Federal Reserve Bank of Minneapolis
Research Department Staff Report 232

Revised March 1998

Application of Weighted Residual Methods to Dynamic Economic Models [†]

Ellen R. McGrattan
Federal Reserve Bank of Minneapolis

[†] This chapter is based on lectures given at the 7th Annual Summer School of the European Economic Association held September 3–13, 1996, at the European University Institute. I thank Patrick Kehoe and an anonymous referee for comments on an earlier draft. The views expressed herein are those of the author and not necessarily those of the Federal Reserve Bank of Minneapolis or the Federal Reserve System.

Many problems in economics require the solution to a functional equation as an intermediate step. Typically, we seek decision functions that satisfy a set of Euler conditions or a value function that satisfies Bellman's equation. In many cases, we cannot derive analytical solutions for these functions and instead must rely on numerical methods. In this chapter, I will show how to apply weighted residual and finite element methods to this type of problem.

In the case of weighted residual methods, the approximate solution to the functional equation is represented as a linear combination of known *basis functions*. In many cases, the basis functions are polynomials. The coefficients on each basis function are the objects to be computed to obtain an approximate solution. These coefficients are found by setting the residual of the equation to zero in an average sense. In other words, a weighted integral of the residual is set to zero.

The finite element method can be viewed as a piecewise application of the weighted residual method. With the finite element method, the first step in solving the functional equation is to subdivide the domain of the state space into nonintersecting subdomains called *elements*. The domain is subdivided because the method relies on fitting low-order polynomials on subdomains of the state space rather than high-order polynomials on the entire state space. The local approximations are then pieced together to get a global approximation. As the dimensionality of the problem increases, higher-order functions can be used where needed, with fewer elements.

My primary goal in this chapter is to illustrate the application of weighted residual and finite element methods by way of examples. I start with a simple differential equation because the coefficients to be computed satisfy a linear system of equations. For this problem, I can work through examples without a computer. I then apply the methods to a deterministic growth model and a stochastic growth model – two standard models in economics.[1] In the growth model examples, the coefficients to be computed satisfy nonlinear systems of equations. Fortunately, these nonlinear equations are exploitably sparse if they are derived from a finite element method.

1. The General Procedure

The problem is to find $d : I\!R^m \rightarrow I\!R^n$ that satisfies a functional equation $F(d) = 0$, where $F : C_1 \rightarrow C_2$ and C_1 and C_2 are function spaces. As an example, I can think of d as decision or policy variables and F as first-order conditions from some maximization problem. My goal here is to find an approximation $d^n(x;\theta)$ on $x \in \Omega$ which depends on a finite-dimensional vector of parameters $\theta = [\theta_1, \theta_2, \ldots, \theta_n]'$. Weighted residual methods assume that d^n is a finite linear combination of known functions, $\psi_i(x)$, $i = 0, \ldots, n$, called *basis functions*:

$$d^n(x;\theta) = \psi_0(x) + \sum_{i=1}^{n} \theta_i \psi_i(x). \tag{1}$$

The functions $\psi_i(x)$, $i = 0, \ldots, n$ are typically simple functions. Standard examples of basis functions include simple polynomials (for example, $\psi_0(x) = 1$, $\psi_i(x) = x^i$), orthogonal polynomials (for example, Chebyshev polynomials), and piecewise linear functions.

In Figure 1, I display the first five polynomials in the class of Chebyshev polynomials, which is a popular choice for the basis functions. *Chebyshev polynomials* are defined on $[-1, 1]$ and are given recursively as follows: $p_0(x) = 1$, $p_1(x) = x$, and

$$p_i(x) = 2x\, p_{i-1}(x) - p_{i-2}(x), \qquad i = 2, 3, 4, \ldots$$

(or, nonrecursively, as $p_i(x) = \cos(i \arccos x)$). The domain Ω is not typically given by $[-1, 1]$. If the domain is instead $[a, b]$, then I can use $\psi_i(x) = p_{i-1}(2(x-a)/(b-a) - 1)$ for $i = 1, 2, \ldots$ and $\psi_0(x) = 0$.

[1] See Taylor and Uhlig (1990) for a summary of alternative algorithms used to solve the stochastic growth model.

Chebyshev polynomials constitute a set of *orthogonal polynomials* with respect to the weight function $w(x) = 1/\sqrt{1-x^2}$, because $\int_{-1}^{1} p_i(x)p_j(x)w(x)dx = 0$ for all $i \neq j$. Using orthogonal polynomials in my representation d^n rather than the simple polynomials x^i may be preferable as n gets large. For large n, it is difficult to distinguish x^n from x^{n+1}. Thus, the approximation is hardly improved when I add x^{n+1}. With orthogonal polynomials, however, p_n is easily distinguished from p_{n+1} because they are orthogonal to each other.

In Figure 2, I display basis functions that can be used to construct a piecewise linear representation for d^n. These basis functions are of the form

$$\psi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{elsewhere.} \end{cases} \tag{2}$$

I do not need to have the points x_i, $i = 1, \ldots, n$ equally spaced. Therefore, if I want to represent a function that has large gradients or kinks in certain places – say, because inequality constraints bind – then I can cluster points in those regions. In regions where the function is near-linear, I do not need many points.

I define the *residual equation* as the functional equation evaluated at the approximate solution d^n:

$$R(x;\theta) = F(d^n(x;\theta)).$$

I want to choose θ so that $R(x;\theta)$ is close to zero for all x. Weighted residual methods get the residual close to zero in the weighted integral sense. That is, I choose θ so that

$$\int_\Omega \phi_i(x)R(x;\theta)dx = 0, \quad i = 1, \ldots, n,$$

where $\phi_i(x)$, $i = 1, \ldots, n$ are *weight functions*. Note that $\phi_i(x)$ and $\psi_i(x)$ can be different functions. Alternatively, the weighted integral can be written

$$\int_\Omega w(x)R(x;\theta)dx = 0, \tag{3}$$

where $w(x) = \sum_i \omega_i \phi_i(x)$ and (3) must hold for any nonzero weights ω_i, $i = 1, \ldots, n$. Therefore, instead of setting $R(x;\theta)$ to zero for all $x \in \Omega$, the method sets a weighted integral of R to zero.

I consider three specific sets of weight functions and, hence, three ways of determining the coefficients $\theta_1, \ldots, \theta_n$.

1. **Least Squares**: $\phi_i(x) = \partial R(x;\theta)/\partial \theta_i$. This set of weights can be derived by calculating the first-order derivatives for the following optimization problem:
$$\min_\theta \int_\Omega R(x;\theta)^2 \, dx.$$

2. **Collocation**: $\phi_i(x) = \delta(x - x_i)$, where δ is the Dirac delta function. This set of weights implies that the residual is set to zero at n points $x_1, \ldots, x_n$ called the *collocation points*: $R(x_i;\theta) = 0$, $i = 1, \ldots, n$. If the basis functions are chosen from a set of orthogonal polynomials with collocation points given as the roots of the nth polynomial in the set, the method is called *orthogonal collocation*.

3. **Galerkin**: $\phi_i(x) = \psi_i(x)$. In this case, the set of weight functions is the same as the basis functions used to represent d. Thus, the Galerkin method forces the residual to be orthogonal to each of the basis functions. As long as the basis functions are chosen from a complete set of functions, then equation (1) represents the exact solution, given that enough terms are included. The Galerkin method is motivated by the fact that a continuous function is zero if it is orthogonal to every member of a complete set of functions.

To illustrate weighted residual methods, I start with a simple problem in which the coefficients θ_i, $i = 1, \ldots, n$ of (1) satisfy a linear system of equations (that is, $A\theta = b$, where A and b do not depend on θ). Once I have tackled the simple problem, I then apply the methods to standard growth models. (See Aiyagari and McGrattan 1997, Braun and McGrattan 1993, and Chari, Kehoe and McGrattan 1997 for other examples.)

