Math 6630: Numerical Solutions of Partial Differential Equations
Polynomial spectral methods, II

See Shen, Tang, and Wang 2011, Chapter 4,

Hesthaven, S. Gottlieb, and D. Gottlieb 2007, Chapters 7, 8

Akil Narayan[1]

[1] Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

April 12, 2023

Non-periodic problems

We have considered PDE's with periodic boundary conditions using Fourier Series.

Our main goal now is to modify our procedures appropriately for nonperiodic problems, e.g., ones of the form,

$$-u''(x) = f(x), \qquad u(-1) = u(1) = 0, \qquad x \in [-1, 1]$$

Our main technique to accomplish will be to approximate the solution as a polynomial.

Orthogonal polynomials are the main computational tool we'll employ.

- For general weight functions ω:
  - three-term recurrence
  - Gaussian quadrature through recurrence coefficients
- For "classical" weight functions ω:
  - Sturm-Liouville operator $\rightarrow$ Best L^2_ω convergence rates
  - Rodrigues' formula $\rightarrow$ explicit formulas for recurrence coefficients
- Classical weight functions yield Legendre, Hermite, Chebyshev, Gegenbauer, Jacobi, and Laguerre polynomial families

Non-periodic problems

We have considered PDE's with periodic boundary conditions using Fourier Series.

Our main goal now is to modify our procedures appropriately for nonperiodic problems, e.g., ones of the form,

$$-u''(x) = f(x), \qquad u(-1) = u(1) = 0,$$

Our main technique to accomplish will be to approximate the solution as a polynomial.

Orthogonal polynomials are the main computational tool we'll employ.

- For general weight functions ω:
  - three-term recurrence
  - Gaussian quadrature through recurrence coefficients
- For "classical" weight functions ω:
  - Sturm-Liouville operator $\rightarrow$ Best L^2_ω convergence rates
  - Rodrigues' formula $\rightarrow$ explicit formulas for recurrence coefficients
- Classical weight functions yield Legendre, Hermite, Chebyshev, Gegenbauer, Jacobi, and Laguerre polynomial families

Now we can address solving differential equations.

The high-level ideas

The particular tools for solving differential equations will not change.

In particular, we'll still use the weighted residual methods strategy:

- For (linear) stationary problems:
  - We introduce an appropriate bilinear form that yields a well-posed weak formulation for the PDE
  - We discretize the bilinear form via Galerkin or collocation methods (or Petrov-Galerkin methods)

The high-level ideas

The particular tools for solving differential equations will not change.

In particular, we'll still use the weighted residual methods strategy:

- For (linear) stationary problems:
  - We introduce an appropriate bilinear form that yields a well-posed weak formulation for the PDE
  - We discretize the bilinear form via Galerkin or collocation methods (or Petrov-Galerkin methods)

- For time-dependent problems:
  - We use "strong form" Galerkin/collocation procedures
  - We discretize the bilinear form via Galerkin or collocation methods (or Petrov-Galerkin methods)

The high-level ideas

The particular tools for solving differential equations will not change.

In particular, we'll still use the weighted residual methods strategy:

- For (linear) stationary problems:
  - We introduce an appropriate bilinear form that yields a well-posed weak formulation for the PDE
  - We discretize the bilinear form via Galerkin or collocation methods (or Petrov-Galerkin methods)
- For time-dependent problems:
  - We use "strong form" Galerkin/collocation procedures
  - We discretize the bilinear form via Galerkin or collocation methods (or Petrov-Galerkin methods)

Of particular note: all the theory we've covered applies quite directly, including Ceá's Lemma, Lax-Milgram, and stability for time-dependent problems.

Overall types of methods

We will use either Galerkin or collocation methods. Both of these approaches require defining a weight function $\omega(x)$.

Bilinear forms will be defined using the L^2_ω inner product, and so our trial/test spaces will be spanned by L^2_ω-orthonormal polynomials $p_0, p_1, \ldots$.

