School on DDEs Dobbiaco, 30 June 2006

RADAR5, a code for implicit delay differential equations

Nicola Guglielmi, L'Aquila

and

Ernst Hairer, Geneva

Outline

(1) The class of considered problems.

(2) Main numerical difficulties.

(3) Basic numerical process, continuous output.

(4) Error control, breaking points, RK equations.

(5) Neutral problems.

(6) Checking termination and bifurcation.

(7) A model problem from immunology.

The problem class

$$\begin{cases} M\,\dot{y}(t) \;=\; f\Big(t, y(t), y(\alpha^{(1)}(t, y(t))), \ldots, y(\alpha^{(m)}(t, y(t)))\Big) \\ \\ \quad y(t_0) \;=\; y_0, \qquad\quad y(t) \;=\; g(t) \quad \text{for } \; t < t_0 \end{cases} \tag{GP}$$

where

- $y \in \mathbf{R}^d \quad (d \geq 1)$;
- f is a real-valued vector function;
- M is a constant (possibly singular) $d \times d$ matrix;
- $\alpha^{(i)}(t, y) \leq t \quad (\alpha^{(i)}(t, y) = t - \tau^{(i)}(t, y))$.

Motivations

(1) If a PDE with delays is discretized in space by finite elements, we obtain an equation of the form (GP) where M is the mass matrix. A multiplication of the equation by M^{-1} would destroy the sparsity pattern.

(2) Allowing M to be singular, all kinds of differential algebraic delay equations are included. For the choice $M = \mathrm{diag}(I, \varepsilon I)$ with a small $\varepsilon > 0$, we get singularly perturbed problems, an important class of stiff problems.

(3) Introducing a new variable $v(t) = \dot{y}(t)$, neutral problems

$$\dot{y}(t) = f\Big(t, y(t), y(\alpha(t,y)), \dot{y}(t), \dot{y}(\beta(t,y))\Big)$$

can be written in the form (GP).

Leading aspects in the integration

(1) By the assumptions the problem requires to be integrated by an implicit continuous scheme.

(2) Since jump discontinuities in the solution or its derivatives are possible, an unconstrained application of high order methods may determine a loss of accuracy.

(3) For small delays the necessary approximation of the solution may be not available.

(4) The non-regularity of the initial function (or the singularity of M) may determine the loss of either the uniqueness or also the existence of the solution of a state-dependent delay differential equation.

Integration by the code RADAR5

For simplicity we consider now a single lag term $\alpha(t, y)$.
For the numerical approximation of the solution on a grid $t_0 < t_1 < t_2 < \ldots$ (determined adaptively) we make use of a continuous method and integrate problems of the form:

$$\begin{cases} M\,\dot{y}(t) &= f(t, y(t), \mathbf{u}(t)), \quad t \in [t_n, t_{n+1}] \\ \\ y(t_n) &= y_n, \end{cases}$$

where $\mathbf{u}(t)$ is a piecewise–polynomial approximation to $y\left(\alpha(t, y(t))\right)$.

The code RADAR5 is developed in FORTRAN-90 and is based on an adaptation of the 3-stage Radau IIA method to implicit delay differential equations.

Basic numerical process (s-stage Radau IIA)

$$M\left(Y_i^{(n)} - y_n\right) = h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j^{(n)}, Z_j^{(n)})$$

$$i = 1, \cdots, s$$

$$y_{n+1} = Y_s^{(n)}$$

with

$$Z_j^{(n)} = \begin{cases} g\left(\alpha_j\right) & \text{if } \alpha_j \leq t_0 \\ \mathbf{u}\left(\alpha_j\right) & \text{if } t_0 < \alpha_j \leq t_{n+1}. \end{cases}$$

where

- $\mathbf{u}(t)$ is a dense approximation to the solution;
- $\alpha_j := \alpha\left(t_n + c_j h, Y_j^{(n)}\right)$.

Dense output in $[t_m, t_{m+1}]$

We can consider two options:

(a) the collocation polynomial

$$u_m(t_m + \theta\, h_m) = \ell_0(\theta)\, y_m + \sum_{i=1}^{s} \ell_i(\theta)\, Y_i^{(m)}, \qquad \theta \in [0, 1)$$

(b) the lower order polynomial

$$v_m(t_m + \theta\, h_m) = \sum_{i=1}^{s} \hat{\ell}_i(\theta)\, Y_i^{(m)}, \qquad \theta \in [0, 1)$$

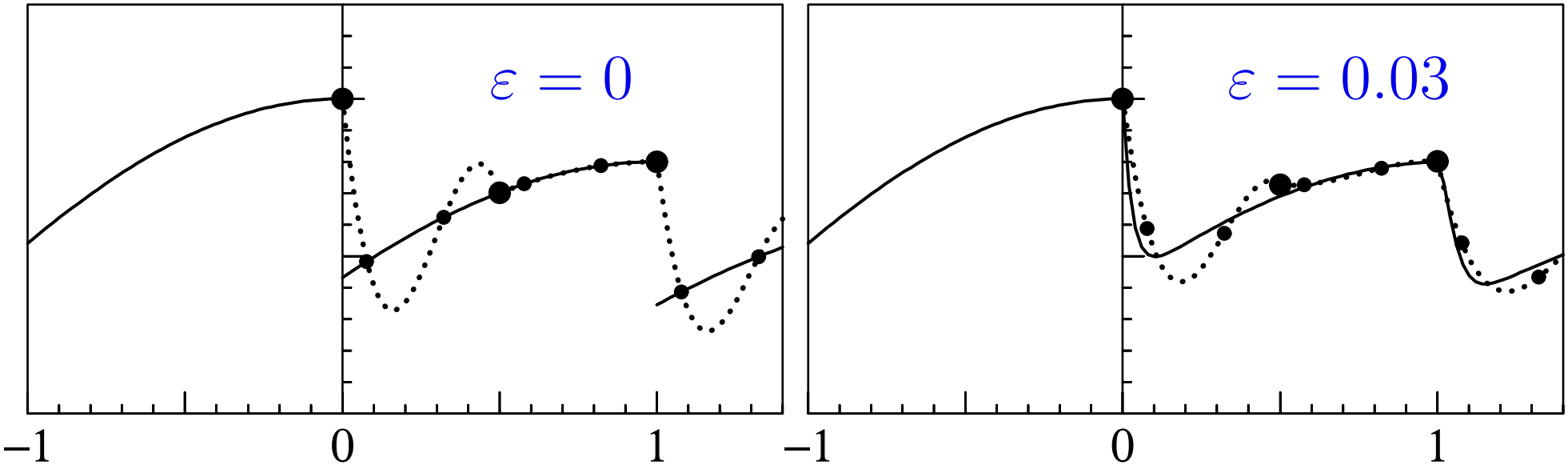
where $\ell_i(\theta)$ is the i-th Lagrange polynomial of degree s on the abscissæ $\{c_0, c_1, \ldots, c_s\}$ $(c_0 = 0)$ and $\hat{\ell}_i(\theta)$ the i-th Lagrange polynomial of degree $s - 1$ on the abscissæ $\{c_1, \ldots, c_s\}$.

Choice of the polynomial

(a) If t_m is a jump-discontinuity the polynomial v_m is chosen.

(b) In other cases the code uses u_m which is more accurate.

Example: $\varepsilon \dot{y}(t) = -y(t) + 0.8y(t-1)$ with $g(t) = \cos t$.



Numerical solution (big bubbles) & internal stages (small); the dotted line is the collocation polynomial u_0 ($h = 0.5$).

Error control

Step size selection strategies for stiff ordinary differential equations are usually based on error estimations at grid points. For delay equations, where the accuracy of the dense output strongly influences the performance, this is not sufficient.

Usual (discrete) error estimation

It is obtained, as in the standard ODE framework, by embedding a method of lower order into the Radau method (see Hairer & Wanner '96).

Continuous error estimation

We use the following quantity as an indicator for the uniform error and denote it as dense component of the local error:

$$\max_{\vartheta\in[0,1]} \|u_n(t_n+h_n\vartheta)-v_n(t_n+h_n\vartheta)\| = \|y_n-v_n(t_n)\| = \mathcal{O}(h_n^s)$$

Breaking points

Initial irregularities may propagate along the integration interval by means of the deviating argument $\alpha(t, y(t))$.

Recursive definition

$$J_0 = \{t_{-\ell}, \ldots, t_{-1}, t_0\} \text{ contains the initial point } t_0$$

and the discontinuity points in the initial function

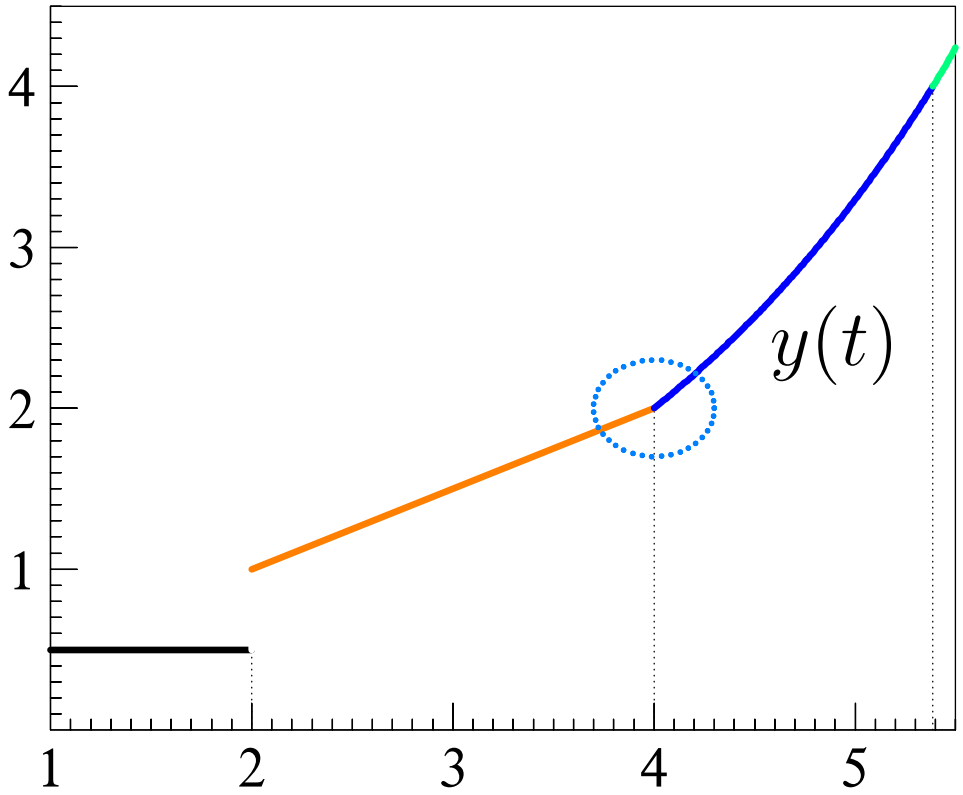
$$J_k = J_{k-1} \bigcup \{\xi : \ \alpha(\xi, y(\xi)) = \zeta \quad \text{for some } \zeta \in J_{k-1}\}$$

Literature on breaking points detection: discontinuity tracking, defect control, error control.

