

2.8 Някои приложения на числените методи за решаване на ОДУ

Ще разгледаме няколко примера, на базата на които ще илюстрираме приложението на разглежданите методи на практика.

2.8.1 Анимация на топче, закачено на пружина

Ще моделираме поведението на топче, закачено на еластична пружина. На топчето му действат 2 сили – силата на тежестта $F_m = mg$ и еластичната реакция на пружината F_e . Последната може да се опише чрез закона на Хук $F_e = -kx$, където k е коефициентът на еластичност, а x е отместването на топчето. От втория закон на Нютон е известно, че:

$$\begin{aligned}ma &= F_m + F_e, \\mx'' &= mg - kx, \\x'' &= g - \frac{k}{m}x.\end{aligned}$$

Така получихме следното диференциално уравнение от втори ред:

$$x'' = g - \frac{k}{m}x.$$

За да анимираме поведението топчето в даден времеви интервал (например $0 < t \leq 10$), са ни необходими 2 начални условия. Нека топчето в момент 0 се намира в позиция 0 и началната му скорост е 0. Така получаваме задачата:

$$\begin{aligned}x'' &= g - \frac{k}{m}x, \quad 0 < t \leq 10, \\x(0) &= 0, \\x'(0) &= 0.\end{aligned}$$

Ще сведем задачата от втори ред към задача на Коши за системата ОДУ от първи ред. За целта ще положим $x' = v$, което води до следната еквивалентна задача:

$$\begin{aligned}x' &= v, \quad 0 < t \leq 10, \\v' &= g - \frac{k}{m}x, \quad 0 < t \leq 10, \\x(0) &= 0, \\v(0) &= 0.\end{aligned}\tag{2.1}$$

Ако означим $\mathbf{u} := (x, v)$, то тогава задачата може да се запише и като

$$\begin{aligned}\frac{d\mathbf{u}}{dt} &= \begin{bmatrix} v \\ g - \frac{k}{m}x \end{bmatrix}, \quad 0 < t \leq 10, \\ \mathbf{u}(0) &= 0.\end{aligned}$$

Примерна имплементация на решението е представена по-долу:

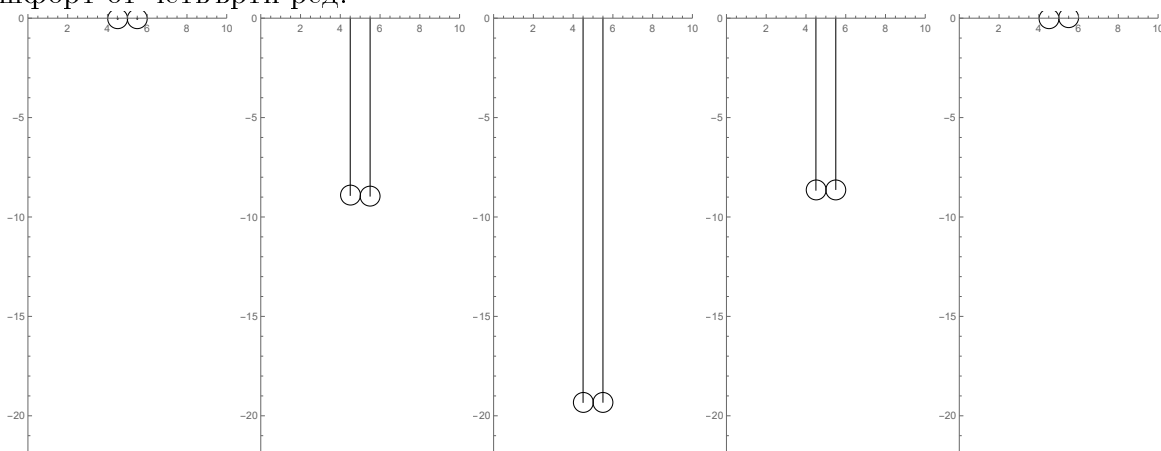
```

In[67]:= Clear[x, t];
exact[t_] = x[t] /. DSolve[{x'[t] == 9.8 - x[t], x[0] == 0, x'[0] == 0}, x[t], t][[1]];
h = 0.01;
t = Table[i*h, {i, 0, 10/h}];
f[x_, p_] := {p[[2]], 9.8 - 1*p[[1]]}
apprPos = AB4[f, h, 0, 10, {0, 0}][[All, 2, 1]];

ballSpring[pos_, x_] := {Circle[{x, -pos}, 0.5], Line[{{x, -pos}, {x, 0}}]}
Animate[Graphics[{ballSpring[exact[t[[i]]], 4.5],
                  ballSpring[apprPos[[i]], 5.5]},
        PlotRange -> {{0, 10}, {-22, 0}}, Axes -> True],
        {i, 1, Length[t], 1}]

