Wykład 14: ZAGADNIENIA SZTYWNE – ciąg dalszy

1 Uzupełnienie zagadnień, interpolacja wyników numerycznych

Będziemy rozwiązywać kilkoma metodami: *ode15s.m , ode113.m , ode45.m* RR postaci

```
function dy = deriv(t,y)
  dy = lambda*y + (1-lambda)*cos(t) - (1+lambda)*sin(t);
  num_fcn_eval = num_fcn_eval + 1;
end % deriv
```

o rozwiązaniu dokładnym

```
function true = true_soln(t)
  true = sin(t) + cos(t);
end % true_soln
```

dla kilku wartości parametru λ: -10 , -1000 , -10000 .

Obejrzyjmy informacje o funkcjach:

ode45 Solve non-stiff differential equations, medium order method.
[TOUT,YOUT] = ode45(ODEFUN,TSPAN,Y0) with TSPAN = [T0 TFINAL]
integrates the system of differential equations y' = f(t,y) from time T0 to TFINAL with initial conditions Y0.
ODEFUN is a function handle.
For a scalar T and a vector Y, ODEFUN(T,Y) must return a column vector corresponding to f(t,y).
Each row in the solution array YOUT corresponds to a time returned in the column vector TOUT.
To obtain solutions at specific times T0,T1,...,TFINAL (all increasing or all decreasing), use
TSPAN = [T0 T1 ... TFINAL].
[TOUT,YOUT] = ode45(ODEFUN,TSPAN,Y0,OPTIONS) solves as above with default integration properties replaced by values in OPTIONS, an argument created with the ODESET function. See ODESET for details.
SOL = ode45(ODEFUN,[T0 TFINAL],Y0...) returns a structure that can be used with DEVAL to evaluate the solution or its first derivative at any point between T0 and TFINAL.

ode113 Solve non-stiff differential equations, variable order method.

ode15s Solve stiff differential equations and DAEs, variable order method.
ode15s can solve problems M(t,y)*y' = f(t,y) with mass matrix M(t,y).
The Jacobian matrix df/dy is critical to reliability and efficiency. Use ODESET to set 'Jacobian' to a function handle FJAC if FJAC(T,Y) returns the Jacobian df/dy or to the matrix df/dy if the Jacobian is constant. If the 'Jacobian' option is not set (the default), df/dy is approximated by finite differences.

1.1 Interpolacja wyników numerycznych

» help deval

DEVAL Evaluate the solution of a differential equation problem.
SXINT = DEVAL(SOL,XINT) evaluates the solution of a differential equation problem at all the entries of the vector XINT. SOL is a structure returned by an initial value problem solver (ODE45, ODE23, ODE113, ODE15S, ODE23S, ODE23T, ODE23TB, ODE15I), the boundary value problem solver (BVP4C), or the solver for delay differential equations (DDE23).
The elements of XINT must be in the interval [SOL.x(1) SOL.x(end)].
For each I, SXINT(:,I) is the solution corresponding to XINT(I).

1.2 Początek programu testowego

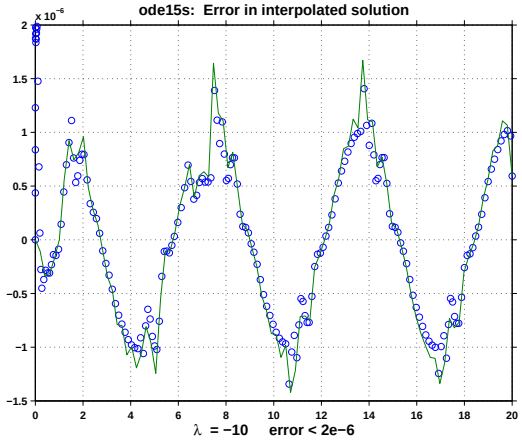
```
function test_ode15s(lambda, relerr, abserr)
options = odeset('RelTol', relerr, 'AbsTol', abserr);
t_begin = 0;  t_end = 20;
y_initial = true_soln(t_begin);
num_fcn_eval = 0;
soln = ode15s(@deriv, [t_begin,t_end], y_initial, options);
h_plot = (t_end-t_begin)/1000;      t_plot = t_begin:h_plot:t_end;
y_plot = deval(soln,t_plot);
figure,    plot(soln.x,soln.y,'o',t_plot,y_plot)
   y_true_nodes = true_soln(soln.x);    error_nodes = y_true_nodes - soln.y;
   y_true = true_soln(t_plot);          error = y_true - y_plot;
figure,    plot(soln.x,error_nodes,'o',t_plot,error)
```

