Métodos implícitos. Aplicación a problemas con características especiales: Parte II

Computación en ciencia e ingeniería: simulación numérica
MÁSTER UNIVERSITARIO EN INGENIERÍA COMPUTACIONAL Y SISTEMAS INTELIGENTES,
Euskal Herriko Unibertsitatea / Universidad del Pais Vasco (UPV/EHU)

Los métodos explícitos como RK4 no son muy apropiados si el sistema de EDO se puede reescribir de la forma

$$\frac{d}{dt}u = J(t) \cdot u + g(t, u), \quad u \in \mathbb{R}^d,$$

donde $J(t)$ es, para cada $t \in \mathbb{R}$, $u \in \mathbb{R}^d$, una matriz de dimensión $d \times d$ con algunas componentes muy grandes con respecto a los valores que toma $g(t, u)$. (En dicho caso, se dice que el sistema de EDOs es "stiff").

Los métodos explícitos como RK4 no son muy apropiados si el sistema de EDO se puede reescribir de la forma

$$\frac{d}{dt}u = J(t) \cdot u + g(t,u), \quad u \in \mathbb{R}^d,$$

donde $J(t)$ es, para cada $t \in \mathbb{R}$, $u \in \mathbb{R}^d$, una matriz de dimensión $d \times d$ con algunas componentes muy grandes con respecto a los valores que toma $g(t,u)$. (En dicho caso, se dice que el sistema de EDOs es "stiff").

Métodos de Runge-Kutta implícitos apropiados para este tipo de problemas:

- Métodos del trapecio y del punto medio (ambos de orden 2),
- Métodos de RK implícitos de Gauss de orden 4, 6, 8, etc,
- Métodos de RK implícitos de Radau de orden 3, 5, 7, etc.

Recordemos la implementación del método del trapecio

$$u_{j+1} = u_j + \frac{h}{2}\, f(t_j, u_j) + f(t_{j+1}, u_{j+1}))$$

Recordemos la implementación del método del trapecio

$$u_{j+1} = u_j + \frac{h}{2} f(t_j, u_j) + f(t_{j+1}, u_{j+1}))$$

Iteración del punto fijo

- $v_j = u_j + \frac{h}{2} (f(t_j, u_j),$
- $u_{j+1}^{[0]} = u_j,$
- Calcular para $k = 0, 1, 2, \ldots$

$$u_{j+1}^{[k+1]} = v_j + \frac{h}{2} f(t_{j+1}, u_{j+1}^{[k]}),$$

- Si $||u_{j+1}^{[k+1]} - u_{j+1}^{[k]}|| \leq \mathrm{itol}$, entonces $u_{j+1} := u_{j+1}^{[k+1]}$.

Estamos considerando sistema de EDOs es de la forma

$$\frac{d}{dt}u = J(t) \cdot u + g(t, u), \quad u \in \mathbb{R}^d,$$

donde $J(t)$ es, para cada $t \in \mathbb{R}$, $u \in \mathbb{R}^d$, una matriz de dimensión $d \times d$ con algunas componentes muy grandes con respecto a los valores que toma $g(t, u)$.

Estamos considerando sistema de EDOs es de la forma

$$\frac{d}{dt}u = J(t) \cdot u + g(t, u), \quad u \in \mathbb{R}^d,$$

donde $J(t)$ es, para cada $t \in \mathbb{R}$, $u \in \mathbb{R}^d$, una matriz de dimensión $d \times d$ con algunas componentes muy grandes con respecto a los valores que toma $g(t, u)$.

De forma más general, vamos a suponer que tenemos un sistema de EDOs de la forma

$$\frac{d}{dt}u = f(t, u), \quad u \in \mathbb{R}^d,$$

y que disponemos de una matriz $J(t, u)$ de dimensión $d \times d$ (para cada $t \in \mathbb{R}$ y cada $u \in \mathbb{R}^d$) tal que, para cada $\delta \in \mathbb{R}^d$ con componentes suficientemente pequeñas,

$$f(t, u + \delta) - f(t, u) \approx J(t, u) \cdot \delta$$

La iteración del punto fijo no es apropiada si algunas componentes de $J(t, u)$ son muy grandes (en ese caso requiere que la longitud de paso sea muy pequeña).

Estamos considerando sistema de EDOs es de la forma

$$\frac{d}{dt}u = J(t) \cdot u + g(t, u), \quad u \in \mathbb{R}^d,$$

donde $J(t)$ es, para cada $t \in \mathbb{R}$, $u \in \mathbb{R}^d$, una matriz de dimensión $d \times d$ con algunas componentes muy grandes con respecto a los valores que toma $g(t, u)$.

De forma más general, vamos a suponer que tenemos un sistema de EDOs de la forma

$$\frac{d}{dt}u = f(t, u), \quad u \in \mathbb{R}^d,$$

y que disponemos de una matriz $J(t, u)$ de dimensión $d \times d$ (para cada $t \in \mathbb{R}$ y cada $u \in \mathbb{R}^d$) tal que, para cada $\delta \in \mathbb{R}^d$ con componentes suficientemente pequeñas,

$$f(t, u + \delta) - f(t, u) \approx J(t, u) \cdot \delta$$

La iteración del punto fijo no es apropiada si algunas componentes de $J(t, u)$ son muy grandes (en ese caso requiere que la longitud de paso sea muy pequeña). Para este tipo de sistemas ("stiff"), es mucho más apropiada la iteración de Newton simplificada.

Iteración de Newton simplificada

- $v_j = u_j + \dfrac{h}{2} f(t_j, u_j)$,
- $u_{j+1}^{[0]} = u_j$,
- $A_j = I - \dfrac{h}{2} J(t_j, u_j)$ (donde I es la matriz identidad $d \times d$),
- Calcular para $k = 0, 1, 2, \ldots$,

$$u_{j+1}^{[k+1]} = u_{j+1}^{[k]} + A_j^{-1} \left(v_j + \frac{h}{2} f(t_{j+1}, u_{j+1}^{[k]}) - u_{j+1}^{[k]} \right),$$

- $u_{j+1} := u_{j+1}^{[k+1]}$ si $||u_{j+1}^{[k+1]} - u_j^{[k]}|| \leq \mathrm{itol}.$

Iteración de Newton simplificada

- $v_j = u_j + \frac{h}{2} f(t_j, u_j)$,
- $u_{j+1}^{[0]} = u_j$,
- $A_j = I - \frac{h}{2} J(t_j, u_j)$ (donde I es la matriz identidad $d \times d$),
- Calcular para $k = 0, 1, 2, \ldots$,

$$u_{j+1}^{[k+1]} = u_{j+1}^{[k]} + A_j^{-1} \left(v_j + \frac{h}{2} f(t_{j+1}, u_{j+1}^{[k]}) - u_{j+1}^{[k]} \right),$$

- $u_{j+1} := u_{j+1}^{[k+1]}$ si $||u_{j+1}^{[k+1]} - u_j^{[k]}|| \leq \mathrm{itol}.$

La iteración de Newton simplificada para otros métodos (el del punto medio, los métodos de RK implícitos de Gauss y de Radau, etc) es similar en esencia a éste.

El método de RK implícito de Radau está implementado (con la iteración de Newton simplificada)

- En la librería deSolve de R, en la función 'ode', con la opción method="radau".
- En el paquete DifferentialEquations.jl (se requiere además ODEInterfaceDiffEq.jl),en la función 'solve' con el método radau5().
- En python, en la librería scipy.integrate, en la función 'scipy.integrate.solve_ivp' con la opción method=radau".