2. The Differential Equation

Consider the following first-order problem: find d such that $d'(x) + d(x) = 0$, with $d(0) = 1$ and $x \in [0, \bar{x}]$. The solution to this problem is $d(x) = exp(-x)$. Using the notation of Section 1, I have

$$F(d)(x) = d'(x) + d(x) = 0. \tag{4}$$

In this section, I consider various choices of basis functions (ψ_i) and weight functions (ϕ_i) to illustrate the weighted residual methods described in Section 1.

Example 1. Suppose that I start with simple polynomials: x^i, $i = 1, \ldots, n$. Then the approximation to $d(x)$ is given by

$$d^n(x;\theta) = 1 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n. \tag{5}$$

Note that I have chosen $\psi_0(x) = 1$ in order to satisfy the boundary condition at $x = 0$. I find the coefficients θ_i, $i = 1, \ldots, n$ by applying a weighted residual method with one of three possible sets of weights. In each case, I will solve a linear system of equations for θ, $A\theta = b$.

1a. Least squares. In this case, the problem is to find θ that minimizes the integral of the squared residual. The residual can be found by substituting equation (5) into equation (4). The first-order conditions of the minimization of the squared residual imply that $\theta_1, \ldots, \theta_n$ satisfy

$$\int_0^{\bar{x}} \frac{\partial R(x;\theta)}{\partial \theta_i} R(x;\theta)\, dx = 0, \quad i = 1, \ldots, n,$$

where the residual and its derivative are given by

$$R(x;\theta) = 1 + \sum_{i=1}^{n} \theta_i \{ i x^{i-1} + x^i \},$$

$$\frac{\partial R(x;\theta)}{\partial \theta_i} = i x^{i-1} + x^i.$$

Suppose that $n = 3$ and $\bar{x} = 6$. Then the following system of equations is solved for θ:

$$\left\{\int_0^6 \begin{bmatrix} 1+x \\ 2x+x^2 \\ 3x^2+x^3 \end{bmatrix} [\, 1+x \quad 2x+x^2 \quad 3x^2+x^3 \,]\, dx \right\} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = -\int_0^6 \begin{bmatrix} 1+x \\ 2x+x^2 \\ 3x^2+x^3 \end{bmatrix} dx$$

or, more simply,

$$\begin{bmatrix} 114.0 & 576.0 & 3067.2 \\ 576.0 & 3139.2 & 17496.0 \\ 3067.2 & 17496.0 & 100643.7 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} -24 \\ -108 \\ -540 \end{bmatrix}.$$

More generally, I can use the fact that

$$R(x;\theta) = (C\vec{x} + e)'\theta + 1,$$

where $\vec{x} = [x, x^2, \ldots, x^n]'$, $e = [1, 0, \ldots, 0]'$, and

$$C = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 3 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & n & 1 \end{bmatrix}.$$

Since the residual R is linear in θ, the derivatives with respect to θ are given by $C\vec{x} + e$. Thus, the system of equations to be solved to compute the coefficients θ for the least squares method is given by

$$\left\{\int_0^{\bar{x}} (C\vec{x} + e)(C\vec{x} + e)'\, dx\right\} \theta = -\int_0^{\bar{x}} (C\vec{x} + e)\, dx$$

or, more succinctly, $A\theta = b$ with

$$A = CMC' + ePC' + CP'e' + \bar{x}ee',$$

$$b = -CP' - \bar{x}e,$$

and

$$M = \int_0^{\bar{x}} \vec{x}\vec{x}'\, dx = \begin{bmatrix} \bar{x}^3/3 & \bar{x}^4/4 & \cdots & \bar{x}^{n+1}/(n+1) \\ \bar{x}^4/4 & \bar{x}^5/5 & \cdots & \bar{x}^{n+2}/(n+2) \\ \vdots & \vdots & \vdots & \vdots \\ \bar{x}^{n+2}/(n+2) & \bar{x}^{n+3}/(n+3) & \cdots & \bar{x}^{2n+1}/(2n+1) \end{bmatrix},$$

$$P = \int_0^{\bar{x}} \vec{x}\, dx = \begin{bmatrix} \bar{x}^2/2 \\ \bar{x}^3/3 \\ \vdots \\ \bar{x}^{n+1}/(n+1) \end{bmatrix}.$$

In Figure 3, I plot the approximate function d^n for $n = 3$ and the exact solution $exp(-x)$. If I had used $n = 5$, then the two lines would be visually indistinguishable.

1b. Collocation. In this case, the problem is to find θ so that the residual is equal to 0 at n points in $[0, \bar{x}]$: $x_1, \ldots, x_n$. Suppose that the x_i are evenly spaced on $[0, 6]$ and that $n = 3$, so that $x_1 = 0$, $x_2 = 3$, and $x_3 = 6$. Then, θ must satisfy the following system of equations:

$$\begin{bmatrix} 1 & 0 & 0 \\ 4 & 15 & 54 \\ 7 & 48 & 324 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}.$$

More generally, I can solve $A\theta = b$ with $(C\vec{x} + e)'$ defined above evaluated at x_i in the ith row of A and b set to a vector of -1's:

$$\begin{bmatrix} (C\vec{x} + e)'|_{x=x_1} \\ (C\vec{x} + e)'|_{x=x_2} \\ \vdots \\ (C\vec{x} + e)'|_{x=x_n} \end{bmatrix} \theta = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}.$$

In Figure 4, I plot the approximate function d^n and the exact solution. If I choose $n = 5$, the two lines are nearly indistinguishable. However, for $n = 3$, the approximation is not as good as the least squares approximation. I will show in a later example how to improve the fit for the collocation method.

1c. Galerkin. In this case, the problem is to find $\theta_1, \ldots, \theta_n$ that satisfy

$$\int_0^{\bar{x}} x^i R(x; \theta)\, dx = 0, \quad i = 1, \ldots, n. \tag{6}$$

Again, consider $n = 3$ and $\bar{x} = 6$. For these choices, the equations in (6) are given by

$$\left\{ \int_0^6 \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} [\, 1 + x \quad 2x + x^2 \quad 3x^2 + x^3 \,]\, dx \right\} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = -\int_0^6 \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} dx. \tag{7}$$

Note that I have written these equations in the form $A\theta = b$. If I compute the integrals in equation (7), then the system of equations becomes

$$\begin{bmatrix} 90.0 & 468.0 & 2527.2 \\ 396.0 & 2203.2 & 12441.6 \\ 1879.2 & 10886.4 & 63318.9 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} -18 \\ -72 \\ -324 \end{bmatrix}.$$

For general n and $\bar{x}$, the coefficients solve $A\theta = b$, where A and b are the following functions:

$$A = MC' + P'e'$$

$$b = -P'$$

with M, C, P, and e as defined above. In Figure 5, I plot the approximate function d^n and the exact solution. The results are similar to those obtained with the least squares method. Again, if I choose $n = 5$, then the approximate and exact solutions are visually indistinguishable.

In Example 1, I assume that the basis functions are the set of simple polynomials, $\{1, x, \ldots, x^n\}$. As I note in Section 1, numerical problems can arise with this set of bases when the order of the approximating function (that is, n) gets large. Suppose instead that I use a set of orthogonal polynomials. In particular, suppose I use the Chebyshev polynomials plotted in Figure 1.

Example 2. Let the approximate solution to (4) be of the following form:

$$d^n(x;\theta) = 1 + x \sum_{i=1}^{n} \theta_i p_{i-1}(2x/\bar{x} - 1), \tag{8}$$

where p_i is the ith Chebyshev polynomial defined on $[-1, 1]$ and $\psi_i(x) = xp_{i-1}(2x/\bar{x} - 1)$ is the ith basis function defined on $[0, \bar{x}]$. Notice that because the polynomials are defined on $[-1, 1]$, I map points in $[0, \bar{x}]$ to points in $[-1, 1]$ by using the transformation $2x/\bar{x} - 1$.

If I apply the method of collocation, then I need to choose points x_i, $i = 1, \ldots, n$ at which to evaluate the residual equation $R(x_i; \theta)$. A sensible choice of points is the n roots of the nth Chebyshev polynomial.[2] For smooth residual functions, mathematical results indicate that choosing the points in this way forces the residual to be close to zero for all the points in the domain of interest. (See Press et al. 1986.)

