Multiple Shooting Revisited

M.R. Osborne

Mathematical Sciences Institute
Australian National University

Optimization and Data Analysis

Outline

Basic multiple shooting

The basic problem addressed by the multiple shooting method is the solution of the linear boundary value problem

$$\frac{d\mathbf{x}}{dt} = M(t)\mathbf{x} + \mathbf{f}(t),$$
$$B_1\mathbf{x}(0) + B_2\mathbf{x}(1) = \mathbf{b}.$$

Let $X(t, t_i)$ be the fundamental matrix satisfying the condition $X(t_i, t_i) = I$ where $0 = t_1 < t_2 < \cdots < t_n = 1$ defines a mesh on $[0, 1]$. Then the basic equation to be solved is:

$$\begin{bmatrix} -X(t_2,t_1) & I & & \\ & \ddots & & \\ & & -X(t_n,t_{n-1}) & I \\ B_1 & & & B_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{n-1} \\ \mathbf{b} \end{bmatrix}$$

where the $\mathbf{v}_i$ correspond to particular integral terms.

Calculation of the $X(t_{i+1}, t_i)$

This requires both an algorithm for integrating the ODE (assume it is of order m), and an algorithm for setting the $\{t_i\}$.

- Use an initial value solver. Sufficient to choose t_{i+1} such that $\|X(t_{i+1}, t_i)\| \leq K$.

Calculation of the $X(t_{i+1}, t_i)$

This requires both an algorithm for integrating the ODE (assume it is of order m), and an algorithm for setting the $\{t_i\}$.

- Use an initial value solver. Sufficient to choose t_{i+1} such that $\|X(t_{i+1}, t_i)\| \leq K$.
- By use of collocation as in, for example, Colsys.

Calculation of the $X(t_{i+1}, t_i)$

This requires both an algorithm for integrating the ODE (assume it is of order m), and an algorithm for setting the $\{t_i\}$.

- Use an initial value solver. Sufficient to choose t_{i+1} such that $\|X(t_{i+1}, t_i)\| \leq K$.
- By use of collocation as in, for example, Colsys.
- By use of a finite difference discretization. For example, the trapezoidal rule.

Calculation of the $X(t_{i+1}, t_i)$

This requires both an algorithm for integrating the ODE (assume it is of order m), and an algorithm for setting the $\{t_i\}$.

- Use an initial value solver. Sufficient to choose t_{i+1} such that $\|X(t_{i+1}, t_i)\| \le K$.
- By use of collocation as in, for example, Colsys.
- By use of a finite difference discretization. For example, the trapezoidal rule.

Methods which attempt an a priori set up of the MS matrix must rely on general properties of the ODE in setting the mesh points $\{t_i\}$. In the case of a mix of fast and slow solutions this is like using a non-stiff solver on a stiff problem.

Calculation of the $X(t_{i+1}, t_i)$

This requires both an algorithm for integrating the ODE (assume it is of order m), and an algorithm for setting the $\{t_i\}$.

- Use an initial value solver. Sufficient to choose t_{i+1} such that $\|X(t_{i+1}, t_i)\| \leq K$.
- By use of collocation as in, for example, Colsys.
- By use of a finite difference discretization. For example, the trapezoidal rule.

Methods which attempt an a priori set up of the MS matrix must rely on general properties of the ODE in setting the mesh points $\{t_i\}$. In the case of a mix of fast and slow solutions this is like using a non-stiff solver on a stiff problem.

Is it possible to find finite difference discretizations which behave in a stiffly stable manner in the context of a mix of fast and slow solutions?

Optimal boundary conditions

Intuitively fast solutions should be fixed at $t = 1$, slow at $t = 0$. Expect good boundary conditions should lead to a relatively well conditioned linear system. Consider factorization:

$$\left[\begin{array}{cccc|c} I & & & & -X_1 \\ -X_2 & I & & & \\ & & \ddots & & \\ \hline & & -X_{n-1} & I & 0 \end{array}\right] \rightarrow Q \left[\begin{array}{ccc|c} & U & & V \\ \hline 0 & \cdots & H & G \end{array}\right]$$