Overall types of methods

We will use either Galerkin or collocation methods. Both of these approaches require defining a weight function $\omega(x)$.

Bilinear forms will be defined using the L^2_ω inner product, and so our trial/test spaces will be spanned by L^2_ω-orthonormal polynomials $p_0, p_1, \ldots$.

We will describe methods with the Galerkin/collocation label along with the name of the basis corresponding to ω. E.g., :

- A Legendre-Galerkin method is a Galerkin approach where $\omega(x) = 1$, and we will use Legendre polynomials as a basis for the trial/test space.
- A Chebyshev-collocation method is a collocation approach where $\omega(x) = (1-x^2)^{-1/2}$, and we will use Chebyshev polynomials as a basis.

Overall types of methods

We will use either Galerkin or collocation methods. Both of these approaches require defining a weight function $\omega(x)$.

Bilinear forms will be defined using the L^2_ω inner product, and so our trial/test spaces will be spanned by L^2_ω-orthonormal polynomials $p_0, p_1, \ldots$.

We will describe methods with the Galerkin/collocation label along with the name of the basis corresponding to ω. E.g., :

- A Legendre-Galerkin method is a Galerkin approach where $\omega(x) = 1$, and we will use Legendre polynomials as a basis for the trial/test space.
- A Chebyshev-collocation method is a collocation approach where $\omega(x) = (1-x^2)^{-1/2}$, and we will use Chebyshevpolynomials as a basis.

The main new (and nontrivial) issue we must contend with are boundary conditions.

Three approaches for boundary conditions

For concreteness, we'll consider the following ODE as an example:

$$-u''(x) = f(x), \qquad\qquad u(-1) = u(1) = 0,$$

However, the procedures we describe next for handling boundary conditions apply to essentially any type of PDE.

Three approaches for boundary conditions

For concreteness, we'll consider the following ODE as an example:

$$-u''(x) = f(x), \qquad\qquad u(-1) = u(1) = 0,$$

However, the procedures we describe next for handling boundary conditions apply to essentially any type of PDE.

- Nonhomogeneous boundary conditions for linear problems: one solution is explicitly constructed to satisfy the boundary conditions, and other is constructed to satisfy the PDE.

Three approaches for boundary conditions

For concreteness, we'll consider the following ODE as an example:

$$-u''(x) = f(x), \qquad u(-1) = u(1) = 0,$$

However, the procedures we describe next for handling boundary conditions apply to essentially any type of PDE.

- Nonhomogeneous boundary conditions for linear problems: one solution is explicitly constructed to satisfy the boundary conditions, and other is constructed to satisfy the PDE.
- "Essential" handling of homogeneous boundary conditions: the trial basis is explicitly constructed to satisfy the boundary conditions.

Three approaches for boundary conditions

For concreteness, we'll consider the following ODE as an example:

$$-u''(x) = f(x), \qquad\qquad u(-1) = u(1) = 0,$$

However, the procedures we describe next for handling boundary conditions apply to essentially any type of PDE.

- Nonhomogeneous boundary conditions for linear problems: one solution is explicitly constructed to satisfy the boundary conditions, and other is constructed to satisfy the PDE.
- "Essential" handling of homogeneous boundary conditions: the trial basis is explicitly constructed to satisfy the boundary conditions.
- "Natural" handling of homogeneous boundary conditions: the weak form is constructed to weakly enforce boundary conditions.

Three approaches for boundary conditions

For concreteness, we'll consider the following ODE as an example:

$$-u''(x) = f(x), \qquad u(-1) = u(1) = 0,$$

However, the procedures we describe next for handling boundary conditions apply to essentially any type of PDE.

- Nonhomogeneous boundary conditions for linear problems: one solution is explicitly constructed to satisfy the boundary conditions, and other is constructed to satisfy the PDE.
- "Essential" handling of homogeneous boundary conditions: the trial basis is explicitly constructed to satisfy the boundary conditions.
- "Natural" handling of homogeneous boundary conditions: the weak form is constructed to weakly enforce boundary conditions.
- "Tau" handling of boundary conditions: trailing (high-frequency) cofficients are altered to satisfy the boundary conditions

We'll mainly describe nonhomogeneous boundary conditions and the essential boundary conditions strategy.

