ISBN 0-315-96013-2

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

To my grandmothers

Abstract

The efficiency with which the numerical solution of ordinary differential equations (ode) can be generated depends to a large extent on the effectiveness of the stepsize adjustment strategy that is used. In this thesis the relative performance of a family of stepsize adjustment strategies is examined. Included in this family is a new strategy. A distinctive feature of its formulation is the incorporation of a mechanism to correct for any persistent deviation of a prescribed solution quality measure from its desired value. The evaluation of the various strategies is undertaken through an extensive set of numerical experiments which use both single processor and multiple processor ode solution procedures.

ACKNOWLEDGEMENTS

The author wishes to express her deepest and sincere thanks to Professor L. G. Birta for his direction, valuable guidance and encouragement throughout the course of this research work and his great effort for helping her to prepare this thesis.

Grateful acknowledgements are extended to Dr. O. Abou-Rabia for his helpful suggestions and stimulating discussions.

The author is also grateful to her husband, for his continual help and encouragement in this work.

Contents

List of Tables

Table

List of Figures

Figures

Chapter 1
Introduction

1.1 Background

The numerical problem we consider is that of generating the solution of a given set of m ordinary differential equations (ode's) of the form:

$$\frac{dy}{dt} := \dot{y}(t) = f(t,y(t)) , \qquad y(t_0) = y_0 , \qquad (*)$$

over a prescribed interval $[t_0,t_f]$, where y and f are both m-vectors and t_0 and y_0 are the given initial values. This problem, which is generally known as the initial value problem, has particular relevance in continuous system simulation studies.

The solution we seek is, in fact, a series of points (t_0,y_0), (t_1,y_1), ..., (t_M,y_M) with $0=t_0<t_1<\cdots<t_M=t_f$. From a procedural point of view, this solution is obtained by solving a series of subproblems. Each such subproblem can be characterized as follows: Assume that the point $P_n=(t_n,y_n)$, $0\le n<M$, is available and that $\hat{y}_n(t)$ is the solution trajectory of (*) which passes through P_n; find the values h_{n+1} (the next stepsize) and y_{n+1} (the next solution value) such that y_{n+1} adequately approximates, $\hat{y}_n(t_{n+1})$ with $t_{n+1}=t_n+h_{n+1}$. There are four separate steps

associated with this process; namely,

(1) establishment of an appropriate value for h_{n+1} ,

(2) generation of the value y_{n+1} ,

(3) assessment of the "quality" of y_{n+1} with respect to $\hat{y}_n(t_{n+1})$,

(4) recovery, when necessary, from a y_{n+1} value that has an unacceptably low quality.

Inasmuch as the spacing between the solution values associated with this specification is not fixed, it describes a variable stepsize process. An alternate solution process is one which maintains a constant predefined value for the solution spacing. Such a fixed stepsize process involves only step (2) and consequently there is no attempt to either monitor, or adapt to, local error along the solution of ode's. It is generally recognized that the effectiveness of the classical formulas for the solution of ode's (e.g. Runge-Kutta, predictor-corrector) that are applied in step (2) is considerably enhanced when they are embedded in a variable stepsize process.

The principal cost of solving a set of differential equation lies in the repeated evaluation of the derivative function associated with the system. The number of such evaluations over the solution interval depends directly on the spacing between solution points. Consequently the efficiency with which the numerical solution of ordinary differential equations can be generated depends to a large extent on the effectiveness of the stepsize adjustment strategy that is used.

Our interest in this study is with three of the four steps in the solution specification given above. Step (2) is simply the matter of applying some underlying solution formula and it is outside the scope of our present interest. Steps (1), (3) and (4) provide a framework for using classical solution formulas in a (computationally) efficient and robust manner.

1.2 Contribution of the Thesis

A fundamental requirement in the continuous system simulation process is the need to solve (possibly repeatedly) the set of ordinary differential equations (ode's) that characterize the system under investigation. Single processor methods for carrying out this numerical task are well known. Recent research efforts have addressed the problem of using multiple processors to carry out this task [1,2,3,4,5]. These studies with parallel methods have focused primarily on fixed stepsize procedures. A few variable stepsize multiprocessor procedures have thus far been reported in the literature [6]. Insights provided in this thesis study will contribute to progress in this area.

The realization of efficient variable stepsize ode solution procedures in both single and multiprocessor environments depends on the quality of the underlying stepsize adjustment strategy that is used. The goal of such a strategy is to avoid unreasonably small stepsizes since this would result in wasted computational effort and, as well, avoid overly optimistic (i.e., large) stepsizes that result in excessive error and hence rejected steps (which again produce wasted computational effort). Problems whose solution trajectories change rapidly and frequently from regions of relative smoothness to regions of rapid change, generally place considerable demands on a stepsize adjustment strategy.

In this thesis, a new strategy for carrying out this critical task is proposed. A distinctive feature of its formulation is the explicit incorporation of a mechanism to correct for any persistent deviation of a prescribed solution quality measure from its desired value. The performance of this new method is compared with that of three other stepsize adjustment strategies that have appeared in the literature. These include the widely used "locally optimal" strategy, the strategy recently proposed by Watts [7] and a strategy proposed in a multiprocessor context by Abou-Rabia [14].

An extensive set of numerical experiments with all four stepsize adjustment strategies was carried out using both single processor and multiprocessor methods. The work reported in [9] is based on our single processor investigations. These test results provide significant insights into the relative behavior of the stepsize adjustment strategies being studied. In particular, they demonstrate some desirable features of the newly proposed strategy.

Chapter 2
Stepsize Adjustment Strategies (Single Processor Case)

2.1 Introduction

The general objective for a numerical solution procedure for solving the initial value problem is to vary the stepsize in a way that efficiently generates a solution having adequate accuracy. The notion of solution quality that was referred to in section 1.1 is measured in terms of an estimate of the solution error at each step. The local error which is made by the solution formula in advancing the solution from y_n to y_{n+1} is $(\hat{y}_n(t_{n+1}) - y_{n+1})$ where y_{n+1} and $\hat{y}_n(t_{n+1})$ are as defined earlier in section 1.1.

A fundamental assumption is that an estimate of the local error is available, either directly or indirectly from the calculations that take place while determining the value of y_{n+1} (we will discuss this in the following section). We denote the error estimate obtained for y_{n+1} by ε_{n+1}. Associated with this error, is a local truncation error tolerance which we denote by $\hat{\varepsilon}_{n+1}$. Typically, in program code, $\hat{\varepsilon}_{n+1}$ has a predefined form which incorporates one or more user assignable parameters. A typical example is

$$\hat{\varepsilon}_{n+1} = \lambda(e + |y_{n+1}|) \tag{2.1.1}$$

where the scalar parameter λ is user assignable and non-negative. Here we use e to denote the m-vector whose entries are all unity and $|y_{n+1}|$ to denote the vector obtained from y_{n+1} by replacing each component by its absolute value.