```

Резултати от експериментите в 5 различни момента от време са показани по-долу. Лявото топче е анимирано, използвайки точното решение на (2.1), а дясното – използвайки приближеното решение, намерено по метода на Адамс-Башфорт от четвърти ред.



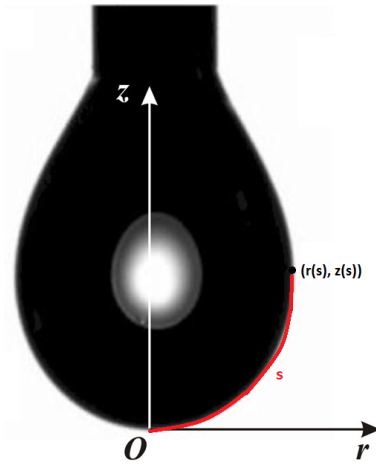
2.8.2 Анализ на формата на осево-симетрична капка.

Методът на Адамс–Башфорт датира от края на 19-ти век, като първо той е бил приложен за определяне профила на капка от даден флуид. Ето защо ние ще го илюстрираме именно върху тази задача.

Разглеждаме системата

$$\begin{aligned}
 \frac{dr}{ds} &= \cos \varphi, \\
 \frac{dz}{ds} &= \sin \varphi, \\
 \frac{d\varphi}{ds} &= 2b + cz - \frac{\sin \varphi}{r},
 \end{aligned}$$

която описва профила на капка, висяща от даден капиляр, под действието на гравитацията. Профилът се описва като параметрична крива $l = (r(s), z(s))$, която е параметризирана по дължината на дъгата, s , измерена от върха на капката където φ е ъгълът, който допирателната в съответната точка сключва с абсцисната ос. Параметърът b описва кривината във върха на капката, а c е параметър, който характеризира веществото.



Началните условия са $r(0) = z(0) = \varphi(0) = 0$. Да отбележим, че системата има особеност при $s = 0$, тъй като дясната страна на последното уравнение не е дефинирана. Може да се покаже обаче, че дясната граница съществува и

$$\left. \frac{d\varphi}{ds} \right|_{0+} = b.$$

Тогава е естествено да приемем последното като дефиниция на дясната страна за $s = 0$.

За да можем да използваме вече написаната в Mathematica функция *AB4*, първо трябва да запишем системата като едно векторно уравнение. Полагайки $\mathbf{u} = (r, z, \varphi)^T$ и $\mathbf{f}(\mathbf{u}) = (f_1, f_2, f_3)^T$, където

$$\begin{aligned} f_1(\mathbf{u}) &= \cos \varphi, \\ f_2(\mathbf{u}) &= \sin \varphi, \\ f_3(\mathbf{u}) &= \begin{cases} 2b + cz - \frac{\sin \varphi}{r}, & s > 0 \\ b, & s = 0 \end{cases}. \end{aligned}$$

получаваме системата

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{f}(\mathbf{u}), \quad s > 0, \\ \mathbf{u}(0) &= (0, 0, 0)^T. \end{aligned}$$

Както казахме, ще използваме функцията *AB4* за нейното решаване, като малко ще я модифицираме. Това се налага, предвид факта, че обикновено задачата е скалирана така (т.е. са избрани такива мерни единици), че радиусът на капилара да бъде единица. Това означава, че трябва да интегрираме, докато r стане 1.

Вземайки предвид, че капката е осево-симетрична, то е достатъчно да вземем само половината от нея.

Прилагаме по-долу модифицирания код, заедно с приложението му за определяне на профила на капката при стойности на параметрите $b = 1.84366$ и $c = -2.9$.

