f'\left(x_{i}\right)h+\frac{f''\left(x_{i}\right)}{2!}h^{2}+\frac{f'''\left(x_{i}\right)}{3!}h^{3}+\mathcal{O}\left(h^{4}\right) \tag{4.19}$$

$$f\left(x_{i-1}\right)=f\left(x_{i}\right)-f'\left(x_{i}\right)h+\frac{f''\left(x_{i}\right)}{2!}h^{2}-\frac{f'''\left(x_{i}\right)}{3!}h^{3}+\mathcal{O}\left(h^{4}\right). \tag{4.20}$$

To eliminate the first derivate term, these two equations can be directly added together resulting in

$$f\left(x_{i+1}\right)+f\left(x_{i-1}\right)=2f\left(x_{i}\right)+f''\left(x_{i}\right)h^{2}+\mathcal{O}\left(h^{4}\right). \tag{4.21}$$

The approximation of the second derivative for a second-order central difference is

$$\boxed{f''_{cen}\left(x_{i}\right)=\frac{f\left(x_{i+1}\right)-2f\left(x_{i}\right)+f\left(x_{i-1}\right)}{h^{2}}-\mathcal{O}\left(h^{2}\right).} \tag{4.22}$$

The second-order central difference will be the main approximation we use for second order derivatives. This is mainly due to the fact that has quadratic convergence. The other reason is that for a given computational node in a reactor, we think of

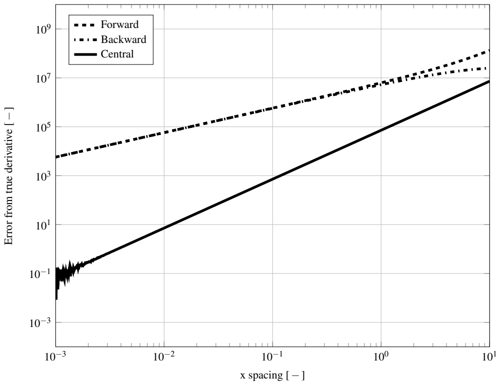


Fig. 4.2 Convergence rate of forward, backward and central difference approximations of a second derivative. The slope of the error as a function of mesh spacing is an estimate of the order of convergence of an approximation scheme.

leakage occuring to the left and to the right. This leakage term in represented with mathematically by a second derivative and by using the central difference approximation, we can couple to both the right and left nodes.

Example - Order of Convergence Second Derivative

In this example, we will verify the order of convergence for second derivatives with each of the approximations above. Similar to the previous example, we can approximate the second derivative of the function, $f(x) = 6x^6 + 4x^3 + 8x + 2$ at $x = 20$. We again, generate a vector of spacings to approximate the second derivative using Eqs. (4.14), (4.18) and (4.22).

A plot of the convergence rates of each of the second derivative approximations is shown in Fig. 4.2. From the figure we can see that the forward and backward approximations converge linearly while the central difference approximation converges quadratically. The is exactly the order of convergence values that were theoretically derived above. MATLAB code to generate Fig. 4.2 is included below.

```
% Simple Approximation of Second Order Derivative

% Function
f = @(x) 6*x.^6 + 4*x.^3 + 8*x + 2;

% Analytical Derivative of Function
df_exact = @(x) 36*x.^5 + 12*x.^2 + 8;
dff_exact = @(x) 180*x.^4 + 24*x;

% Create a log space of h (width between x values)
h = logspace(-3,1,1000);

% Perform approximation with forward finite difference
dff_for = (f(20+2*h) - 2*f(20+h) + f(20))./(h.^2);

% Perform approximation with backward finite difference
```

```
dff_bac = (f(20) - 2*f(20-h) + f(20-2*h))./(h.^2);

% Perform approximation with central finite difference
dff_cen = (f(20+h) -2*f(20) + f(20-h))./(h.^2);

% Compute Error
err_for = abs(dff_for - dff_exact(20));
err_bac = abs(dff_bac - dff_exact(20));
err_cen = abs(dff_cen - dff_exact(20));

% Plot Results
loglog(h,err_for,'k--','LineWidth',2);
hold on
loglog(h,err_bac,'k-.','LineWidth',2);
loglog(h,err_cen,'k-','LineWidth',2);
grid
grid minor
xlabel('x spacing [-]','LineWidth',2);
ylabel('Error from true derivative [-]');
legend('Forward','Backward','Central','Location','NorthWest');
```

4.4 Higher Order Finite Difference

In certain situations it may be advantageous to use a higher order finite difference method. In this section we will not derive any of these expressions, but just include them for completeness. In order to derive these expressions just Taylor expand the function to higher order and include more neighboring points.