Example 1 (C. Paul testset)

$$\dot{y}(t) = y(y(t)) \qquad \text{for} \qquad t \geq 2$$

with initial condition $y(t) = 0.5$ for $t < 2$, and $y(2) = 1$.



Solution is only continuous at $t = 4$ (where $y(t) = 2$).

Solution is only C^1-continuous at $t \approx 5.38$ (where $y(t) = 4$).

Related literature on computing breaking points

- Neves & Feldstein (1984) propose to check in every succesful step if (for some previous breaking point ζ)

$$\alpha(t, u_n(t)) - \zeta$$

changes sign in $[t_n, t_{n+1}]$.

Remark: the use of $u_n(t)$ is dangerous because if the solution is not smooth $u_n(t)$ may be a bad approximation.

- Hauber (1997), Enright & Hayashi (1997) check instead

$$\alpha(t, u_{n-1}(t)) - \zeta \qquad \text{for } t \in [t_n, t_{n+1}].$$

extrapolating (which may lead to large errors)

$$u_{n-1}(t_{n-1} + \vartheta h_{n-1}) \qquad \text{for } \vartheta > 1.$$

Detecting breaking points

We expect a breaking point in the interval $[t_n, t_n + h]$ if the following two conditions occur:

1. the step is rejected, i.e.,
   - the iterative solver for the nonlinear system fails to converge

   or
   - the local error estimate is not small enough;
2. there exists a previous breaking point ζ such that

$$\alpha(t, u_{n-1}(t)) - \zeta$$

changes sign on $[t_n, t_n + h]$, where $u_{n-1}(t)$ is the continuous output polynomial of the preceding step.

Computing breaking points

The main idea is that of solving the Runge–Kutta equations coupled with the scalar equation determining the breaking point. This means to consider the Runge–Kutta equations with an a priori unknown step-size.

Since applying a Newton process to the coupled system would destroy the block structure of the Jacobian in the simplified Newton iteration (see later), we apply a splitting method.

The resulting method turns out to converge linearly.
But convergence is very fast due to a very small error constant.

Algorithm C (computes breaking points)

$$(1) \qquad M\left(Y_i^{(n)} - y_n\right) = h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j^{(n)}, Z_j^{(n)})$$

$$i = 1, \cdots, s$$

$$(2) \qquad \alpha(t_n + h, u_n(t_n + h)) = \zeta$$

Till convergence, we alternatively

- solve system (1) with fixed h by simplified Newton iteration, which determines

$$u_n(t_n + \vartheta h) = \ell_0(\theta)\, y_n + \sum_{i=1}^{s} \ell_i(\theta)\, Y_i^{(n)}, \qquad \theta \in [0, 1)$$

- update h with a root finding algorithm applied to (2) with fixed $u_n(\cdot)$.

Convergence of Algorithm C

Theorem 1

Let $u_n^h(t)$ be the continuous approximation of the solution obtained from the solution of (1) with stepsize h, and denote by $h^* = h^*(h)$ the solution of (2), which we assume to be simple, obtained by using such an approximation of the solution $u_n^h(t)$, that is

$$\alpha(t_n + h^*, u_n^h(t_n + h^*)) = \zeta.$$

Then

$$|h^* - \hat{h}| \leq \mathit{Const} \cdot h^s \cdot |h - \hat{h}|,$$

where $\hat{h}$ is the exact solution of (1)-(2), i.e. $t_n + \hat{h}$ is the searched numerical breaking point.

Proof of Theorem 1

We let $g(h, h^*) := \alpha(t_n + h, u_n^h(t_n + h^*)) - \zeta$, and we notice that $g(\hat{h}, \hat{h}) = 0$. Taylor expansion shows that

$$0 = \frac{\partial g}{\partial h}(\hat{h}, \hat{h})(h - \hat{h}) + \frac{\partial g}{\partial h^*}(\hat{h}, \hat{h})(h^*(h) - \hat{h}) + \text{higher order terms.}$$

By the non-degeneracy assumption of the breaking point, we have $\partial g / \partial h^*(\hat{h}, \hat{h}) \neq 0$. Hence, only the dependence on h of $g(h, h^*)$ and thus of $u_n^h(t_n + h^*)$ has to be studied for fixed h^*. By definition of the method, $u_n^h(t)$ is a polynomial of degree s interpolating y_n and $Y_1^{(n)}, \ldots, Y_s^{(n)}$, which approximates the local solution at $t_n + c_i\, h$ with an error of size $\mathcal{O}(h^{s+1})$. Since polynomials of degree s are reproduced without error and independent of h, we have $u_n^h(t) - u_n^{\hat{h}}(t) = \mathcal{O}(h^s(h - \hat{h}))$.

Accuracy of numerical breaking points

Theorem 2

Let $y(t)$ be the exact solution and let ζ and ξ be exact breaking points of the problem such that $\alpha(\xi, y(\xi)) = \zeta$. Further, let ζ_h be an approximation of ζ obtained with sufficiently small step sizes, and let $\xi \in (t_n, t_{n+1})$.

If

$$\frac{d}{dt}(\alpha(t, y(t)))|_{t=\xi} \neq 0,$$

then the breaking point ξ_h computed by Algorithm C satisfies

$$|\xi_h - \xi| \leq C(\|y_{n+1} - y(t_{n+1})\| + |\zeta_h - \zeta|).$$

Proof of Theorem 2

For given h, we denote the solution of the nonlinear system of Runge–Kutta equations (1) by $Y_i^{(n)}(h)$. Since we consider only stiffly accurate Runge–Kutta methods, the numerical approximation at $t_n + h$ is $y_{n+1}(h) := Y_s^{(n)}(h)$.

The equation (2) can thus be written as

$$\alpha(t_n + h, y_{n+1}(h)) = \zeta_h,$$

and an application of the implicit function theorem proves the statement.

Considerations on the global error

For problems (GP) with state dependent delays it is not possible to obtain reasonable error bounds for the global error $y_n - y(t_n)$. It may happen that t_n is a numerically computed breaking point, and the corresponding exact breaking point is slightly different. If at this point the solution has a jump discontinuity, the global error is there $\mathcal{O}(1)$.

It is possible to circumvent this difficulty by introducing a time transformation that is close to the identity.
In the following statement we denote by $y_h(t)$ the numerical solution also if variable step sizes are employed. The value h represents the maximal step size.

Accuracy of the numerical solution

Theorem 3

Assume that f is smooth and that breaking points are simple and well separated and assume that the global error of the Runge–Kutta method is of size $\mathcal{O}(h^r)$ provided that all exact breaking points are included in the mesh.

If, instead of the exact breaking points, those obtained by Algorithm C are used, then we have on bounded intervals

$$\|y_h(t) - y(s)\| \;=\; \mathcal{O}(h^r) \qquad \text{(EB)}$$

where the function $s = s(t)$ satisfies $s = t + \mathcal{O}(h^r)$.

Proof outline of Theorem 3

As soon as one encounters the first breaking point larger than t_0, it may happen that the computed value ξ_h slightly differs from the exact breaking point ξ. Introducing a time transformation $t \leftrightarrow s$ mapping $[t_0, \zeta_h]$ onto $[t_0, \zeta]$ (e.g., a linear transformation), we obtain (EB) for $t \leq \zeta_h$, because, by Theorem 2, $\zeta_h - \zeta = \mathcal{O}(h^r)$.

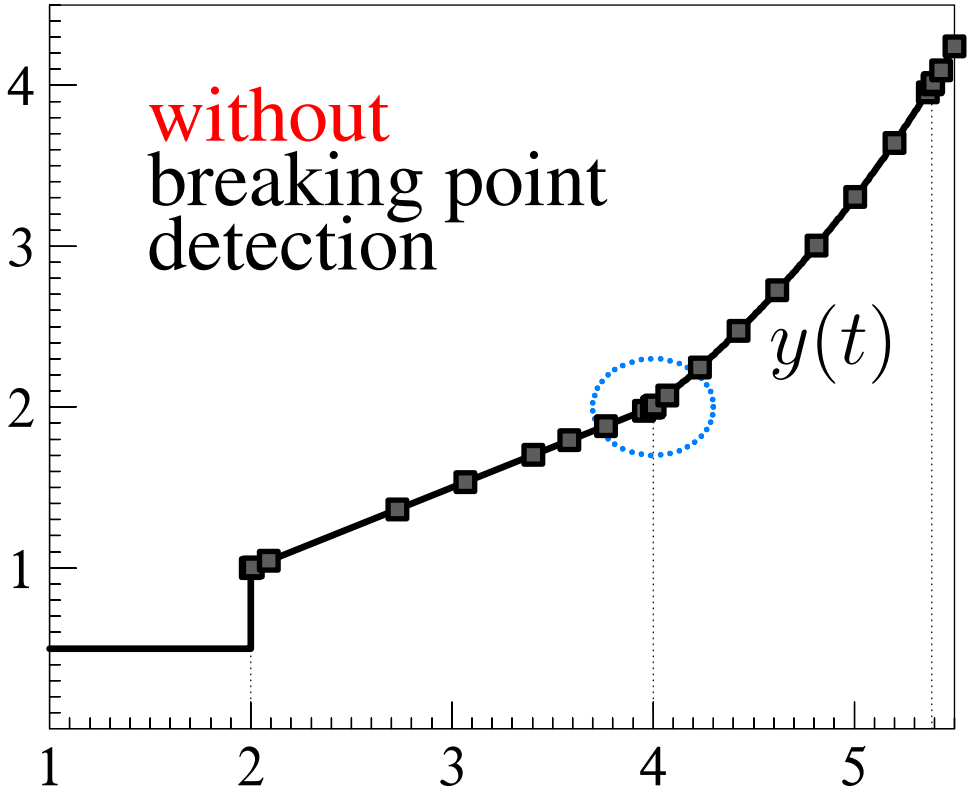
The interval from ζ_h to the next computed breaking point has to be treated in the same way, and so on (this happens a finite number of times).

