Lösung von Randwertproblemen mit bvp4c.m (II)

$$\left.\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}\tau}(\tau) &= T\cdot\Big(u(1-u)-\frac{auv}{u+d}\Big) \\ \frac{\mathrm{d}v}{\mathrm{d}\tau}(\tau) &= T\cdot\Big(bv\cdot(1-\frac{v}{u})\Big) \\ \frac{\mathrm{d}T}{\mathrm{d}\tau}(\tau) &= 0 \end{aligned}\right\}, \quad (\tau\in(0,1)), \quad \left\{\begin{aligned} u(0) &= 0.3, \\ u(0) &= u(1), \\ v(0) &= v(1). \end{aligned}\right.$$

Anfangsnäherung

```
% -> get initial guess for bvp4c.m by solving an initial value problem
options = odeset ( 'AbsTol', atol, 'RelTol', rtol );
solode = ode45 ( @wperrhs, [ 0 1 ], [ u0init, v0init, Tinit ]', options, u0init );

Ninit  = 50;           tauini = linspace ( 0, 1, Ninit+1 );
solinit.x = tauini;    solinit.y = deval ( solode, tauini );
```

Randbedingungen

```
function r = perbc ( Y0, Y1, u0init );
u0 = Y0(1);    u1 = Y1(1);    v0 = Y0(2);    v1 = Y1(2);    % read input data
r = zeros ( size ( Y0 ) );                                   % memory allocation
r(1) = u0 - u0init;   r(2) = u1 - u0;    r(3) = v1 - v0;    % evaluate residual in b.c.
```