Approximation of First Derivatives

- Second Order Forward Difference

$$f'\left(x_i\right) = \frac{-f\left(x_{i+2}\right) + 4f\left(x_{i+1}\right) - 3f\left(x_i\right)}{2h} + \mathscr{O}\left(h^2\right) \tag{4.23}$$

- Second Order Backward Difference

$$f'\left(x_i\right) = \frac{3f\left(x_i\right) - 4f\left(x_{i-1}\right) + f\left(x_{i-2}\right)}{2h} + \mathscr{O}\left(h^2\right) \tag{4.24}$$

- Fourth Order Central Difference

$$f'\left(x_i\right) = \frac{-f\left(x_{i+2}\right) + 8f\left(x_{i+1}\right) - 8f\left(x_{i-1}\right) + f\left(x_{i-2}\right)}{12h} + \mathscr{O}\left(h^4\right) \tag{4.25}$$

Approximation of Second Derivatives

- Second Order Forward Difference

$$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2} + \mathcal{O}\left(h^2\right) \tag{4.26}$$

- Second Order Backward Difference

$$f''(x_i) = \frac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2} + \mathcal{O}\left(h^2\right) \tag{4.27}$$

- Fourth Order Central Difference

$$f''(x_i) = \frac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{12h^2} + \mathcal{O}\left(h^4\right) \tag{4.28}$$

4.5 Nonuniform Spacing

When we derive the finite difference multigroup diffusion equation, we will include the option of having a nonuniform mesh spacing. This is straightforward, but affects the order of convergence of the approximations derived above. Here, we will re-derive the central finite difference approximations of the first and second derivatives. We can then study how the order of convergence is affected.

Beginning with the central difference approximation of the first derivative, we can write the Taylor expansion to the left and right of x_i,

$$f(x_{i+1}) = f(x_i) + f'(x_i)h_i + \frac{f''(x_i)}{2!}h_i^2 + \mathcal{O}\left(h_i^3\right) \tag{4.29}$$

$$f(x_{i-1}) = f(x_i) - f'(x_i)h_{i-1} + \frac{f''(x_i)}{2!}h_{i-1}^2 - \mathcal{O}\left(h_{i-1}^3\right). \tag{4.30}$$

In the above equations we define $h_i = x_{i+1} - x_i$ and $h_{i-1} = x_i - x_{i-1}$. We can subtract Eq. (4.30) from (4.30) to yield,

$$f(x_{i+1}) - f(x_{i-1}) = (h_i - h_{i-1})f'(x_i) + \left(h_i^2 - h_{i-1}^2\right)\frac{f''(x_i)}{2} + \mathcal{O}\left(h_i^3 + h_{i-1}^3\right). \tag{4.31}$$

Unlike in the uniform spacing case, the second derivative cannot be cancelled out when we consider nonuniform spacing. Therefore, the leading term for the truncation error is the second order term. We can rewrite the above equation so that

$$f(x_{i+1}) - f(x_{i-1}) = (h_i + h_{i-1})f'(x_i) + \mathcal{O}\left(h_i^2 - h_{i-1}^2\right). \tag{4.32}$$

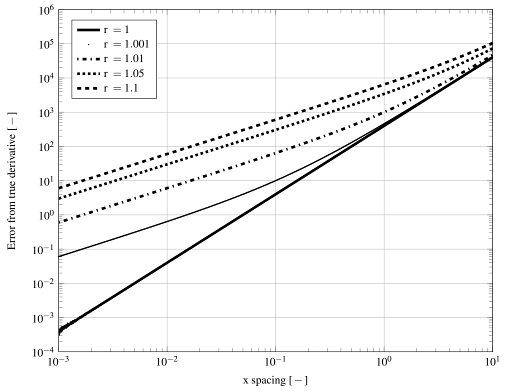


Fig. 4.3 Convergence central finite difference approximation of a first derivative for nonuniform spacing. The plot is shown for various values of a grid multiplier *r*. The slope of the error as a function of mesh spacing is an estimate of the order of convergence of an approximation scheme.