The quality of the generated solution estimate y_{n+1}, is then established by comparing the values of ε_{n+1} and $\hat{\varepsilon}_{n+1}$. A typical procedure is to define the scalar quality measure $R_{n+1} = \|r_{n+1}\|$ where $r_{n+1} = \varepsilon_{n+1} / \hat{\varepsilon}_{n+1}$. (For m-vectors a and b, a/b is the vector whose j^{th} component is a_j/b_j. Through-out the sequel the specific vector norm we use is $\|a\| = \max_j\{|a_j|\})$.

Clearly, if $R_{n+1} < 1$, then the solution estimate y_{n+1} would, for all practical purposes, be regarded as having acceptable quality; i.e., a quality which is consistent with the specified user tolerance. Conversely, if R_{n+1} is greater than 1, then it is normal practice to conclude that the quality of y_{n+1} is unacceptable and hence an appropriate recovery procedure is activated; i.e. step (4) of the general specification. To strive for a value of 1 for R_{n+1} has obvious intuitive appeal. However there is no intrinsic reason to conclude that if R_{n+1} is "somewhat bigger" than 1, then the quality of the total solution trajectory will be seriously compromised by accepting y_{n+1}. (Two of the stepsize adjustment procedures considered in the sequel rely on this point of view since they do allow excursions beyond 1.)

The above discussion has, in fact, outlined the general aspects of step (3) of the general solution specification given in section 1.1. The remaining discussion in this section examines a number of different alternatives relating to steps 1 and 4 of this specification.

2.2 The Standard Stepsize Adjustment Strategy

A widely used approach for stepsize adjustment (often referred as to the "locally optimal" stepsize adjustment procedure) is based on the assumption that, for a p^{th} order solution method, the local error approximation, ε_n, at (t_n, y_n) can be written as:

$$\varepsilon_n = h_n^{p+1} \phi_n$$

where $\phi_n = \phi(t_n, y_n)$ and $\phi(t, y)$ is a principal error function and h_n is the step size used to arrive at t_n. With this assumption, $R_n = \|r_n\|$ becomes

$$R_n = h_n^{p+1} g_n \qquad (2.2.1)$$

where $g_n = \|\phi_n / \hat{\varepsilon}_n\|$.

Assume now that the value of R_n satisfies the assessment criterion; i.e., the solution value (t_n, y_n) has acceptable quality. The immediate problem then is to determine h_{n+1}; namely, the stepsize to be used in moving forward to produce the next solution value y_{n+1}. Observe that once this step is taken, the relationship that will hold is:

$$R_{n+1} = h_{n+1}^{p+1} g_{n+1}$$

If the nominal desired value for R_{n+1} is 1, we have

$$h_{n+1} = \left(\frac{1}{g_{n+1}}\right)^{\frac{1}{p+1}} \qquad (2.2.2)$$

Unfortunately this specification for h_{n+1} has no direct value because the right hand side of (2.2.2) is not known. One standard approach at this point is to assume that $g_{n+1} \approx g_n$; then by incorporating (2.2.1), we obtain

$$h_{n+1} = \left(\frac{1}{R_n}\right)^{\frac{1}{p+1}} h_n \qquad (2.2.3)$$

which provides a usable specification. This is usually referred to as the "locally optimal stepsize".

In practice, ode codes which rely on this approach generally introduce an element of conservatism whereby a somewhat smaller step than that resulting from (2.2.3) is used. A typical implementation has the form

$$h_{n+1} = \left(\mu \frac{1}{R_n}\right)^{\frac{1}{p+1}} h_n \qquad (2.2.4)$$

where a common value for the "safety factor" μ is 0.5 (see, for example, [10], page 116). Note that (2.2.4) applies only for the case where $n>0$ and also that the first step, h_1, is established by other means.

In order to assess the "quality" of the solution value y_{n+1} we introduce the threshold value R_T (a value great than or equal to 1) to denote the boundary between acceptable and unacceptable values of the solution quality measure. If $R_{n+1} \le R_T$ then the step is successful and the procedure is repeated from (t_{n+1}, y_{n+1}). However if $R_{n+1} > R_T$ then (t_{n+1}, y_{n+1}) must be rejected and a new solution value must be computed with a reduced stepsize $\bar{h}_{n+1}$; i.e. at the point $\bar{t}_{n+1} = t_n + \bar{h}_{n+1}$. The value used for R_T with the locally optimal stepsize adjustment procedure is 1.

2.2.1 Handling a Failure

To deal with the failed case (i.e. $R_{n+1} > R_T$), observe that the specification of (2.2.2) will continue to hold at $(\bar{t}_{n+1}, \bar{y}_{n+1})$ i.e.,

$$\bar{h}_{n+1} = (1/\bar{g}_{n+1})^{\frac{1}{p+1}}$$

Again there is the need to replace $\bar{g}_{n+1}$ with some appropriate approximation. In this case the obvious choice is

$$\bar{g}_{n+1} \approx g_{n+1} = R_{n+1} / h_{n+1}^{p+1}$$

which gives

$$\bar{h}_{n+1} = \left(\frac{1}{R_{n+1}}\right)^{\frac{1}{p+1}} h_{n+1}$$

We again incorporate a safety factor and obtain

$$\bar{h}_{n+1} = \left(\mu \frac{1}{R_{n+1}}\right)^{\frac{1}{p+1}} h_{n+1} \qquad (2.2.5)$$

Notice that because $R_{n+1} > 1$, $\bar{h}_{n+1} < h_{n+1}$ as required. Assuming that $\bar{h}_{n+1}$ yields an acceptable step; i.e., the associated R-value, $\bar{R}_{n+1} < 1$, then $\bar{h}_{n+1}$ and $\bar{R}_{n+1}$ are used as the values h_{n+1} and R_{n+1} respectively in the specifications for the subsequent step. This approach is, in fact, a widely used and effective strategy for handling step

(4) of the general process outlined earlier.

The specification given by (2.2.4) together with this mechanism (2.2.5) for dealing with a failure, are referred to as strategy S1 throughout our discussions.