"Lifting" for non-homogeneous boundary conditions

Linearity is really nice.

Consider solving,

$$-u''(x) = f(x), \qquad u(-1) = 3, \qquad u(1) = -2,$$

One strategy to address such boundary conditions is to realize that

$$w(x) = -2\frac{1+x}{2} + 3\frac{1-x}{2} = -\frac{5}{2}x + \frac{1}{2},$$

satisfies the boundary conditions.

"Lifting" for non-homogeneous boundary conditions

Linearity is really nice.

Consider solving,

$$-u''(x) = f(x), \qquad u(-1) = 3, \qquad u(1) = -2,$$

One strategy to address such boundary conditions is to realize that

$$w(x) = -2\frac{1+x}{2} + 3\frac{1-x}{2} = -\frac{5}{2}x + \frac{1}{2},$$

satisfies the boundary conditions. Also: $w''(x) = 0$

With this information, we can define

$$U(x) := u(x) - w(x)$$

which implies that U satisfies the equation,

$$-U''(x) = f(x), \qquad U(-1) = U(1) = 0.$$

"Lifting" for non-homogeneous boundary conditions

Linearity is really nice.

Consider solving,

$$-u''(x) = f(x), \qquad u(-1) = 3, \qquad u(1) = -2,$$

One strategy to address such boundary conditions is to realize that

$$w(x) = -2\frac{1+x}{2} + 3\frac{1-x}{2} = -\frac{5}{2}x + \frac{1}{2},$$

satisfies the boundary conditions.

With this information, we can define

$$U(x) := u(x) - w(x)$$

which implies that U satisfies the equation,

$$-U''(x) = f(x), \qquad U(-1) = U(1) = 0.$$

One we solve this equation with *homogeneous* boundary conditions for U, then we simply define

$$u(x) = U(x) + w(x).$$

"Essential" treatment of boundary conditions

Because of the previous result we can, for many cases, specialize to homogeneous boundary conditions,

$$-u''(x) = f(x), \qquad u(-1) = u(1) = 0.$$

A standard Galerkin approach would utilize the trial/test space,

$$P_N = \operatorname{span}\{x^j \mid j = 0, \ldots, N\},$$

but $u \in P_N$ need not satisfy the boundary conditions.

"Essential" treatment of boundary conditions

Because of the previous result we can, for many cases, specialize to homogeneous boundary conditions,

$$-u''(x) = f(x), \qquad\qquad u(-1) = u(1) = 0.$$

A standard Galerkin approach would utilize the trial/test space,

$$P_N = \operatorname{span}\{x^j \mid j = 0, \ldots, N\},$$

but $u \in P_N$ need not satisfy the boundary conditions.

One way to address boundary conditions is to define a space so that functions must be from this space:

$$P_{N,0} = \left\{p \in P_N \;\middle|\; p(-1) = 0, \;\; p(1) = 0\right\}.$$

The dimension of this space is $\dim P_{N,0} = \dim P_N - 2 = N - 1$.

Building the boundary conditions explicitly into the approximation space is called an essential treatment of boundary conditions.

Essential boundary conditions

$$P_{N,0} = \left\{ p \in P_N \;\middle|\; p(-1) = 0, \;\; p(1) = 0 \right\}.$$

For concreteness in what follows, suppose we wish to derive a Legendre-Galerkin approach.

Essential boundary conditions

$$P_{N,0} = \left\{ p \in P_N \;\middle|\; p(-1) = 0, \;\; p(1) = 0 \right\}.$$

For concreteness in what follows, suppose we wish to derive a Legendre-Galerkin approach.