Solving for the first order derivative and dividing by the difference in mesh spacing, we are left with

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{h_i + h_{i-1}} + \mathscr{O}\left(\frac{h_i^2 - h_{i-1}^2}{h_i + h_{i-1}}\right). \tag{4.33}$$

We can simply the polynomial in the leading truncation error by factorizing the numerator,

$$\boxed{f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{h_i + h_{i-1}} + \mathscr{O}(h_i - h_{i-1})\,.} \tag{4.34}$$

From the result, we observe that the approximation has a linear convergence when the spacing is nonuniform. When the spacing is uniform we can readily see that the leading term in the error will disappear and we will be left with a second order approximation.

Example - Central Difference Approximation of First Derivative with Nonuniform Spacing

In this example, we extend the previous example of first derivative approximations to nonuniform spacing. Here, we only show the result for the central difference approximation. We again approximate the value of the first derivative of the function $f(x) = x^4$ at $x = 100$. We define another variable, *r* which is the ratio of the spacing on the right side of $x = 100$ to the left side of $x = 100$. We generate two vectors of spacing values, one for the left and one for the right side of $x = 100$ while keeping the ratio *r* the same at any given element of these two vectors.

Figure 4.3 shows the convergence rate of the central difference approximation of the first derivative. A range of grid multipliers, *r*, are shown. We can observe that depending on the value of *r*, the order of convergence ranges from linear to qudratic. The convergence is of course purely quadratic when the grid multiplier is unity which is for a uniform grid. We can also observe that as the grid multiplier increases the starting error is much larger and shows less and less quadratic convergence. On the other hand, as r approaches unity, more of the range is governed by quadratic convergence, but does eventually turn into linear. MATLAB code to generate Fig. 4.3 is included below.

```
% Simple Approximation of First Order Derivative - Nonuniform Spacing

% Function
f = @(x) x.^4;

% Analytical Derivative of Function
df_exact = @(x) 4*x.^3;

% vector non-uniform grid multiplier
r = [1,1.001,1.01,1.05,1.1];

% preallocate error vector
err_cen = zeros(1000,length(r));

% begin loop around grid multipliers
for i = 1:length(r)

    % Create a log space of widths and a muliplier (width between x values)
    hl = logspace(-3,1,1000);
    hr = r(i)*hl;

    % Perform approximation with central finite difference
    df_cen = (f(100+hr) - f(100-hl))./(hl + hr);

    % Compute Error
    err_cen(:,i) = abs(df_cen - df_exact(100));

end

% Plot Results
figure1 = figure;
axes1 = axes('Parent',figure1,'ZMinorGrid','on','YScale','log',...
    'YMinorTick','on','XScale','log','XMinorTick','on');
box(axes1,'on');
grid(axes1,'on');
grid minor
hold(axes1,'all');
loglog1 = loglog(hl,err_cen,'Parent',axes1,'LineWidth',2,'Color',[0 0 0]);
set(loglog1(1),'DisplayName','r = 1');
set(loglog1(2),'MarkerSize',2,'Marker','.','LineStyle','none',...
    'DisplayName','r = 1.001');
set(loglog1(3),'LineStyle','-.','DisplayName','r = 1.01');
set(loglog1(4),'LineStyle',':','DisplayName','r = 1.05');
set(loglog1(5),'LineStyle','--','DisplayName','r = 1.1');
xlabel('x spacing [-]','LineWidth',2);
ylabel('Error from true derivative [-]');
legend1 = legend(axes1,'show');
set(legend1,'Location','NorthWest');
```

Similarly, we can derive the nonuniform spacing version of the central difference approximation of the second derivative. This approximation is the most common in finite difference approximaitons of the diffusion equation. To start the derivation we Taylor expand to the left and right of x_i to fourth order as before,

$$f\left(x_{i+1}\right)=f\left(x_i\right)+f'\left(x_i\right)h_i+\frac{f''\left(x_i\right)}{2!}h_i^2+\frac{f'''\left(x_i\right)}{3!}h_i^3+\mathscr{O}\left(h_i^4\right) \tag{4.35}$$

$$f\left(x_{i-1}\right)=f\left(x_i\right)-f'\left(x_i\right)h_{i-1}+\frac{f''\left(x_i\right)}{2!}h_{i-1}^2-\frac{f'''\left(x_i\right)}{3!}h_{i-1}^3+\mathscr{O}\left(h_{i-1}^4\right). \tag{4.36}$$