2.3 An Alternate Stepsize Adjustment Strategy

This alternate approach is a variation on the stepsize adjustment strategy outlined above. We re-write (2.2.3) as

$$h_{n+1} = \sigma_n h_n \tag{2.3.1}$$

where $\sigma_n = (1/R_n)^{\frac{1}{p+1}}$. Notice that there is no safety factor in this formula and that it applies only for $n > 0$ (the first step, h_1, is established by other means). The "quality" of the solution y_{n+1} which results from the use of (2.3.1) is evaluated using $\sigma_{n+1} = (1/R_{n+1})^{1/(p+1)}$. The boundary between an acceptable and unacceptable value for y_{n+1} is defined by a value of 0.5 for σ_{n+1}; i.e. if $\sigma_{n+1} \geq 0.5$, then the step is successful and the procedure is repeated from (t_{n+1}, y_{n+1}); otherwise the solution (t_{n+1}, y_{n+1}) must be rejected and a new solution value must be computed with a reduced stepsize $\bar{h}_{n+1}$ as described below.

2.3.1 Handling a Failure

The approach here is almost the same as that used with the standard strategy, except the reject criterion is $\sigma_{n+1} < 0.5$. The specification that corresponds to (2.2.5) is:

$$\bar{h}_{n+1} = \sigma_{n+1} h_{n+1} \qquad (2.3.2)$$

where $\sigma_{n+1} = (1/R_{n+1})^{\frac{1}{p+1}}$. Also notice that because $\sigma_{n+1}<0.5$, $\bar{h}_{n+1}<h_{n+1}$ as required. Assuming that $\bar{h}_{n+1}$ yields an acceptable step; i.e., the associated value $\bar{\sigma}_{n+1}\geq 0.5$, then $\bar{h}_{n+1}$ and $\bar{R}_{n+1}$ are used as the values h_{n+1} and R_{n+1} respectively in the specifications for the subsequent step.

The specification given by (2.3.1) together with (2.3.2) for dealing with a failure, are referred to as strategy S2 in the sequel. The approach was first used by Abou-Rabia [14].

2.4 The Stepsize Adjustment Strategy Proposed by Watts

For the case where the solution quality of (t_n, y_n) is acceptable, another approach for transforming (2.2.2) into a usable form was recently suggested by Watts [7]. His suggestion is to assume an equality of ratios among three values of the function g; i.e.,

$$\frac{g_{n+1}}{g_n} \approx \frac{g_n}{g_{n-1}}$$

which allows (2.2.2) to be written as

$$h_{n+1} = \left(\frac{g_{n-1}}{g_n^2}\right)^{\frac{1}{p+1}}$$

Then, by incorporating (2.2.1), we obtain:

$$h_{n+1} = (\frac{R_{n-1}}{R_n^2})^{\frac{1}{p+1}} (\frac{h_n}{h_{n-1}}) h_n \qquad (2.4.1)$$

The implementation of (2.4.1) recommended by Watts incorporates a safety factor μ which is embedded in the following way:

$$h_{n+1} = (\mu \frac{R_{n-1}}{R_n^2})^{\frac{1}{p+1}} (\frac{h_n}{h_{n-1}}) h_n \qquad (2.4.2)$$

The value suggested by Watts for μ is $-0.1*\log\lambda$ where λ is the error tolerance parameter of (2.1.2) and *log* is to the base 10. This assignment is clearly not meaningful when $\lambda \le 10^{-10}$. Accordingly, in our implementation of the procedure we use $\mu = \min(0.9, -0.1*\log\lambda)$.

In the implementation, the first step, h_1, is established by other means, and the second step, h_2, is obtaind by using the standard approach S1; i.e., formula (2.2.4). The first application of formula (2.4.1) start from $n=2$, with the condition that $R_1 \le R_T$ and $R_2 \le R_T$. We again use R_T to denote the threshold value for R; i.e., if $R \le R_T$ then the step is accepted; otherwise it is rejected and a new solution value must be computed with a reduced stepsize $\bar{h}_{n+1}$, which will be discussed in following. The value used for R_T is 1.

2.4.1 Handling a Failure

Watts makes no proposal for handling a failed step. To deal with this situation, we use the same procedure as with the standard stepsize adjustment strategy, i.e., repeat the step using a reduced stepsize $\bar{h}_{n+1}$ as prescribed by formula (2.2.5). Assuming that $\bar{h}_{n+1}$ yields an acceptable step; i.e., the associated R-value, $\bar{R}_{n+1} < 1$, then $\bar{h}_{n+1}$ and $\bar{R}_{n+1}$ are used as the values h_{n+1} and R_{n+1} respectively in the specifications for the subsequent step. Our experiments with this approach have shown it to give good performance. The specification given by (2.4.1) together with this mechanism for dealing with a failure, are referred to in the sequel as strategy S3.

2.5 A New Adaptive Stepsize Adjustment Strategy

The point of view which we adopt here is that the formulae of the underlying ode solver together with the ode's that are being solved, form a dynamical system. The relevant "output" of this system, however, is not the solution of the ode's but rather the solution quality measure, R. This system, furthermore, has an input through which a control action is applied. This control input is the integration stepsize h. The associated control problem is to choose h in a way which maintains the system output at a specified desired value R^*. Access to the input and output of the system is available only at the set of discrete points in time $t_i, i = 1,2,...,M$, which correspond to the mesh points at which solution values are generated.

The particular features of our system are as follows:

a) The value chosen for the input h at time t_n (i.e., h_{n+1}) affects the value of the output R at t_{n+1}; i.e., the value R_{n+1}.

b) It is a simple matter to recover from a particularly poor choice for the control variable. In other words, the time evolution of the system is not monotonic inasmuch as "replays" are possible.

c) The system input-output transformation has the property that $\delta R/\delta h > 0$. This assumed relationship implies that, at any point in time, the output can be increased/decrease by increasing/decreasing the value of the control input. This feature provides the basis for the control strategy.

An intuitively appealing control strategy is to select $h_{n+1} = \alpha_n h_n$ with some suitable choice for the weighting factor α_n. From an adaptive control point of view, the value chosen for α_n should reflect the goal of achieving a system output which is equal to R^* taking into account the underlying assumption that $\delta R/\delta h > 0$. For the case where $R^* = 1$, the choice

$$\alpha_n = \left(\frac{\mu}{R_n}\right)^{\frac{1}{p+1}}$$

(which corresponds to (2.2.4)) has this property inasmuch as the resulting value of h_{n+1} will increase (decrease) with respect to h_n depending on whether h_n generated a system output R_n that was less (greater) than the desired value of 1.

This particular choice, however, is restrictive inasmush as α_n reflects the error on only the most recent step. An alternate approach is to formulate a strategy that strives to detect, and correct for, a persistent tendency of the system output to deviate from its desired value. In a classical control context, this requirement is typically achieved by embedding an "integral" component within the control action. Our intent is to achieve an equivalent effect through an appropriate specification for α_n. Such a specification should, in particular, provide a reflection of past errors in the system output.