[2] The method of orthogonal collocation picks the points x_i, $i = 1, \ldots, n$ to be the zeros of the nth basis function. This is not quite what I am doing here. Since the boundary condition needs to be satisfied by my candidate solution, I have chosen $\psi_i(x) = xp_{i-1}(2x/\bar{x} - 1)$ rather than $\psi_i(x) = p_i(2x/\bar{x} - 1)$. But I choose the zeros of the nth Chebyshev polynomial as the points at which to evaluate the residual.

Suppose that $n = 3$ and $\bar{x} = 6$. In this case, I choose the collocation points to be the roots of the polynomial $p_3(2x/\bar{x} - 1) = 4(\frac{1}{3}x - 1)^3 - 3(\frac{1}{3}x - 1)$. The roots, therefore, are 0.4, 3, and 5.6, and the system of equations is

$$\begin{bmatrix} 1.4 & -1.1 & 0.2 \\ 4.0 & 1.0 & -4.0 \\ 6.6 & 7.6 & 9.8 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}. \tag{9}$$

The solution to (9) is substituted into (8). In Figure 6, I plot this approximation along with the solution from Example 1b and the exact solution. Notice that the approximation using orthogonal polynomials is closer to the exact solution at all points in the domain.

In Examples 1 and 2, I use polynomials in the representation of the approximate solution. These polynomials are nonzero on most of the domain of x. Next I will work with basis functions that are nonzero on only small regions of the domain of x. The resulting representations of d^n will be piecewise functions (for example, piecewise linear, piecewise quadratic). In the terminology of numerical analysts, I will be applying a *finite element method*.[3]

The idea behind the finite element method is to break up the domain of x into smaller pieces, use low-order polynomials to get good local approximations for the function d, and then piece the local approximations together to get a good global approximation. In effect, one can think of the finite element method as a piecewise application of a weighted residual method. Thus, to apply a finite element method, I first divide the domain into smaller nonoverlapping subdomains. On each of the subdomains, I construct a local approximation to the function d. For the problem in (4), Ω is one-dimensional, and therefore, division of Ω means coming up with some partition, say, $[x_1, x_2, \ldots, x_n]$ on $I\!R$. Each subinterval $[x_i, x_{i+1}]$ is called an *element*.[4]

---

[3] Weighted residual methods can be divided into two categories: spectral methods and finite element methods. Spectral methods use basis functions that are smooth (that is, C^∞) and nonzero on most of the domain of x (for example, sets of polynomials such as those drawn in Figure 1). Finite element methods use basis functions that are only nonzero on small regions of the domain of x (for example, the tent functions drawn in Figure 2).

[4] In Section 4, I consider a two-dimensional problem. In such cases, the finite element discretization

Suppose, for example, that I want to represent d as a piecewise linear function; that is, over each element, I assume that the approximation is of the form $a + bx$. Suppose also that I want the function d to be continuous on the whole domain Ω. How would I construct basis functions $\psi_i(x)$ so that I can write d^n as in (1)?

The first step is to assign *nodes* on the element. For the finite element method, nodes are points on an element that are used to define the geometry of the element and to uniquely define the order of the polynomial being used to approximate the true solution over the element. Since I am assuming that an element is some interval $[x_i, x_{i+1}]$, two nodes – in particular, the two endpoints x_i and x_{i+1} – are needed to define the geometry. And only two points are needed to uniquely define a linear function. Therefore, the nodes on a one-dimensional element with linear bases are the two endpoints of the element.

The second step in constructing the basis functions is to assume that the undetermined coefficients are equal to the approximate solution at the nodal points. Assume that the numbering of elements and nodes is such that element i is the interval $[x_i, x_{i+1}]$: the first element is $[x_1, x_2]$, the second element is $[x_2, x_3]$, and so on. Assume also that the approximate solution on element i, $d_i^n(x;\theta)$, satisfies $d_i^n(x_i) = \theta_i$ and $d_i^n(x_{i+1}) = \theta_{i+1}$. In other words, assume that the undetermined coefficients represent the solution at the nodes. The approximation of d on element i, d_i^n, is therefore uniquely given by

$$d_i^n(x;\theta) = \theta_i \psi_i(x) + \theta_{i+1}\psi_{i+1}(x), \quad x \in [x_i, x_{i+1}],$$

where the basis functions are given by equation (2) and drawn in Figure 2. Since elements are connected to each other at nodal points on the element boundaries, this choice of basis functions guarantees that the approximation is continuous across elements. Notice also that any linear function (and, hence, any continuous piecewise linear d^n) can be represented with the basis functions given in (2).

consists of simple two-dimensional (nonoverlapping) subdomains, such as triangles, rectangles, and quadrilaterals.

In the examples below, I will focus on only the simplest finite element approximations that are used in practice. For the one-dimensional examples, I will use linear and quadratic basis functions. For the two-dimensional examples, I will use rectangularly shaped elements with linear and quadratic basis functions. Extensions to higher dimensions and higher-order polynomials are relatively straightforward. (See, for example, Hughes 1987.)

Example 3. Let the approximate solution to (4) be of the form

$$d^n(x;\theta) = \sum_{i=1}^{n} \theta_i \psi_i(x),$$

with $\psi_i(x)$, $i = 1, \ldots, n$ given by (2). To impose the boundary condition $d^n(0;\theta) = 1$, I need to set θ_1 to one. Here I apply a Galerkin method. Therefore, the weight functions are given by the bases $\psi_i(x)$, $i = 1, \ldots, n$.

3a. Three elements. Suppose that there are three elements with nodes at 0, 1, 3, and 6. Then the residual equation is given by

$$R(x;\theta) = \sum_{i=1}^{4} \theta_i(\psi_i'(x) + \psi_i(x))$$

$$= \begin{cases} \theta_1\,(-x) + \theta_2\,(1+x) & \text{if } x \in [0,1] \\ \theta_2\,(1 - \frac{1}{2}x) + \theta_3\,(\frac{1}{2}x) & \text{if } x \in [1,3] \\ \theta_3\,(\frac{5}{3} - \frac{1}{3}x) + \theta_4\,(-\frac{2}{3} + \frac{1}{3}x) & \text{if } x \in [3,6]. \end{cases} \tag{10}$$

If I substitute the residual (10) into the weighted integral (3) with $\phi_i(x) = \psi_i(x)$, then I get the following system of equations:

$$\left\{ \int_0^1 \begin{bmatrix} 1-x \\ x \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} -x & 1+x & 0 & 0 \end{bmatrix} dx \right.$$

$$+ \int_1^3 \begin{bmatrix} 0 \\ \frac{3}{2} - \frac{1}{2}x \\ -\frac{1}{2} + \frac{1}{2}x \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 - \frac{1}{2}x & \frac{1}{2}x & 0 \end{bmatrix} dx$$

$$\left. + \int_3^6 \begin{bmatrix} 0 \\ 0 \\ 2 - \frac{1}{3}x \\ -1 + \frac{1}{3}x \end{bmatrix} \begin{bmatrix} 0 & 0 & \frac{5}{3} - \frac{1}{3}x & -\frac{2}{3} + \frac{1}{3}x \end{bmatrix} dx \right\} \begin{bmatrix} 1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

or if I compute the integrals,

$$\begin{bmatrix} -1/6 & 2/3 & 0 & 0 \\ -1/3 & 1 & 5/6 & 0 \\ 0 & -1/6 & 5/3 & 1 \\ 0 & 0 & 0 & 3/2 \end{bmatrix} \begin{bmatrix} 1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{11}$$

Note that I need to drop the first equation because I have to impose that $\theta_1 = 1$ for the boundary condition to be satisfied.[5] Therefore, the system of equations reduces to

$$\begin{bmatrix} 1 & 5/6 & 0 \\ -1/6 & 5/3 & 1 \\ 0 & 0 & 3/2 \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 0 \\ 0 \end{bmatrix},$$

with three equations and three unknowns. In Figure 7, I plot the finite element approximation and the exact solution. By construction, the approximate function is piecewise linear. As in the case of Examples 1–3, with five degrees of freedom, it is difficult to distinguish the approximate from the exact solutions when they are plotted.