As with the uniform spacing example, we would like to cancel out the first derivative term and then solve for the second derivative. To do this we must divide Eq. (4.36) by h_i and Eq. (4.30) by h_{i-1}. This results in

$$\frac{f\left(x_{i+1}\right)}{h_i}=\frac{f\left(x_i\right)}{h_i}+f'\left(x_i\right)+\frac{f''\left(x_i\right)}{2}h_i+\frac{f'''\left(x_i\right)}{6}h_i^2+\mathscr{O}\left(h_i^3\right) \tag{4.37}$$

$$\frac{f\left(x_{i-1}\right)}{h_{i-1}}=\frac{f\left(x_i\right)}{h_{i-1}}-f'\left(x_i\right)+\frac{f''\left(x_i\right)}{2}h_{i-1}-\frac{f'''\left(x_i\right)}{6}h_{i-1}^2+\mathscr{O}\left(h_{i-1}^3\right). \tag{4.38}$$

Now we can see that if we add Eq. (4.38) and (4.38) the first derivative will cancel out. We can also see that after the addition, the third derivative term will remain, thus making it the leading term in the truncation error. These operations are reflected in the following equation:

$$\begin{aligned}\frac{f\left(x_{i+1}\right)}{h_i}+\frac{f\left(x_{i-1}\right)}{h_{i-1}}=\left(\frac{1}{h_i}+\frac{1}{h_{i-1}}\right)f\left(x_i\right)+\left(h_i+h_{i-1}\right)\frac{f''\left(x_i\right)}{2}\\+\mathscr{O}\left(h_i^2-h_{i-1}^2\right).\end{aligned} \tag{4.39}$$

Finally, we can solve for the second derivative and reduce the polynomial in the leading term of the truncation error,

$$\boxed{f''\left(x_i\right)=\frac{\frac{f\left(x_{i+1}\right)}{h_i}-\left(\frac{1}{h_i}+\frac{1}{h_{i-1}}\right)f\left(x_i\right)+\frac{f\left(x_{i-1}\right)}{h_{i-1}}}{\frac{h_i+h_{i-1}}{2}}-\mathscr{O}\left(h_i-h_{i-1}\right).} \tag{4.40}$$

We observe that the order of convergence is now linear for nonuniform spacing with the central difference approximation. Also, if the spacings are equivalent, we can readily see that Eq. (4.40) reduces to Eq. (4.22).

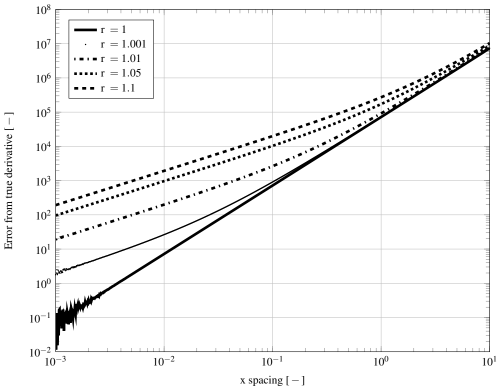


Fig. 4.4 Convergence central finite difference approximation of a second derivative for nonuniform spacing. The plot is shown for various values of a grid multiplier r. The slope of the error as a function of mesh spacing is an estimate of the order of convergence of an approximation scheme.