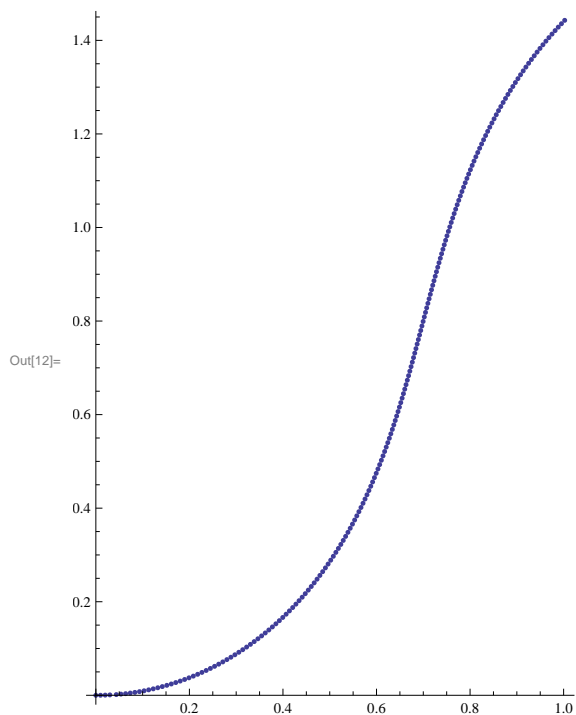
```

In[7]:= AB4[f_, u0_, h_] := (
  y = Table[0, {4}];
  fi = Table[0, {4}]; (*list of values of the right-hand side f*)
  y[[1]] = u0;
  (*Compute the first approximate values, using the RK4 method*)
  For[i = 1, i ≤ 3, i++,
    k1 = h f[y[[i]]];
    k2 = h f[y[[i]] + k1 / 2];
    k3 = h f[y[[i]] + k2 / 2];
    k4 = h f[y[[i]] + k3];
    y[[i + 1]] = y[[i]] +  $\frac{1}{6}$  (k1 + 2 k2 + 2 k3 + k4);
    fi[[i + 1]] = f[y[[i + 1]]];
  ];
  i = 4;
  (*Compute the remaining values, using the AB4 formula*)
  While[y[[i, 1]] ≤ 1,
    y = Append[y,
      y[[i]] +  $\frac{h}{24}$  (55 fi[[i]] - 59 fi[[i - 1]] + 37 fi[[i - 2]] - 9 fi[[i - 3]])];
    fi = Append[fi, f[y[[i + 1]]]];
    i++;
  ];
  y
)

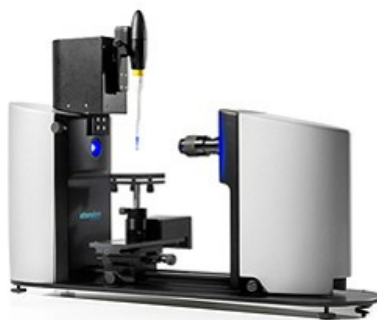
In[8]:= b = 1 / 0.5424
c = -2.9;
f[u_] := {
  Cos[u[[3]]],
  Sin[u[[3]]],
  If[u[[1]] ≠ 0, 2 b + c u[[2]] -  $\frac{\text{Sin}[u[[3]]]}{u[[1]]}$ , b]
}
res = AB4[f, {0, 0, 0}, 0.01];
ListPlot[Table[{res[[i, 1]], res[[i, 2]]}, {i, 1, Length[res]}],
  AspectRatio → Automatic]

```

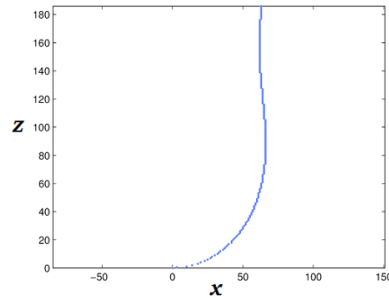
В резултат от изпълнението на горния код, получаваме следния профил на капка.



Решаването на горната система ОДУ е част от т.нар. Axisymmetric Drop Shape Analysis. Това е метод, на базата на който работят много уреди за определяне на повърхностно напрежение на течности (тензиометри).