There are two popular computational ways to realize $P_{N,0}$. The first defines,

$$q_n(x) := p_n(x) - p_n(-1)\frac{1-x}{2} - p_n(1)\frac{1+x}{2}, \qquad n \geqslant 2,$$

where p_n are Legendre polynomials. Thus, $q_n(\pm 1) = 0$, and we take,

$$P_{N,0} = \operatorname{span}\left\{ q_n \;\middle|\; n = 2, \ldots, N \right\}$$

Essential boundary conditions

$$P_{N,0} = \left\{ p \in P_N \;\middle|\; p(-1) = 0, \;\; p(1) = 0 \right\}.$$

For concreteness in what follows, suppose we wish to derive a Legendre-Galerkin approach.

There are two popular computational ways to realize $P_{N,0}$. The first defines,

$$q_n(x) := p_n(x) - p_n(-1)\frac{1-x}{2} - p_n(1)\frac{1+x}{2}, \qquad n \geqslant 2,$$

where p_n are Legendre polynomials. Thus, $q_n(\pm 1) = 0$, and we take,

$$P_{N,0} = \operatorname{span}\left\{ q_n \;\middle|\; n = 2, \ldots, N \right\}$$

The second approach is similar, using instead high-degree Legendre polynomials,

$$r_n(x) := p_n(x) - \gamma_n p_{n-1}(x) - \delta_n p_{n-2}(x), \qquad n \geqslant 2,$$

where γ_n and δ_n are chosen to satisfy,

$$r_n(\pm 1) = 0 \quad \Longrightarrow \quad \begin{pmatrix} p_{n-1}(-1) & p_{n-2}(-1) \\ p_{n-1}(1) & p_{n-2}(1) \end{pmatrix} \begin{pmatrix} \gamma_n \\ \delta_n \end{pmatrix} = \begin{pmatrix} p_n(-1) \\ p_n(1) \end{pmatrix}.$$

Finally,

$$P_{N,0} = \operatorname{span}\left\{ r_n \;\middle|\; n = 2, \ldots, N \right\}$$

There are other options: e.g.

$$S_n(x) = (1-x^2)\,p_n(x), \quad n = 0, 1, 2, \ldots N-2$$

$$\text{span}\{S_n\}_{n=0}^{N-2} \subset P_N$$

A simple example

Let's construct a Legendre-Galerkin method for,

$$-u''(x) + u(x) = f(x), \qquad u(-1) = u(1) = 0.$$

To identify an appropriate bilinear form, we consider

$$H_0^1([-1,1]) := \left\{ u \in H^1([-1,1]) \;\middle|\; u(\pm 1) = 0 \right\}.$$

A simple example

Let's construct a Legendre-Galerkin method for,

$$-u''(x) + u(x) = f(x), \qquad\qquad u(-1) = u(1) = 0.$$

To identify an appropriate bilinear form, we consider

$$H_0^1([-1,1]) := \left\{ u \in H^1([-1,1]) \;\middle|\; u(\pm 1) = 0 \right\}.$$

With $\langle \cdot, \cdot \rangle$ the standard L^2 inner product on $[-1,1]$ and taking $u, v \in H_0^1([-1,1])$, we have

$$\left\langle -u'', v \right\rangle + \langle u, v \rangle = \langle f, v \rangle \quad \implies \quad \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle .$$

Note that the boundary term vanishes only because $v \in H_0^1$.

A simple example

Let's construct a Legendre-Galerkin method for,

$$-u''(x) + u(x) = f(x), \qquad\qquad u(-1) = u(1) = 0.$$

To identify an appropriate bilinear form, we consider

$$H_0^1([-1,1]) := \left\{ u \in H^1([-1,1]) \;\middle|\; u(\pm 1) = 0 \right\}.$$

With $\langle \cdot, \cdot \rangle$ the standard L^2 inner product on $[-1,1]$ and taking $u, v \in H_0^1([-1,1])$, we have

$$\left\langle -u'', v \right\rangle + \langle u, v \rangle = \langle f, v \rangle \quad \Longrightarrow \quad \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle .$$

Note that the boundary term vanishes only because $v \in H_0^1$.