Example - Central Difference Approximation of Second Derivative with Nonuniform Spacing

We now show an example of approximating the second derivative of the function $f(x) = 6x^6 + 4x^3 + 8x + 2$ at $x = 20$ using the central difference approximation. We again generate curves for a fixed ratio between the grid size to the right and left of $x = 20$. The results are shown in Fig. 4.4. Similar to the nonuniform example of the first derivative, we see a range of convergence rates from linear to quadratic. MATLAB code to generate Fig. 4.4.

```
% Simple Approximation of Second Order Derivative - Nonuniform ↩
    Spacing

% Function
f = @(x) 6*x.^6 + 4*x.^3 + 8*x + 2;

% Analytical Derivative of Function
df_exact = @(x) 36*x.^5 + 12*x.^2 + 8;
dff_exact = @(x) 180*x.^4 + 24*x;

% vector non-uniform grid multiplier
r = [1,1.001,1.01,1.05,1.1];

% preallocate error vector
err_cen = zeros(1000,length(r));

% begin loop around grid multipliers
for i = 1:length(r)

    % Create a log space of widths and a muliplier (width ↩
        between x values)
    hl = logspace(-3,1,1000);
    hr = r(i)*hl;

    % Perform approximation with central finite difference
    dff_cen = (f(20+hr)./hr -(1./hr+1./hl).*f(20) + f(20-hl)./↩
        hl)./((hl + hr)/2);

    % Compute Error
    err_cen(:,i) = abs(dff_cen - dff_exact(20));

end

% Plot Results
figure1 = figure;
axes1 = axes('Parent',figure1,'ZMinorGrid','on','YScale','log'↩
     ,...
    'YMinorTick','on','XScale','log','XMinorTick','on');
box(axes1,'on');
grid(axes1,'on');
grid minor
hold(axes1,'all');
loglog1 = loglog(hl,err_cen,'Parent',axes1,'LineWidth',2,'Color↩
    ',[0 0 0]);
set(loglog1(1),'DisplayName','r = 1');
set(loglog1(2),'MarkerSize',2,'Marker','.','LineStyle','none'↩
     ,...
    'DisplayName','r = 1.001');
set(loglog1(3),'LineStyle','-.','DisplayName','r = 1.01');
set(loglog1(4),'LineStyle',':','DisplayName','r = 1.05');
set(loglog1(5),'LineStyle','--','DisplayName','r = 1.1');
xlabel('x spacing [-]','LineWidth',2);
ylabel('Error from true derivative [-]');
legend1 = legend(axes1,'show');
set(legend1,'Location','NorthWest');
```

4.6 Estimation of Order of Convergence

From all of the examples shown in this chapter, we can see the expected convergence rates for each type of approximations. To verify that codes are consistent with discretization methods, we should use a uniform mesh and determine the order of convergence. In all of the MATLAB examples, in order to determine the error and be able to plot the convergence rates, we needed to evaluate the *exact* derivative at the point of interest. In real applications this derivative will be unknown and so we need a different method in estimating the order of convergence, denoted as p.

In all of the boxed approximations above we always represent the derivative with an equals (=) sign. This is because we include the truncation error at the end of the expression. When we program these approximations into a code, we will not include this truncation error term. If we define $u \equiv f'(x)$ or $u \equiv f''(x)$ we can write any of the apprximations above as

$$u = u_h + \mathcal{O}\left(h^p\right). \tag{4.41}$$

Therefore, if u is the true answer that we are looking for, our code will give us the answer u_h. In order to determine the order of convergence, we need to write out the leading term in the truncation error. We know from the taylor series expansion it will have the form

$$\mathscr{O}(h^p) = \beta h^p + R, \tag{4.42}$$

where β is some constant multiplying the leading truncation term and R represents the rest of the truncation error. We now write the true answer u in terms of 3 uniform discretization grids with spacings of h, $2h$ and $4h$,

$$u = u_h + \beta h^p + R \tag{4.43}$$

$$u = u_{2h} + \beta' (2h)^p + R' \tag{4.44}$$

$$u = u_{4h} + \beta'' (4h)^p + R''. \tag{4.45}$$

The terms u_h, u_{2h} and u_{4h} are the results that we would get from output of a code. We see that we have 8 unknowns but only 3 equations. The first approximation that we can make is that the leading term in the truncation error dominates such that $R = R' = R'' = 0$. Next, we can assume that the multipliers, β are all equivalent, $\beta = \beta' = \beta''$. Now, we have reduce the number of unknowns to 3: u, β and p such that the system is closed.