The specification which we propose has the following form:

$$h_{n+1} = \alpha_n h_n$$
$$\alpha_n = \beta_n \left(\frac{\mu}{R_n}\right)^{\frac{1}{p+1}}$$
$$\beta_n = 0.5(1+\theta_n)$$
$$\theta_n = \varphi_n \theta_{n-1}$$
$$\varphi_n = 0.6 + 0.4 * \min\{\Psi(0.5), \Psi(R_n)\}$$

where the function $\Psi(x) = x^{-1/3}$.

For the first application of this formula (i.e., $n=1$), the resultant value of h_2 is identical to the value provided by (2.2.4). (The first step, h_1, is established by other means). Subsequent steps, however provide adaptive adjustments to A_n (and hence to β_n) which reflect the nature of the system output error on previous steps. In particular, if a sequence of R_i values persists in being less than $R^* = 1$ then the corresponding sequence of β_n values will increase in a cumulative fashion. Likewise, if a sequence of R_i values persists in being greater than R^* then the β_n sequence will rapidly decrease.

It is important to observe that, in order to be effective, this strategy must permit excursions of the system output beyond the nominal desired value of 1. It is only in this way that the necessary contraction of β_n can take place. Thus an intrinsic feature of the approach is that R_T, the threshold value of the system output which defines the boundary of unacceptable solution quality, must be taken to be greater than 1. A satisfactory choice for R_T was found to be 2. Thus, with the proposed approach the rejection/recovery procedure is activated only when $R_n > 2$. This will be discussed in a later section.

2.5.1 Handling a Failure

The case of a failed step (i.e., $R_n > 2$) requires the recalculation of the solution value using a reduced value for h_{n+1}. This value is established as follows:

$$
\begin{aligned}
&\bar{h}_{n+1} = \bar{\alpha}_n h_{n+1} \\
&\bar{\alpha}_n = \delta_n \left(\frac{\mu}{R_{n+1}}\right)^{\frac{1}{p+1}} \\
&\delta_n = \min(1, 0.5(1+\bar{\theta}_n)) \\
&\bar{\theta}_n = \bar{\varphi}_n \theta_n \\
&\bar{\varphi}_n = 0.6 + 0.4 * \Omega(R_{n+1})
\end{aligned}
$$

where the function $\Omega(x) = x^{-3}$.

Assuming that h_{n+1} yields an acceptable solution value, then $\bar{h}_{n+1}$ and $\bar{A}_n$ are used as the values for h_{n+1} and A_n respectively in the specifications for the subsequent step. The adaptive stepsize adjustment strategy outlined in this section is referred to as strategy S4.

Chapter 3
Stepsize Adjustment Strategies (Multiple Processor Case)

3.1 Overview of the BPC and PPC Parallel Solution Methods

Approaches for achieving parallelism in the solution of (*) fall into two broad categories; namely, the equation segmentation approach and the parallel algorithm approach, as described in [11]. Most of the research has focused on the parallel algorithm approach. In the parallel algorithm approach, parts of the integration procedure itself are distributed over the set of available processors. The two most common approaches in this category are the Block Predictor-Corrector (BPC) and Parallel Predictor-Corrector (PPC). [1,2,8,12,13].

With both approaches, we divide the integration interval $[t_0, t_f]$ into a sequence of adjacent, non-overlapping, sub-intervals or blocks, with block length sh where h is the integration stepsize (in this section we take h to have a fixed value over the whole solution interval) and s is the number of solution points in the block. For the BPC method $s=N$ and for the PPC method $s=N/2$, where N is the number of available processors.

Obviously, an even number of processors is required in the PPC approach.

We assume a particular reference state in the computation as shown in Figure 3.1 and we focus on two adjacent blocks called the current block (block n) and the forward block (block n+1). The blocks to the left of the current block are referred to as "previous" blocks. Suppose the right-hand end of the current block occurs at $t=t_{n,s}=t_0+nsh$. A set of equally spaced points in the forward block occurs at $t=t_{n+1,i}=t_{n,s}+ih,\ i=1,2,...s$. Correspondingly, the points $t=t_{n,i}=t_{n-1,s}+ih,\quad i=1,2,...s$, (or equivalently, $t=t_{n,s}-jh,\quad j=0,1,2,...(s-1)$) represent a set of s equally spaced points in the current block.

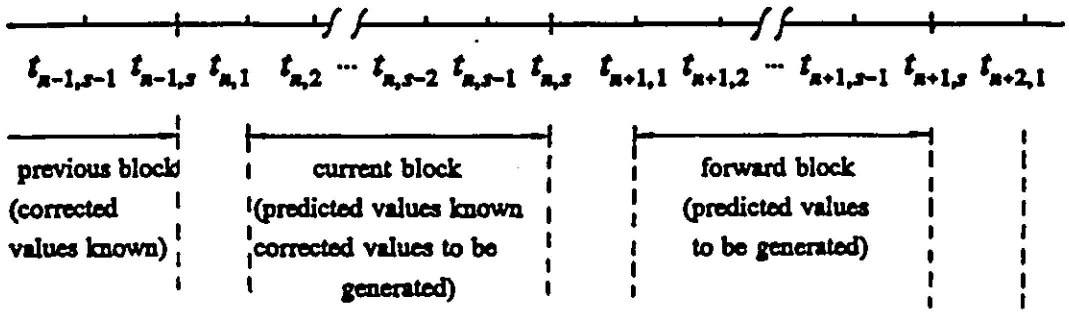


Figure 3.1

The following assumptions are associated with the reference state:

(1) a set of predicted (tentative) solution values is available at the s points within the current block; i.e. the solution pairs $(y^p_{n,i}, f^p_{n,i}), i=1,2,...s$, where $y^p_{n,i}$ corresponds to the predicted solution value at $t_{n,i}$ and $f^p_{n,i}=f(t_{n,i}, y^p_{n,i})$.

(2) corrected (final) solution values are available at the points in the previous blocks; i.e., solution pairs $(y_{n-k,i}, f_{n-k,i})$, $k=1,2,...(n-1)$,

$i=1,2,...s$, where $y_{n-k,i}$ corresponds to the i[th] solution value within block n-k; i.e., at $t_{n-k,i}$ and $f_{n-k,i}=f(t_{n-k,i},y_{n-k,i})$.

The two computational tasks that must be carried out with respect to the reference state are:

(a) the set of predicted solution pairs $(y^p_{n,i},f^p_{n,i})$ in the current block n need to be transformed to corrected solution pairs; i.e., $(y_{n,i},f_{n,i})$, $i=1,2,...s$.

(b) a set of predicted solution pairs need to be generated at the points in the forward block n+1; i.e., $(y^p_{n+1,i},f^p_{n+1,i})$, $i=1,2,...s$.

