

Collocation Methods - TPBVP

we will discuss (briefly).

- The last major class of methods^A for solving the TPBVPs are collocation methods, in particular the pseudospectral (PS) method

- Model problem: Linear w/ Dirichlet B.C.'s.

$$p(x)y''(x) + q(x)y'(x) + s(x)y(x) = f(x)$$

$$a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta$$

- Similar to the Galerkin method we approximate the solution with a finite linear combination of some basis functions

$$\Rightarrow y(x) \approx y_N(x) = \sum_{j=0}^{N+1} c_j \phi_j(x)$$

and consider the residual

$$r_N(x) = p(x) \sum_{j=0}^{N+1} c_j \phi_j''(x) + q(x) \sum_{j=0}^{N+1} c_j \phi_j'(x) + s(x) \sum_{j=0}^{N+1} c_j \phi_j(x) - f(x)$$

We choose the coefficients c_j so that $y_N(x)$

1) Satisfies the boundary conditions: $y_N(a) = \alpha, \quad y_N(b) = \beta$

2) The residual is zero at N suitably chosen nodes x_j :

$$r_N(x_j) = 0 \quad j = 1, 2, \dots, N$$

- There are three popular choices for the ^{type of} basis functions ϕ_j to be used.

Collocation Methods - TPRVP

- 1) Trigonometric functions: These are the perfect choice for periodic problems and produce highly accurate results. ^(spectral accuracy) For the node points, one use equally spaced points.
- 2) splines: These are suitable for non-periodic problems and are good when the true solution is not that smooth. This method works for nodes spaced in any fashion.
- 3) Orthogonal polynomials: These are also suitable for non-periodic problems and produce highly accurate results (spectral accuracy) when the true solution is smooth. Common choices for ^{the} polynomials include: Chebyshev and Legendre. The node points are selected as either the roots or the extrema of the polynomials used.

- We will briefly discuss the collocation method based on the Chebyshev orthogonal polynomials. In the literature, this is called the Chebyshev Pseudospectral Method

- Chebyshev Pseudospectral method is viewed from finite difference

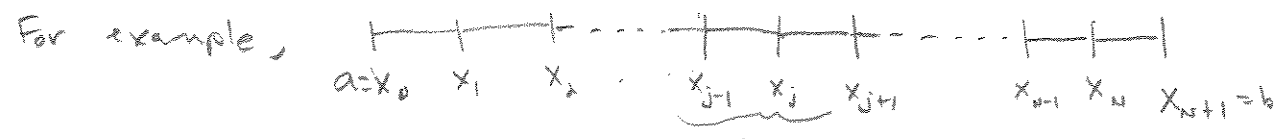
- We could proceed to discuss the method by a brute force approach on determining linear systems that arise from the requirement that the residual is zero at some node points.

Collocation Methods - TPBVP

- However, there is a much more elegant way to proceed that connects the method to the FD method. In fact, we will see that the collocation PS approach can be viewed as FD method taken to as high of accuracy as possible.

- Recall that the FD method works by replacing the continuous differential operators with discrete ^(approximate) derivative operators.

At each node (or grid point) in our domain, we approximate derivatives with FD formulas.



$$y'(x_j) = \frac{1}{h} \left[-\frac{1}{2}y(x_{j-1}) + \frac{1}{2}y(x_{j+1}) \right] + \mathcal{O}(h^2)$$

Recall:

We generated the formula by differentiating the quadratic polynomial that interpolated the function values $y(x_{j-1}), y(x_j), y(x_{j+1})$.

Similarly, for the second derivative we obtained

$$y''(x_j) = \frac{1}{h^2} [y(x_{j-1}) - 2y(x_j) + y(x_{j+1}))] + \mathcal{O}(h^2)$$

We then used these formulas to approximate the TPBVP at each interior node points:

$$= \left(\frac{p(x_j)}{h^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \frac{q(x_j)}{h} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} + r(x_j) \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \right) \begin{bmatrix} y(x_{j-1}) \\ y(x_j) \\ y(x_{j+1}) \end{bmatrix} = f(x_j)$$

$j=1, 2, \dots, N$

- Note that this is entirely the same as substituting the approximation

$$y(x) \approx y_2(x) = \sum_{k=0}^2 C_k x^k = C_0 + C_1 x + C_2 x^2$$

into the the TPBVP and requiring that the residual be exact for $x=x_j$. Thus, the FD method can be thought of as a kind of local collocation method.

- To get better accuracy with the FD method (we use higher degree polynomials to interpolate the $y(x_j)$ values and differentiate these. For example

$$y'(x_j) = \frac{1}{h} \begin{bmatrix} \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} \end{bmatrix} \begin{bmatrix} y(x_{j-2}) \\ y(x_{j-1}) \\ y(x_j) \\ y(x_{j+1}) \\ y(x_{j+2}) \end{bmatrix} + \mathcal{O}(h^4) \quad j=2,3,\dots,N-1$$

which comes from differentiating a 4th degree polynomial interpolant

- Suppose we wanted to push this idea to its limit. That is if we have a $N+2$ equally spaced nodes on $[a,b]$, ($h = \frac{b-a}{N+1}$, $x_j = a + jh$), we interpolate at each node point with an $N+1$ degree polynomial, differentiate it, and collect the terms in a FD formula.



$$y(x) = \sum_{k=0}^{N+1} L_k(x) y(x_k) \Rightarrow y'(x_j) = \sum_{k=0}^{N+1} \underbrace{L'_k(x_j)}_{\text{FD formula}} y(x_k) \quad j=0,1,\dots,N+1$$

$$-L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{N+1} (x-x_i) / \prod_{\substack{i=0 \\ i \neq k}}^{N+1} (x_k-x_i) \quad \text{(Lagrange basis functions)}$$

- What is wrong with this idea?

