

Diferenciálne rovnice – numerické metódy

Použijeme:

```
[t_out, x_out] = ode45(FUN,tspan,x0,pair,ode_fcn_format,tol,trace,count,hmax)
```

Príklad Nájdime riešenie diferenciálnej rovnice $y' = y + t$, ktoré spĺňa začiatočnú podmienku $y(0) = 1$.

Riešenie:

```
octave:1> function dy=fundif(t,y)           % zápis diferenciálnej rovnice pomocou m - súboru
> dy=[y(1)+t];
> endfunction
```

```
octave:2> [t_out,y_out]=ode45('fundif',[0,1],1,0,0,0.001,0)
t_out =                                     % výsledkom sú hodnoty premenných t a y
    0.00000
    0.01000
    0.35292
    0.75292
    1.00000
y_out =
    1.0000
    1.0101
    1.4935
    2.4935
    3.4366
```

Príklad Nájdime riešenie diferenciálnej rovnice $y'' - 3y' + 2y = 2e^{3x}$, ktoré spĺňa začiatočné podmienky $y(0) = 1, y'(0) = 3$.

Riešenie:

```
octave:5> function dy=fundif(t,y)           % zápis diferenciálnej rovnice pomocou m-súboru
> dy=[y(2);3*y(2)-2*y(1)+2*exp(3*t)];
> endfunction
```

```
octave:6> [t_out,y_out]=ode45('fundif',[0,1],[1,3],0,0,0.01,0)
t_out =                                     % výsledkom sú hodnoty premenných t, y a y'
    0.00000
    0.01000
    0.27051
    0.67051
    1.00000
y_out =
    1.0000    3.0000
    1.0305    3.0914
    2.2514    6.7541
    7.4749   22.4245
   20.0861   60.2578
```

Ďalšie možnosti sú uvedené v: Usage:

```
[t_out, x_out] = ode45(FUN,tspan,x0,pair,ode_fcn_format,tol,trace,count,hmax)
[tout, xout] = rk8fixed(FUN, tspan, x0, Nsteps, ode_fcn_format, trace, count)
```

```

[tout, xout] = rk4fixed(FUN, tspan, x0, Nsteps, ode_fcn_format, trace, count)
[tout, xout] = rk2fixed(FUN, tspan, x0, Nsteps, ode_fcn_format, trace, count)
[tout, xout] = ode78(FUN,tspan,x0,ode_fcn_format,tol,trace,count,hmax)
[tout, xout] = ode23(FUN,tspan,x0,ode_fcn_format,tol,trace,count,hmax)

```

Kde

INPUT:

```

% FUN - String containing name of user-supplied problem description.
%      Call: xprime = fun(t,x) where FUN = 'fun'.
%      t    - Time (scalar).
%      x    - Solution column-vector.
%      xprime - Returned derivative COLUMN-vector; xprime(i) = dx(i)/dt.
% tspan - [ tstart, tfinal ]
% x0    - Initial value COLUMN-vector.
% ode_fcn_format - this specifies if the user-defined ode function is in
%      the form:  xprime = fun(t,x) (ode_fcn_format=0, default)
%      or:       xprime = fun(x,t) (ode_fcn_format=1)
%      Matlab's solvers comply with ode_fcn_format=0 while
%      Octave's lsode() and sdirk4() solvers comply with ode_fcn_format=1.
% tol   - The desired accuracy. (optional, default: tol = 1.e-6).
% trace - If nonzero, each step is printed. (optional, default: trace = 0).
% count - if nonzero, variable 'rhs_counter' is initialized, made global
%      and counts the number of state-dot function evaluations
%      'rhs_counter' is incremented in here, not in the state-dot file
%      simply make 'rhs_counter' global in the file that calls ode23
% hmax  - limit the maximum stepsize to be less than or equal to hmax
%

```

% OUTPUT:

```

% tout - Returned integration time points (column-vector).
% xout - Returned solution, one solution column-vector per tout-value.

```