Повърхностното напрежение е характеристика на течностите, която ги кара да възприемат форма, при която повърхността им е минимална. Направата на измервателни уреди за измерване на повърхностно напрежение е голяма индустрия. За тази цел капка от дадено вещество се пуска от капиляр. На капката се прави снимка, която се дигитализира и така профилът се описва с множество от точки (заради симетрията разглеждаме само половината).



От друга страна, отчитайки силите, действащи върху капката, както видяхме, може да се определи профилът при зададени параметри. Софтуерът, управляващ уредите, определя стойността на параметрите (в частност параметърът c , в който участва повърхностното напрежение) така, че експерименталният и теоретичният профил да съвпадат.

2.8.3 Решаване на твърди задачи.

Нека разгледаме задачата на Коши

$$\begin{aligned} \frac{du}{dt} &= \lambda(-u + \sin t), \quad 0 < t \leq 10, \\ u(0) &= 0, \end{aligned}$$

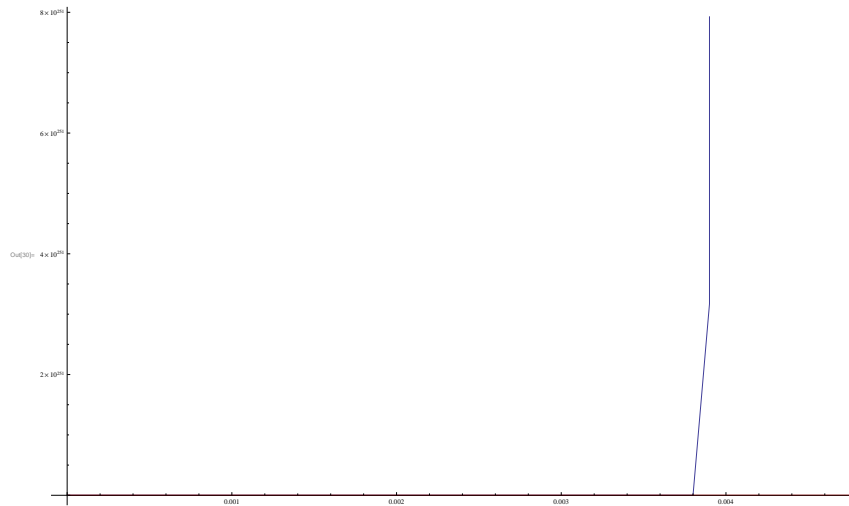
имаща точно решение

$$u(t) = \frac{\lambda}{1 + \lambda^2} e^{-\lambda t} + \frac{\lambda^2}{1 + \lambda^2} \sin t - \frac{\lambda}{1 + \lambda^2} \cos t.$$

Да решим задачата при $\lambda = 1000000$. Първата ни идея може да бъде да използваме явен метод, поради относителната простота и бързодействие спрямо неявните. Нека приложим например метода на Рунге-Кута от четвърти ред.

```
In[24]:= λ = 1 000 000;
f[t_, y_] := λ (-y + Sin[t]);
y = RK4[f, 0, 0.0001, 0, 10];
plotAppr = ListLinePlot[Table[{t[[i+1]], y[[i+1]]}, {i, 0, n}]];
c = λ / (1 + λ^2);
plotExact = Plot[c E^-λ t + λ^2 / (1 + λ^2) Sin[t] - λ / (1 + λ^2) Cos[t], {t, 0, 10}, PlotStyle -> Red];
Show[plotAppr, plotExact]
```

Да отбележим, че сме използвали доста малка стъпка -10^{-4} . Резултатът от изпълнението на горния код е следният:



Ясно е, че причината за получения резултат е условната устойчивост на метода. За конкретната задача условието за устойчивост означава, че стъпката трябва да е твърде малка, за да бъде методът устойчив, което го прави на практика неприложим.