- How can we fix it?

- Chebyshev nodes: $x_k = -\cos(\pi k / (N+1))$ $k=0, 1, 2, \dots, N+1$

Legendre nodes: x_k are the roots (or extrema) of the $N+2$ degree Legendre polynomial
(there is no closed form solution for these)

- Once we have computed all the values $L_k'(x_j)$ $k=0, 1, \dots, N+1$
 $j=0, 1, \dots, N+1$

We collect these in a differentiation matrix D_{N+1} just like we did for the FD method. Thus, we can replace the continuous $\frac{d}{dx}$ operator with D_{N+1}

- Fortunately, we can write down the entries in D_{N+1} in closed form for all nodes.

Let $w_k = \prod_{j=0}^{N+1} (x_k - x_j)$, then

$$D_{ij} = \frac{1}{w_j} \prod_{\substack{k=0 \\ k \neq i,j}}^{N+1} (x_i - x_k) = \frac{w_i}{w_j (x_i - x_j)} \quad (i \neq j)$$

$$D_{jj} = \sum_{\substack{k=0 \\ k \neq j}}^{N+1} (x_j - x_k)^{-1}$$

- (A similar analysis can be done for any order derivative)

Collocation methods - TPBVP

- Using the notation from above, we can solve the TPBVP

$$p(x)y''(x) + q(x)y'(x) + s(x)y(x) = f(x) \quad a \leq x \leq b$$

$$y'(a) = \alpha$$

$$y(b) = \beta$$

The first step is to change variables in the differentiation matrix formulation. ^(DM)

Since the chebyshev nodes are defined over $[-1, 1]$, we must change the variable so that the DMs are defined over $[a, b]$

If t is the variable for $[-1, 1]$, then

$$x = \frac{b-a}{2}t + \frac{b+a}{2}, \quad t = \frac{1}{b-a}(2x - b - a)$$

$$\Rightarrow \frac{d}{dx}y(x) = \frac{d}{dt}y(t) \frac{dt}{dx} = \frac{2}{b-a}y'(t) \quad \& \quad \frac{d^2}{dx^2}y(x) = \left(\frac{2}{b-a}\right)^2 y''(t)$$

Thus, we can write the chebyshev PS approximation to (*)

by
$$x_k = \frac{b-a}{2}(-\cos(\frac{\pi(k-1)}{N+1})) + \frac{b+a}{2} \quad k=1, 2, \dots, N+2$$

$$\left[\left(\frac{2}{b-a}\right)^2 \tilde{P} \tilde{D}_N^{(2)} + \frac{2}{b-a} \tilde{Q} \tilde{D}_N + S \right] \underline{y} = \underline{f} - \left(\frac{2}{b-a}\right)^2 \tilde{P} \text{diag}(D_N^{(2)}(2:N, 1)) \alpha$$

$$- \left(\frac{2}{b-a}\right) \tilde{Q} \text{diag}(D_N(2:N, 1)) \beta - \left(\frac{2}{b-a}\right)^2 \tilde{P} \text{diag}(D_N^{(2)}(2:N, N+2)) \alpha$$

$$- \left(\frac{2}{b-a}\right) \tilde{Q} \text{diag}(D_N(2:N, N+2)) \beta$$

Where $P = \text{diag}(p(x_k))$
 $Q = \text{diag}(q(x_k)) \quad k=2, 3, \dots, N+1$
 $S = \text{diag}(s(x_k))$

- Example: Consider the TPBVP used to test the Galerkin FE method:

$$-(1-x)y' - (1-x)\pi^2 y = -\pi(3x-2)\sin(\pi x) + \cos(\pi x)$$

$$\Leftrightarrow (x-1)y'' + y' + (x-1)\pi^2 y = -\pi(3x-2)\sin(\pi x) + \cos(\pi x)$$

$$0 \leq x \leq 1, \quad y(0) = 0, \quad y(1) = -1$$

So, $p(x) = x-1$, $q(x) = 1$, $S(x) = x-1$, $f(x) = \cos(\pi x) - \pi(3x-2)\sin(\pi x)$

The exact solution is $y(x) = x\cos(\pi x)$.

N	$\ PS \text{ Error}\ _{\infty}$	$\ Galerkin \text{ FE}\ _{\infty}$
5	1.14×10^{-4}	3.73×10^{-2}
10	1.18×10^{-9}	1.09×10^{-2}
15	5.72×10^{-15}	5.94×10^{-3}
•	machine precision	
•		
•		
•		
22,000		6.57×10^{-9}

cannot get lower than this value do to roundoff errors.

- We see that the Chebyshev (PS) results in a far more accurate answer with far fewer nodes than the piecewise linear Galerkin FE method. This is because the PS method converges spectrally (or exponentially) whereas the Galerkin converges like $O(h^2)$.

- Note that the PS method requires inverting a full $N \times N$ matrix, whereas the Galerkin required inverting a tridiagonal matrix.

\Rightarrow FLOPs PS $\sim O(N^3)$ FLOPs Galerkin $\sim O(N)$ \rightarrow For this problem we see the PS method is more efficient & stable.