To solve this system of equations for p we first eliminate u in Eqs. (4.43) and (4.44) by equating them,

$$u_h + \beta h^p = u_{2h} + 2^p \beta h^p. \tag{4.46}$$

In Eq. (4.46) we have already factored out the 2^p. Solving for βh^p we get,

$$\beta h^p = \frac{u_h - u_{2h}}{2^p + 1}. \tag{4.47}$$

Equation (4.47) can be substituted into Eqs. (4.44) and (4.45) yielding respectively,

$$u = u_{2h} + 2^p \left(\frac{u_h - u_{2h}}{2^p + 1} \right) \tag{4.48}$$

$$u = u_{4h} + 4^p \left(\frac{u_h - u_{2h}}{2^p + 1} \right) \tag{4.49}$$

To eliminate u such that p is the only unknown, we can equate the above expressions. We also split the 4^p into $2 \cdot 2^p$ so that

$$u_{2h} + 2^p \left(\frac{u_h - u_{2h}}{2^p + 1} \right) = u_{4h} + 2 \cdot 2^p \left(\frac{u_h - u_{2h}}{2^p + 1} \right). \tag{4.50}$$

We can rearrange the equation and factor out some common terms conveniently leaving,

$$u_{2h} - u_{4h} = 2^p \left(\frac{u_h - u_{2h}}{2^p + 1} \right) (2^p + 1). \tag{4.51}$$

We can then cancel out the common terms and solve for p,

$$\boxed{p = \frac{\log\left(\frac{u_{2h}-u_{4h}}{u_h-u_{2h}}\right)}{\log 2}}. \tag{4.52}$$

Example - Order of Convergence Estimation

For this example, we will estimate the order of convergences for the three approximations for the first derivative. However, this time we will assume that we do not know the value of the first derivative at $x = 100$. Instead, we will estimate the order of convergence with Eq. (4.52). We arbitrarily choose the grid spacing such that $h = 0.001$. This therefore gives us the other 2 spacings of 0.002 and 0.004. We evaluate the first derivative with each finite difference approximation and compute p, respectively. The results from the MATLAB code included below are

$$\begin{aligned} p_{for} &\approx 1.000289 \\ p_{bac} &\approx 0.997114 \\ p_{cen} &\approx 2.000043 \end{aligned}$$

From the results, we can see very good agreement with the theoretically derived convergence orders. In practice it is common that results contain an array of values and not just scalars a shown in all of these examples. The parameter u_h should then be taken as the norm of the results. Be cautious when doing this that different parts of your domain and solution may converge at different rates. This is especially true when including boundary conditions that are first order while the interior discretization is of second order. Thus, when performing the norm of the solution, just consider a subset of the results where you expect second order convergence.

```
% Simple Approximation of First Order Derivative
% Estimation of Convergence Order using 3 grid sizes

% Function
f = @(x) x.^4;

% Create a log space of h (width between x values)
h = 0.01;

% Perform approximation with forward finite difference
df_for(1) = (f(100+h) - f(100))./h;
df_for(2) = (f(100+2*h) - f(100))./(2*h);
df_for(3) = (f(100+4*h) - f(100))./(4*h);

% Perform approximation with backward finite difference
df_bac(1) = (f(100) - f(100-h))./h;
df_bac(2) = (f(100) - f(100-2*h))./(2*h);
df_bac(3) = (f(100) - f(100-4*h))./(4*h);
```

```
% Perform approximation with central finite difference
df_cen(1) = (f(100+h) - f(100-h))./(2*h);
df_cen(2) = (f(100+2*h) - f(100-2*h))./(2*2*h);
df_cen(3) = (f(100+4*h) - f(100-4*h))./(2*4*h);

% calculate order of convergence
p_for = log((df_for(2) - df_for(3))/(df_for(1) - df_for(2)))/↩
    log(2);
p_bac = log((df_bac(2) - df_bac(3))/(df_bac(1) - df_bac(2)))/↩
    log(2);
p_cen = log((df_cen(2) - df_cen(3))/(df_cen(1) - df_cen(2)))/↩
    log(2);

% display results
fprintf('Order of Convergence for Forward FD: %d\n',p_for);
fprintf('Order of Convergence for Backward FD: %d\n',p_bac);
fprintf('Order of Convergence for Central FD: %d\n',p_cen);
```

4.7 Finite Difference Multigroup Diffusion Equation

Chapter 5
Finite Volume Methods

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Chapter 6
Finite Element Methods

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Chapter 7
Stationary Iterative Methods

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

This chapter will contain the idea of iterative methods, and talk about Jacobi and Gauss - Siedel, example should be provided either for fission source iterations or energy group sweep. Also should include SOR method.

