Systems of ODEs (and higher-order ODEs)

Sunday, October 25, 2020 2:19 PM

Section 5.9 in Burden and Faires

Previously, defined IVP: $y' = f(t,y)$, $a \le t \le b$, $y(a) = y_0$

and we derived Euler's Method w_i approximates $y(t_i)$

$$w_0 = y_0$$
$$w_{i+1} = w_i + h\, f(t_i, w_i)$$

Now extend to coupled systems of ODEs

Change notation (to be consistent w/ book) from $y(t)$ to $u(t)$,

and in particular $\vec{u}(t)$ to denote that it's a vector

An m^{th} order system of 1^{st}-order IVP is

$$\left.\begin{aligned} \frac{du_1}{dt} &= f_1(t, u_1, \ldots, u_m) \\ \frac{du_2}{dt} &= f_2(t, u_1, \ldots, u_m) \\ &\vdots \\ \frac{du_m}{dt} &= f_m(t, u_1, \ldots, u_m) \end{aligned}\right\} \text{ or just } \vec{u}' = \vec{f}(t, \vec{u})$$

$$\vec{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix}, \quad \vec{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

Most theory and algorithms extend straight-forwardly

Theory for systems

Before, we wanted $f(t,u)$ to be Lipschitz in u

Now, define $\vec{f}(t, \vec{u})$ to be L-Lipschitz in $\vec{u}$ on a region $D = \{(t, \vec{u}) : a \le t \le b,\ \vec{u} \in \mathbb{R}^m\}$ to mean $\exists\, L < \infty$ s.t.

$$(\forall\, (t, \vec{u}), (t, \vec{v}) \in D), \quad |\vec{f}(t, \vec{u}) - \vec{f}(t, \vec{v})| \le L \cdot \|\vec{u} - \vec{v}\|,$$

where $\|\vec{z}\|_1 := \sum_{i=1}^{m} |z_i|$ $\quad \ell_1$ norm

Supplementary

(FACT: $\|\vec{z}\|_2 := \sqrt{\sum_{i=1}^{m} |z_i|^2}$ $\quad \ell_2$ norm

$\|\vec{z}\|_p := \left(\sum_{i=1}^{m} |z_i|^p\right)^{1/p}$ $\quad \ell_p$ norm $1 \le p \le \infty$

$$\|\vec{z}\|_\infty := \max_{i=1,\dots,m} |z_i| \qquad \ell_\infty \text{ norm} \quad)$$

For Lipschitz, you can use a different norm than ℓ_1 and theory still works, though the value of L needs to be adjusted

Theorem 5.17 Existence and uniqueness for systems

Consider the system of ODEs / IVP, and assume each $f_i(t,\vec{u})$ is continuous and $\vec{f}$ is Lipschitz with respect to $\vec{u}$, uniformly in t, on D. Then the system IVP has a unique solution.

Special Case: Linear ODEs

Supplementary (not in book)

A linear 1st order ODE is $y' = \overbrace{a(t)}^{\text{coefficient}} \cdot y + b(t)$

We can usually solve this in closed form as we can reduce it to an integration problem. General strategy:

① Solve $y_{\text{homogeneous}}$, $y_h' = a(t)\cdot y_h + \underline{\underline{0}}$

② Find $y_{\text{particular}}$ via variation of parameters

Let's simplify: constant coefficient, homogeneous

$y' = a\cdot y$ then $y(t) = c\cdot e^{at}$ ... and if $y(0)=y_0$, $y(t)=y_0 e^{at}$

For systems, equiv. linear, 1st order system w/ const. coeff., homogeneous

is $\vec{u}(t)' = A\cdot\vec{u}$, A is a matrix $(m\times m)$

We cannot solve for each component separately unless A is a diagonal matrix.

One trick: if A is diagonalizable $A = V\cdot\Lambda V^{-1}$

eigenvectors $V = [\vec{v}_1, \dots, \vec{v}_m]$

eigenvalues $\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}$

do a change-of-variables

$\vec{w}(t) = V^{-1}\cdot\vec{u}(t)$ or $\vec{u}(t) = V\,\vec{w}(t)$

so ODE is now $\vec{u}\,' = (\overbrace{V\Lambda V^{-1}}^{A})\vec{u}$, with $\underbrace{V^{-1}\vec{u}}_{w}$

V^{-1}

$\vec{w}\,' = \Lambda\,\vec{w}$

diagonal!