[5] Recall that the integral equation can be written as in (3), where in this case, $w(x) = \sum_i \omega_i \psi_i(x)$. The function $w(x)$ must satisfy the homogeneous counterpart of the boundary condition $d(0) = 1$, that is, $w(0) = 0$. For those familiar with the calculus of variations, w is like the variation of the solution and thus must satisfy the homogeneous counterparts of boundary conditions for d. Enforcing the condition $w(0) = 0$ is equivalent to dropping the first equation in (11).

3b. m elements, linear bases. Because the same calculations are made for each element, derivation of the linear system of equations for θ can be simplified greatly. The trick is to work with one *master element* and use the calculations for all elements. Consider the following master element. Assume that it has length ℓ_e, that node 1 is placed at 0, and that node 2 is placed at ℓ_e. To construct the approximation d^n on this master element, I use a linear combination of two functions:

$$\psi_1^e(\bar{x}) = 1 - \frac{\bar{x}}{\ell_e} \quad \text{and} \quad \psi_2^e(\bar{x}) = \frac{\bar{x}}{\ell_e},$$

where $\bar{x}$ lies in $[0,\ell_e]$. If I map element e given by $[x_e, x_{e+1}]$ to the master element, then the relationship between the *local coordinate* $\bar{x}$ and the *global coordinate* x (which lies somewhere between x_e and x_{e+1}) is given by $\bar{x} = x - x_e$.

Let

$$\Psi(\bar{x}) = [\, 1 - \tfrac{\bar{x}}{\ell_e} \quad \tfrac{\bar{x}}{\ell_e} \,]$$

$$\frac{d\Psi(\bar{x})}{d\bar{x}} = [\, -\tfrac{1}{\ell_e} \quad \tfrac{1}{\ell_e} \,].$$

Define K^e to be the following integral:

$$\begin{aligned} K^e &= \int_0^{\ell_e} \left\{ \Psi(\bar{x})' \left(\Psi(\bar{x}) + \frac{d\Psi(\bar{x})}{d\bar{x}} \right) \right\} d\bar{x} \\ &= \int_0^{\ell_e} \left\{ \begin{bmatrix} 1 - \frac{\bar{x}}{\ell_e} \\ \frac{\bar{x}}{\ell_e} \end{bmatrix} \left([\, 1 - \tfrac{\bar{x}}{\ell_e} \quad \tfrac{\bar{x}}{\ell_e} \,] + [\, -\tfrac{1}{\ell_e} \quad \tfrac{1}{\ell_e} \,] \right) \right\} d\bar{x} \\ &= \begin{bmatrix} \frac{\ell_e}{3} - \frac{1}{2} & \frac{\ell_e}{6} + \frac{1}{2} \\ \frac{\ell_e}{6} - \frac{1}{2} & \frac{\ell_e}{3} + \frac{1}{2} \end{bmatrix}. \end{aligned} \tag{12}$$

Then I can show that the residual equation is given by

$$\begin{bmatrix} \begin{bmatrix} K^1 & \\ & \end{bmatrix} & & & & 0 \\ & \begin{bmatrix} K^2 & \\ & \end{bmatrix} & & & \\ & & \begin{bmatrix} K^3 & \\ & \end{bmatrix} & & \\ & & & \ddots & \\ 0 & & & & \begin{bmatrix} K^m & \\ & \end{bmatrix} \end{bmatrix} \begin{bmatrix} 1 \\ \theta_2 \\ \vdots \\ \theta_{m+1} \end{bmatrix} = 0. \tag{13}$$

With the exceptions of (1,1) and $(m+1, m+1)$, the diagonal elements of the matrix multiplying θ in (13) are the sum of two numbers. For example, diagonal element (2,2) is found by summing $K^1(2,2)$ and $K^2(1,1)$, (3,3) is found by summing $K^2(2,2)$ and $K^3(1,1)$, and so on. Again, I drop the first equation and solve a linear system of $m = n-1$ equations in $n-1$ unknowns.

Consider again the case in which $m = 3$ (that is, Example 3a). The first element is [0,1], which has length 1; the second element is [1,3], which has length 2; and the third element is [3,6], which has length 3. Using the formula in equation (12) I have

$$K^1 = \begin{bmatrix} -1/6 & 2/3 \\ -1/3 & 5/6 \end{bmatrix}, \quad K^2 = \begin{bmatrix} 1/6 & 5/6 \\ -1/6 & 7/6 \end{bmatrix}, \quad K^3 = \begin{bmatrix} 1/2 & 1 \\ 0 & 3/2 \end{bmatrix}.$$

Substituting K^1, K^2, and K^3 into equation (13) with $K^1(2,2)$ added to $K^2(1,1)$ and $K^2(2,2)$ added to $K^3(1,1)$ yields the matrix in (11).

What is evident from (13) is that the system I am solving is sparse. The matrix that will be inverted is $m \times m$, but it has only $3m-2$ nonzeros. If $m = 20$, then 15% of the matrix elements are nonzero. If $m = 100$, 3% of the matrix elements are nonzero. Typically, a sparse matrix is defined as a matrix with very few nonzero elements. However, what is most relevant is whether or not I can apply methods to take advantage of the number and positioning of the zero elements. In the example above, the A matrix in $A\theta = b$ is tridiagonal and, therefore, has a structure that lends itself well to numerical techniques designed for solving sparse linear systems.

Although the computed examples in this chapter have a small number of unknowns and, therefore, can be solved without resorting to sparse equation solvers, problems such as that described in Chari, Kehoe, and McGrattan (1997) require solving systems on the order of 25,000 equations. In such cases, the memory limitations make standard (non-sparse) matrix inversion routines infeasible to use. In their example, Chari, Kehoe, and McGrattan (1997) have to invert a 24,800×24,800 matrix in which 0.14% of the elements are nonzero. As in the example above, the positions of the nonzero elements in their example are such that the system of equations is exploitably sparse. Saad (1996) describes a variety of methods for solving large sparse systems of equations.

In Example 3, I assume that the approximate solution on each element is a linear function. There are two possible ways to improve the approximation: increase the number of elements or increase the order of the polynomial approximation on each element. I next consider an approximation with quadratic basis functions. In this case, a typical element has three nodes: one at each endpoint and one in the interior. The nodes at the endpoints are needed to define the geometry of the element (which is the same as in the case of linear bases). The interior point is needed to uniquely determine the order of the polynomial; that is, there are three degrees of freedom if the approximate solution is of the form $a + bx + cx^2$ on the element. To allow flexibility, I do not necessarily assume that the third nodal point is at the midpoint of the element. However, because I want to avoid repeating the same calculations, I should calculate the functions and residuals for a master quadratic element just as I did in the linear case in Example 3b. Consider the following master element. Assume that it has length ℓ_e. Let $\bar{x}$ be the local coordinate of the element, which has node 1 located at x_1^e. Then $\bar{x} = x - x_1^e$, where x is the global coordinate. Locally, assume that the nodes are located at $\bar{x} = 0$, $\bar{x} = \alpha\ell_e$, and $\bar{x} = \ell_e$, with $0 < \alpha < 1$.

