Jurnal Komputer, Informasi dan Teknologi Vol: 4, No 2, 2024, Page: 1-7

rihi Mesh Placement Algorithm In Collocation Methods

Edy Hermansyah[1], Annisa F. Edriani[2]

[1,2] Universitas Dehasen Bengkulu

DOI: https://doi.org/10.53697/jkomitek.v4i2.1852

*Correspondence: Edy Hermansyah
Email: edhermanz@unived.ac.id

Received: 11-10-2024
Accepted: 13-11-2024
Published: 15-12-2024

Abstract: Various adaptive mesh selection strategies for solving single higher order two-points boundary value problems (BVPs) by using collocation methods are intensively investigated for along time and they are now well established. In this work we concern with numerical investigations of adaptive mesh selection algorithms using the criterion function rihi for solving first order system of BVPs and developing some algorithms. The algorithms perform quite nicely and appear competitive with De Boor algorithm.

Keywords: Collocation Solution, Mesh Selection, First Order System Bvps, Criterion Functions.

Introduction

The first order linear system of n differential equations considered is

$$\mathbf{x}'(t) = A(t)\,\mathbf{x}(t) + \mathbf{q}(t), \quad a < t < b \qquad \ldots(1)$$

where $\mathbf{x}(t)$ and $\mathbf{q}(t)$ are vector valued functions and $A(t)$ is a matrix (nxn).

The differential equation (1) has n associated homogeneous boundary conditions. The interval $[a,b]$ is subdivided in to w subintervals by partition π : $a=t_1 < t_2 < \ldots < t_{w+1}=b$. For each subinterval $I_j = [t_j,t_{j+1}]$ there are q collocation points which are chosen as $\xi_{jk} = h_k\xi^*_j+(t_k+t_{k-1})$, for $j = 1, 2, \ldots, q$; $k = 1, 2, \ldots, w$; where ξ^*_j, $j = 1, 2, \ldots, q$, are given reference points in the interval $[-1,1]$. These reference points will be the zeros of some orthogonal polynomials, particularly Chebyshev or Gauss points. The approximate solution $\mathbf{x}_{wq}$ is a polynomial with degree less than $(q+1)$ in each subinterval. If $\mathbf{x}$ denotes the exact solution, the residual $\mathbf{r}_{wq}$ and the error $\mathbf{e}_{wq}$ then can be defined by $\mathbf{r}_{wq} = (D-T)\,\mathbf{x}_{wq} - \mathbf{y}$ and $\mathbf{e}_{wq} = \mathbf{x}_{wq} - \mathbf{x}$.

In mesh selection strategies the aim is to determine w such that w is sufficiently small and the approximate solution satisfies some tolerance TOL. There are two main types in constructing mesh selection strategies, i.e. mesh placements/equidistributing and mesh subdivision algorithms(Anitescu, 2019; Cuomo, 2022; Nazeer, 2022; Parks, 2015; Samaniego, 2020). This paper concerns with mesh selection algorithms based on multiplication the residual $\mathbf{r}_i$ and its associated subinterval h_i, where $1 \le i \le w$. This criterion will be used in two types of strategies.

Methodology

Adaptive Mesh Selection Algorithm

The algorithms for adaptive refinement of a mesh and a redistribution of meshes can be distinguished into two types, firstly equidistribution / mesh placement algorithms where a new mesh in chosen at each stage so that some criterion function has the same value in each subinterval. This was considered in details by De Boor [3] and Pereyra & Sewell [6]. The second one is called mesh subdivision algorithm where additional knots are inserted into a given mesh. For solving single higher order boundary value problems, a number of criterion functions have been suggested. Some of these aim to reflect some measure of smoothness of the approximate solution, for example the magnitude of a particular derivative in each subinterval. White suggested the use of arc-length for this purpose(Darby, 2011; Nguyen, 2015; Patterson, 2014). Other criterion functions relate to some measure of error. Carey & Humphrey in [7] suggest using an estimate of the maximum residual. In [3] De Boor derives a criterion function based on the error analysis, which is discussed in details in [4]. Wright [9] investigated the criterion function based on error estimates derived in [1]. The following facts about collocation approximation process are described more explicitly and proved in [2] and [8].