Solving a DIAGONAL system of equations is easy
... since it's uncoupled.

$$w_i(t) = c_i \cdot e^{\lambda_i t}$$

Undoing the change-of-variables and adding in $\vec{u}(0) = \vec{u}_0$ as the initial condition, we find

$$\vec{u}(t) = \underbrace{V \cdot \begin{bmatrix} e^{\lambda_1 t} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_m t} \end{bmatrix} V^{-1}}_{\text{this is called the matrix exponential of } A:} \cdot \vec{u}_0 = e^{At} \cdot \vec{u}_0$$

Def matrix exponential of A:

(1) if A is diagonalizable, $A = V \Lambda V^{-1}$ is eigenvalue decomp.,

$$e^A = V \cdot \begin{bmatrix} e^{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_m} \end{bmatrix} \cdot V^{-1}$$

(2) if A isn't diagonalizable, e^A still exists
(define via Taylor Series $e^A = I + A + \frac{1}{2} A^2 + \frac{1}{3!} A^3 + \dots$
or via Jordan form)

Facts $e^{At} = V \cdot \begin{bmatrix} e^{\lambda_1 t} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_m t} \end{bmatrix} V^{-1}$

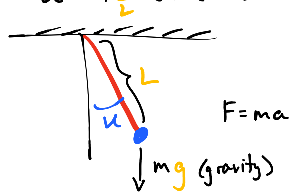
Computation in Matlab, expm (exp(A) does element-wise exponential)
in Python, scipy.linalg.expm

... but expensive computationally, not often used to solve ODEs

(*) Special Case: HIGHER-ORDER ODEs

Ex: Motion of a pendulum (ultra-classic)

$$u'' + \frac{g}{L} \sin(u) = 0$$



→ typical physics/intro ODE course
we assume small displacements,
$\sin(u) \approx u$

$$u'' + g/L \cdot u = 0$$

Solutions are $u(t) = a_1 \cdot \sin(\omega_0 t) + a_2 \cos(\omega_0 t)$
$\omega_0 = \sqrt{g/L}$

w/o the small angle approximation, how to solve numerically?

TRICK Introduce $v(t)$, "velocity", $v = u'$

So... $u'' + g/L \sin(u) = 0 \implies$ $u' = v$, $v' = -g/L \cdot \sin(u)$

or...
$$u_1' = u_2$$
$$u_2' = -g/L \sin(u_1)$$

where $u_1 =$ "u" from before, $u_2 =$ "v" from before

$$\vec{u}' = \vec{f}(t, \vec{u}), \quad \vec{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad f_1(t, \vec{u}) = u_2, \quad f_2(t, \vec{u}) = -g/L \sin(u_1)$$

⊛ This trick works for any higher-order ODE

i.e., to convert $\underbrace{u'''(t)}_{u_3'} + 3t^2 \underbrace{u''(t)}_{u_3} + 5(\underbrace{u'(t)}_{u_2})^2 + e^t \underbrace{u(t)}_{u_1} = \sin(t)$

into a single 1st order ODE (with 3 equations),

$$u_1' = u_2$$
$$u_2' = u_3$$
$$u_3' = -3t^2 u_3 - 5u_2^2 - e^t u_1 + \sin(t)$$

Numerical Methods for Systems

Euler:
$$w_0 = y_0$$
$$w_{i+1} = w_i + h f(t_i, w_i)$$

1D IVP $y(a) = y_0$, $y' = f(t, y)$

w_i approximates $y(t_i)$

Now for a system
$$\vec{w}_0 = \vec{u}_0$$
$$\vec{w}_{i+1} = \vec{w}_i + h \vec{f}(t_i, \vec{w}_i)$$

m-dimensions IVP $\vec{u}(0) = \vec{u}_0$, $\vec{u}' = \vec{f}(t, \vec{u})$

$\vec{w}_i$ approximates $\vec{u}(t_i)$

... so not really any different!

The other fancier methods we'll talk about also extend easily, so usually we'll just consider the scaler case without loss of generality.

Conceptually, numerically solving a system of m ODEs is just as simple as a single ODE. *

This in contrast to finding analytic solutions which is usually harder with systems

* However, there are practical issues, as we'll see when we talk about stiff ODES, where one variable has a fast timescale and another has a slow timescale

  (Ex. climate simulations: fast timescale for weather, but need to run for a long time)