As in the linear case, I can write the approximation on element e as a linear combination of basis functions that have the property that they are equal to 1 at one node and 0 at the other nodes; that is,

$$d_e^n(\bar{x};\theta) = \theta_1^e\psi_1^e(\bar{x}) + \theta_2^e\psi_2^e(\bar{x}) + \theta_3^e\psi_3^e(\bar{x}),$$

where the ψ_i^e's are constructed to satisfy $d_e^n(0) = \theta_1^e$, $d_e^n(\alpha\ell_e) = \theta_2^e$, and $d_e^n(\ell_e) = \theta_3^e$. The only functions that satisfy these conditions are given by

$$\begin{aligned}
\psi_1^e(\bar{x}) &= \left(1 - \frac{\bar{x}}{\ell_e}\right)\left(1 - \frac{\bar{x}}{\alpha \ell_e}\right) \\
\psi_2^e(\bar{x}) &= \frac{\bar{x}}{\alpha(1-\alpha)\ell_e}\left(1 - \frac{\bar{x}}{\ell_e}\right) \\
\psi_3^e(\bar{x}) &= -\frac{\alpha \bar{x}}{(1-\alpha)\ell_e}\left(1 - \frac{\bar{x}}{\alpha \ell_e}\right).
\end{aligned} \tag{14}$$

Notice that there are only three elements of the vector θ used to approximate the function on any particular element. I use the superscript e to indicate the element. In the example below, I will relate the coefficients θ_i^e to the elements in the $n \times 1$ vector θ.

Example 4. Let the approximate solution be piecewise quadratic. Assume that the interior nodes on each element are at the midpoints of the elements (so that $\alpha = 1/2$.) Let

$$\Psi(\bar{x}) = [\,\psi_1^e(\bar{x}) \quad \psi_2^e(\bar{x}) \quad \psi_3^e(\bar{x})\,]$$

where $\psi_i^e(\bar{x})$ is given in (14), and define K^e to be the following integral:

$$\begin{aligned}
K^e &= \int_0^{\ell_e} \left\{ \Psi(\bar{x})' \left(\Psi(\bar{x}) + \frac{d\Psi(\bar{x})}{d\bar{x}} \right) \right\} d\bar{x} \\
&= \begin{bmatrix} \frac{2}{15}\ell_e - \frac{1}{2} & \frac{1}{15}\ell_e + \frac{2}{3} & -\frac{1}{30}\ell_e - \frac{1}{6} \\ \frac{1}{15}\ell_e - \frac{2}{3} & \frac{8}{15}\ell_e & \frac{1}{15}\ell_e + \frac{2}{3} \\ -\frac{1}{30}\ell_e + \frac{1}{6} & \frac{1}{15}\ell_e - \frac{2}{3} & \frac{2}{15}\ell_e + \frac{1}{2} \end{bmatrix}.
\end{aligned}$$

Then I can show that if there are m elements, the residual equations are given by

$$\begin{bmatrix} \begin{bmatrix} K^1 \end{bmatrix} & & & & 0 \\ & \begin{bmatrix} K^2 \end{bmatrix} & & & \\ & & \begin{bmatrix} K^3 \end{bmatrix} & & \\ & & & \ddots & \\ 0 & & & & \begin{bmatrix} K^m \end{bmatrix} \end{bmatrix} \begin{bmatrix} 1 \\ \theta_2 \\ \vdots \\ \theta_{2m+1} \end{bmatrix} = 0. \tag{15}$$

Diagonal element (3,3) is the sum of $K^1(3,3)$ and $K^2(1,1)$, diagonal element (5,5) is the sum of $K^2(3,3)$ and $K^3(1,1)$, and so on. The system of equations to be solved for $\theta_2, \ldots, \theta_{2m+1}$ is $A\theta = b$, which is the second through $(2m+1)$th equation of the system in (15). (Recall that the first equation is dropped, because I impose that $\theta_1 = 1$.) Again, it is evident that the system of equations to be solved is sparse. In this example, the matrix to be inverted is $2m \times 2m$. The number of nonzero elements in A is $8m - 4$. If $m = 20$, then 10% of the elements of A are nonzero. If $m = 100$, then 2% of the elements of A are nonzero.

Note that a relationship exists between the θ_i's and the θ_i^e's. The θ_i^e denotes the ith unknown coefficient on element e, $i = 1, 2, 3$, whereas θ_i denotes the unknown coefficient for node i, $i = 1, \ldots, 2m+1$. If I label nodes consecutively, then $\theta_2 = \theta_2^1$, $\theta_3 = \theta_3^1 = \theta_1^2$, $\theta_4 = \theta_2^2$, $\theta_5 = \theta_3^2 = \theta_1^3$, and so on.

If $m = 3$ and the elements are the same as in Example 3, namely, [0,1], [1,3], and [3,6], then the approximate solution satisfies

$$\begin{bmatrix} 16 & 22 & 0 & 0 & 0 & 0 \\ -18 & 12 & 24 & -7 & 0 & 0 \\ 0 & -16 & 32 & 24 & 0 & 0 \\ 0 & 3 & -16 & 20 & 26 & -8 \\ 0 & 0 & 0 & -14 & 48 & 26 \\ 0 & 0 & 0 & 2 & -14 & 27 \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{bmatrix} = \begin{bmatrix} 18 \\ -4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Note that I have multiplied all the elements in the system by 30 to avoid writing them as fractions. In Figure 8, I plot the approximate solution (the line marked "quadratic bases") along with the exact solution and the approximate solution with linear basis functions found in Example 3. Notice that the fit is better with quadratic functions. However, in this example, the number of unknowns doubles.

In general, the finite element approximations must satisfy certain conditions to guarantee convergence of the method as the number of elements is increased. The approximate solution should be continuous and differentiable over each element – with the order of differentiability such that all terms in the residual equation are nonzero. The polynomials used to represent the approximate solution on an element should be complete, and the approximation should be continuous across element boundaries.

Although the finite element method – as I applied it in Examples 3 and 4 – can be viewed as a Galerkin weighted residual method, a lot of structure is put on the bases and the element geometries. In fact, one criticism of spectral methods is that, unlike with the finite element method, the selection of basis and weight functions seems arbitrary. This is why the finite element method and spectral weighted residual methods are typically treated separately in textbooks. (See, for example, Reddy 1993.)

In the next two sections, I apply weighted residual and finite element methods to standard problems in economics. Unfortunately, these problems do not have the feature that the residual equations are linear in the unknown vector θ. However, as I will show, most of the computation will again involve solving a potentially large linear system of equations.

3. The Deterministic Growth Model

I now turn to the deterministic growth model:

$$\begin{aligned} &\max_{\{c_t\}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\ &\text{subject to} \;\; c_t + k_t = f(k_{t-1}), \end{aligned} \tag{16}$$

where c_t is consumption at date t, k_t is the capital stock at date t, $u(\cdot)$ is the utility function, $f(\cdot)$ is the production function, and $\beta < 1$ is a discount factor.[6] From the Euler equation, the functional equation is given by

$$F(c)(k) = \beta \frac{u'(c(f(k) - c(k)))}{u'(c(k))} f'(f(k) - c(k)) - 1 = 0,$$

[6] See Sargent (1987) for a detailed discussion of the problems described here and in the next section.

and the boundary condition is given by $c(0) = 0$. In this case, I want to compute an approximation $c^n(k;\theta)$ to the consumption function that sets $F(c)$ approximately equal to zero for all k.

Example 5. Let $u(c) = \ln(c)$ and $f(k) = \lambda k^\alpha$. In this case, the functional equation is

$$F(c)(k) = \frac{\beta\alpha\lambda\left(\lambda k^\alpha - c(k)\right)^{\alpha-1}\, c(k)}{c\left(\lambda k^\alpha - c(k)\right)} - 1$$

The solution for consumption in this case is

$$c(k) = (1-\beta\alpha)\lambda k^\alpha.$$

Suppose that I want to obtain an approximate solution of the form

$$c^n(k;\theta) = \theta_1 k + \theta_2 k^2 + \ldots + \theta_n k^n$$

which satisfies the boundary condition at $k = 0$. The residual equation is therefore

$$R(k;\theta) = \frac{\beta\alpha\lambda\Big(\lambda k^\alpha - \sum_{j=1}^n \theta_j k^j\Big)^{\alpha-1} \sum_{j=1}^n \theta_j k^j}{\sum_{i=1}^n \theta_i \Big(\lambda k^\alpha - \sum_{j=1}^n \theta_j k^j\Big)^i} - 1.$$

To apply weighted residual methods, I have to compute integrals of the form

$$\int_0^{\bar{k}} \phi_i(k) R(k;\theta)\, dk, \quad i = 1, \ldots, n \tag{17}$$

where $\bar{k}$ is the upper bound of the domain for the capital stock. Since the residual R is a nonlinear function of θ, it makes sense to do numerical integration. If I apply Gaussian quadrature (which is typically done), then equation (17) is replaced by

$$\sum_l \omega_l \phi_i(k_l) R(k_l;\theta), \quad i = 1, \ldots, n \tag{18}$$

where ω_l are the *quadrature weights* and the grid points k_l are the *quadrature abscissas*. (See Press et al. 1986 for the quadrature formulas and a description of how they are derived.)

