Numerical Treatment of Singular Implicit BVPs – New Matlab Code bvpsuite

Ewa B. Weinmüller

We consider boundary value problems for systems in ODEs which exhibit singular points in the interval of integration. These problems typically have the form

$$z'(t) = 1/t^\alpha f(t, z(t)), \;\; t \in (0,1], \quad g(z(0), z(1)) = 0,$$

where α is non-negative. Depending on the value of α one distinguishes between singularity of the first kind, $\alpha = 1$, and essential singularity, $\alpha > 1$. Another interesting class of problems

$$z'(\tau) = \tau^\beta f(\tau, z(\tau)), \quad \tau \in [1, \infty), \quad \beta \geq -1$$

is also covered since it can be transformed to a finite interval by $t := 1/\tau$,

$$z'(t) = -\frac{1}{t^{\beta+2}} f(1/t, z(1/t)), \quad t \in (0,1], \quad \beta + 2 \geq 1.$$

We first discuss the analytical properties of such problems, especially the existence and uniqueness of bounded solutions, an important prerequisite for the well-posedness of the system. We also point out typical difficulties arising in convergence theory for standard discretization methods.

A `Matlab` code `colimp` (a new version of `sbvp1.0`) for the numerical treatment of the above problem class is presented. This solver is based on polynomial collocation, equipped with a posteriori estimates of the global error of the solution $z(t)$, and features an adaptive mesh selection procedure based on the equidistribution of the global error. Moreover, in the present version of the code a pathfollowing strategy based on pseudo-arclength parametrization applied for the computation of solution branches with turning points of parameter-dependent equations,

$$z'(t) = 1/t^\alpha f(t, z(t), \lambda), \;\; t \in (0,1], \quad g(z(0), z(1)) = 0,$$

has been implemented. This pathfollowing procedure is well-defined under realistic assumptions, and a numerical solution is possible with a stable, high-order discretization method.

Finally, we demonstrate the reliability and efficiency of our code by solving a few problems relevant in applications.