This step is independent of the boundary conditions. Must solve system with matrix $\begin{bmatrix} H & G \\ B_2 & B_1 \end{bmatrix}$ in order to compute $\mathbf{x}_1$, $\mathbf{x}_n$. Let

$$\begin{bmatrix} H & G \end{bmatrix} = \begin{bmatrix} L & 0 \end{bmatrix} \begin{bmatrix} S_1^T \\ S_2^T \end{bmatrix} \Rightarrow \begin{bmatrix} B_2 & B_1 \end{bmatrix} = S_2^T.$$

Bob Mattheij's example

Consider the differential system defined by

$$M(t) = \begin{bmatrix} 1 - 19\cos 2t & 0 & 1 + 19\sin 2t \\ 0 & 19 & 0 \\ -1 + 19\sin 2t & 0 & 1 + 19\cos 2t \end{bmatrix},$$

$$\mathbf{f}(t) = \begin{bmatrix} e^t\left(-1 + 19\left(\cos 2t - \sin 2t\right)\right) \\ -18e^t \\ e^t\left(1 - 19\left(\cos 2t + \sin 2t\right)\right) \end{bmatrix}.$$

Here the right hand side is chosen so that $\mathbf{x}(t) = e^t\mathbf{e}$ satisfies the differential equation. The fundamental matrix displays the fast and slow solutions:

$$X(t,0) = \begin{bmatrix} e^{-18t}\cos t & 0 & e^{20t}\sin t \\ 0 & e^{19t} & 0 \\ -e^{-18t}\sin t & 0 & e^{20t}\cos t \end{bmatrix}.$$

Bob Mattheij's example

For boundary data with two terminal conditions and one initial condition :

$$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \mathbf{b} = \begin{bmatrix} e \\ e \\ 1 \end{bmatrix},$$

the trapezoidal rule discretization scheme gives the following results.

	$\Delta t = .1$			$\Delta t = .01$		
$\mathbf{x}(0)$	1.0000	.9999	.9999	1.0000	1.0000	1.0000
$\mathbf{x}(1)$	2.7183	2.7183	2.7183	2.7183	2.7183	2.7183

Table: Boundary point values - stable computation

These computations are apparently satisfactory.

Bob Mattheij's example

For two initial and one terminal condition:

$$B_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \mathbf{b} = \begin{bmatrix} 1 \\ e \\ 1 \end{bmatrix}.$$

The results are given in following Table.

	$\Delta t = .1$			$\Delta t = .01$		
$\mathbf{x}(0)$	1.0000	.9999	1.0000	1.0000	1.0000	1.0000
$\mathbf{x}(1)$	-7.9+11	2.7183	-4.7+11	2.03+2	2.7183	1.31+2

Table: Boundary point values - unstable computation

The effects of instability are seen clearly in the first and third solution components.

Bob Mattheij's example

The "optimal" boundary matrices corresponding to $h = .1$ are given in the Table. These confirm the importance of weighting the boundary data to reflect the stability requirements of a mix of fast and slow solutions. The solution does not differ from that obtained when the split into fast and slow was correctly anticipated.

B_1			B_2		
.99955	0.0000	.02126	-.01819	0.0000	-.01102
0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
.02126	0.0000	.00045	.85517	0.0000	.51791

Table: Optimal boundary matrices when $\Delta t = .1$

The problem setting

Let the ODE (model) have the general form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}\left(t, \mathbf{x}, \beta\right)$$

where $\beta \in R^p$ is a vector of parameters with "true" value β^* which is to be estimated from problem data:

$$\mathbf{y}_i = H\mathbf{x}\left(t_i, \beta\right), \quad i = 1, 2, \cdots, \quad n, \quad \gg m + p$$

where $H : R^m \to R^k$, $\mathbf{y} \in R^k$, $1 \leq k \leq m$, and $m + p$ is the number of degrees of freedom in the model. If $\mathbf{y}_i \sim N\left(H\mathbf{x}\left(t_i, \beta^*\right), V\right)$ are independent observations then the appropriate objective is

$$F\left(\beta\right) = \sum_{i=1}^{n} \left(\mathbf{y}_i - H\mathbf{x}\left(t_i, \beta\right)\right)^T V^{-1} \left(\mathbf{y}_i - H\mathbf{x}\left(t_i, \beta\right)\right).$$

The problem setting

Mesh selection for integrating the ODE system is conditioned by two important considerations:

- The asymptotic analysis of the effects of noisy data on the parameter estimates shows that this gets small no faster than $O\left(n^{-1/2}\right)$.
- It is not difficult to obtain ODE discretizations that give errors at most $O\left(n^{-2}\right)$.