The values for ω_l and k_l do not depend on the function being integrated ($\phi_i(k)R(k;\theta)$ in this case). In other words, once I know the bounds of integration (for example, 0 and $\bar{k}$) and the number of quadrature points, I can look up the ω_l's and k_l's in a standard quadrature table.[7] Depending on the specific quadrature rule (for example, Legendre, Chebyshev, Hermite) used, ω_l's and k_l's will differ, but the calculations of R and ϕ will look the same no matter what quadrature rule is used.

The final step is to solve the system of equations in (18). In this case, the system is nonlinear. The problem is to find θ such that $G(\theta) = 0$, where G has the same dimension as θ. Applying Newton's method to $G(\theta) = 0$ means iterating on

$$\theta^{j+1} = \theta^j - \left[\left. \frac{\partial G(\theta)}{\partial \theta} \right|_{\theta=\theta^j} \right]^{-1} G(\theta^j), \quad j = 1, 2, \ldots$$

with some initial guess θ^0, where θ^j is the vector of unknown coefficients at the jth iteration. Notice that as I iterate, I solve a sequence of problems of the following form: find θ such that $A\theta = b$, where A is the Jacobian matrix $\partial G/\partial\theta$ evaluated at θ^j and b is the function itself, $G(\theta^j)$.

For the three weighted residual applications below, assume that $\alpha = 0.25$, $\beta = 0.96$, $\lambda = 1/(\alpha\beta)$, and $\bar{k} = 2$. For this set of parameters, the steady-state capital stock is equal to one. Assume also that the quadrature rule is Legendre with 20 quadrature abscissas used to approximate the integral in (17). In this case, $\omega_l = \int_{-1}^{1} \prod_{i=1, i\neq l}^{20} (x - x_i)/(x_l - x_i) dx$, $l = 1, \ldots, 20$, and $x_1, \ldots, x_{20}$ are the roots of the 20th Legendre polynomial found recursively as follows: $p_0(x) = 1$, $p_1(x) = x$, $ip_i(x) = (2i-1)xp_{i-1}(x) - (i-1)p_{i-2}(x)$ for $i = 2, \ldots, 20$. Since $\bar{k} = 2$, the points k_l are given by $k_l = x_l + 1$, $l = 1, \ldots, 20$.

5a. Least squares. To apply the method of least squares, I set $\phi_i(k) = \partial R(k;\theta)/\partial\theta_i$,

[7] The weights and abscissas are chosen so that the n-point quadrature rule is exact for integrals of all polynomials of order $2n - 1$ times some weight function, which depends on the specific rule. For example, Gauss-Legendre quadrature uses a weight function of 1 and Gauss-Chebyshev quadrature uses a weight function of $1/\sqrt{1 - x^2}$, where x is defined on $(-1, 1)$.

where the derivative of the residual is given by

$$\frac{\partial R(k;\theta)}{\partial \theta_l} = -\frac{\beta\alpha\lambda\tilde{k}^{\alpha-1}k^\ell}{c^n(\tilde{k};\theta)}\left\{1+\frac{(\alpha-1)c^n(k;\theta)}{\tilde{k}} - \frac{c^n(k;\theta)}{c^n(\tilde{k};\theta)}\left(\tilde{k}^\ell - k^\ell\sum_{i=1}^n i\theta_i\tilde{k}^{i-1}\right)\right\}$$

$l = 1,\ldots,n$, and $\tilde{k} = \lambda k^\alpha + \sum_j \theta_j k^j$. In Figure 9, I plot the approximate solution c^n for $n = 5$ along with the exact solution. Since the derivative of the true function is infinite at $k = 0$ and relatively small for high values of k, I must add more polynomials to completely resolve the solution at all capital stocks. I also plot the result for a more restricted grid on the capital stocks, namely, $[\frac{1}{3}, \frac{5}{3}]$. This is the grid Judd (1992) uses when evaluating weighted residual methods for the deterministic growth model. For both approximations, I assume that $n = 5$. Notice that although the approximation on $[\frac{1}{3}, \frac{5}{3}]$ is very close to the true solution, the exact solution is very smooth – almost linear.

5b. Collocation. To apply the collocation method, I set $\phi_i(k) = \delta(k - k_i)$, where k_i, $i = 1,\ldots,n$ are collocation points in $[0,\bar{k}]$. In Figure 10, I plot two approximations: one with five evenly spaced collocation points between 0.1 and 2 and one with five evenly spaced collocation points between $\frac{1}{3}$ and $\frac{5}{3}$. The problem of fitting functions with steep gradients becomes acute in this case, which is why I avoid the region of capital stocks below 0.1. Even so, the approximation on [0.1,2] is not very accurate. It is clear that I need better choices for basis functions and collocation points to make this method competitive with least squares. On $[\frac{1}{3}, \frac{5}{3}]$, I find that the approximation is not quite as good as that for least squares, but it is not too different from the exact solution. Here again, the fit is good because the exact solution is very smooth on $[\frac{1}{3}, \frac{5}{3}]$.

5c. Galerkin. To apply the Galerkin method, I set $\phi_i(k) = k^i$, $i = 1,\ldots,n$. In Figure 11, I plot approximate functions on [0,2] and $[\frac{1}{3}, \frac{5}{3}]$ along with the exact solution. The results here are similar to the results of the least squares method.

Because I need to include more polynomials, which in the case of k^i, $i = 1,\ldots,n$ become similar to each other as n gets large, it makes sense to use a class of orthogonal polynomials. Judd (1992) uses a representation for consumption of the form

$$c^n(k;\theta) = \sum_{i=1}^{n} \theta_i \psi_i(k), \tag{19}$$

where $\psi_i(k) = p_{i-1}(2(k - \underline{k})/(\bar{k} - \underline{k}) - 1)$, $\underline{k}$ is a lower bound on the capital stocks, and $p_i(x)$ is the ith Chebyshev polynomial defined in equation (19).[8]

Example 6. In this case, assume that $u(c) = c^{1-\tau}/(1-\tau)$ and $f(k) = \lambda k^\alpha + (1-\delta)k$. Let $\tau = 5$, $\alpha = 0.25$, $\delta = 0.025$, $\beta = 0.99$, and $\lambda = (1-\beta(1-\delta))/(\alpha\beta)$ (so that the steady-state capital is equal to 1). Let c^n take the form of (19) with $n = 10$. In Figure 12, I plot the approximate solutions for $\underline{k} = 0.03$, $\bar{k} = 2$ (marked with a square), and $\underline{k} = 0.1$, $\bar{k} = 1.9$ (marked with a circle) along with the exact solution.[9] The location of the points marked by squares or circles are the quadrature abscissas.

It is clear from Figure 12 that more polynomials are needed for a good approximation on [0.03,2]. This is because I am trying to approximate a very steep part of the function and a very flat part of the function using the same basis functions. When I restrict the domain to [0.1,1.9], there is a significant improvement in the approximation over this region of the state space. The approximation is visually indistinguishable from the exact solution. In this restricted region of the domain, the function does not have any large gradients.

Suppose that, instead of using Chebyshev polynomials, I apply the Galerkin method with piecewise linear basis functions as is done for the finite element method.