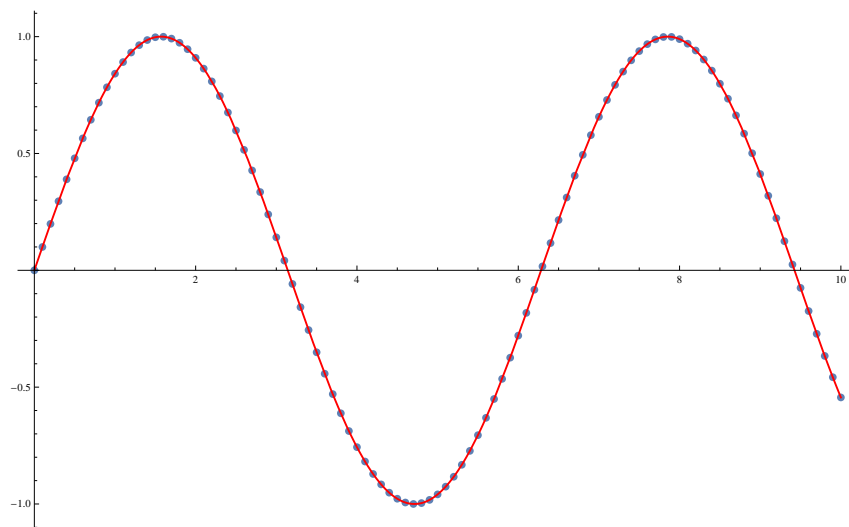
Това е пример за т.нар. **твърди задачи**. Естествен подход тогава е да опитаме да решим задачата с неявен метод. Ще изпробваме неявния метод на Ойлер за целта.

```

λ = 1 000 000;
f[t_, y_] := λ (-y + Sin[t]);
y = implicitEuler[f, 0, 0.1, 0, 10];
plotAppr = ListPlot[Table[{t[[i + 1]], y[[i + 1]]}, {i, 0, n}],
  PlotRange → All];
c =  $\frac{\lambda}{1 + \lambda^2}$ ;
plotExact = Plot[c E-λ t +  $\frac{\lambda^2}{1 + \lambda^2}$  Sin[t] -  $\frac{\lambda}{1 + \lambda^2}$  Cos[t], {t, 0, 10}, PlotStyle → Red];
Show[plotAppr, plotExact, PlotRange → All]

```

Резултатът (при използвана стъпка $h = 0.1$) е следният:



Виждаме, че дори неявният метод на Ойлер, който има първи ред на сходимост, се справя добре с решаването на задачата.

2.8.4 Решаване на неустойчиви по начални данни задачи.

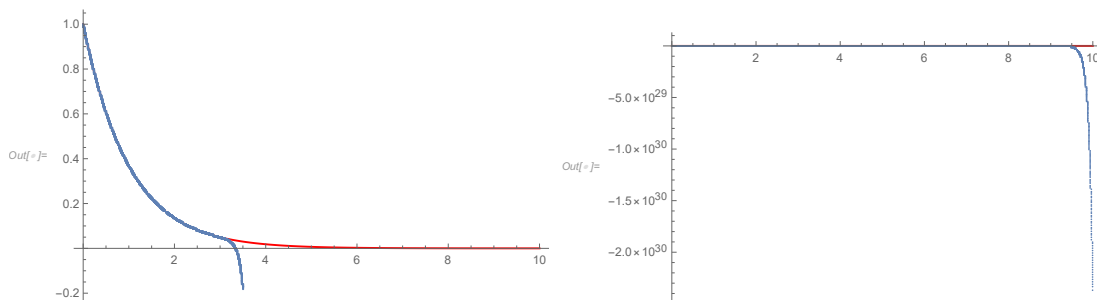
Нека разгледаме задачата на Коши

$$\begin{aligned} \frac{d^2u}{dt^2} - 10\frac{du}{dt} - 11u &= 0, \quad 0 < t \leq 10, \\ u(0) &= 1, \\ u'(0) &= -1 \end{aligned}$$