For $t \in (t_j , t_{j+1})$, the global error $|| e ||_j$ in some cases is known that for some $d > q$ it satisfies the local inequality :

$$|| \mathbf{e} ||_j = || \mathbf{x}_{wq}(t) - \mathbf{x}(t) ||_j \le C h_j^{q+1} || \mathbf{x}^{(q+1)}(t) ||_j + O(h_j^{q+2}) + O(h^d)$$

and for the mesh points it satisfies :

$$|| \mathbf{x}_{wq}(t_j) - \mathbf{x}(t_j) ||_j \le O(h^d) ,\ 1 \le j \le w+1,$$

where $h = \max h_i = \max |t_{i+1} - t_i |$, C is a constant.

A closer examination of the error reveals that the following equality :

$$|| \mathbf{x}_{wq}(t) - \mathbf{x}(t) ||_j = C h_j^{q+1} || \mathbf{x}^{(q+1)}(t) ||_j + O(h^{q+2}) \quad \ldots(2)$$

Mesh Placement Algorithms

For comparison purpose, we shall briefly take a look de Boor's idea [3] in constructing an adaptive mesh placement algorithm and afterward we introduce a criterion function involved the residual. The bound (2) above implies that, for sufficiently small h: $||\mathbf{e}|| = ||\mathbf{x}_{wq} - \mathbf{x}|| \le C \max h_j^{q+1} ||\mathbf{x}^{(q+1)}(t)||_j$ and therefore suggest that break points $t_2, t_3, \ldots, t_w$ be placed so as to minimize the local terms $\max h_j^{q+1} || \mathbf{x}^{(q+1)}(t)||_j$. This can be achieved by requiring $h_j^{q+1} || \mathbf{x}^{(q+1)}(t)||_j$ = constant, $j=1,2,\ldots,w$. which is equivalent to determining $t_2, t_3, \ldots, t_w$, so that

$$h_j || \mathbf{x}^{(q+1)}(t)||_j^{1/(q+1)} = \text{constant}, \quad j = 1, 2, \ldots, w \quad \ldots(3)$$

and produces therefore asymptotically the same distribution of t_i's as the problem of determining $t_2, t_3, \ldots, t_w$ so that

$$\int_{t_i}^{t_{i+1}} || \mathbf{x}^{(q+1)}(t)||^{(1/(q+1))}\, dt = (1/w) \int_a^b || \mathbf{x}^{(q+1)}(t)||^{(1/(q+1))}\, dt \quad \ldots(4)$$

This latter problem is very easy to solve if $|| \mathbf{x}^{(q+1)}(t) ||$ is replaced by piecewise constant approximation $DB(t) = || \mathbf{x}^{(q+1)}(t) ||$. For then $I(t) = \int_a^t DB^{(1/(q+1))}\,dt$ is an easily computable continuous and monotone increasing piecewise linear function, hence so is I^{-1}. By evaluating the piecewise linear function I^{-1} at the *(w-1)* points we have mesh points :

$$t_i = I^{-1}(\, i\, I(b)/w), \quad i = 1,2, \ldots, w-1 \qquad \ldots(5)$$

De Boor also suggests a numerical approach to obtain piecewise constant approximation DB(t) using values from neighbouring subintervals.

The equation (3) can be regarded as definition of equidistributed mesh points and formally defined as follows

Definition : A mesh points is called asymptotically equidistributing with respect to the function T(t) if and only if $h_j \,|| \mathbf{T}(t) ||_j$ = constant, $j = 1, 2, \ldots, w$.

From this point of view, since the residual $\mathbf{r}(t)$ is local in nature, it is reasonable to attempt to equidistribute the meshes with respect to $\mathbf{r}(t)$:

$$h_j \,|| \mathbf{r}(t) ||_j = \text{constant}, \; j = 1, 2, \ldots, w \qquad \ldots(6)$$

Moreover, in constructing error estimate E_3 we have to evaluate the residuals and construct an approximate $\mathbf{r}^*$. Hence it can be used it to approximate $|| \mathbf{r}(t) ||$. Another reason is that an error estimate can be derived using the residual as follows :

The analysis for collocation at Gauss points yields

$\mathbf{e}^{(j)}(t) = \mathbf{x}_{wq}{}^{(j)}(t) - \mathbf{x}^{(j)}(t) = O(h^{q-j}),\ 0 \le j \le (q+1)$.