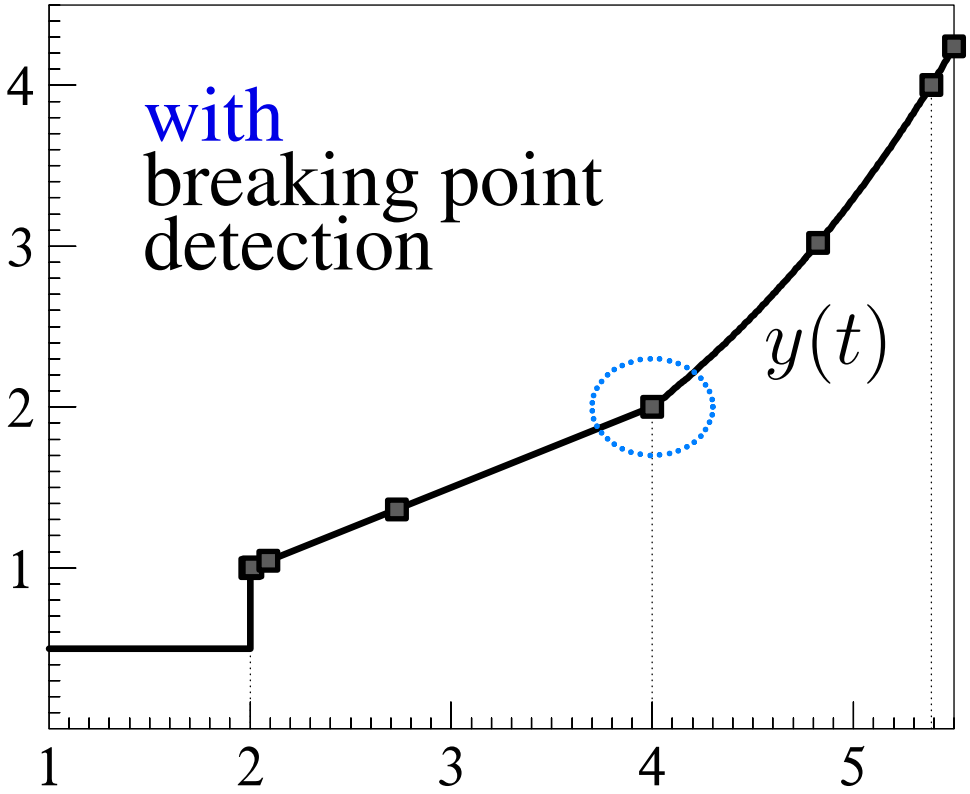
Finally the standard convergence proof (propagation of local errors and accumulation of these errors) has to be applied; see for example Theorem 4.3.5 of (Bellen & Zennaro, 2003).

Numerical example 1 (C. Paul testset)

$$\dot{y}(t) = y(y(t)) \qquad \text{for} \qquad t \geq 2$$

with initial condition $y(t) = 0.5$ for $t < 2$, and $y(2) = 1$.





A quantitative comparison

	RADAR5 – old version				RADAR5 – new version			
Tol	feval	accept	reject	error	feval	accept	reject	error
10^{-3}	175	12	9	$2.8 \cdot 10^{-4}$	80	7	3	$2.6 \cdot 10^{-5}$
10^{-6}	355	24	17	$5.6 \cdot 10^{-6}$	141	13	4	$3.9 \cdot 10^{-8}$
10^{-9}	691	55	29	$2.2 \cdot 10^{-7}$	208	23	5	$1.3 \cdot 10^{-9}$
10^{-12}	1390	152	33	$3.4 \cdot 10^{-10}$	478	61	6	$5.6 \cdot 10^{-13}$

In the old version the breaking points were only implicitly determined by means of the error control. Their direct computation (new version) improves accuracy and efficiency.

Solving the Runge–Kutta equations

$$(1) \qquad M\left(Y_i^{(n)} - y_n\right) = h\sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j^{(n)}, Z_j^{(n)})$$

For stiff problems (or when M is singular), the system cannot be solved by fixed point iteration.

As common in the implementation of implicit Runge-Kutta methods, we pre-multiply the system (1) by $A^{-1} = (\omega_{ij})$ and so obtain the nonlinear system $F(Y) = 0$, where $Y = (Y_1, \ldots, Y_s)^T$ and the i-th component of $F(Y)$ is

$$F_i(Y) = \sum_{j=1}^{s} \omega_{ij} M(Y_j - y_n) - hf(t_n + c_i h, Y_i, Z_i)$$

Structure of the Jacobian

$$J = A^{-1} \otimes M - h\, I_s \otimes \left(\frac{\partial f}{\partial y} + \frac{\partial f}{\partial z} \frac{\partial \alpha}{\partial y} u'_m(\alpha_0) \right)$$

$$- \; h\, I_s \cdot \mathcal{U} \otimes \frac{\partial f}{\partial z}$$

where I_s is the $s \times s$ identity, $\alpha_0 = \alpha(t_n, y_n) \in [t_m, t_{m+1}]$ and $\mathcal{U}$ is the matrix given by

$$\mathcal{U}_{jk} = \begin{cases} \ell_k(\tau_j) & \text{if} \quad \tau_j = \dfrac{\alpha(t_n + c_j h, Y_j) - t_n}{h} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Simplified Newton iteration?

The Jacobian term

$$J_0 = A^{-1} \otimes M - h\, I_s \otimes \left(\frac{\partial f}{\partial y} + \frac{\partial f}{\partial z} \frac{\partial \alpha}{\partial y} u_m{}'(\alpha_0) \right)$$

can be block-diagonalized. But the second term in general prevents this possibility for the whole Jacobian J.

Possible cases

(1) Large delays. The deviating arguments fall on the left of t_n, that is $\mathcal{U} \equiv 0$.
Transforming the matrix A^{-1} to diagonal form, the linear system with matrix $J = J_0$ can be solved efficiently.

(2) Small delays. Some deviating arguments fall into the current step, that is $\mathcal{U} \not\equiv 0$.

Preserving the tensor structure

The case of small delays.

Since, in general, the matrices A^{-1} and $\mathcal{U}$ are not simultaneously diagonalizable, the tensor product structure cannot easily be exploited for an efficient solution of the linear systems with matrix J.

However, for very small delays, $\mathcal{U} \approx I_s$. Consequently, the second and third terms in J can be considered together, and the idea of diagonalizing the matrix A^{-1} can again be applied.

Optimizing the iteration

The tensor structure can be maintained in all steps by approximating $\mathcal{U} \approx \gamma\, I_s$, e.g. with

$$\gamma \longrightarrow \min_{\gamma \in \mathbf{R}} \|\mathcal{U} - \gamma\, I_s\|^2$$

Example 2 (Enright & Hayashi)

$$\begin{aligned} p'(t) &= \cos(t)(1 + p\left(t\,p^2(t)\right)) + L\,p(t)\,p'(tp^2(t)) \\ &+ (1-L)\sin(t)\,\cos\left(t\,\sin(t)^2\right) - \sin\left(t + t\,\sin(t)^2\right) \end{aligned}$$

with $p(0) = 0$ and $p'(0) = 1$.

This is a scalar neutral delay equation, with vanishing delays at $t = 0, \pi/2, 3/2\,\pi$ (Enright & Hayashi, '96, Castleton & Grimm, '73). The true solution is $p(t) = \sin(t)$.
Furthermore, when $L = 1$, the problem is singular at $t = \pi/2$.

The presence of vanishing delays is delicate for the numerical integration.

Example 2 formulated as implicit problem

We introduce a new variable $q(t) = p'(t)$ and obtain

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} p'(t) \\ q'(t) \end{pmatrix} = \begin{pmatrix} q(t) \\ \Phi\Big(t, p(t), p(\alpha(t, p(t))), q(t), q(\alpha(t, p(t)))\Big) \end{pmatrix}$$

where $\alpha\,(t, p) = t\,p^2$ and

$$\Phi \;\; = \;\; \cos\,(t)\,p\,(\alpha\,(t, p(t))) + L\,p(t)\,q\,(\alpha\,(t, p(t))) + \varphi(t) - q(t)$$

with $\varphi(t)$ a suitable function of t.

Set $y(t) = (p(t), q(t))^{\mathrm{T}}$, $z(t) = (p(\alpha(t, p(t))), q(\alpha(t, p(t))))^{\mathrm{T}}$ and denote the right-hand side of the system by $f\,(t, y(t), z(t))$.

Example 2: numerical integration

For simplicity consider the 1-stage Radau IIA method, that is backward Euler method ($s = 1$, $a_{11} = 1$, $c_1 = 1$). Hence we provide the method by following linear interpolation,

$$u_m(t) = y_m + (t - t_m)\,\frac{(y_{m+1} - y_m)}{h_m}, \quad \text{if } t_m \le t < t_{m+1}.$$

For t_n close to $\pi/2$ we compute

$$J = \begin{pmatrix} 1 & -h \\ hL\pi & h\,(1 - L) \end{pmatrix} + \mathcal{O}(h^2), \qquad J_0 = \begin{pmatrix} 1 & -h \\ hL\pi & h \end{pmatrix} + \mathcal{O}(h^2)$$

The leading terms in the lower-right coefficients are different.

Example 2: using J_0 instead of J

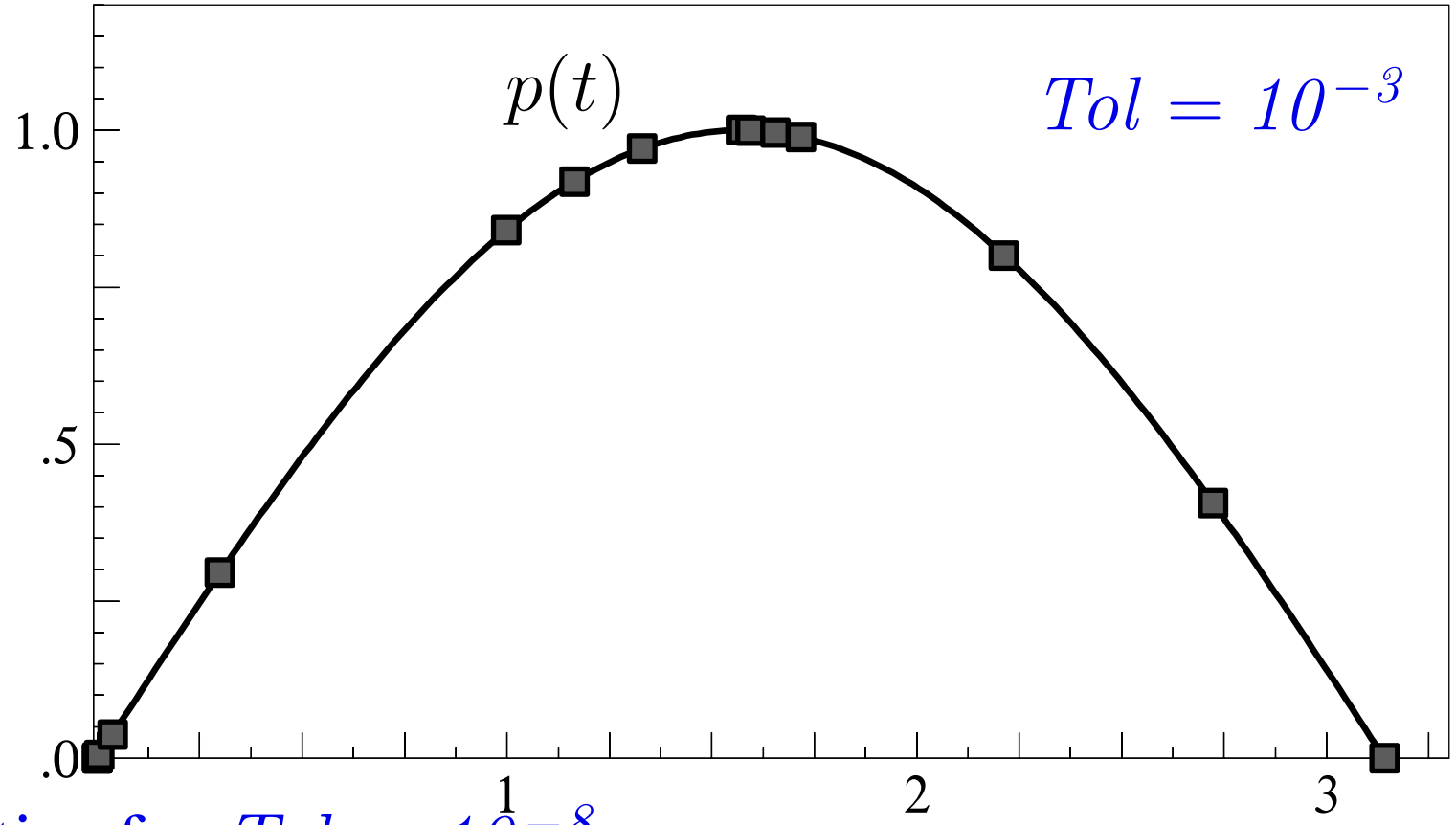
A numerical experiment for Newton iteration.