Upon completion of these two steps, the assumed reference state has been advanced by a distance sh along the t axis and an iterative process is thus established.

Both the BPC and PPC methods share a common mechanism for carrying out steps (a) and (b). This is based on the use of the linear multistep formula given by:

$$\sum_{j=0}^{M}\alpha_j y_j + \sum_{j=0}^{M^*}\beta_j f_j = 0 \qquad (3.1.1)$$

Where each of the y_j corresponds to some $y_{i,j}$ in the notation of the preceding discussion (similarly for f_j). The α_j, β_j and M, M^* are parameters that are determined by the requirements arising from the desired order of the resulting formula. (A relevant discussion is provided in Appendix 2). The BPC and PPC approaches are, however, different in how the available processors are used to transform the reference state into the sequel state.

With the BPC approach the following four steps are carried out sequentially:

1. The N processors are used in a "corrector mode" to evaluate the final (corrected) solution values at the $s\,(=N)$ points in the current block. Each

processor is assigned one point. The computation carried out by each processor is based on a specific version of the general formula (3.1.1), having a common prescribed order and tailored to the assigned point. All processors complete the computational task in a synchronized way because the formulas have identical structure and differ only in coefficient values.

2. The processors then evaluate the deriative function, f, at their respective points, using the values obtained in step 1. Again the computation is identical over all processors (only the data values are different),and consequently the processors complete their work in synchronism.

3. The N processors are then used in a "predictor mode", where each is assigned the calculation of a tentative (predicted) solution value at one of the s points in the forward block. Once again we use a specific version of formula (3.1.1) together with data generated in step 1 and 2. The calculation remains in synchronism because of the identically structured computation.

4. Each processor then evaluates the derivative function, f, at its assigned point in the forward block, using the values obtained in step 3. Synchronism is maintained because of the identically structured calculation.

The completion of these four steps of the BPC cycle achieves a transformation of the original reference state into a sequel state which has advanced $sh = Nh$ units along the time axis.

The principal difference between the BPC and PPC methods is in the sequencing of the steps outlined above. More specifically, the PPC method carries out the steps 1 and 3 concurrently, and then carries steps 2 and 4 concurrently. This concurrency feature is achieved by separating the available processors into two distinct sets each with $s = N/2$ processors. One set of the processors consistently carries out in the "corrector mode" calculations, while the other set carries out "predictor mode" calculation. The PPC method involves the following steps:

1. The s processors operating in the "corrector mode" compute final corrected solution values at the s points in the current block. Each processor is assigned one point in the block. The computation of each processor is again based on a specific version of the general formula (3.1.1), having a common prescribed order and tailored to the assigned point. Simultaneously, the s processors operating in the "predictor mode" compute the predicted (tentative) solution values at the s points in the forward block. Again this is achieved by using appropriately specified instances of (3.1.1).

2. Each of the s processors then evaluates the derivative function f, at its assigned point, making use of the solution value produced in step 1.

For the same reasons as given earlier for the BPC method, these steps of the PPC cycle are such that the processors remain in synchronism.

3.2 Stepsize Adjustment for the BPC Approach

In this section we examine further the BPC procedure from the point of view of formulating a variable stepsize approach. We begin by considering the local truncation error. A particular approximation for the local trancation error of the i^{th} point within a reference block n; i.e., $\varepsilon_{n,i}$, can be defined as the difference between the corrected solution value $y_{n,i}$ and the predicted solution value $y^{p}_{n,i}$, at time $t_{n,i}$, $i=1,2,\dots s$, as noted in section 3.1.

In conventional integration procedures, changes in the integration stepsize are based on a solution quality criterion (as outlined in section 2.1). More specifically, we have $R_{n,i} = |r_{n,i}|$ where $r_{n,i} = \varepsilon_{n,i} / \hat{\varepsilon}_{n,i}$ and $\hat{\varepsilon}_{n,i}$ is a local truncation error tolerance derived from a user-specified error tolerance parameter, λ; i.e., $\hat{\varepsilon}_{n,i} = \lambda(e + |y_{n,i}|)$. (see (2.1.1)).

A variable stepsize control strategy for the BPC method can be formulated by extending the ideas outlined in chapter 2. We define a reference state in terms of the three adjacent blocks, designated as B_{n-1}, B_n and B_{n+1}, as shown in Figure 3.2 below:

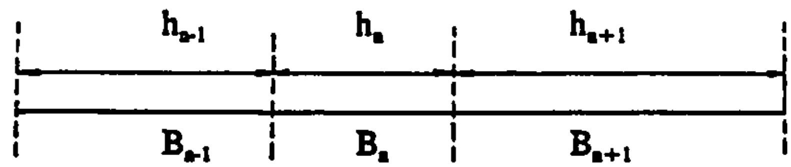


Figure 3.2

The stepsize separating the s points within block B_{n-1} is denoted by h_{n-1}, while the stepsizes within blocks B_n and B_{n+1} are denoted as h_n and h_{n+1} respectively.

We consider a state in the computation where acceptable corrected values $y_{n-1,i}$, $i = 1,2,\ldots s$, have been generated in block B_{n-1} with the stepsize h_{n-1} (i.e., the solution is known up to the right-hand endpoint of block B_{n-1}) and a set of predicted values $y^p_{n,i}$, $i = 1,2,\ldots s$, has been generated in block B_n (with the stepsize h_n). The computation proceeds according to the following two steps:

Step 1. Compute the corrected values $y_{n,i}$, $i = 1,2,\ldots s$, simultaneously at the s points within B_n by specific versions of the formula (3.1.1) as outlined in section 3.1. (Some examples of such formulas are given in

Appendix 1.) Using the the predicted and corrected values within block B_n, we can obtain the local truncation error estimates $\varepsilon_{n,i}$ and the local error tolerances, $\hat{\varepsilon}_{n,i}$ at the s points within B_n. Then compute the quality criterion, R_n, for block B_n as $R_n = \max_{1 \le i \le s} R_{n,i}$.

Step 2. Two separate cases have to be considered.

(a). Successful Case

For the case where $R_n \le R_T$, (where R_T is a threshold value. associated with the different strategies), the corrected values $y_{n,i}$, $i = 1, 2, \dots s$, within B_n are considered acceptable. A specific version of a general formula of the form $h_{n+1} = \xi h_n$ is used to compute the stepsize h_{n+1} that will be used as the spacing within block B_{n+1}. As described in section 2.2 through 2.5, the value of ξ for strategies S1, S2, and S4, depends mainly on R_n. For strategy S3 it depends both on R_n and on R_{n-1}. Then, with the stepsize h_{n+1} a set of predicted values $y^p_{n+1,i}$, $i = 1,2,\dots s$, can be generated simultaneously within block B_{n+1} by the formulae outlined in (3.1.1) (see Appendix 1). Thus the computational state has been transformed into a sequel state which has advanced $sh_n = Nh_n$ units along the time axis; i.e., the solution is now known up to the right-hand endpoint of block B_n. The iteration procedure has thus been established.