Example 7. Assume that $u(\cdot)$, $f(\cdot)$, and the parameterization are the same as in Example 6. Let $x_1 = 0$, $x_{11} = 2$, and $x_i = x_{i-1} + 0.005\exp(0.574(i-2))$. This partition implies that there are 10 elements with lengths that increase exponentially. Thus, there will be

---

[8] Note that this approximation will not satisfy the boundary condition at $c(0) = 0$ if $\underline{k} = 0$ for any θ. However, if I make a slight modification, namely, $\psi_i(k) = kp_{i-1}(2k/\bar{k} - 1)$ defined on $[0, \bar{k}]$, then the boundary condition is satisfied for all possible choices of θ.

[9] What I call the *exact solution* here is actually a finite element approximation with a large number of elements. Although this itself is an approximation, doubling the number of elements leaves Figure 12 unchanged.

more points near the origin, where the function has a large (infinite in this case) gradient. To compute the weighted integral, I use a Legendre quadrature rule with two quadrature points per element. On an element of length ℓ_e, the Legendre quadrature rule with two quadrature points implies the following weights and abscissas for (18): $\omega_l = \ell_e/2$, $l = 1, 2$, and $k_1 = k_e + 0.211\ell_e$, $k_2 = k_e + 0.789\ell_e$, where k_e is the first endpoint of the element.

In Figure 13, I plot the finite element approximation along with the exact solution. Because the finite element method is a piecewise application of a weighted residual method, it is possible to get a more accurate approximation over the entire [0,2] domain – I am not using the same basis functions in the very steep region and the very flat region of the consumption function.

To obtain the approximation in Figure 13, the main computational task is the inversion of a 10×10 matrix. In this matrix, 68 of the 100 elements are zeros, and the structure of the matrix is band diagonal. As the number of unknowns becomes large, it becomes expensive and, in some cases, infeasible to invert the matrix without using inversion routines that exploit the fact that the matrix is band diagonal. (See Saad 1996.)

4. The Stochastic Growth Model[10]

Suppose that, instead of the deterministic growth model, I want to calculate the decision functions for the stochastic growth model in which decisions depend on the capital stock and a stochastic shock. The stochastic growth model assumes that output at date t can be allocated either to current consumption c_t or to current investment i_t. The consumption/savings decision is assumed to be optimal in that the preferences of households are maximized. The preferences are given by

$$E\left[\sum_{t=0}^{\infty} \beta^t\, u(c_t) \,\middle|\, k_{-1}\right], \qquad 0 < \beta < 1, \tag{20}$$

[10] See Judd (1992) for more details on spectral methods as applied to this problem and McGrattan (1996) for more details on the finite element method as applied to this problem.

where k_t is the capital stock at t and k_{-1} is known. The maximization of equation (20) is done subject to the feasibility constraints

$$c_t + k_t - (1-\delta)k_{t-1} = \lambda_t k_{t-1}^{\alpha}, \qquad 0 < \alpha < 1,\ 0 \leq \delta \leq 1, \tag{21}$$

the nonnegativity constraints $c_t \geq 0$, $k_t \geq 0$ for all $t \geq 0$ and subject to the process for the technology shock,

$$\ln \lambda_t = \rho \ln \lambda_{t-1} + \varepsilon_t, \qquad -1 < \rho < 1, \tag{22}$$

where ε_t is a serially uncorrelated, normally distributed random variable with mean zero and variance σ^2. Because ε is normally distributed, it does not have a compact support. The technology shock in this case takes on values between 0 and infinity. On the computer, I cannot specify an upper bound of infinity. Instead, I can either specify a large upper bound (in which the probability of observing a larger value is small) or make a transformation of variables and work with a bounded interval. Let $z = \tanh(\ln(\lambda))$, which is defined on $[-1, 1]$. Then I can rewrite equation (22) as follows:

$$z_t = \tanh(\rho \tanh^{-1}(z_{t-1}) + \sqrt{2}\sigma\nu_t),$$

where $\nu_t = \varepsilon_t/(\sqrt{2}\sigma)$.

Because the stochastic shock takes on a continuum of values, I need to solve a two-dimensional problem. The representation of the approximate solution is then

$$c^n(k, z; \theta) = \sum_{i=1}^{n} \theta_i \psi_i(k, z).$$

A simple set of basis functions is all products of the elements of $\{1, k, k^2, \ldots, k^{n_k}\}$ and $\{1, z, z^2, \ldots, z^{n_z}\}$. Alternatively, I can use all products of the elements of two sets of orthogonal polynomials. In either case, however, the number of unknowns starts to add up quickly, especially if a large number of polynomials are needed to approximate consumption at both high and low values of the capital stock.

One way to keep the problem tractable is to use the set of complete polynomials rather than all products of terms in $\{k^i\}_{i=0}^{n_k}$ and $\{z^i\}_{i=0}^{n_z}$ (for example, bases $\{1, k, z, k^2, kz, z^2\}$ rather than $\{1, k, z, kz, k^2, z^2, k^2z, kz^2, k^2z^2\}$). Using the set of complete polynomials allows me to approximate higher-order functions but limits the number of unknown coefficients.[11] Another way to keep the problem tractable is to apply a finite element method. As earlier examples show, the system of equations to be solved for the unknown coefficients θ is typically very sparse. Therefore, in big problems, I do not need as much storage as in a typical spectral method, and I can apply algorithms for solving sparse systems of equations.

Consider application of the finite element method to the stochastic growth model. The first step is to write out the residual equation using the first-order condition for the problem in (20):

$$R(k, z; \theta) = \frac{\beta}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{c^n(\tilde{k}, \tilde{z}; \theta)^{-\tau}}{c^n(k, z; \theta)^{-\tau}} \left(\alpha \tilde{k}^{\alpha-1} \sqrt{\frac{1+\tilde{z}}{1-\tilde{z}}} + 1 - \delta \right) e^{-\nu^2} \, d\nu - 1 = 0, \tag{23}$$

where

$$\tilde{k} = k^\alpha \sqrt{(1+z)/(1-z)} + (1-\delta)k - c^n(k, z; \theta)$$
$$\tilde{z} = \tanh(\rho \tanh^{-1}(z) + \sqrt{2}\sigma\nu),$$

$c^n(0, z; \theta) = 0$, ν is distributed normally with mean zero and variance $1/2$, and the domain for the state space is $\Omega = [0, \bar{k}] \times [-1, 1]$. If I apply a Gauss-Hermite quadrature rule when computing the integral in equation (23), then the residual equation becomes

$$R(k, z; \theta) \simeq \frac{\beta}{\sqrt{\pi}} \sum_{l=1}^{m_\nu} \frac{c^n(\tilde{k}, \tilde{z}_l; \theta)^{-\tau}}{c^n(k, z; \theta)^{-\tau}} \left(\alpha \tilde{k}^{\alpha-1} \sqrt{\frac{1+\tilde{z}_l}{1-\tilde{z}_l}} + 1 - \delta \right) \omega_l - 1,$$

where $\tilde{z}_l = \tanh(\rho \tanh^{-1}(z) + \sqrt{2}\sigma\nu_l)$ and ν_l, ω_l, $l = 1, \ldots, m_\nu$ are the abscissas and weights for an m_ν-point quadrature rule. (For the quadrature formulas, see Press et al. 1986.)

[11] See Judd (1992) for a comparison of complete polynomials and tensor products in the stochastic growth model example.

The second step in applying the finite element method is to divide up the domain into smaller nonoverlapping subdomains called elements. In this problem, the domain is two-dimensional and rectangular: $\Omega = [0, \bar{k}] \times [-1, 1]$. A reasonable choice for the element shape, therefore, is a rectangle. Suppose that I divide the domain into smaller rectangular subdomains which do not overlap.[12] Each element will be a rectangle in Ω, say, $[k_i, k_{i+1}] \times [z_j, z_{j+1}]$, where k_i is the ith grid point for the the capital stock and z_j is the jth grid point for the technology shock.