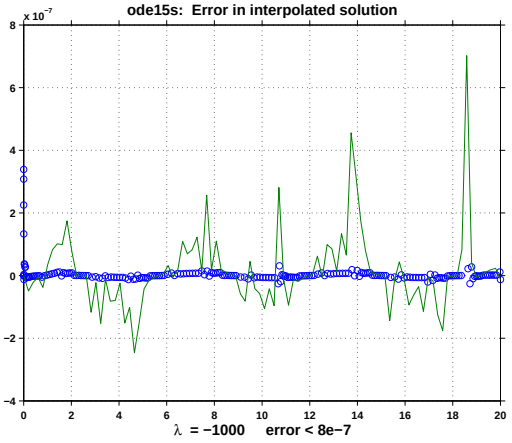
» help odeset

odeset Create/alter ODE OPTIONS structure.
OPTIONS = odeset('NAME1',VALUE1,'NAME2',VALUE2,...) creates an integrator options structure OPTIONS in which the named properties have the specified values.
odeset with no input arguments displays all property names and their possible values.

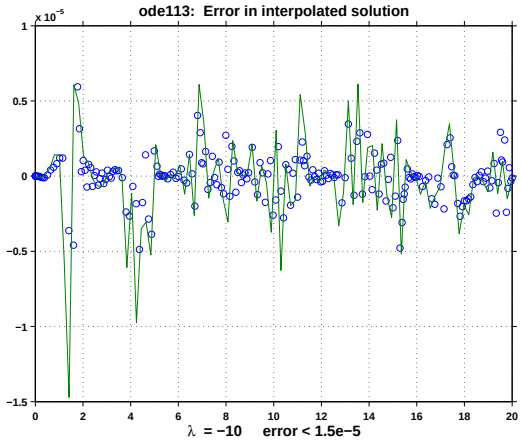
1.3 Wyniki programów dla $\lambda = -10$, $\lambda = -1000$ i $\lambda = -10000$