(b). Unsuccessful Case

For the case where $R_n > R_T$, the corrected values $y_{n,i}$, $i = 1,2,\dots s$, in B_n are considered unacceptable and must be rejected. Because the corrected values in block B_n depend on the predicted values in block B_n, a new set of predicted values needs to be produced with a reduced spacing $\bar{h}_n$. Consequently a specific version of a general formula $\bar{h}_n = \bar{\xi} h_n$ is used to compute a stepsize $\bar{h}_n$ (where $\bar{\xi} < 1$ is dependent on the strategy being used but mainly depends on R_n). This $\bar{h}_n$ value replaces the current value of h_n as an updated spacing within block B_n, and a new set of predicted values $y^p_{n,i}$, $i = 1,2,...s$, is produced using an appropriate version of (3.1.1). The procedure then returns to step 1.

Note that the application of the general formula $h_{n+1} = \xi h_n$ can start from n=1 because the spacing within the first block, h_1, is predefined. An alternate procedure, however, needs to be carried out to produce the s solution values in the first block B_1.

3.3 Stepsize Adjustment for the PPC Approach

In terms of varying stepsize the PPC method presents a somewhat more complex problem than is the case with the BPC method. Both use the same definition for the local trancation error as well as the solution quality criterion. For the PPC approach, we need to consider a reference state with five adjacent blocks, which we designate as B_{n-2}, B_{n-1}, B_n, B_{n+1}, and B_{n+2}. The associated solution spacing within these blocks is $h_{n-2}, h_{n-1}, h_n, h_{n+1}, h_{n+2}$ (see Figure 3.3).

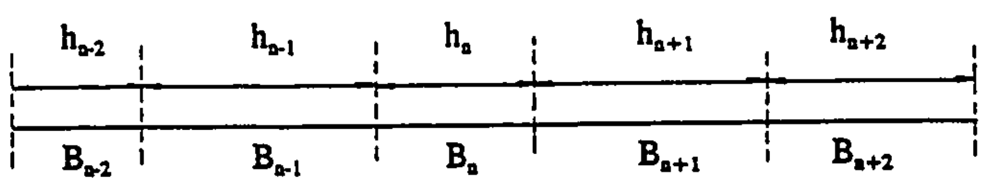


Figure 3.3

We Assume that acceptable solution values in block B_{n-2} and B_{n-1} are available, with spacing h_{n-2} and h_{n-1} respectively (i.e. the solution is known up to the right-hand endpoint of block B_{n-1}). We also assume that a set of predicted values $y^p_{n,i}$, $i=1,2,...s$, is available within block B_n with known spacing of h_n. The solution spacing in block B_{n+1} (denoted by h_{n+1}) is also assumed to be known. The computation proceeds according to the following two steps:

Step 1. Compute the corrected values $y_{n,i}$, $i=1,2,...s$, at the s points within B_n by a specific version of formula (3.1.1) using $s=N/2$ processors and at the same time, generate the predicted values $y^p_{n+1,i}$, $i=1,2,...s$, at the s points within B_{n+1}, using the other s processors. With the predicted and corrected values in B_n, we obtain the local trunction error estimates $\varepsilon_{n,i}$, $i=1,2,...s$. at the s points within B_n. Then compute the quality criterion, R_n, for block B_n as $R_n = \max_{1\le i\le s} R_{n,i}$ where the definition of $R_{n,i}$ is the same as given in section 3.2.

Step 2. Two separate cases have to be considered.

(a). Successful Case

For the case where $R_n \le R_T$, (where R_T is a threshold value associated with the different strategies), the corrected values $y_{n,i}$, $i=1,2,...s$, within B_n are considered acceptable. A specific version of a general formula of the form $h_{n+2}=\xi h_n$ is used to compute a stepsize h_{n+2} that will be used as the spacing within block B_{n+2}. (Notice the difference here with the BPC method). The computational state is now transformed into a sequel state which has advanced $sh_n = \frac{N}{2}h_n$ units along the time axis; i.e., the solution is now known up to the right-hand endpoint of block B_n. The next phase of the procedure is to simultaneously calculate the corrected values in B_{n+1} and the predicted values in B_{n+2}.

(b). Unsuccessful Case

For the case where $R_n > R_T$, the corrected values $y_{n,i}$, $i = 1,2,...s$, in B_n are considered unacceptable and must be rejected. Furthermore, the s predicted values in block B_{n+1} nust also be rejected. Because the corrected values in block B_n depend on the predicted values in block B_n, a new set of predicted values needs to be produced with a reduced spacing $\bar{h}_n$. Consequently a specific version of a general formula $\bar{h}_n = \bar{\xi} h_n$ is used to compute a stepsize $\bar{h}_n$ (where $\bar{\xi} < 1$ is dependent on the strategy being used but mainly depends on R_n). This $\bar{h}_n$ value replaces the current value of h_n as an updated spacing within block B_n, and a new set of predicted values $y^p_{n,i}$, $i = 1,2,...s$, is produced by an appropriate version of (3.1.1) using $s = N/2$ processors. Note furthermore that during this time interval the other s processors are not doing useful work. The procedure then returns to step 1. Note however that although the stepsize in B_{n+1} remains unchanged, the specific time points in B_{n+1} are shifted because the right-hand endpoint of B_n has shifted to the left.

Also note that the application of the general formula $h_{n+2} = \xi h_n$ starts from n=2; i.e., the spacing within the first and second blocks, h_1, and h_2, are predefined. Some alternate precedures need to be carried out to produce the s solution values in the first block B_1 and the s predicted values in the second block B_2.

Chapter 4
Experimental Evaluation

4.1 Scope of the Experiments

The testing activity was concerned with evaluating the performance of a collection of variable stepsize single processor and multiprocessor methods which utilize the stepsize adjustment procedures S1, S2, S3 and S4. The activity was carried out using a Fortran-based program package which included modules for all relevent procedures.

