12.3.1 Implementation of Implicit Methods for Linear Systems

MO2.7 MO2.8

Discussions

All posts sorted by recent activity

Let's next look at the implementation of an implicit method for an IVP for which $\underline{f}(\underline{u}, t)$ involves a linear system as described in Section 10.1.2. Recall that for such an IVP that $\underline{f}(\underline{u}, t)$ has the following form,

$$\underline{f}(\underline{u}, t) = A\underline{u} + \underline{b}(t) \tag{12.32}$$

where A is an $M \times M$ matrix, M is the number of states, and $\underline{b}$ is a vector of known functions of time (but do not depend on $\underline{u}$). Let's now compare the Forward Euler and Backward Euler implementations for this $\underline{f}(\underline{u}, t)$. For Forward Euler, an iteration is,

$$\underline{v}^{n+1} = \underline{v}^n + \Delta t \left[A\underline{v}^n + \underline{b}(t^n)\right] \tag{12.33}$$

The computational cost of a Forward Euler iteration will be dominated by the matrix-vector multiplication $A\underline{v}^n$ (unless $\underline{b}(t)$ is an extremely complex function to evaluate). The asymptotic computational complexity of a matrix-vector multiply is $2M^2$ (try deriving this yourself by determining the total number of mathematical operations, in particular multiplications and additions, required to multiply a matrix and a vector).

For Backward Euler we have the following iteration,

$$\underline{v}^{n+1} = \underline{v}^n + \Delta t \left[A\underline{v}^{n+1} + \underline{b}(t^{n+1})\right] \tag{12.34}$$

Now, re-arranging this iteration produces the following linear system of equations to be solved for $\underline{v}^{n+1}$,

$$(I - \Delta t A)\,\underline{v}^{n+1} = \underline{v}^n + \Delta t \underline{b}(t^{n+1}) \tag{12.35}$$

where I is an $M \times M$ identity matrix. On the left-hand side of this equation, we have a matrix ($I - \Delta t A$) multiplying the vector $\underline{v}^{n+1}$. On the right-hand side, we have a known vector, $\underline{v}^n + \Delta t \underline{b}(t^{n+1})$. Thus, a Backward Euler iteration requires the solution of a linear $M \times M$ system of equations. If we use Gaussian elimination (see Section 10.4) to solve this system of equations, the asymptotic computational complexity is $\frac{2}{3}M^3$

© All Rights Reserved

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap
Cookie Policy
Your Privacy Choices

< Previous Next >

Connect

Blog
Contact Us
Help Center
Security
Media Kit

GET IT ON Google play
Download on the App Store

© 2023 edX LLC. All rights reserved.
深圳市恒宇博科技有限公司 粤ICP备17044299号-2