

Beépített köz.diff. megoldók

A Matlab programrendszerben léteznek beépített programok, amelyek alkalmasak akár nagy pontossággal és hatékonyan megoldani a kezdetiérték-feladatokat. Ilyen például a gyakran alkalmazott `ode45` rutin, ami egy ún. beágyazott Dormand–Prince-módszer. Ez egy egylépéses, váltakozó lépésközü módszer. A módszer lényege, hogy kiszámol egy negyed- és egy ötödrendű Runge–Kutta-módszert, és úgy választ lépésközt, hogy a hiba a negyedrendű módszer hibája legyen.

Nézzük meg, hogyan tudjuk használni az `ode45` rutint. Szeretnénk a `diffegy.m` fájlban megadott differenciálegyenletet megoldani a $[0, 1]$ intervallumon, az $y_0 = 1$ kezdőértékből indulva. Lépünk be a `diffegy.m` fájlt tartalmazó könyvtárba, és a parancsablakba gépeljük be:

```
»ode45(@diffegy,[0 1],1)
```

Erre a Matlab megoldja a feladatot, és kirajzolja a numerikus megoldást.

Természetesen szükségünk lehet a megoldás számszerű értékeire is. Ha az `y` jelölést szeretnénk használni a megoldás számszerű értékeire, és a `t` jelölést azon időpontok vektorára, amelyekben a Matlab előállítja a megoldást, akkor ezt írjuk:

```
»[tki,yki]=ode45(@diffegy,[0 1],1)
```

Erre megkapjuk a `t` és `y` vektort, ábra viszont nem készül, azt magunk rajzoltathatjuk ki a

```
»plot(t,y)
```

paranccsal. A két kimenő vektor hosszúságát az szabja meg, hogy a Matlabnak milyen lépésközzel kell dolgoznia ahhoz, hogy az alapértelmezésben szereplő előírt pontosságot elérje. Ennek átdefiniálásáról hamarosan szó lesz.

Ha megadott lépésközönként van szükségünk a numerikus megoldás értékeire, akkor definiáljuk ezen időpontok vektorát, pl.

```
»tspan =0:0.2:1
```

módon (növekvő vagy csökkenő sorrendben). Ezután gépeljük be a következő parancsot:

```
» [t,y]=ode45(@diffegy,tspan,1)
```

Az eredmény:

```
t =  
    0  
    0.2000  
    0.4000  
    0.6000  
    0.8000  
    1.0000
```

```
y =  
    1.0000  
    1.0187  
    1.0703  
    1.1488  
    1.2493  
    1.3679
```

A Matlab alapértelmezésben $1e-3$ -as relatív hibával és $1e-6$ -os abszolút hibával dolgozik. Ha ezeket szeretnénk átállítani, pl. mindkettőt két nagyságrenddel csökkenteni, akkor gépeljük be:

```
»opts = odeset('reltol',1.e-5, 'abstol',1.e-8)
```

Ezután a következőképpen hívjuk meg az `ode45` függvényt:

```
» [t,y]=ode45(@diffegy,[0,1],1,opts)
```

Vannak a Matlabnak egyéb beépített módszerei is. Egy másik, ugyancsak gyakran használt beépített módszer, az `ode23` ugyancsak beágyazott Runge–Kutta típusú módszer, amelyet Bogacki–Shampine-módszernek is nevezünk. Ez egy explicit (2; 3)-típusú Runge–Kutta-módszer. Igazából akkor hatékony, amikor olcsón szeretnénk kevésbé pontos megoldást kapni. (Csak kis merevségű feladatokra alkalmazható, és az `ode45` módszerhez hasonlóan nem merev feladatok numerikus megoldására alkalmazzuk.)

Többlépéses módszerek algoritmusai is megtalálhatók a Matlabban. Ilyen az `ode113`,

amelynek a rendje 1-től 13-ig változhat, és az Adams–Bashforth–Moulton-módszeren alap-
szik. Összehasonlítva az `ode45` módszerrel, kevésbé pontos, de kisebb számításigényű és
különösen előnyös, amikor az f függvény kiértékelése költséges.

Befejezésül megjegyezzük, hogy merev feladatok megoldására is vannak rutinok a
Matlabban. Ilyenek a `ode15s`, ill. az `ode23s`. A `ode123t` és az `ode123tb` rutinok az
enyhén merev rendszerek megoldására javasolhatók. Ezek a módszerek alacsony pon-
tosságúak. Az `ode15s` módszer a backward differentiation formulákon (BDF) (amelyet
Gear-módszernek is szokásos nevezni) alapul, és változó lépéshosszú módszer. Különösen
akkor javasolt, amikor az `ode45` módszer nagyon lassú vagy egyáltalán nem működik. Az
`ode23s` egy másodrendű, módosított, egylépéses Rosenbrock-módszer. Mivel egylépéses,
ezért általában hatékonyabb, mint az `ode15s` módszer.

Összefoglalva a beépített megoldók legfontosabb jellemzői:

`ode45`:

Nem merev feladatokra

Közepes pontosságú

Általában ezt használjuk, ezt érdemes először kipróbálni.

`ode23`:

Nem merev feladatokra

Kis pontosságú

Nagy előírt hibahatár és mérsékelten merev rendszerek esetén használjuk

`ode113`:

Nem merev feladatokra

Kis pontosságtól nagy pontosságig.

Nagy előírt pontosság ill. nagy számításigényű függvények esetén javasolt.

`ode15s`:

Merev feladatokra

Kis és közepes pontosság

Ha az `ode45` lassú, ill. tömegmátrix van, akkor érdemes ezzel próbálkozni.

`ode23s`:

Merev feladatokra

Kis pontosságú

Nagy előírt hibahatár esetén merev feladatra érdemes használni, ill. ha konstans tömegmátrix van.

ode23t:

Mérsékelten merev feladatokra

Kis pontosságú

Mérsékelten merev feladatokra alkalmazzuk, ahol numerikus csillapítás nélküli megoldásra van szükség.

ode23tb:

Merev feladatokra

Kis pontosságú

Nagy előírt hibahatár esetén merev feladatra alkalmazzuk, ill. ha tömegmátrix van.