The evaluation was based on a set of five test problems which have been widely used in literature. These problems, which are described in the Appendix 1, have known analytic solutions. Results were obtained over a variety of circumstances; i.e.,

a) Three separate ode solution formulas were used in the single processor case; namely the Runge-Kutta-Fehlberg process of order 4/5 as, for example, provided in [10], the Runge-Kutta-Verner process of order 5/6, and a fourth order Adams predictor-corrector process. To facilitate the subsequent discussion, we denote these cases as methods RKF, RKV and PC respectively.

b) Two separate ode solution formulas were used in the multiprocessor

(parallel) tests; i.e., the Block Predictor-Corrector approach and the Parallel Predictor-Corrector approach (as described in chapter 3) which we denoted as BPC and PPC respectively. Three different number of processors (NP=2, 4 and 6) are used in the experiments with these parallel approaches. The integration order used in these experiments was $p=NP+1$. This choice was motivated primarily by the fact that it corresponds to the original formulation of the BPC method that appeared in the literature [12].
c) Three different user prescribed error tolerances were also considered for each problem/solution formula combination; i.e., three values of λ in eq'n (2.1.1). These values were chosen to provide final global solution accuracies which could be classified as low, medium and high.

The results of these experiments are presented in Tables 1-15 and Figures 1-43. In each of the Figures, the horizontal coordinate is time t and the vertical coordinate is the solution quality measure, R. Each of the Tables summarizes the results obtained for a particular problem using three different ode solution methods. Each Table includes the results for three values of λ and the four stepsize adjustment strategies that have been outlined. The meaning of the remaining columns in the Tables is described in the following section.

With the single process experiments, the specification of a solution method (RKF, RKV or PC) and a value for the error tolerance parameter λ, identifies a group of four results that corresponds to the four stepsize adjustment strategies being investigated. For convenience, we refer to such a set of four results as a "Q-set". This notion of a Q-set applies also to the multiprocessor experiments. However in that case, the specification of a solution method requires the identification of the specific approach (BPC or PPC) <u>and</u> the number of processors being used.

4.2 Comparison Criteria

In the experimental results, several comparison criteria were considered; namely,

NRHS: The number of derivative function evaluations per processor required over the solution interval.

ERR: The maximum global solution error that occurred (with respect to the known analytic solution).

NSS: The number of successful solution steps over the solution interval (for a single processor solution, this corresponds to the number of solution points, while for a multiprocessor case, this corresponds to the number of blocks).

NFS: The number of failed solution steps (or blocks); i,e, the number of occurrences of an R value which exceeded the admissible threshold, thereby causing a step rejection and a subsequent repetition with a reduced stepsize.

Avg R: The average of the R values taken over all successful steps.

In general, performance focuses on the achievement of high accuracy (low value of ERR) with low computational effort (low value of NRHS). In accessing the relative performance of two solutions, the unambiguous situation is where the value of both ERR and NRHS for one case is lower than for the other, thereby clearly identifying the superior case. This simple situation does not often occur. More typically, the value of ERR of *Sol'n-1* may be less than that for *Sol'n-2* while the value of NRHS for *Sol'n-1* is greater than for *Sol'n-2*. The superior solution is thus difficult to identify.

To deal with the problem, a simple blended assessment criterion or metric has been conceived. Its form is as follows:

$$METRIC = \frac{NRHS}{-\log(ERR)}$$

Here *log* is to the base 10. Low values of METRIC are desirable; i.e., indicate superior performance. The value of this criterion is included in the tabular results that are presented.

4.3 Discussion of the Results

4.3.1 Test Results for Single-processor Methods

A number of significant conclusions can be formulated from the results in Table 1 through 5 which summarize the results obtained with single processor solution methods.

1) The average value obtained for *R* (Avg R) with the newly proposed strategy S4 is almost always equal to the desired value of one, or is very close to one. The significant exception to this observation is the low accuracy result with problem TP1 & TP4 (all solution methods; i.e., RKF, RKV, PC). Note however that the other three strategies also have Avg R values significantly different from one for these two problems.

The results in these Tables also show that Avg R tends to be less than 0.9 with strategy S3 and is generally even less for strategy S1 (no bigger than 0.5). Frequently the Avg R values produced by S2 are in the region of one.

2) In Tables 1-5, within a Q-set (see section 4.1), the value of NRHS for S1 is usually the highest. Strategy S2 and S4 generally share the lowest

value in NRHS. Frequently the value of NRHS for S3 is the second largest. There are however, some cases where the NRHS value for S3 significantly exceeds that obtained with S1. These occurrences correspond to situations where the S3 strategy gives rise to a large number of failed steps (i.e., large NFS value). These tend to occur when the Predictor-Corrector method is used.

3) The global solution accuracy obtained within a Q-set is remarkably similar. Generally, however, strategy S1 provides the highest accuracy, and strategy S4 provides the lowest.

4) Unfortunately, within a Q-set, none of the four stepsize adjustment strategies consistently has the smallest value for METRIC (thereby showing best performance). For problem TP1, TP3 and TP5, the proposed strategy S4 usually performs the best, and for problem TP2 and TP4, strategy S2 has superior behavior (lowest METRIC value).

A particularly interesting presentation of some of the results is provided in Figures 1-13. These Figures display some representative "R-trajectories"; i.e. the successive values of the solution quality measure, R, over the solution interval.

The results given in Figure 1-2 and Figure 6-7 are for test problem TP1 and TP3 respectively using the RKF and PC solution methods. Each of these Figures corresponds to one particular value for the error tolerance parameter λ and shows the R-trajectories associated with the four stepsize adjustment strategies S1, S2, S3 and S4. The results shown in this group of Figures is representative of one category of observed R-trajectory behavior where all four stepsize adjustment strategies maintain a reasonably smooth constant value for the solution quality measure, R, after an initial transient interval.

The erratic behavior which occurs in Figure 6-7 in the region of t=1 corresponds to a severe derivative change in the y_2 component of the solution with problem TP3 (see Figure 43 in Appendix 1). It is intetesting to note in Figure 7, the oscillatory behavior of strategy S3 which occurs in the t=0 to t=2 region with the PC method. The capability of strategy S4 to maintain a value of one is clearly apparent.

The results given in Figures 3-5, 8-10 and 11-13 are for test problems TP2, TP4 and TP5 respectively, using the RKF, RKV and PC solution methods. The results shown in this group of Figures is representative of one category of observed R-trajectory behavior where all four stepsize adjustment strategies result in a periodic oscillation in the value of R. In Figures 3-5, the value for solution quality measure R varies slightly around 0.5 with strategy S1, around 0.9 with strategy S3, around 1.0 with both S2 and S4. Note that with strategy S4 most of the points do occur at the value of one for the solution quality measure. Similar behavior occurs in Figure 8-13, except for an abnormal oscillatory performance in Figure 10 with strategy S3 which corresponds to the case where S3 experiences many failed steps. Also notice in Figure 12 the vertical scale of the solution quality measure R with strategy S2 is double that used with the other strategies.

In this group of Figures, the cyclic pattern arising with strategy S1 and S2 is very closely synchronized with a similar cyclic pattern in the solution trajectories (see Appendix 1 solution trajectory figures). This periodicity is also present with strategies S3 and S4 but in the form of periodic spikes in the R-trajectories.