The error can be expressed as

$$\mathbf{e}(t) = \sum_{i=1}^{w} \int_{t_i}^{t_{i+1}} G(t,s)\mathbf{r}(s)ds,$$

where $\mathbf{r}(s)$ is the residual and $G(t,s)$ is the Green's function.

Using the fact that the residual $\mathbf{r}$ is zero at the collocation points t_{ij}, we have the relationship

$$\mathbf{r}(t) = (\mathbf{x}^{(q+1)}(t_{i+1/2}) / (q!) \prod_{j=1}^{q} (t - t_{ij}) + O(h^{q+1}),$$

solving this equation for $\mathbf{x}^{(q+1)}$ in term of $\mathbf{r}(t_{i+1/2})$ and substituting into (2) gives an error estimate

$$|| \mathbf{e} ||_j = C \,|| \mathbf{r}(t_{i+1/2}) ||_j \, h_j + O(h^{q+2}) \qquad \ldots(7)$$

Obviously, equiditributing equation (7) does introduce a mesh selection algorithm with respect to function $\mathbf{r}_i(t) = \mathbf{r}(t_{i+1/2})$, hence equation (6) is a more general form of equation (7). The equation (6) will be used in constructing mesh adaptive algorithm and this procedure will be called RH algorithms. Note that direct attempt to equidistribute the local terms in equation (6) by using de Boor procedure, the earlier numerical experiments show that it fails to produce a sensible results. This is not surprising since the residuals $\mathbf{r}_j(t)$ also depend on size of h_j and it can be related with h_j by : $|| \mathbf{r} ||_j = k_j h_j^s$, for some constant integer *s*. Multiplying both side by h_j gives $||\mathbf{r}||_j h_j = k_j h_j^{s+1}$. Therefore, equidistributing (6) is equivalent to equidistributing $k_j h_j^{s+1}$.

Having computed $\mathbf{r}$ on initial mesh points π, a new partition π^* π^* : $a = t^*_1 < t^*_2 < t^*_3 < \ldots < t^*_{w^*+1} = b$, where $w^* > w$ producing a more accurate solution is desired.

Suppose that $h^*_i = |t^*_{i+1} - t^*_i|$. In order to equidistribute (6) then we have
$k_1 h^{*s+1}_1 = k_2 h^{*s+1}_2 = \ldots = k_w h^{*s+1}_w$,
hence $(h^*_1 / h^*_2) = (k_2 / k_1)^{1/(s+1)}$, $(h^*_1 / h^*_3) = (k_3 / k_1)^{1/(s+1)}$, …, $(h^*_1 / h^*_w) = (k_w / k_1)^{1/(s+1)}$.
By taking $(k_j / k_1)^{1/(s+1)}$ as slopes of the piecewise linear constant RH and the slope in the first subinterval is set to be one, then we have RH as follows :

$$RH = \begin{cases} t - t_1 & a < t < t_2 \\ h_1 + (k_2 / k_1)^{1/(s+1)} (t - t_2) & t_2 < t < t_3 \\ \vdots & \\ h_1 + (k_2 / k_1)^{1/(s+1)} h_2 + \ldots + (k_w / k_1)^{1/(s+1)} (t - t_w), & t_w < t < b \end{cases}$$

where $k_i = ||\mathbf{r}||_i / h_i^s$ $i = 1, 2, \ldots, w$. The new mesh points are determined by evaluating RH^{-1} at the $(w^* - 1)$ points $(i\, RH(b) / w^*)$.

Mesh Subdivision Algorithms

In this procedure it is expected that the subinterval with maximum $||\mathbf{e}_c||$ determined using some criterion function gives maximum effect on the error $||\mathbf{e}||$. The procedure can be described as follows.

1. Solve the bvps using a crude initial mesh points
2. Evaluate the criterion $||\mathbf{e}_c||_j$, $j = 1, 2, \ldots, w$
3. Searching for the subinterval which has maximum $||\mathbf{e}_c||_j$
4. Halve this subinterval
5. Repeat first step till either $||\mathbf{e}_{wq}|| <$ a desired tolerance TOL or $w > w_{max}$.