We choose: $t_n = \pi/2$, $L = 0.95$, $h = 10^{-3}$. Exact solution: $p(t_{n+1}) = 0.9999995274\ldots$, $q(t_{n+1}) = 0.0000274335\ldots$

k	Θ_k	$\mathbf{E}_k$	$p_{n+1}^{[k]}$
2	0.8949919	$0.919739 \cdot 10^{-3}$	1.000000399884
3	0.9511827	$0.872451 \cdot 10^{-3}$	1.000000354905
5	0.9506469	$0.784701 \cdot 10^{-3}$	1.000000271474
9	0.9497251	$0.633801 \cdot 10^{-3}$	1.000000128108
10	0.9495226	$0.600675 \cdot 10^{-3}$	1.000000096655

Notation: Θ_k is the measured contractivity of the error during the iteration, $\mathbf{E}_k$ is the 2-norm of the error and $p_{n+1}^{[k]}$ the k-th iteration for p_{n+1}. Observe: $p_{n+1}^{[k]} > 1 \Longrightarrow$ advanced argument!

Example 2: numerical integration by RADAR5



Statistics for $Tol = 10^{-8}$

L	nr. of steps	iterations with $J \neq J_0$	error at $t = \pi$
0.3	120	2	$0.10 \cdot 10^{-8}$
0.7	144	14	$0.53 \cdot 10^{-9}$
1.0	164	19	$0.43 \cdot 10^{-7}$

Neutral problems

$$\begin{cases} \dot{y}(t) &= f\Big(t, y(t), y(\alpha(t, y(t))), \dot{y}(\beta(t, y(t)))\Big), t \in [t_0, t_f] \\ y(t) &= g(t), \quad t \leq t_0 \end{cases} \qquad \text{(NP)}$$

Assumptions

- $y, f \in \mathbf{R}^d \quad (d \geq 1)$;
- $f : [t_0, t_f] \times \mathbf{R}^d \times \mathbf{R}^d \times \mathbf{R}^d \longrightarrow \mathbf{R}^d$;
- g continuously differentiable vector function;
- $\alpha(t, y(t)) \leq t, \beta(t, y(t)) \leq t.$

Breaking points are not smoothed

The set of breaking points

$J_0 \;=\; \{t_0\}$ contains the initial point t_0

$$J_k \;=\; J_{k-1} \bigcup \{\xi :\ \alpha(\text{or } \beta)(\xi, y(\xi)) = \zeta \quad \text{for some } \zeta \in J_{k-1}\}$$

In the case of non neutral DDEs, jump discontinuities are typically smoothed out along the integration interval. For neutral equations instead jump discontinuities may appear in the same derivative of the solution at all breaking points.

Existence and uniqueness

If the deviating arguments α, β are state dependent, existence and uniqueness of the solution are not always guaranteed on the right-hand side of a breaking point.

Checking termination/bifurcation

Assume that the derivative $\dot{y}(t)$ of the solution has a jump discontinuity at ζ and that the solution uniquely exists up to a breaking point $\xi > \zeta$ satisfying $\beta(\xi, y(\xi)) = \zeta$. Further assume the existence of two smooth functions $x^+(t), x^-(t)$, defined on a neighborhood of ζ, such that

$$
\begin{aligned}
x^+(t) &= y(t) \qquad \text{for} \quad t > \zeta \\
x^-(t) &= y(t) \qquad \text{for} \quad t < \zeta
\end{aligned}
\tag{3}
$$

and smoothly prolonged in a full neighborhood of ζ.

Existence and uniqueness...

For x^+ and x^- we consider the delay differential equations

$$\begin{aligned} \dot{y}(t) &= f(t, y(t), y(\alpha(t, y(t))), \dot{x}^+(\beta(t, y(t)))) \\ \dot{y}(t) &= f(t, y(t), y(\alpha(t, y(t))), \dot{x}^-(\beta(t, y(t)))) \end{aligned} \tag{4}$$

with suitable initial values. Assuming they have unique solutions, which we denote by $y^+(t)$ and $y^-(t)$, on a non-empty interval $(\xi, \xi + \varepsilon)$, we have (for $t > \xi$ close to ξ)

(5) $y^+(t)$ is a solution of (NP) $\iff$ $\beta(t, y^+(t)) > \zeta$

(6) $y^-(t)$ is a solution of (NP) $\iff$ $\beta(t, y^-(t)) < \zeta$.

If both conditions (3) and (4) are satisfied we have that two solutions exist for $t > \xi$; on the other hand if none of them is satisfied, the solution terminates.

Example of termination

We consider the equation (with initial function $g(t) = 1 - t$)

$$\dot{y}(t) \;=\; -\,\dot{y}(\beta(t, y(t)))\;\; \beta(t, y(t)) = y(t) - 2 \qquad \text{for } \; t > 0.$$

It has the solution $y(t) = 1 + t$ on the interval $(0, 1)$, and a breaking point at $\xi = 1$ created by the discontinuity at $\zeta = 0$. In a neighborhood of $\zeta = 0$ we consider the smooth functions

$$x^+(t) = 1 + t, \qquad x^-(t) = 1 - t,$$

and the corresponding differential equations (2) which are obtained by replacing the term $\dot{y}(y(t) - 2)$ by $\dot{x}^+(y(t) - 2)$ and $\dot{x}^-(y(t) - 2)$, respectively.

Example of termination...

The ordinary differential equations (2) have the solutions

$$y^+(t) = 3 - t, \qquad y^-(t) = 1 + t$$

close to $\xi = 1$, respectively. As a consequence we have

$$\begin{aligned} \beta(t, y^+(t)) = \quad & 1 - t < 0 = \zeta \qquad \text{for } t > 1 \\ \beta(t, y^-(t)) = & -1 + t > 0 = \zeta \qquad \text{for } t > 1, \end{aligned}$$

so that neither condition (3) nor condition (4) is satisfied. This proves that the solution terminates at $t = 1$.

Neutral problems as implicit DDEs

Problem (NP) can be written as an index 1 implicit system

$$\left\{ \begin{array}{rcl} M\,\dot{w}(t) & = & g\Big(t, w(t), w(\alpha(t, w(t))), w(\beta(t, w(t)))\Big) \\ \\ w(t) & = & \psi(t), \qquad \text{for } \; t < t_0 \end{array} \right.$$

In fact by introducing a new variable $v(t) = \dot{y}(t)$ in (NP), we get the following equivalent implicit system

$$\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{y} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v(t) \\ v(t) - f(t, y(t), y(\alpha(t, y(t))), v(\beta(t, y(t)))) \end{pmatrix}$$

for $t_0 \leq t \leq t_f$ (where I denotes the identity matrix).

Basic implicit numerical process

$$\begin{pmatrix} Y_i^{(n)} \\ 0 \end{pmatrix} = \begin{pmatrix} y_n + h\ \Sigma_{j=1}^{s} a_{ij} V_j^{(n)} \\ \Sigma_{j=1}^{s} a_{ij} \left(V_j^{(n)} - f\left(t_n + c_j h, Y_j^{(n)}, Z_j^{(n)}, W_j^{(n)} \right) \right) \end{pmatrix}$$

where (being η and λ dense approximations of y and v)

$$Z_j^{(n)} = \begin{cases} g\left(\alpha_j\right) & \text{if } \alpha_j < t_0 \\ \eta\left(\alpha_j\right) & \text{if } t_0 \leq \alpha_j \leq t_{n+1} \end{cases}$$

$$W_j^{(n)} = \begin{cases} g'\left(\beta_j\right) & \text{if } \beta_j < t_0 \\ \lambda\left(\beta_j\right) & \text{if } t_0 \leq \beta_j \leq t_{n+1} \end{cases}$$

with $\alpha_j = \alpha(t_n + c_j h, Y_j^{(n)})$ and $\beta_j = \beta(t_n + c_j h, Y_j^{(n)})$.

Basic implicit numerical process...

Remark

This means that the method converges according to the results stated in Theorem 3. In order to apply it it is crucial that the breaking points are computed exactly or, more realistically, to a precision which does not affect the convergence result.

Linear algebra

Given the special structure the linear algebra associated to the Newton process can be implemented in a more efficient way.

Dense output in $[t_m, t_{m+1}]$

$$\eta_m(t_m + \vartheta h_m) = \ell_0(\vartheta)\, y_m + \sum_{i=1}^{s} \ell_i(\vartheta) Y_i^{(m)}, \quad \vartheta \in [0,1]$$

$$\lambda_m(t_m + \rho h_m) = \ell_0(\rho)\, v_m + \sum_{i=1}^{s} \ell_i(\rho) V_i^{(m)}, \quad \rho \in [0,1]$$

with $\ell_i(\theta)$ i-th Lagrange polynomial on $\{c_0 = 0, c_1, \ldots, c_s\}$.

Choice for $\lambda_m(t)$ at a jump discontinuity t_m.

One of the following choice is recommended.

- Neglecting the interpolation condition in t_m (represented by the the term $\ell_0(\rho)\, v_m$);
- Replacing v_m by v_m^* through a suitable projection (see later).

Checking existence and bifurcation

Let $\xi = t_n$ and $\zeta = t_m$ be a numerical breaking point and its ancestor. Then, the polynomial functions

$$\lambda_m(t) \quad \text{and} \quad \lambda_{m-1}(t)$$

are well-defined in a neighbourhood of $\zeta = t_m$ and satisfy the discrete analogue of x^+ and x^- (see (3)).

With a small step size $\varepsilon > 0$ we then compute one step of some numerical method applied to (NP), where $x^+(s)$ and $x^-(s)$ are replaced by $\lambda_m(s)$ and $\lambda_{m-1}(s)$, respectively.

The numerical results y_n^+ and y_n^- are then used to check the conditions (5) and (6).

Checking existence and bifurcation...