The problem setting

Mesh selection for integrating the ODE system is conditioned by two important considerations:

- The asymptotic analysis of the effects of noisy data on the parameter estimates shows that this gets small no faster than $O\left(n^{-1/2}\right)$.
- It is not difficult to obtain ODE discretizations that give errors at most $O\left(n^{-2}\right)$.

This suggests:

- That the trapezoidal rule provides an adequate integration method.
- That it should be possible even to integrate the ODE on a mesh coarser than that provided by the observation points $\{t_i\}$ (here we wont!).

Method 1 - embedding

Here boundary conditions

$$B_1\mathbf{x}(0) + B_2\mathbf{x}(1) = \mathbf{b}$$

are adjoined to the ODE. The solutions $\mathbf{x}(t, \beta, \mathbf{b})$ of the resulting BVP provide comparison solutions for minimizing F. Gauss-Newton (scoring) provides an appropriate algorithm. In this formulation the estimation problem is unconstrained with variables $\beta,\ \mathbf{b}$.

It is necessary to choose $B_1,\ B_2$ to ensure stability in these integrations. Selection of appropriate conditions appears to require structural information about the ODE system. This is where the optimal boundary conditions enter.

Method 1 - embedding

Let $\nabla_{(\beta,b)}\mathbf{x} = \left[\frac{\partial \mathbf{x}}{\partial \boldsymbol{\beta}}, \frac{\partial \mathbf{x}}{\partial \mathbf{b}}\right]$, $\mathbf{r}_i = \mathbf{y}_i - H\mathbf{x}(t_i, \boldsymbol{\beta}, \mathbf{b})$ then the gradient of F is

$$\nabla_{(\beta,b)}F = -2\sum_{i=1}^{n} \mathbf{r}_i^T V^{-1} H \nabla_{(\beta,b)}\mathbf{x}_i.$$

The gradient terms wrt β are found by solving the BVP's

$$B_1 \frac{\partial \mathbf{x}}{\partial \beta}(0) + B_2 \frac{\partial \mathbf{x}}{\partial \beta}(1) = 0,$$

$$\frac{d}{dt}\frac{\partial \mathbf{x}}{\partial \beta} = \nabla_x \mathbf{f} \frac{\partial \mathbf{x}}{\partial \beta} + \nabla_\beta \mathbf{f},$$

Method 1 - embedding

Let $\nabla_{(\beta,b)}\mathbf{x} = \left[\frac{\partial \mathbf{x}}{\partial \boldsymbol{\beta}}, \frac{\partial \mathbf{x}}{\partial \mathbf{b}}\right]$, $\mathbf{r}_i = \mathbf{y}_i - H\mathbf{x}\left(t_i, \boldsymbol{\beta}, \mathbf{b}\right)$ then the gradient of F is

$$\nabla_{(\beta,b)}F = -2\sum_{i=1}^{n} \mathbf{r}_i^T V^{-1} H \nabla_{(\beta,b)}\mathbf{x}_i.$$

while the gradient terms wrt **b** satisfy the BVP's

$$B_1 \frac{\partial \mathbf{x}}{\partial \mathbf{b}}(0) + B_2 \frac{\partial \mathbf{x}}{\partial \mathbf{b}}(1) = I,$$
$$\frac{d}{dt}\frac{\partial \mathbf{x}}{\partial \mathbf{b}} = \nabla_x \mathbf{f} \frac{\partial \mathbf{x}}{\partial \mathbf{b}}.$$

Embedding: Again the Mattheij example

Consider the modification of the Mattheij problem with parameters $\beta_1^* = \gamma$, and $\beta_2^* = 2$ corresponding to the solution $\mathbf{x}(t, \beta^*) = e^t \mathbf{e}$:

$$M(t) = \begin{bmatrix} 1 - \beta_1 \cos \beta_2 t & 0 & 1 + \beta_1 \sin \beta_2 t \\ 0 & \beta_1 & 0 \\ -1 + \beta_1 \sin \beta_2 t & 0 & 1 + \beta_1 \cos \beta_2 t \end{bmatrix},$$

$$\mathbf{f}(t) = \begin{bmatrix} e^t \left(-1 + \gamma \left(\cos 2t - \sin 2t\right)\right) \\ -(\gamma - 1)e^t \\ e^t \left(1 - \gamma \left(\cos 2t + \sin 2t\right)\right) \end{bmatrix}.$$

In the numerical experiments optimal boundary conditions are set at the first iteration. The aim is to recover estimates of $\beta^*, \mathbf{b}^*$ from simulated data $e^{t_i} H\mathbf{e} + \varepsilon_i$, $\varepsilon_i \sim N(0, .01I)$ using Gauss-Newton, stopping when $\nabla F \mathbf{h} < 10^{-8}$.