Chapter 8
Nonstationary Iterative Methods - Krylov Subspace Methods

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

REF: http://www.netlib.org/utk/papers/templates/node9.html

Intro to Krylov Methods Arnoldi Iterations - Gram-Schmidt etc?

Chapter 9
Conjugate Gradient

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

REF: http://www.netlib.org/utk/papers/templates/node9.html

Specific example - Conjugate Gradient

Chapter 10
GMRES

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the ’starred’ version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

REF: http://www.netlib.org/utk/papers/templates/node9.html

Specifically derive out GMRES with givens rotations. Preconditioning JFNK?

Chapter 11
Power Iteration

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Derive out the power iteration method and give example.

Chapter 12
Nonlinear Iteration

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the ’starred’ version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Newton Iteration - with GMRES JFNK

Chapter 13
Chebyshev Acceleration Method

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Chebyshev Acceleartion of Power iteration

Chapter 14
Time Stepping Methods

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Forward Euler (Explicit) Backward Euler (Implicit) Runge-Kutta (4th order mostly used in spatial kinetics) Adams-Moulton Adams-Bashforth

Part II
Reactor Statics

Lorem ipsum...

Chapter 15
Classical Nodal Methods - Flare Model

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the ’starred’ version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Summer course on nodal methods (Herman office)

Chapter 16
Analytic Nodal Method

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the ’starred’ version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

Derivation of Analytic Nodal Method with example code Smith Master Thesis

Chapter 17
Nodal Expansion Method

Abstract Each chapter should be preceded by an abstract (10–15 lines long) that summarizes the content. The abstract will appear *online* at www.SpringerLink.com and be available with unrestricted access. This allows unregistered users to read the abstract as a teaser for the complete chapter. As a general rule the abstracts will not appear in the printed version of your book unless it is the style of your particular book or that of the series to which your book belongs.

Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript. Use the plain `abstract` command if the abstract is also to appear in the printed version of the book.

- Bandini Thesis

Part III
Reactor Dynamics

Lorem ipsum...

Appendix A
Chapter Heading

All's well that ends well

Use the template *appendix.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style appendix of your book in the Springer layout.

A.1 Section Heading

Instead of simply listing headings of different levels we recommend to let every heading be followed by at least a short passage of text. Furtheron please use the LATEX automatism for all your cross-references and citations.

A.1.1 Subsection Heading

Instead of simply listing headings of different levels we recommend to let every heading be followed by at least a short passage of text. Furtheron please use the LATEX automatism for all your cross-references and citations as has already been described in Sect. A.1.

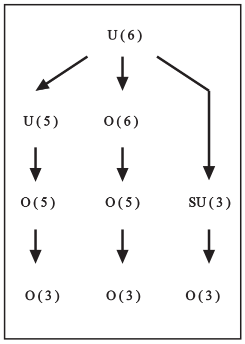
For multiline equations we recommend to use the `eqnarray` environment.

$$\mathbf{a} \times \mathbf{b} = \mathbf{c}$$
$$\mathbf{a} \times \mathbf{b} = \mathbf{c} \tag{A.1}$$

A.1.1.1 Subsubsection Heading

Instead of simply listing headings of different levels we recommend to let every heading be followed by at least a short passage of text. Furtheron please use the

Fig. A.1 Please write your figure caption here



LaTeX automatism for all your cross-references and citations as has already been described in Sect. A.1.1.

Please note that the first line of text that follows a heading is not indented, whereas the first lines of all subsequent paragraphs are.

Table A.1 Please write your table caption here

Classes	Subclass	Length	Action Mechanism
Translation	mRNA^a	22 (19–25)	Translation repression, mRNA cleavage
Translation	mRNA cleavage	21	mRNA cleavage
Translation	mRNA	21–22	mRNA cleavage
Translation	mRNA	24–26	Histone and DNA Modification

^{a} Table foot note (with superscript)

Glossary

Use the template *glossary.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your glossary in the Springer layout.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.