Consequently, if both conditions

$$\beta(t_n + \varepsilon, y_n^+) > t_m \quad \text{and} \quad \beta(t_n + \varepsilon, y_n^-) < t_m \tag{7}$$

are satisfied, the solution bifurcates at t_n; if none of them is satisfied, the solution terminates to exist at t_n.
For the considered case of neutral delay equations, we can consider for example a one-step application of Euler method

$$\begin{aligned} y_n^+ &= y_n + \varepsilon f(t_n, y_n, y_m, \lambda_m(t_m)), \\ y_n^- &= y_n + \varepsilon f(t_n, y_n, y_m, \lambda_{m-1}(t_m)). \end{aligned}$$

If just one of conditions (7) is satisfied the solution continues to exist for $t > t_n$.

Projection

If just one of conditions (7) is satisfied, so that the solution continues to exist, the integration continues according to one of the following choices for v at t_n.

- If the first of (7) is fulfilled we set

$$v_n^* = f(t_n, y_n, y_m, \lambda_m(t_m))$$

- If the second of (7) is fulfilled we set

$$v_n^* = f(t_n, y_n, y_m, \lambda_{m-1}(t_m))$$

Modi£ed dense output

$$\lambda_n(t_n + \rho h_n) = \ell_0(\rho)\, v_n^* + \sum_{i=1}^{s} \ell_i(\rho) V_i^{(n)}, \quad \rho \in [0, 1]$$

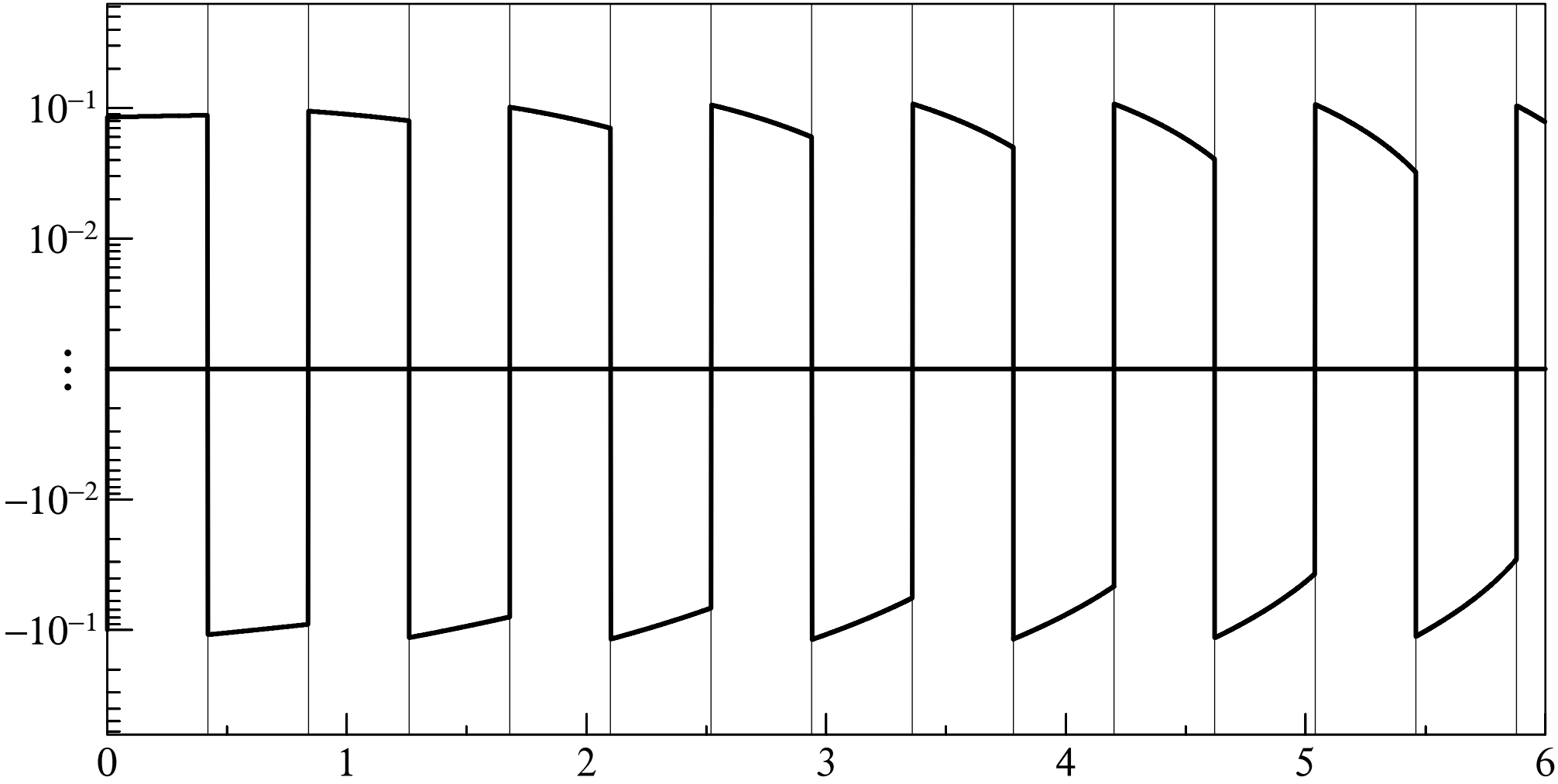
Numerical example 3 (Y. Kuang)

We consider a neutral problem studied by Y. Kuang and formulate it as a differential-algebraic problem (of index 1)

$$\begin{aligned} \dot{y}_1(t) &= y_3(t) \\ \dot{y}_2(t) &= y_2(t)(F(y_1(t)) - \alpha) \\ 0 &= y_1(t)(1 - y_1(t-\tau) - \rho y_3(t-\tau)) - y_2(t)F(y_1(t)) - y_3(t), \end{aligned}$$

where $F(y) = y^2/(y^2+1)$, $\alpha = 0.1$, $\rho = 2.9$, and $\tau = 0.42$.

Numerical example 3: graph of y_3



A quantitative comparison

	RADAR5 – old version				RADAR5 – new version			
Tol	feval	accept	reject	error	feval	accept	reject	error
10^{-3}	1049	98	30	$4.0 \cdot 10^{-3}$	2647	186	138	$4.5 \cdot 10^{-5}$
10^{-6}	7411	676	292	$3.0 \cdot 10^{-5}$	4382	351	117	$2.3 \cdot 10^{-8}$
10^{-9}	16112	1505	513	$1.6 \cdot 10^{-7}$	10160	917	137	$1.4 \cdot 10^{-9}$
10^{-12}	32023	2977	752	$2.5 \cdot 10^{-7}$	28313	2718	163	$7.2 \cdot 10^{-12}$

Numerical example 4

Following Castleton and Grimm (1973) we consider

$$\dot{y}(t) = \cos(t)(1 + y(t\, y^2(t))) + L\, y(t)\, \dot{y}(t y^2(t)) + \varphi(t)$$

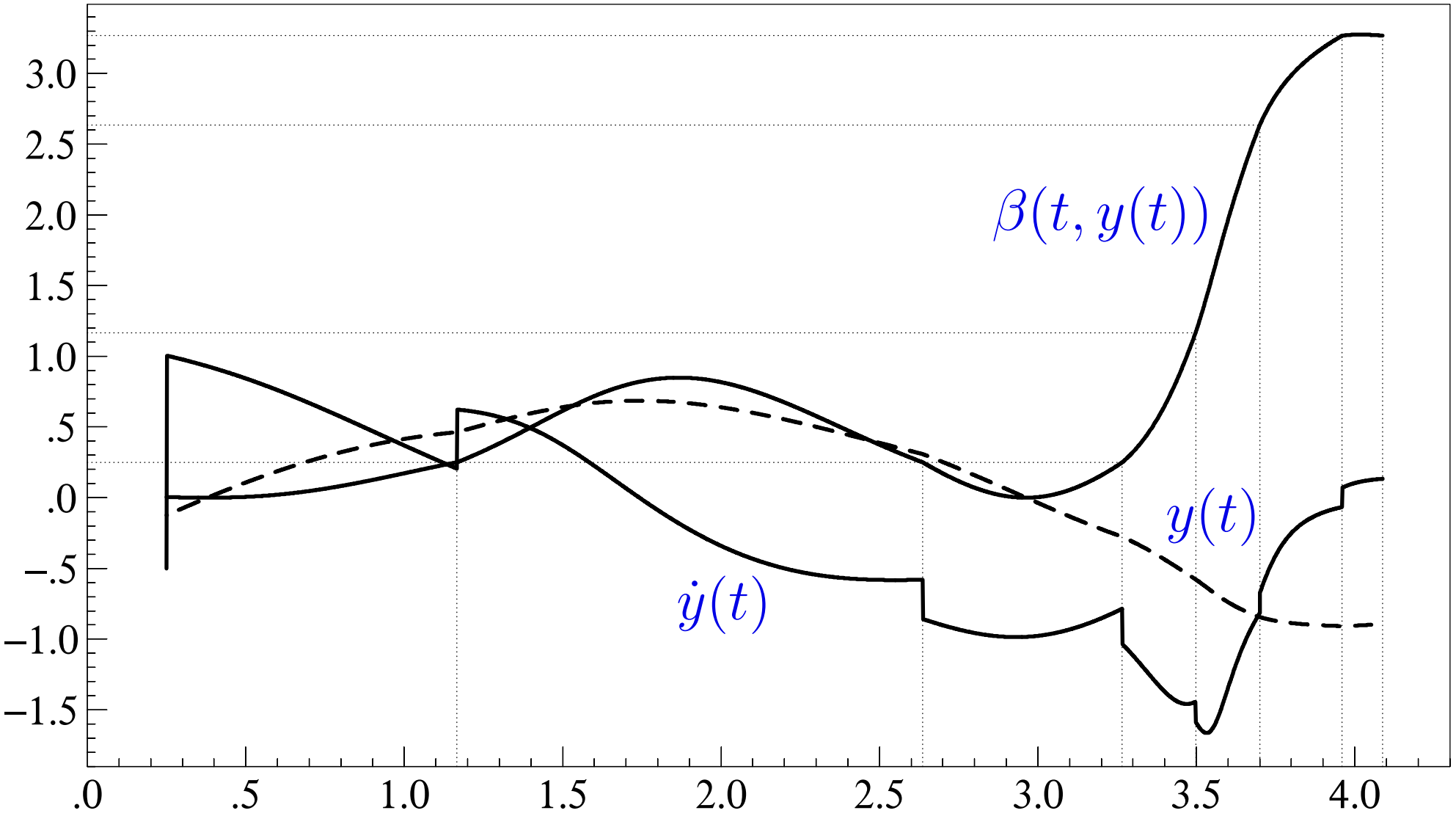
with a given function $\varphi(t)$.