Embedding: Again the Mattheij example

$$H = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$H = \begin{bmatrix} .5 & 0 & .5 \\ 0 & 1 & 0 \end{bmatrix}$$

$H = [1/3 \;\; 1/3 \;\; 1/3]$
$n = 51,\ \gamma = 10,\ \sigma = .1$
14 iterations
$n = 51,\ \gamma = 20,\ \sigma = .1$
11 iterations
$n = 251,\ \gamma = 10,\ \sigma = .1$
9 iterations
$n = 251,\ \gamma = 20,\ \sigma = .1$
8 iterations

$H = \begin{bmatrix} .5 & 0 & .5 \\ 0 & 1 & 0 \end{bmatrix}$
$n = 51,\ \gamma = 10,\ \sigma = .1$
5 iterations
$n = 51,\ \gamma = 20,\ \sigma = .1$
9 iterations
$n = 251,\ \gamma = 10,\ \sigma = .1$
4 iterations
$n = 251,\ \gamma = 20,\ \sigma = .1$
5 iterations

Here $\left\| \begin{bmatrix} B_1 & B_2 \end{bmatrix}_1 \begin{bmatrix} B_1 & B_2 \end{bmatrix}_k^T - I \right\|_F < 10^{-3}$, $k > 1$.

The simultaneous method

This formulates a constrained estimation problem:

$$\min_{\mathbf{x},\boldsymbol{\beta}} F(\mathbf{x},\beta)\,;\;\; \mathbf{c}_i(\mathbf{x},\beta) = 0,\; i = 1, 2, \cdots, n-1,$$

where

$$\mathbf{c}_i(\mathbf{x},\beta) = \mathbf{x}_{i+1} - \mathbf{x}_i - \frac{\Delta t}{2}\left[\mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}, \beta) + \mathbf{f}(t_i, \mathbf{x}_i, \beta)\right].$$

This has the advantage - which could translate into faster execution speeds - that repeated solution of BVP's is not required with the solution being part of the problem variables (contrast embedding). Does it have stability advantages?

The simultaneous method

This formulates a constrained estimation problem:

$$\min_{\mathbf{x},\boldsymbol{\beta}} F(\mathbf{x},\boldsymbol{\beta})\,;\ \mathbf{c}_i(\mathbf{x},\boldsymbol{\beta}) = 0,\ i = 1, 2, \cdots, n-1,$$

where

$$\mathbf{c}_i(\mathbf{x},\boldsymbol{\beta}) = \mathbf{x}_{i+1} - \mathbf{x}_i - \frac{\Delta t}{2}\left[\mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}, \boldsymbol{\beta}) + \mathbf{f}(t_i, \mathbf{x}_i, \boldsymbol{\beta})\right].$$

Possible disadvantages are the potentially very large constraint set - at least in theory - as $n \to \infty$, and the more complex algorithmic questions associated with the constrained problem.

The basic NLBVP algorithm

In outline, a modified Newton algorithm would go something like this:

- Provide an initial guess at the solution.
- In estimation by embedding need to estimate (and check) appropriate BC's using linearised equations.
- Solve the linearized problem for the Newton correction $\mathbf{h}$.
- Compute an improved solution estimate by line-searching in the direction determined by the estimated correction.
- Update the solution estimate.
- Repeat iterative step if convergence test not satisfied.

The line search

The new feature is the line-search. This needs an objective function Φ to reduce in order to gauge improvement. Possibilities include:

- Let $\mathbf{r}$ be the composite vector whose components are the ODE residuals at the mesh points. Then $\Phi(\lambda) = \|\mathbf{r}(\mathbf{x} + \lambda \mathbf{h})\|^2$.