Then the (infinite-dimensional) weak form is,

$$\text{Find } u \in H_0^1 \text{ such that } \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle \text{ for all } v \in H_0^1$$

A simple example

Let's construct a Legendre-Galerkin method for,

$$-u''(x) + u(x) = f(x), \qquad u(-1) = u(1) = 0.$$

To identify an appropriate bilinear form, we consider

$$H_0^1([-1,1]) := \left\{ u \in H^1([-1,1]) \;\middle|\; u(\pm 1) = 0 \right\}.$$

With $\langle \cdot, \cdot \rangle$ the standard L^2 inner product on $[-1,1]$ and taking $u, v \in H_0^1([-1,1])$, we have

$$\left\langle -u'', v \right\rangle + \langle u, v \rangle = \langle f, v \rangle \quad \Longrightarrow \quad \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle.$$

Note that the boundary term vanishes only because $v \in H_0^1$.

Then the (infinite-dimensional) weak form is,

$$\text{Find } u \in H_0^1 \text{ such that } \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle \text{ for all } v \in H_0^1$$

To discretize things, we'll use the trial/test space $P_{N,0}$ with basis q_n. Then our scheme is

$$\text{Find } u \in P_{N,0} \text{ such that } \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle \text{ for all } v \in P_{N,0}$$

Implementation

$$\text{Find } u \in P_{N,0} \text{ such that } \langle u', v' \rangle = \langle f, v \rangle \text{ for all } v \in P_{N,0}$$

This scheme can be implemented as follows:

$$u(x) = \sum_{j=2}^{N} \widehat{u}_j q_j(x),$$

and setting $v = q_n$ for each $n = 2, \ldots, N$, we obtain a linear system for the coefficients $\boldsymbol{u} = (\widehat{u}_j)_{j=2}^N$:

$$(\boldsymbol{S} + \boldsymbol{M})\,\boldsymbol{u} = \boldsymbol{f}, \qquad \boldsymbol{f} = (f_n)_{n=2}^N,$$

where the stiffness and mass matrices have entries, respectively,

$$(S)_{n,j} = \langle q_n', q_j' \rangle, \qquad (M)_{n,j} = \langle q_n, q_j \rangle, \qquad n, j = 2, \ldots, N.$$

Implementation

$$\text{Find } u \in P_{N,0} \text{ such that } \langle u', v' \rangle = \langle f, v \rangle \text{ for all } v \in P_{N,0}$$

This scheme can be implemented as follows:

$$u(x) = \sum_{j=2}^{N} \widehat{u}_j q_j(x),$$

and setting $v = q_n$ for each $n = 2, \ldots, N$, we obtain a linear system for the coefficients $\boldsymbol{u} = (\widehat{u}_j)_{j=2}^N$:

$$(\boldsymbol{S} + \boldsymbol{M})\,\boldsymbol{u} = \boldsymbol{f}, \qquad \boldsymbol{f} = (f_n)_{n=2}^N,$$

where the stiffness and mass matrices have entries, respectively,

$$(S)_{n,j} = \langle q_n', q_j' \rangle, \qquad (M)_{n,j} = \langle q_n, q_j \rangle, \qquad n, j = 2, \ldots, N.$$

There are two main options to compute these matrices:

- Analytical methods – use the three-term recurrence and some related properties to compute exact expressions.
- Computational methods – use Legendre-Gauss quadrature.
  - The integrand is a polynomial of degree at most $2N$, so an $N+1$-point rule is sufficient.
  - To evaluate q_n', we need to evaluate p_n' $\longrightarrow$ differentiate the three-term recurrence to obtain a modified recurrence relation that can evaluate $p_n'(x)$.