In contrast to previous authors (Paul, Enright & Hayashi, ...) we choose the initial functions

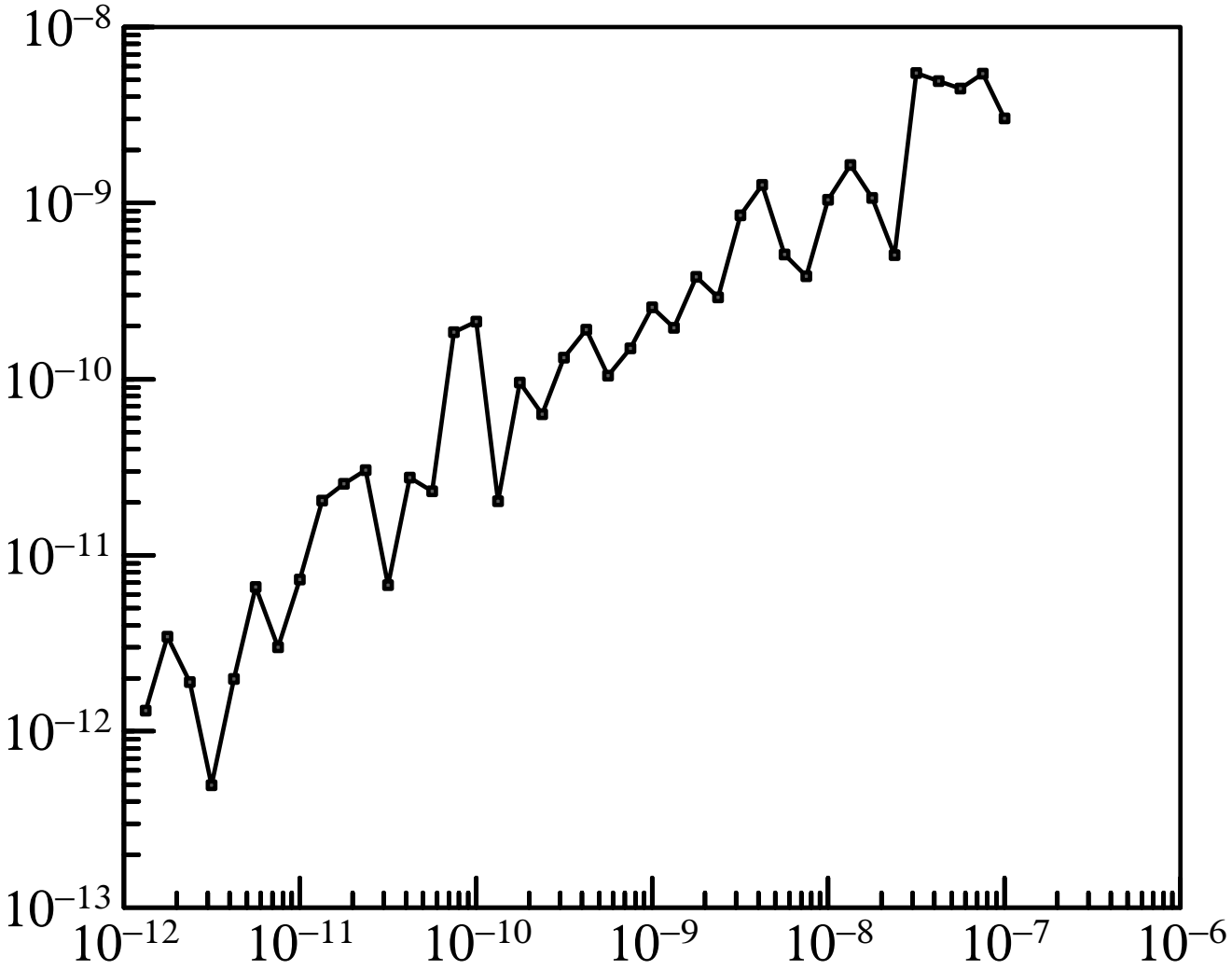
$$y(t) = -t/2, \quad \dot{y}(t) = -1/2 \quad \text{for } t \leq 0.25,$$

and we consider the equation with $L = 0.6$ for $t \geq 0.25$. This modification produces breaking points, and provokes a termination of the solution at $T = 4.09$.

Numerical example 4: graph of the solution



Example 4: precision diagram for breaking points



Mean accuracy versus Tolerance of computed breaking points

$$\mu = \frac{1}{6} \sum_{i=1}^{6} \frac{|\xi_i - \xi_i^*|}{|\xi_i^*|}$$

A mathematical model for antibody production

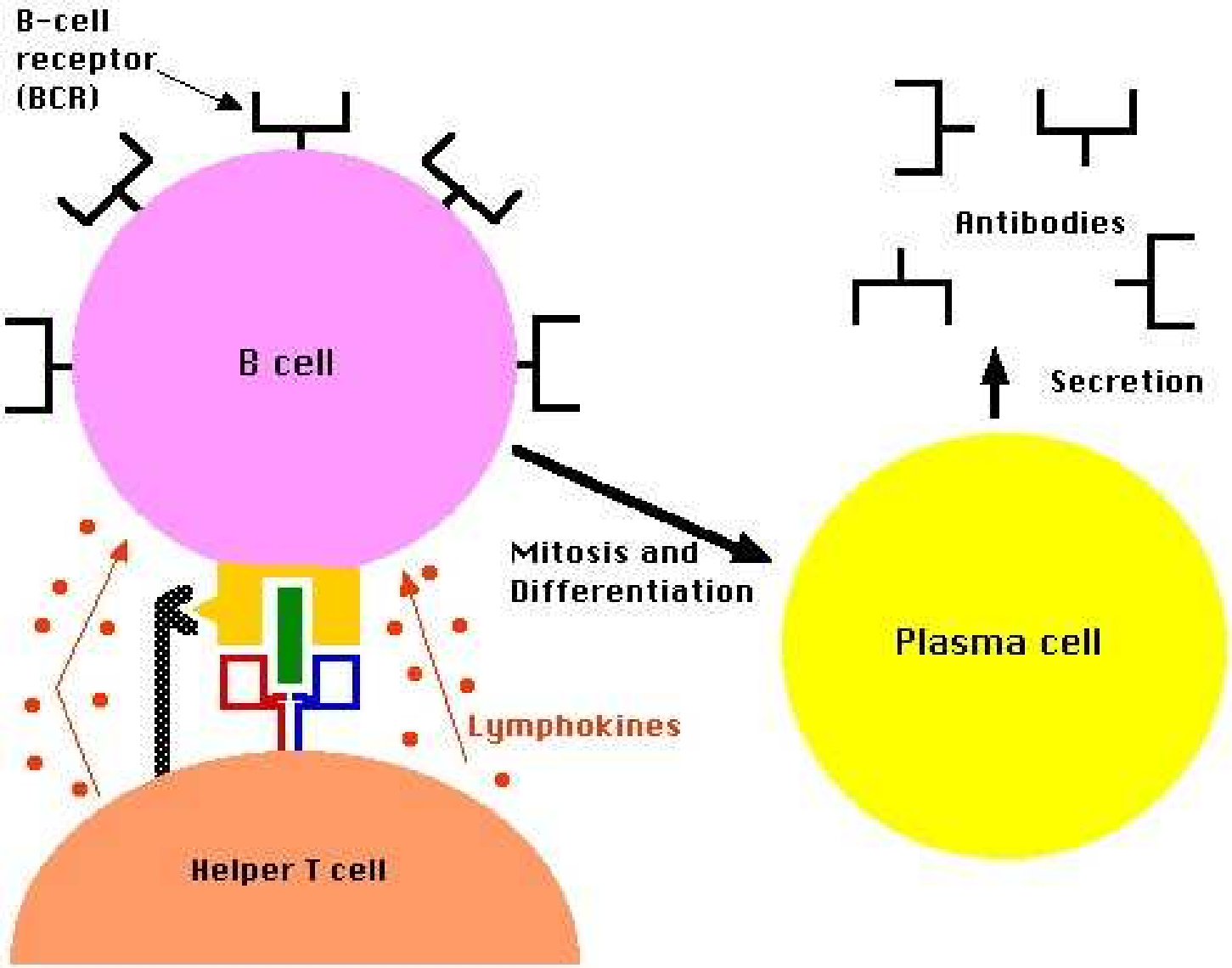
The goal is that of presenting a schematic model (due to Waltman) of antigen stimulated antibody production.

The model makes use of threshold criteria with the aim of determining the proliferation phases of lymphocytes and antibody proliferation.

This translates into suitable memory effects in the governing eauations.

A lower level analysis is very important in order to better estimate the parameters used by the model.

Basic diagram



Main phases

Notation

- $\mathrm{Ag}(t)$ concentration of unbound antigen molecules;
- $\mathrm{Rf}(t)$ concentration of unbound receptor molecules;
- $\mathrm{Rb}(t)$ concentration of bound antigen molecules;
- $\mathrm{Ab}(t)$ concentration of unbound antibodies.

Main mechanisms of interaction.

- Initial phase: combination of antigen and receptor molecules on the surface of B-lymphocytes;
- Intermediate phase: generation of new receptor molecules by a memory mechanism;
- Decisive phase: production of antibodies.

Initial dynamics

First phase ($t \in [0, t_0]$)

$$\begin{aligned} \dot{\mathrm{Ag}}(t) &= -r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t), \\ \dot{\mathrm{Rf}}(t) &= -r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t), \\ \dot{\mathrm{Rb}}(t) &= \quad r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t), \end{aligned}$$

with r suitable association constant.

Intermediate phase ($t \in [t_0, t_1]$)

$$\dot{\mathrm{Rf}}(t) = -r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t) + a\,r\,\mathrm{Ag}\left(\alpha_1(t)\right)\,\mathrm{Rf}\left(\alpha_1(t)\right)$$

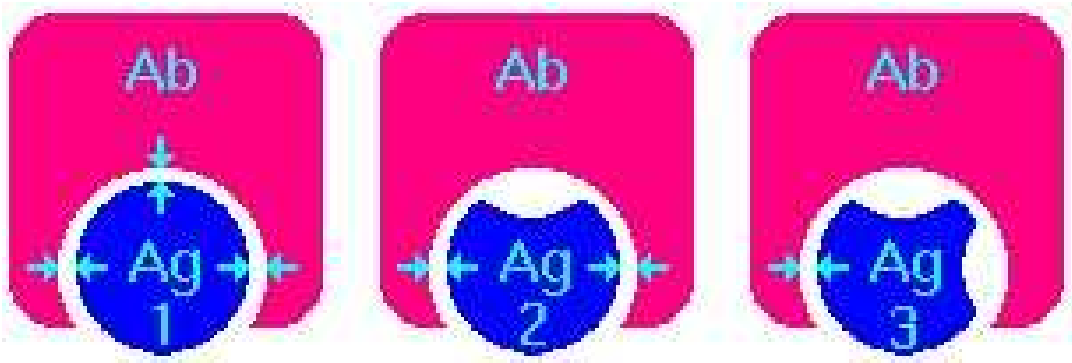
where a is an amplification factor and $\alpha_1(t) \leq t$ models a memory effect.