To make a clearer comparison, we slightly modify de Boor's algorithm by searching subinterval which has maximum $h_j\, ||\mathbf{x}^{(q+1)}(t)||_j^{1/(q+1)}$, and $\mathbf{x}^{(q+1)}(t)$ is approximated by piecewise DB. This procedure is called De Boor's mesh subdivision algorithm. For the RH mesh subdivision algorithm, it is simply by searching subinterval which has maximum h_j $||\mathbf{r}(t)||_j$.

Result and Discussion

In this section we demonstrate the practicality of our criterion function RH for both types mesh placement and mesh subdivision algorithms.

Two first order system test problems representing of some badly behaved boundary value problems have been chosen. These are

BVP 1 : $$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\sigma \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\sigma \end{bmatrix} \qquad \text{BCs : } x_1(0) = x_1(2) = 0$$

BVP 2 : $$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \sigma & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma\cos^2(\pi t) + 2\pi^2\cos(2\pi t) \end{bmatrix} \qquad \text{BCs : } x_1(0) = x_1(1) = 0$$

The BVP 1 has a severe layer at the left-hand boundary. In the meantime, the problem 2 has two layers at both sides. Implementing RH mesh placement algorithm, the table 1 displays the numerical results for problem 1 with σ = 100 and max number of subinterval w_{max} = 100.

Table 1

q	2		3				4				5				
TOL	E-2	E-3	E-2	E-3	E-4	E-5	E-2	E-3	E-4	E-5	E-2	E-3	E-4	E-5	E-8
S=1	33	54	21	26	38	58	18	21	-	-	15	15	21	-	-
S=2	36	73	16	18	32	56	13	15	18	29	12	12	15	19	45
S=q	36	73	15	20	33	61	9	13	18	25	8	11	12	15	40
S=q+1	39	77	14	19	30	57	9	12	17	27	8	9	11	15	37
S=q+2	38	78	14	18	35	48	9	13	17	28	9	9	10	15	36
deBoor	32	73	13	22	41	78	10	13	17	30	9	9	15	17	56

By varying s, the results indicate that the RH mesh placement algorithm performs very well for s = q+1 and s = q+2, though for small q, it gives a better results if s = 1, but it completely fails for q = 4 and TOl = 10^{-4}. The results also show that the RH algorithm gives a better results compare to those using De Boor algorithm, especially if the number of collocation points *q* increases, as shown for q = 5 and TOL = 10^{-8}, where the RH algorithm's performance is much better compare to De Boor's algorithm. For smaller *q*'s De Boor algorithm might be better, since the RH algorithm rely on accuracy in approximating **r,** which is more reliable if *q* is sufficiently big.

Results for problem 2 with σ = 400 and w_{max} = 100 is given by following table :

Table 2

q	2			3				4				5				
TOL	E-2	E-3	E-4	E-2	E-3	E-4	E-5	E-2	E-3	E-4	E-5	E-2	E-3	E-4	E-5	E-8
S=1	24	43	81	-	-	-	-	6	-	-	-	17	-	-	-	-
S=q	24	43	81	10	17	29	49	6	9	13	21	5	6	8	12	37
S=q+1	24	43	81	9	17	28	49	6	9	13	20	5	6	8	12	36
S=q+2	24	43	81	9	17	28	49	6	9	13	20	5	6	8	12	36
deboor	26	67	-	12	22	38	76	7	10	16	24	5	7	10	15	49

As in the problem 1, the RH algorithm demonstrates a better performance compare to De Boor's algorithm. It is interesting to note that for q = 4 and q = 5, both algorithms produce the similar results, though De Boor needs more subintervals for TOL = 10^{-8}.

Similar results can be seen in the table 3 and table 4 where we employ the RH and De Boor mesh subdivision algorithms for both problems.