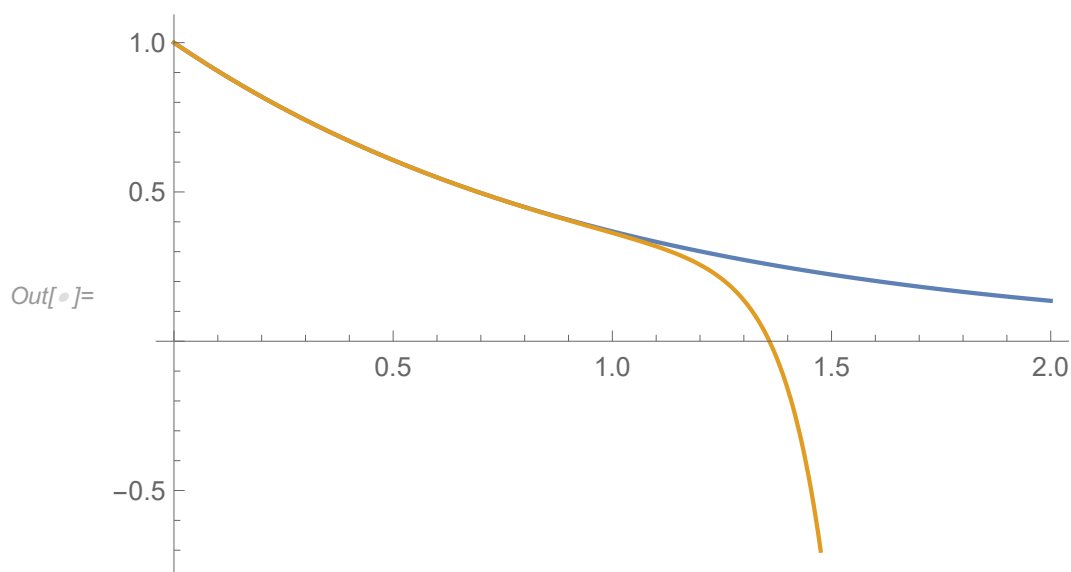
имаща точно решение

$$u(t) = e^{-t}.$$

Без значение кой от методите приложим за задачата приближеното решение ще се различава много от точното от някакво място нататък. Например за четиристъпковия метод на Адамс–Башфорт със стъпка $h = 0.001$ получаваме следния резултат:



Резултатите от решаването на тази задача са лоши, тъй като задачата е неустойчива по начални данни. Това значи, че ако променим малко началното условие (например $u(0) = 0.999999$), решението се променя много:



Тъй като при решаване на задачата приближено на всяка стъпка внасяме грешка, то не е възможно да възстановим решението в достатъчно голям интервал.

2.8.5 Сравнение на методите за числено решаване на ОДУ

В настоящия параграф ще направим едно кратко сравнение на база на няколко признака между методите за числено решаване на ОДУ, без да претендираме за каквато и да е изчерпателност.

1. Явни методи

(а) Метод на Ойлер

- + Бързи и лесни за имплементиране
- Бавно сходящи.

(б) Методи на Рунге-Кута

- + Съществуват методи с висок ред на сходимост
- + По-добри свойства от гледна точка на устойчивост спрямо методите на Адамс-Башфорт
- + Удобни са в случаи, когато искаме да изпълним дадена точност и за използване с адаптивен избор на стъпката
- По-бавни от методите на Адамс-Башфорт

(в) Методи на Адамс-Башфорт

- + Съществуват методи с висок ред на сходимост
- + По-бързи спрямо методите на Рунге-Кута
- + Често се комбинират с неявните методи на Адамс-Мултон като предикторно-коректорен метод
- По-лоши свойства от гледна точка на устойчивост спрямо методите на Рунге-Кута

2. Явни срещу неявни методи

- Явните методи са значително по-бързи.
- За твърди задачи явните методи са практически неприложими.
- Често неявните методи се използват заедно с явен метод в двойка предиктор-коректор.

На база на горното сравнение още веднъж ще подчертаем, че познаването на различни методи ни позволява да използваме метод, който е подходящ, от гледна точка на изискванията на конкретната приложна задача, която решаваме.

Библиография

- [1] J. Butcher, Numerical Methods for Ordinary Differential Equations. Wiley, 2nd edition, 2009.
- [2] E. Coddington, An Introduction to Ordinary Differential Equations. Dover Publications, Unabridged edition, 1989.
- [3] С. Димова, Т. Черногорова, А. Йотова, Числени методи за диференциални уравнения. Университетско издателство “Св. Климент Охридски”, София, 2010.
- [4] J. Hale, Ordinary Differential Equations. Dover Publications, 2009.
- [5] M. Hirsh, S. Smale, R. Devaney, Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic Press, 3rd edition, 2012.
- [6] M.H. Holmes, Introduction to Numerical Methods in Differential Equations. Springer, 2007.
- [7] J. Murray, Mathematical Biology I. An Introduction. Springer, 3rd edition, 2002.
- [8] M. Tenenbaum, H. Pollard, Ordinary Differential Equations. Dover Publications, Revised ed., 1985.