Antibody production

Final phase ($t \in [t_1, T]$)

$$\begin{aligned}
\dot{\mathrm{Ag}}(t) &= -r\,\mathrm{Ag}\,\mathrm{Rf}(t) - s\,\mathrm{Ag}(t)\,\mathrm{Ab}(t) \\
\dot{\mathrm{Ab}}(t) &= -s\,\mathrm{Ag}(t)\,\mathrm{Ab}(t) - \gamma\,\mathrm{Ab}(t) + \\
&\quad + b\,r_1\,\mathrm{Ag}\left(\alpha_2(t)\right)\,\mathrm{Rf}\left(\alpha_2(t)\right)
\end{aligned}$$

where s is a combination factor, b is an amplification factor related to antibody secretion capacity of plasma cells, γ is a catabolic factor and $\alpha_2(t) \leq t$ is a second memory effect.



Memory effects

Memory effects are defined by the integral equations measuring accumulation phenomena:

$$\int_{\alpha_1(t)}^{t} f_1\left(\mathrm{Ag}(s), \mathrm{Rf}(s), \mathrm{Rb}(s)\right)\, ds = m_1, \qquad t \geq t_0$$

$$\int_{\alpha_2(t)}^{t} f_2\left(\mathrm{Rf}(s), \mathrm{Rb}(s)\right)\, ds = m_2, \qquad t \geq t_1$$

where m_1 and m_2 are suitable biological thresholds.
Example: $f_1(x, y, w) = xy + w$ and $f_2(y, w) = y + w$ yield a possible choice of the biologic measure functions.

The whole differential-functional system

Antigens, receptors and antibodies ($H_s(t)$ denotes a step function)

$$\begin{cases} \dot{\mathrm{Ag}}(t) &= -r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t) - s\,\mathrm{Ag}(t)\,\mathrm{Ab}(t) \\ \dot{\mathrm{Rf}}(t) &= -r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t) + a\,r\,\mathrm{Ag}(\alpha_1(t))\mathrm{Rf}(\alpha_1(t))H_{t_0}(t) \\ \dot{\mathrm{Rb}}(t) &= \quad r\,\mathrm{Ag}(t)\,\mathrm{Rf}(t) \\ \dot{\mathrm{Ab}}(t) &= \left(-s\,\mathrm{Ag}(t)\,\mathrm{Ab}(t) - \gamma\,\mathrm{Ab}(t) + b\,r\,\mathrm{Ag}(\alpha_2(t))\,\mathrm{Rf}(\alpha_2(t)\right)H_{t_1}(t) \end{cases}$$

Memory effects

$$\begin{cases} \dot{\alpha}_1(t) &= H_{t_0}(t)\,\dfrac{f_1\left(\mathrm{Ag}(t), \mathrm{Rf}(t), \mathrm{Rb}(t)\right)}{f_1\left(\mathrm{Ag}\left(\alpha_1(t)\right), \mathrm{Rf}\left(\alpha_1(t)\right), \mathrm{Rb}\left(\alpha_1(t)\right)\right)} \\ \dot{\alpha}_2(t) &= H_{t_1}(t)\,\dfrac{f_2\left(\mathrm{Rf}(t), \mathrm{Rb}(t)\right)}{f_2\left(\mathrm{Rf}\left(\alpha_2(t)\right), \mathrm{Rb}\left(\alpha_2(t)\right)\right)} \end{cases}$$

Mathematical considerations

(i) the deviating arguments are state-dependent and hence are not known in advance;

(ii) the delays $t - \alpha_1(t, y(t))$ and $t - \alpha_2(t, y(t))$ become very small as time grows; this does not allow to integrate the problem step-by-step as an ordinary differential equation;

(iii) solution components have very different magnitudes and have very steep variations in correspondence of triggers;

(iv) the presence of discontinuities in the right-hand side determines a certain number of low order breaking points, which have to be computed to prevent a loss of accuracy;

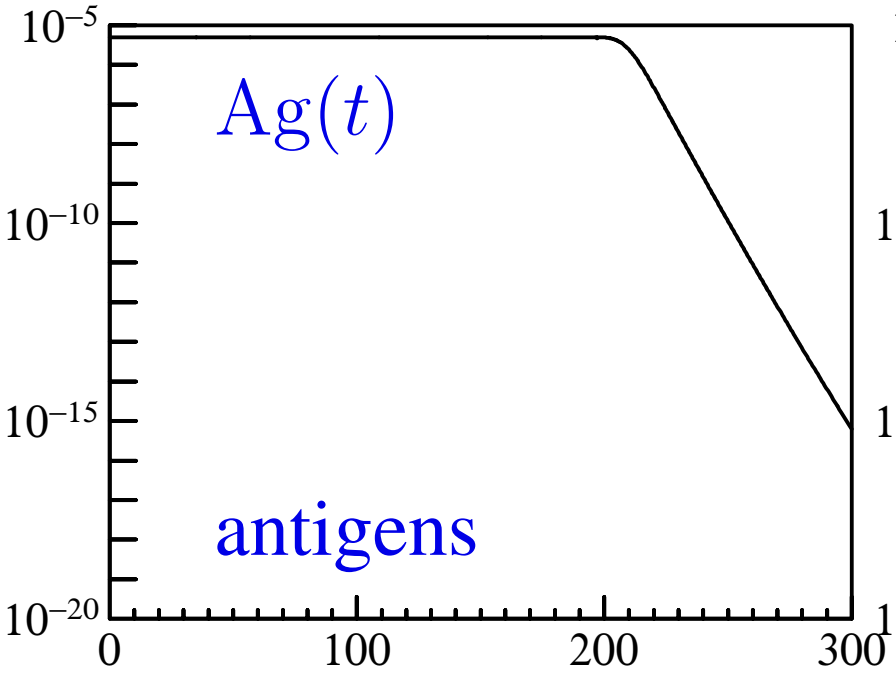
(v) the system is stiff

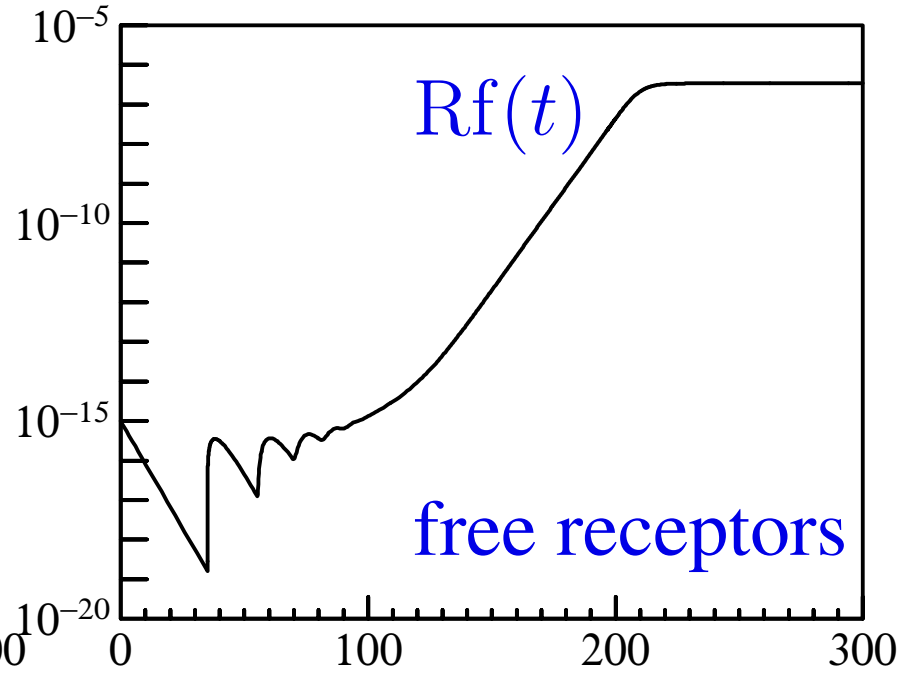
Numerical simulation

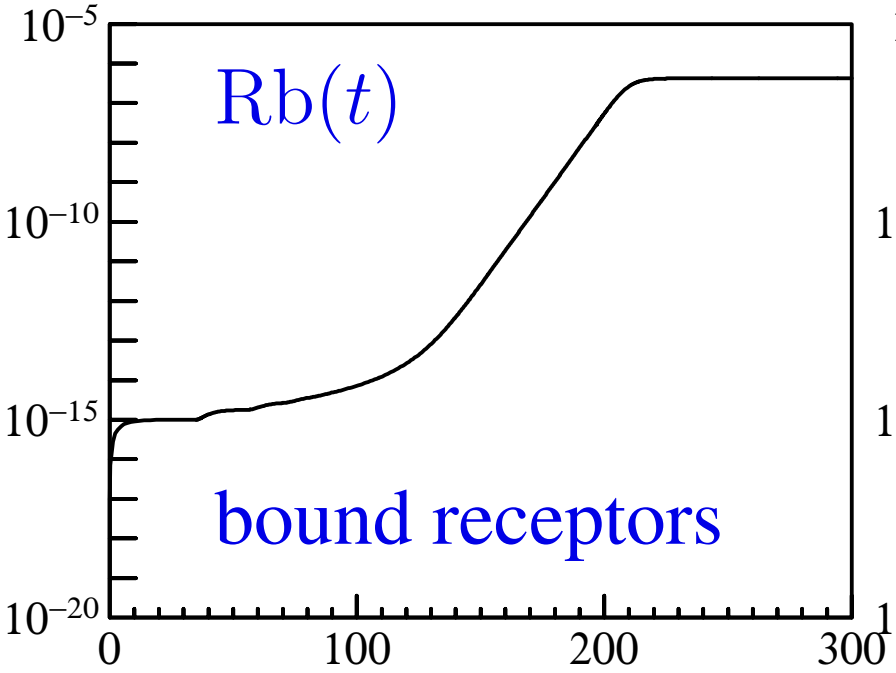
We consider the following example problem.