The line search

The new feature is the line-search. This needs an objective function Φ to reduce in order to gauge improvement. Possibilities include:

- Let $\mathbf{r}$ be the composite vector whose components are the ODE residuals at the mesh points. Then $\Phi(\lambda) = \|\mathbf{r}(\mathbf{x} + \lambda\mathbf{h})\|^2$.
- Let the current step of the Newton iteration be written $J(\mathbf{x})\mathbf{h} = -\mathbf{r}$, and set

$$J(\mathbf{x})\widetilde{\mathbf{h}}(\lambda) = -\mathbf{r}(\mathbf{x} + \lambda\mathbf{h}).$$

  In this case $\Phi(\lambda) = \|\widetilde{\mathbf{h}}(\lambda)\|^2$.

It seems agreed that the "affine invariant" second case should be superior to the first in general. However, convergence can be proved in the first case but not the second.

An example: rotating discs flow

The governing ODE's for the similarity solutions to the flow between two infinite rotating discs are:

$$\frac{dx_1}{dt} = -2x_2,$$
$$\frac{dx_2}{dt} = x_3,$$
$$\frac{dx_3}{dt} = x_1x_3 + x_2^2 - x_4^2 + x_6,$$
$$\frac{dx_4}{dt} = x_5,$$
$$\frac{dx_5}{dt} = 2x_2x_4 + x_1x_5,$$
$$\frac{dx_6}{dt} = 0.$$

+ boundary conditions:

$$x_1(0) = 0,$$
$$x_2(0) = 0,$$
$$x_4(0) = 1,$$
$$x_1(b) = 0,$$
$$x_2(b) = 0,$$
$$x_4(b) = s.$$

An example: rotating discs flow

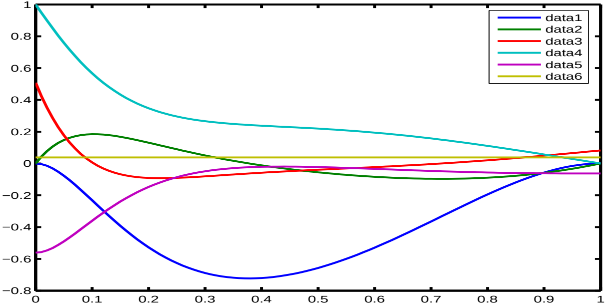
The next slide gives results of numerical computations. The case reported corresponds to $s = 0.0, \ b = 9$. Starting values are $\mathbf{x}_i = 0, \ i = 1, 2, \cdots, n$. Other settings are $n = 101$, iteration tolerance $1.e - 10$, and Armijo parameter $\theta = .25$. The iteration tolerance is applied to the objective function which is defined as $\sqrt{\Delta t \Phi(\mathbf{x})}$ where $\Phi(\mathbf{x})$ is the affine covariant objective in the first case, and the sum of squares of residuals in the second. In general, difficulty increases with increasing separation b and decreasing rotation speed ratio s, but this is by no means the full story.

An example: rotating discs flow

it	affine cov			ss	
	λ	objs	objt	λ	obj
s=0.0					
0					3.999-01
1	.25	1.7769 00	1.4134 00	1	1.2075-01
2	1	2.2370 00	1.5573 00	1	1.8768-02
3	.25	1.5529 00	5.6232-01	1	1.0154-03
4	.25	1.5421 00	1.0481 00	1	1.5290-04
5	1	5.3675-01	7.1059-02	1	1.6576-07
6	1	6.3186-02	3.5046-03	1	3.3204-12
7	1	3.4912-03	4.4195-06		
8	1	4.4287-06	1.6675-11		

Table: Rotating disc flow: numerical results for $b = 9$

An example: rotating discs flow



The Mattheij example yet again

go MatEx3 The size of the errors for the two meshes suggests something more than instability is involved. For the DE

$$\frac{dx}{dt} = \lambda x$$

the trapezoidal rule gives

$$\left(1 - \frac{\lambda \Delta t}{2}\right) x_{i+1} = \left(1 + \frac{\lambda \Delta t}{2}\right) x_i .$$

Stiff stability for $\lambda \leq 0$ follows immediately. For $\lambda > 0$ it seems a different story - the amplification factor passes through $+\infty$ to oscillate in sign, eventually tending to -1 as $\lambda \to \infty$. Sometimes called "super stability". go MatEx2