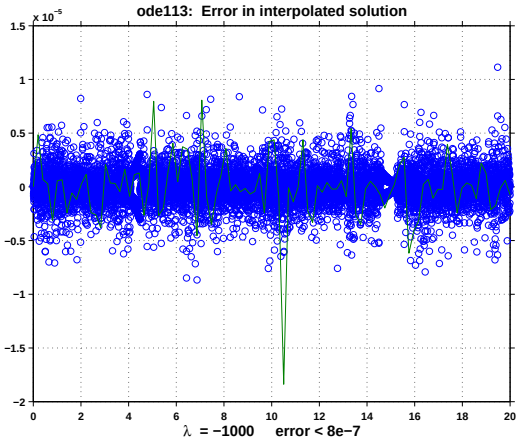
ode15s



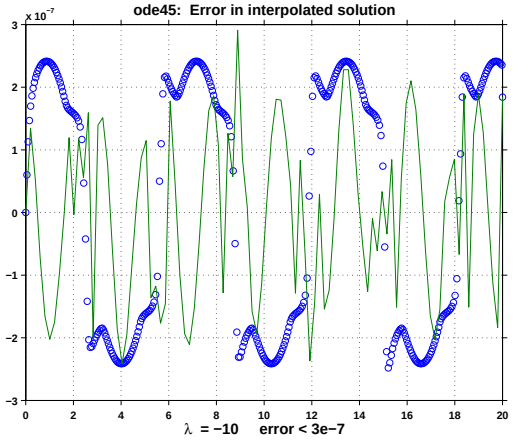


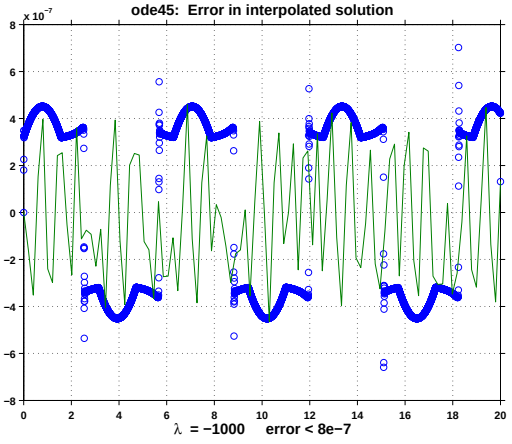
ode113





ode45





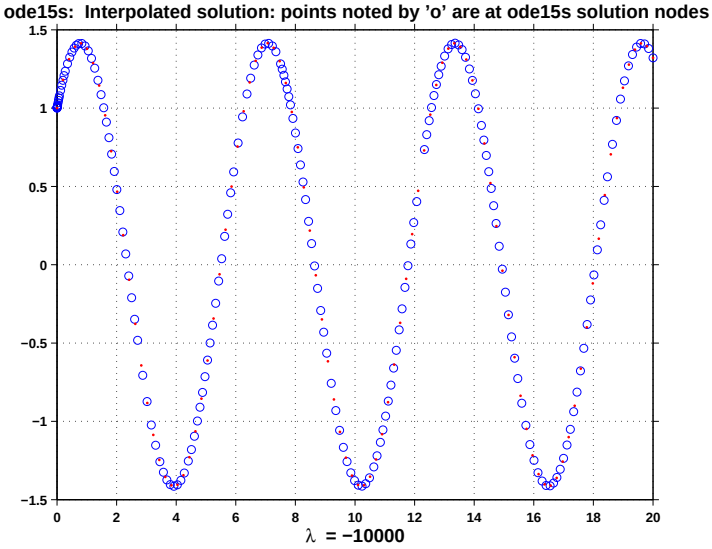
Nr	Parameters	λ = -10			λ = -1000			λ = -10000		
		ode45	ode15s	ode113	ode45	ode15s	ode113	ode45	ode15s	ode113
1	NSuccSteps	362	188	217	7316	204	12489	60267	198	124125
2	NFailed	6	14	10	8	26	1689	3315	18	16991
3	NFEVals	2209	407	445	43945	463	26668	381493	435	265242
4	N-DF/Dy	0	1	0	0	1	0	0	1	0
5	N-LU-decomp	0	38	0	0	54	0	0	43	0
6	NSolLinEq	0	404	0	0	460	0	0	432	0
7	Time [s]	0.80	0.77	0.71	7.78	0.78	11.58	53.35	0.77	189.38

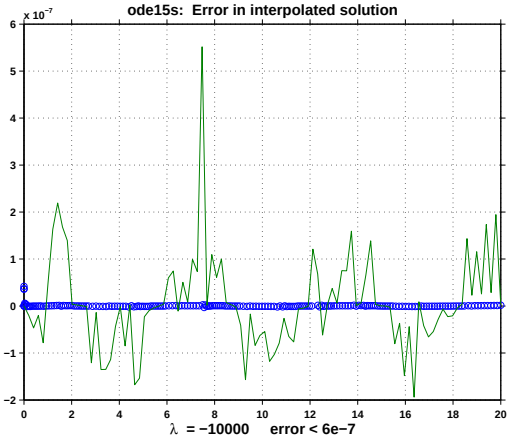
1.4 Wywołując *test_ode15s(-10000, 1e-6, 1e-6)* otrzymujemy

```
soln =
     solver: 'ode15s'
    extdata: [1x1 struct]
          x: [1x189 double]
          y: [1x189 double]
      stats: [1x1 struct]
      idata: [1x1 struct]
```

```
soln.stats =
      nsteps: 198
    nfailed: 18
    nfevals: 435
       npds: 1
   ndecomps: 43
    nsolves: 432
```

z następującymi wykresami





2 Zadania na ćwiczenia lub na pracownię

1. Proszę spróbować rozwiązać metodą strzałów ***nieliniowe zagadnienie brzegowe*** dla funkcji $Q = Q(x)$:

$$\left(\frac{Q'}{x^2}\right)' - \frac{QQ'}{x^4} = 0$$

na odcinku $(0, 1)$ z warunkami:

$$Q(0) = 0 \quad \text{i} \quad Q(1) = M \quad \text{dla zadanej stałej} \quad M > 0 \text{ , (np. } M = 10,\ 50,\ 500, ...).$$

* * *