Implementation

$$\text{Find } u \in P_{N,0} \text{ such that } \langle u', v' \rangle = \langle f, v \rangle \text{ for all } v \in P_{N,0}$$

This scheme can be implemented as follows:

$$u(x) = \sum_{j=2}^{N} \widehat{u}_j q_j(x),$$

and setting $v = q_n$ for each $n = 2, \ldots, N$, we obtain a linear system for the coefficients $\boldsymbol{u} = (\widehat{u}_j)_{j=2}^N$:

$$(\boldsymbol{S} + \boldsymbol{M})\,\boldsymbol{u} = \boldsymbol{f}, \qquad \boldsymbol{f} = (f_n)_{n=2}^N,$$

where the stiffness and mass matrices have entries, respectively,

$$(S)_{n,j} = \langle q_n', q_j' \rangle, \qquad (M)_{n,j} = \langle q_n, q_j \rangle, \qquad n, j = 2, \ldots, N.$$

There are two main options to compute these matrices:

- Analytical methods – use the three-term recurrence and some related properties to compute exact expressions.
- Computational methods – use Legendre-Gauss quadrature.
  - The integrand is a polynomial of degree at most $2N$, so an $N+1$-point rule is sufficient.
  - To evaluate q_n', we need to evaluate p_n' $\longrightarrow$ differentiate the three-term recurrence to obtain a modified recurrence relation that can evaluate $p_n'(x)$.

In implementation, choosing q_n versus r_n as a basis does make a difference. E.g., choosing r_n in this case makes $\boldsymbol{S}$ diagonal.

Beyond essential boundary conditions

The previous approach works similarly for time-dependent problems.

We'll *briefly* describe alternative approaches. For things like Robin/mixed boundary conditions, e.g.,

$$\alpha u(1) + \beta u'(1) = 0, \qquad \alpha, \beta \neq 0,$$

then it's more appropriate to "naturally" enforce boundary conditions. In particular if u, v satisfy this condition, then via integration by parts:

$$\begin{aligned}
\left\langle -u'', v \right\rangle &= \left\langle u', v' \right\rangle - u'(x)v(x)\big|_{-1}^{1} \\
&= \left\langle u', v' \right\rangle - u'(1)v(1) + \dots \\
&= \left\langle u', v' \right\rangle + \frac{\alpha}{\beta} u(1)v(1) + \dots,
\end{aligned}$$

where $\dots$ corresponds to the boundary condition at $x = -1$.

Beyond essential boundary conditions

The previous approach works similarly for time-dependent problems.

We'll *briefly* describe alternative approaches. For things like Robin/mixed boundary conditions, e.g.,

$$\alpha u(1) + \beta u'(1) = 0, \qquad \alpha, \beta \neq 0,$$

then it's more appropriate to "naturally" enforce boundary conditions. In particular if u, v satisfy this condition, then via integration by parts:

$$\begin{aligned}\left\langle -u'', v \right\rangle &= \left\langle u', v' \right\rangle - u'(x)v(x)\big|_{-1}^{1} \\ &= \left\langle u', v' \right\rangle - u'(1)v(1) + \dots \\ &= \left\langle u', v' \right\rangle + \frac{\alpha}{\beta} u(1)v(1) + \dots,\end{aligned}$$

where $\dots$ corresponds to the boundary condition at $x = -1$.

Note that this bilinear form has constant (boundary) terms, but is still linear in both u and v.

An appropriate trial/test space would *not* enforce any conditions essentially at $x = +1$.

Instead, the scheme would enforce the condition through the bilinear form.

Such a treatment of boundary terms is a natural treatment of boundary conditions.

"Tau" methods

Another approach to handling boundary conditions is through a "tau" approach.

The main idea: strictly enforce a boundary condition in the scheme, but remove one weak Galerkin condition.

E.g., for our favorite problem,

$$-u''(x) = f(x), \qquad u(\pm 1) = 0,$$

then a(n unrecommended) tau scheme would be,

$$\text{Find } u \in P_N \text{ such that } \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle \text{ for all } v \in P_{N-2},$$

along with the conditions,

$$u(-1) = 0, \qquad u(+1) = 0.$$

I.e., the PDE weak form imposes $N-3$ conditions on expansion coefficients $\widehat{u}_n$ (because $\dim P_{N-2,0} = N-3$), and an additional 2 constraints on these coefficients through the above two conditions.