Table 3

	RH	de boor	TOL / q
w:	48	37	1.0e-2 / 2
	67	98	1.0e-3 / 2
	16	16	1.0e-2 / 3
	24	24	1.0e-3 / 3
	38	40	1.0e-4 / 3
	62	64	1.0e-5 / 3
	12	13	1.0e-2 / 4
	15	17	1.0e-3 / 4
	21	21	1.0e-4 / 4
	30	32	1.0e-5 / 4
	9	9	1.0e-2 / 5
	12	13	1.0e-3 / 5
	13	15	1.0e-4 / 5
	19	22	1.0e-5 / 5
	47	67	1.0e-8 / 5

Table 4

	RH	de boor	TOL / q
w:	26	28	1.0e-2 / 2
	54	84	1.0e-3 / 2
	14	14	1.0e-2 / 3
	26	24	1.0e-3 / 3
	32	32	1.0e-4 / 3
	62	62	1.0e-5 / 3
	8	8	1.0e-2 / 4
	12	12	1.0e-3 / 4
	16	16	1.0e-4 / 4
	30	30	1.0e-5 / 4
	6	6	1.0e-2 / 5
	8	8	1.0e-3 / 5
	12	12	1.0e-4 / 5
	18	18	1.0e-5 / 5
	52	56	1.0e-8 / 5

These tables display the numerical results for s = q+1. For problem 1 the results shows that apart for q = 2 and TOL = 10^{-2}, in term of number of subintervals needed in most cases the RH algorithm's performance is more competitive than De Boor's algorithm, at least it is as good as De Boor algorithm performance. The similar results are indicated by table 4 for problem 2. It is notable that for q = 2 and Tol = 10^{-3} , the RH algorithm gives a significant improvement for two problems.

Conclusion

The numerical results show that in most cases the criterion rihi appears to be a competitive algorithm in both type of strategies: mesh placement and mesh subdivision.

References

Ahmed, AH&Wright, K. 1986 Error Est for Coll Sol of Lin. ODEs. Comp. and Maths with Applics 12B

Ascher,UM&Mattheij,RMM&Russell, RD, 1988 Numerical Solution of BVPs for ODEs, Prentice Hall.

de Boor, C. 1973 Good Approx by Spline with Variable Knots II, Conference on the Solution of DEs, lecturer note in mathematics, Springer Verlag, New York.

de Boor, C. & Swartz, B. 1973 Collocation at Gaussian points, SIAM J. Num.Anal., 18

Hermansyah, E. 2001 An Investigation of Collocation Algorithms for Solving Boundary Value Problems for System of ODEs, Ph.D. thesis, University of Newcastle upon Tyne, 2001.

Pereyra, V. & Sewell, 1974 Mesh Selection for Discrete Solution of BVPs, J. Num. Math 23

Russell, RD & Christiansen, J. 1978 Adaptive Mesh Selection Strategies for Solving BVPs, SIAM J

Num. Anal. 15

Russell, R.D & Shampine, L.F 1972 A Collocation Method for BVPs, J. Num. Math 19

Wright, K & Ahmed, AH & Seleman. 1991 Mesh Selection in Collocation for BVPs, IMA J. Num. Analysis 11.

Anitescu, C. (2019). Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials and Continua*, *59*(1), 345–359. https://doi.org/10.32604/cmc.2019.06641

Cuomo, S. (2022). Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *Journal of Scientific Computing*, *92*. https://doi.org/10.1007/s10915-022-01939-z

Darby, C. L. (2011). An hp-adaptive pseudospectral method for solving optimal control problems. *Optimal Control Applications and Methods*, *32*(4), 476–502. https://doi.org/10.1002/oca.957

Nazeer, M. (2022). Theoretical study of MHD electro-osmotically flow of third-grade fluid in micro channel. *Applied Mathematics and Computation*, *420*. https://doi.org/10.1016/j.amc.2021.126868

Nguyen, V. P. (2015). Isogeometric analysis: An overview and computer implementation aspects. *Mathematics and Computers in Simulation*, *117*, 89–116. https://doi.org/10.1016/j.matcom.2015.05.008

Parks, D. H. (2015). CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, *25*(7), 1043–1055. https://doi.org/10.1101/gr.186072.114

Patterson, M. A. (2014). GPOPS - II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software*, *41*(1). https://doi.org/10.1145/2558904

Samaniego, E. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, *362*. https://doi.org/10.1016/j.cma.2019.112790