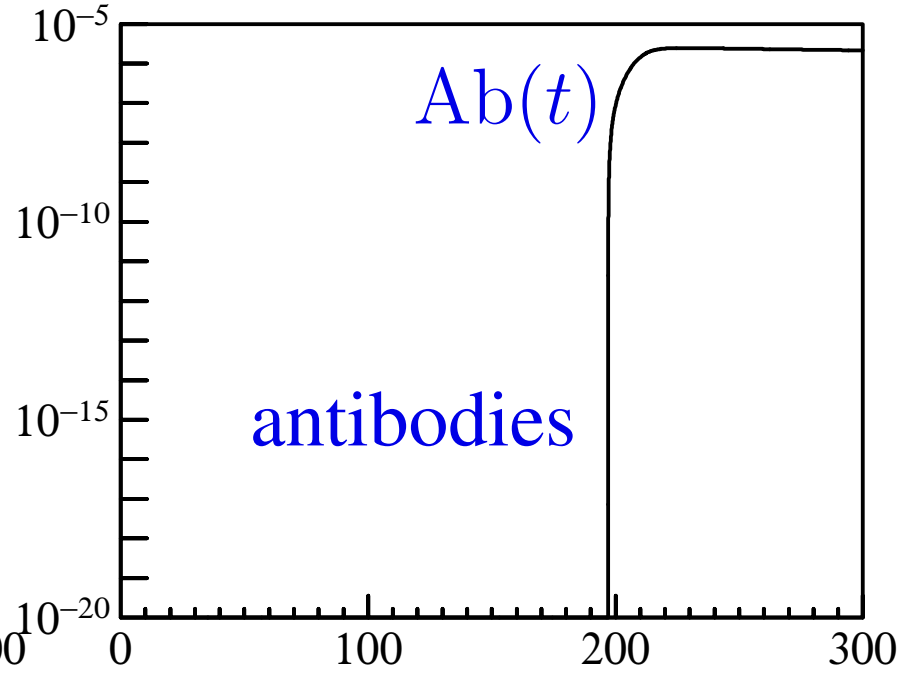
- $f_1(x, y, w) = xy + w$ and $f_2(y, w) = y + w$ are the functions modeling the accumulation effects which determine the delays;
- $a = 1.8$ and $b = 20$ are the amplification factors, $\gamma = 0.002$ is the catabolic factor, $r = 5 \cdot 10^4$ and $s = 10^5$ are the combination factors;
- the initial values and initial functions are given by $\mathrm{Ag}(t) = 5 \cdot 10^{-6}$, $\mathrm{Rf}(t) = 10^{-15}$, and $\mathrm{Rb}(t) = \mathrm{Ab}(t) = \alpha_1(t) = \alpha_2(t) = 0$ for $t \leq 0$.

Solution components (semi-logarithmic scale)

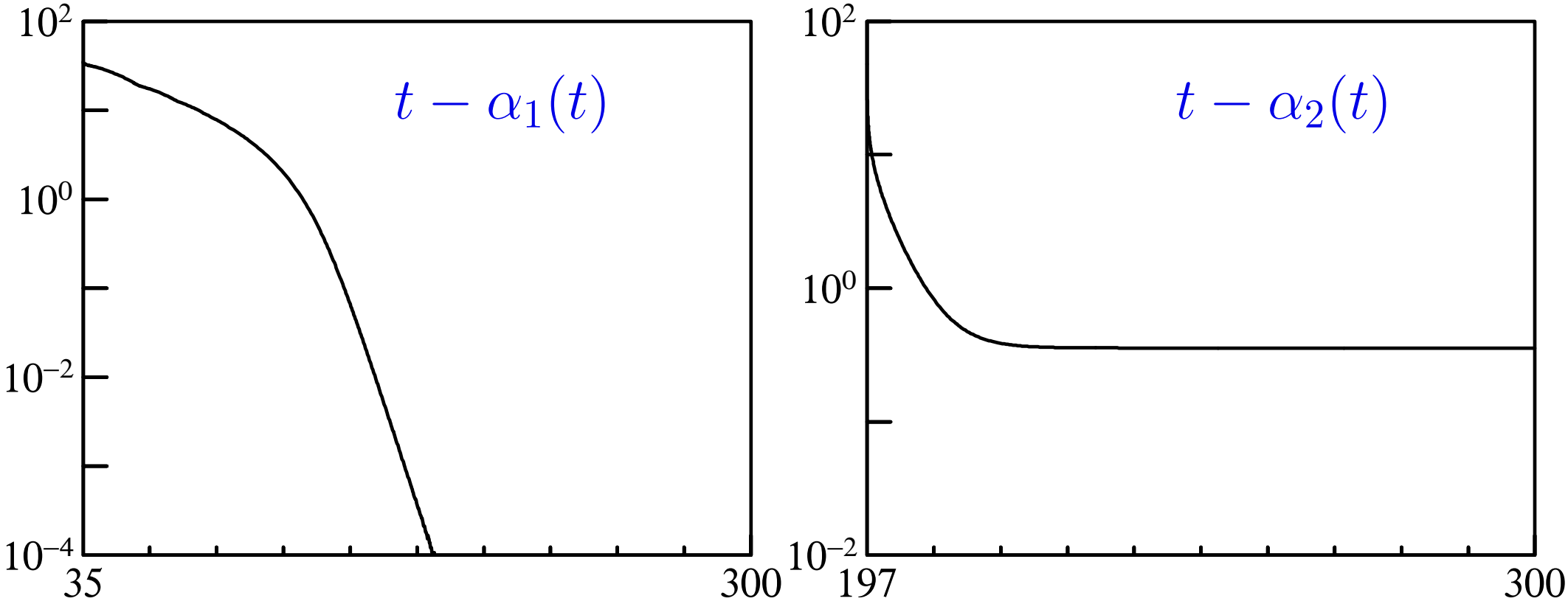








Computed delays

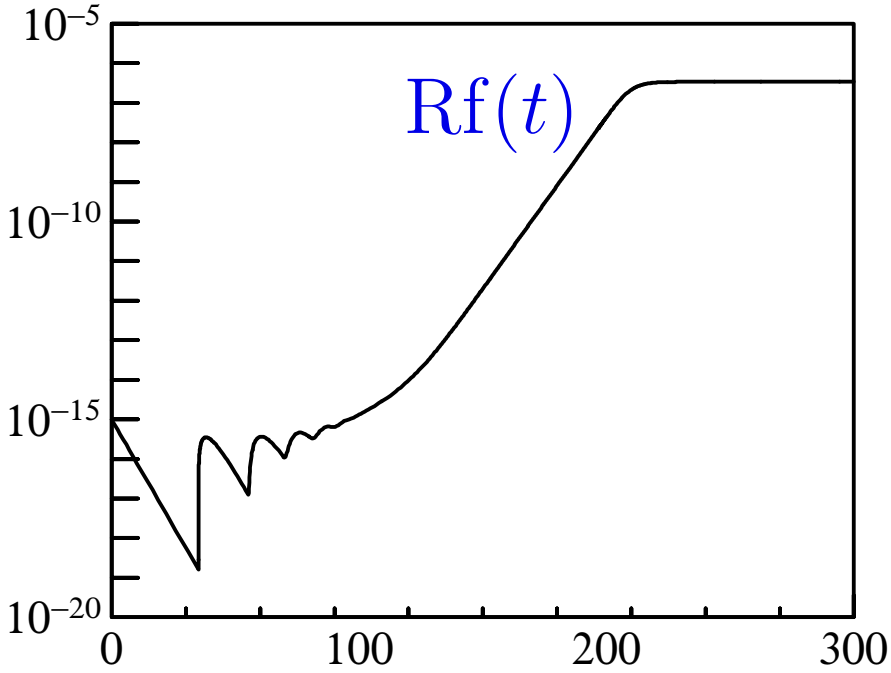


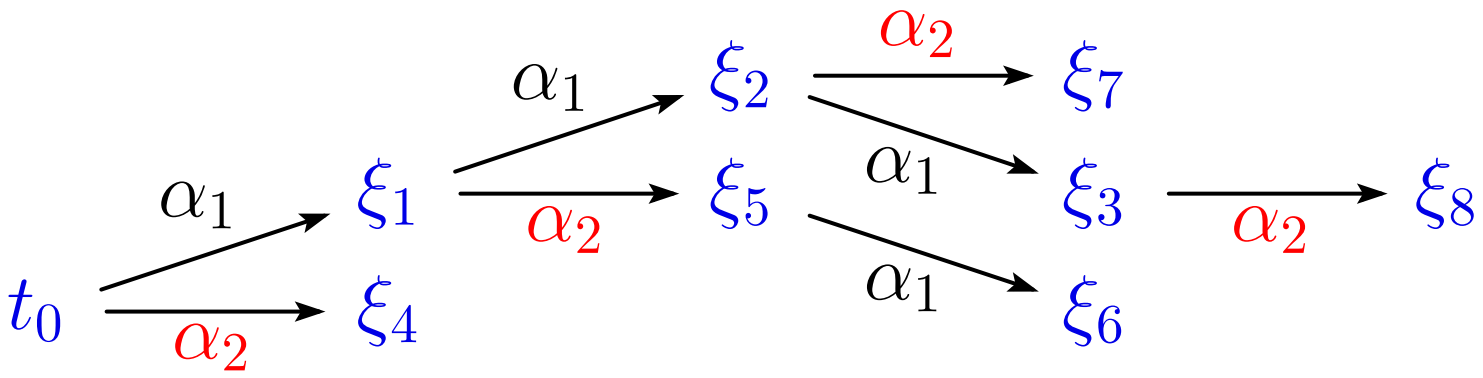
The first delay becomes extremely small.

Computed breaking points

An oscillating behavior of free receptors is known in the medical literature.

$$\xi_1 = 55.21325176$$
$$\xi_2 = 69.26718167$$
$$\xi_3 = 79.63960593$$
$$\dots$$
$$\xi_8 = 197.0000173$$





Efficiency of the code

Comparison between version 1 (which does not compute automatically breaking points) and version 2.

	RADAR5 – ver. 2		RADAR5 – ver. 1	
Tol	feval	err	feval	err
10^{-3}	2227	0.218	2800	0.778
10^{-6}	3409	6.85e-4	4244	1.05e-2
10^{-9}	7939	3.32e-6	8537	2.48e-4
10^{-12}	22694	3.66e-8	—-	——

Notation: feval gives the number of function evaluations and err a measure of the error.

Software

A new release of the code RADAR5 is presently being distributed at the web-sites

http://univaq.it/∼guglielm
http://www.unige.ch/∼hairer/software.html

(where the first released version is actually available).
It contains drivers for nine examples including problems simulating enzyme kinetics, acute hepatitis B virus infection, and the examples of the present talk.

Some bibliographic references

The general theory concerning numerical integration of DDEs is discussed in

Bellen & Zennaro, Numerical methods for delay differential equations, Oxford University Press, 2003

The basic numerical process and implementation issues concerning RADAR5 are discussed in

G. & Hairer, Implementing Radau IIa methods for stiff delay differential equations, Computing, vol. 67, pp. 1–12, 2001

The subject relevant to the numerical approximation of breaking points is discussed in

G. & Hairer, Computing breaking points in implicit delay differential equations, preprint, 2005, available at http://www.unige.ch/ hairer/preprints.html.

Some bibliographic references...

The general theory on the numerical integration of stiff ordinary differential equations is discussed in

Hairer & Wanner, Solving ordinary differential equations II. Stiff and differential algebraic problems, Springer Verlag, 1996.

The optimization technique to preserve the tensor structure in the Newton process applied to the RK equations is also considered in

G., On the Newton iteration in the application of collocation methods to implicit delay equations, *Proceedings of the Conference Numdiff-03, Halle, September 2003*, Applied Numerical Mathematics, vol. 185, pp. 261–277, 2006.

Some bibliographic references...

The integration of the model of antigen-antibody dynamics is described in

Bellen, G. & Maset, Numerical methods for delay models in biomathematics, in Integration of Complex Systems in Biomedicine, editor A. Quarteroni, Springer, in press.

The theory of threshold models in population dynamics is described in

Hoppensteadt & Waltman, Did something change? Thresholds in population models, Trends in nonlinear analysis, Springer, pp. 341–374 (2003).