"Tau" methods

Another approach to handling boundary conditions is through a "tau" approach.

The main idea: strictly enforce a boundary condition in the scheme, but remove one weak Galerkin condition.

E.g., for our favorite problem,

$$-u''(x) = f(x), \qquad u(\pm 1) = 0,$$

then a(n unrecommended) tau scheme would be,

$$\text{Find } u \in P_N \text{ such that } \left\langle u', v' \right\rangle + \langle u, v \rangle = \langle f, v \rangle \text{ for all } v \in P_{N-2},$$

along with the conditions,

$$u(-1) = 0, \qquad u(+1) = 0.$$

I.e., the PDE weak form imposes $N-3$ conditions on expansion coefficients $\widehat{u}_n$ (because $\dim P_{N-2,0} = N-3$), and an additional 2 constraints on these coefficients through the above two conditions.

One generally should not use tau methods if there are better approaches (e.g., essential handling of boundary conditions).

However, tau methods are very useful for time-dependent problems with time-depedent boundary conditions, e.g., $u(1,t) = h(t)$.

Collocation methods

If using a collocation method, e.g., Legendre-collocation, for,

$$-u''(x) + u(x) = f(x), \qquad\qquad u(-1) = u(1) = 0,$$

with the strong form, then satisfying the boundary conditions in collocation form is easy if ± 1 are among our quadrature points.

Collocation methods

If using a collocation method, e.g., Legendre-collocation, for,

$$-u''(x) + u(x) = f(x), \qquad u(-1) = u(1) = 0,$$

with the strong form, then satisfying the boundary conditions in collocation form is easy if ± 1 are among our quadrature points.

Unfortunately, there are never any Gauss quadrature points lying on ± 1.

One can instead sacrifice quadrature accuracy for the freedom to place nodes at ± 1. This results in the Gauss-Lobatto rule. In particular, this quadrature rule satisfies,

$$\int_{-1}^{1} p(x)\omega(x)\mathrm{d}x = \sum_{j=1}^{N} w_j p(x_j), \qquad p \in P_{2N-3}, x_1 = -1, x_N = 1.$$

Collocation methods

If using a collocation method, e.g., Legendre-collocation, for,

$$-u''(x) + u(x) = f(x), \qquad u(-1) = u(1) = 0,$$

with the strong form, then satisfying the boundary conditions in collocation form is easy if ± 1 are among our quadrature points.

Unfortunately, there are never any Gauss quadrature points lying on ± 1.

One can instead sacrifice quadrature accuracy for the freedom to place nodes at ± 1. This results in the Gauss-Lobatto rule. In particular, this quadrature rule satisfies,

$$\int_{-1}^{1} p(x)\omega(x)\mathrm{d}x = \sum_{j=1}^{N} w_j p(x_j), \qquad p \in P_{2N-3}, x_1 = -1, x_N = 1.$$

If we use the Legendre-Gauss-Lobatto points $\{x_j\}_{j=1}^N$ for collocation, along with the Lagrange polynomial representation,

$$u(x) = \sum_{j=1}^{N} u_j \ell_j(x), \qquad \ell_j(x_k) = \delta_{j,k},$$

then the scheme:

- Enforces zero PDE residual at x_j for $j = 2, \ldots, N-1$
- Sets $u_1 = 0$, and $u_N = 0$.

The inverse transform (back to orthogonal polynomial coefficient space) on the Gauss-Lobatto points is not unitary, but it's well-conditioned.

References I


Hesthaven, Jan S., Sigal Gottlieb, and David Gottlieb (2007). *Spectral Methods for Time-Dependent Problems*. Cambridge University Press. ISBN: 0-521-79211-8.


Shen, Jie, Tao Tang, and Li-Lian Wang (2011). *Spectral Methods: Algorithms, Analysis and Applications*. Springer Science & Business Media. ISBN: 978-3-540-71041-7.