I consider two types of approximations over the rectangular elements: linear and quadratic. Suppose the representation for consumption on some element e is linear,

$$c_e^n(k, z) = a + b\,k + c\,z + d\,kz. \tag{24}$$

Because there are four unknowns, I require an element with four nodes. If I place the four nodes at the corners of the rectange, then I can uniquely define the geometry of the element and use the values of the solution at the four nodes to pin down the constants in equation (24). That is, as in the one-dimensional case, I can rewrite the approximation in (24) so that $c_e^n(k, z; \theta) = \sum_i \theta_i^e \psi_i^e(k, z)$, $i = 1, \ldots, 4$, where the basis functions are such that ψ_i^e is 1 at node i and zero at the other three nodes on the element.

Before I give formulas for the basis functions, it is convenient to first consider a mapping from global coordinates (k, z) to local coordinates (ξ, η) defined on a master element. This is done for convenience, since the master element has a fixed set of coordinates, while each element in Ω has a different set of coordinates. Thus, I can construct basis functions once but use them for each element. Consider functions $\xi(k)$ and $\eta(z)$ that map a typical element $[k_i, k_{i+1}] \times [z_j, z_{j+1}]$ to the square $[-1, 1] \times [-1, 1]$; that is, $\xi(k) = (2k - k_i - k_{i+1})/(k_{i+1} - k_i)$ and $\eta(z) = (2z - z_j - z_{j+1})/(z_{j+1} - z_j)$. Assume that the four nodes of the master element are $(-1, -1)$, $(1, -1)$, $(1, 1)$, and $(-1, 1)$

[12] Extensions to non-rectangular element shapes require additional work but are not as useful in economic problems as in engineering problems, which sometimes involve irregularly shaped domains. (See, for example, Hughes 1987 and Reddy 1993.)

using the local coordinates. In this case, the basis functions are constructed so that $c_e^n(\xi,\eta;\theta) = \sum_i \theta_i^e \psi_i^e(\xi,\eta)$ with $\theta_1^e = c_e^n(-1,-1;\theta)$, $\theta_2^e = c_e^n(1,-1;\theta)$, $\theta_3^e = c_e^n(1,1;\theta)$, and $\theta_4^e = c_e^n(-1,1;\theta)$. These restrictions imply that

$$
\begin{aligned}
c_e^n(\xi,\eta;\theta) = &\frac{1}{4}(1-\xi)(1-\eta)\,\theta_1^e + \frac{1}{4}(1+\xi)(1-\eta)\,\theta_2^e \\
&+\frac{1}{4}(1+\xi)(1+\eta)\,\theta_3^e + \frac{1}{4}(1-\xi)(1+\eta)\,\theta_4^e.
\end{aligned}
\tag{25}
$$

To attain a more accurate approximation, I can increase the number of elements while retaining linear basis functions or use higher-order polynomials. Consider, for example, quadratic functions in two dimensions. One simple way to construct these functions is to take the product of one-dimensional quadratic polynomials. A unique set of coefficients for the polynomial requires that there be nine nodes and, hence, nine interpolation functions. In this case, the approximation on the master element $[-1,1]\times[-1,1]$ is given by

$$
\begin{aligned}
c_e^n(\xi,\eta;\theta) = &\frac{1}{4}\xi(\xi-1)\eta(\eta-1)\,\theta_1^e + \frac{1}{4}\xi(\xi+1)\eta(\eta-1)\,\theta_2^e + \frac{1}{4}\xi(\xi+1)\eta(\eta+1)\,\theta_3^e \\
&+\frac{1}{4}\xi(\xi-1)\eta(\eta+1)\,\theta_4^e + \frac{1}{2}(1-\xi^2)\eta(\eta-1)\,\theta_5^e + \frac{1}{2}\xi(\xi+1)(1-\eta^2)\,\theta_6^e \\
&+\frac{1}{2}(1-\xi^2)\eta(\eta+1)\,\theta_7^e + \frac{1}{2}\xi(\xi-1)(1-\eta^2)\,\theta_8^e + (1-\xi^2)(1-\eta^2)\,\theta_9^e.
\end{aligned}
\tag{26}
$$

Example 8. Let $\tau = 1$, $\delta = 0$, $\beta = 0.95$, $\alpha = 0.33$, $\rho = 0.95$, and $\sigma = 0.1$. Assume that the partition on z is given by $[-0.391,\ -0.123,\ 0.123,\ 0.391]$ and that the partition on k is given by $[0,\ 0.010,\ 0.036,\ 0.102,\ 0.273,\ 0.714,\ 1.85]$. I set the number of quadrature points on each element to nine, that is, three points for integration with respect to the capital stock and three points for integration with respect to the technology shock. For integration over ν, I set the number of quadrature points, m_ν, equal to 10.

In Figure 14, I plot the approximate piecewise linear solution (marked with a square) along with the exact solution. Even though there are only 18 elements, it is hard to distinguish the two.

Example 9. Suppose that I use the same parameterization as in Example 8, but instead of linear basis functions, I use the quadratic functions in equation (26). In Figure 14, the solution is marked with a circle. Notice that the fit with quadratic bases is slightly better than that with linear bases – however, since the coarse piecewise linear approximation is very accurate, there is not much room for improvement.

5. Conclusions

My intent in this chapter was to demonstrate the potential power of weighted residual methods and to show that they are relatively easy to implement. I have shown that the main computational task is to solve a system of equations, $A\theta = b$, for θ – once if the problem is linear or several times if the problem is nonlinear. As with most algorithms, choices must be made when implementing a weighted residual method. In particular, a set of basis functions must be chosen to represent the approximate solution and a set of weight functions must be chosen for the weighted integral. I have discussed the most widely used basis and weight functions.

I have also distinguished the methods according to the type of basis functions used. Spectral methods use smooth functions, such as polynomials, that are nonzero on most of the domain. Finite element methods, however, use basis functions that are equal to zero on most of the domain and nonzero on only a few subdivisions of the domain. For the simple differential equation in Section 2, both types of basis functions generated very accurate solutions even with a small number of bases in the approximation.

The distinction becomes more obvious in higher-order or highly nonlinear problems. In these cases, a finite element method has advantages. With spectral methods, the system of equations that is solved for the unknown coefficients θ is dense because each element of θ will in general affect every residual equation. For problems with many state variables, there are typically many coefficients to compute. Inversion of a large, dense matrix may not be feasible. With the finite element method, however, A is sparse and its structure can typically be exploited. With spectral methods, the same functions are used on all regions of the state space. For problems that are nonlinear, such as the growth examples in Sections 3 and 4, it is better to use different approximations in different regions of the state space. For this reason, I have concluded that a finite element method may be better suited to problems in which the solution is nonlinear or kinked in regions where inequality constraints bind.

References

Aiyagari, S. Rao and Ellen R. McGrattan. 1997. The optimum quantity of debt. *Journal of Monetary Economics*, forthcoming.

Braun, Richard A. and Ellen R. McGrattan. 1993. The macroeconomics of war and peace. *NBER Macroeconomics Annual 1993*. Cambridge: MIT Press.

Chari, V.V., Patrick Kehoe, and Ellen R. McGrattan. 1997. The poverty of nations: A quantitative investigation. Staff Report #204, Federal Reserve Bank of Minneapolis.

Hughes, Thomas J.R. 1987. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Englewood Cliffs: Prentice-Hall

Judd, Kenneth. 1992. Projection methods for solving aggregate growth models, *Journal of Economic Theory* 58(2): 410-52.

McGrattan, Ellen. 1996. Solving the Stochastic Growth Model with a Finite Element Method. *Journal of Economic Dynamics and Control* 20: 19-42.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. 1986. *Numerical recipes: The art of scientific computing*. Cambridge: Cambridge University Press.

Reddy, J.N. 1993. *An introduction to the finite element method*. New York: McGraw-Hill.

Saad, Yousef. 1996. *Iterative Methods for Sparse Linear Systems*. Boston: PWS.

Sargent, Thomas J. 1987. *Dynamic Macroeconomic Theory*. Cambridge: Harvard University Press.

Taylor, John B. and Harald Uhlig. 1990. Solving nonlinear stochastic growth models: A comparison of alternative solution methods. *Journal of Business and Economic Statistics* 8: 1-17.