4.3.2 Test Results for BPC Approach

The BPC results are summarized in Table 6-10 and Figures 14-28. Some interesting observations can be made from these results:

1) As with the single processor solution methods, the average value obtained for R (Avg R) with the proposed strategy S4 is almost always equal to the desired value of one or is very close to one. The exceptions to this observation are the test results where the number of processors is 6 (NP=6) with problem TP1 and TP4 (see Table 6 and 9). In fact, for these two problems, none of the four strategies achieves a value of one for Avg R. For example, the Avg R value for strategy S1, S3 and S4 is in the range betweem 0.4-0.7 and for strategy S2 it is usually over 1.2 (this high Avg R value tends to appear in the lowest accuracy cases; see for example Table 6 where Avg R = 2.65).

Generally, the results in these Tables show that the Avg R value tends to be less than 0.9 with strategy S3 and is typically even less for strategy S1 (no bigger than 0.5). Frequently the Avg R values produced by S2 are in the region of one.

2) In Tables 6-10, within a Q-set, the value of NRHS for S1 is usually the highest. Strategy S2 and S4 generally share the lowest value in NRHS. Frequently the value of NRHS for S3 is the second largest. There are however, some cases where the NRHS value for S3 significantly exceeds that obtained with S1. These occurrences correspond to situations where the S3 strategy gives rise to a large number of failed steps (i.e., large NFS value).

3) The global solution accuracy obtained within a Q-set is also

remarkably similar. However strategy S1 generally provides the highest accuracy, and S4 provides the lowest.

4) Generally, when NP increases with λ fixed, the value of METRIC tends to monotonically decrease for any particular strategy. One particular exception to this occurs with problem TP1 and TP4 in the lowest accuracy cases; i.e., highest λ .

Further interesting results are presented in Figure 14-28, using the BPC approach. Each of these Figures corresponds to the experiments within a particular Q-set, and shows the R-trajectories associated with the four stepsize adjustment strategies.

The Figures 14-28 can be separated into two groups corresponding to NP=2 & 4 (Figures 14, 15, 17, 18, 20, 21, 23, 24, 26, 27) and NP=6 (Figures 16, 19, 22, 25, 28). Notice in the first group a consistently smooth behavior of the R-trajectories for strategies S1, S2 and S4. On the other hand, the R-trajectory for S3 frequently behaves erratically. This behavior corresponds to the cases where S3 experiences many failed steps. The capability of strategy S2 & S4 to maintain a value of one are clearly apparent. In the second group, the R-trajectories for all four stepsize strategies show significant oscillatory behavior. Erratic behavior with S3 is particularly evident and results from the large number of failed steps.

The "complexity" of the solution trajectories for the problems influences the smoothness of the R-trajectories, but this influence is not particularly apparent with lower order (lower NP) cases.

It is worth noticing that, for all the stepsize adjustment strategies, the steady smooth behavior of R-trajectories starts to deteriorate when NP increases. And also notice that the cyclic patterns arising with strategies S1, S2 and S4 are very closely synchronized with a similar cyclic pattern in the solution trajectories of the associated problems TP2, TP4 and TP5 (see Figures A1.2, A1.4, A1.5 in Appendix 1).

4.3.3 Test Results for PPC Approach

A number of significant observations can be drawn from Tables 11-15, which summarize the results obtained with PPC approach.

1) The newly proposed stepsize adjustment strategy is not as successful in maintaining a value of one for Avg R with the PPC approach as it is with the BPC approach. Difficulties are encountered with problems TP1 and TP4 with the lowest accuracy case (i.e., $\lambda = 10^{-4}$), and with all problems with NP=6. It should be noted, however, that the other strategies also have difficulty with the low accuracy experiments with TP1 and TP4. The Avg R value for the NP=6 case (all problems) obtained by the other three strategies also differs signficantly from one, (although S2 provides a few exceptions). The S2 results are frequently quite high (e.g. Avg R = 3.8 in Table 11 with NP=6, $\lambda = 10^{-4}$) while the other strategies provide relatively low value for Avg R (e.g. Avg R = 0.04 in Table 14 with S3 and NP=6, $\lambda = 10^{-4}$).

The test results obtained suggest that with the PPC approach, there is an inherent difficulty in maintaining an Avg R value at the desired value of one.

2) In Tables 11-15, the value of NRHS for S2 is almost always the lowest within a Q-set. The highest value of NRHS usually occurs either with S3 or S4. These occurrences correspond to situations where S3 or S4 results in a large number of failed steps (i.e., large NFS value).

3) The global solution accuracy obtained within a Q-set is still very

similar. Generally, the highest accuracy is provided by strategy S1 and the lowest by S2.

4) Generally, when NP increases with λ fixed, the value of METRIC tends to monotonically decrease for any particular strategy. An exception to this occurs with the low accuracy case with problem TP1.

Further interesting results of the PPC approach are presented in Figures 29-43, which display some representative "R-trajectories". Each of these Figures corresponds to the experiments within a particular Q-set.

These Figures can be separated into two groups corresponding to the number of processors, i.e., NP=2 & 4 , and NP=6. Notice in the first group a generally smooth behavior of the R-trajectories for strategies S1, S2 and S4. On the other hand, the R-trajectory for S3 always behaves erratically.

The rapid changes in the solution trajectories in problem TP5 which occur at (approximately) t=6.5, 12.5, 18 are reflected in the erratic behavior of the R-trajectories (Figures 42 and 43). A similar relationship can be detected between the solution behavior of problem TP4 and the associated R-trajectories of Figures 38 and 39. However, the "complexity" of the solution trajectories for the problems influences the smoothness of the R-trajectories, but this influence is not particularly apparent with lower order (lower NP) cases.

A particularly undesirable behavior with strategy S3 can be noted in almost all the cases with NP=4 and 6, for all the test problems. (The exception is NP=4 with problem TP1). In these cases, the R-trajectories repeatedly acquire very low values.

With all the PPC experiments, erratic behavior of the R-trajectories becomes more pronounced as the value of NP increases.

Chapter 5
Concluding Remarks

5.1 General Observations

In this thesis, we have investigated the performance of a family of stepsize adjustment techniques for ordinary differential equation solvers, within the context of both single and multiple processor procedures. Three of these strategies have appeared previously in the literature while the fourth is a newly proposed strategy.

The results obtained in this study suggest that the widely used "locally optimal" stepsize adjustment strategy (designated as S1 in our study) almost always provides inferior performance relative to the other alternatives. The design of the newly proposed strategy S4 was based on the goal of maintaining a value of unity for the solution quality measure, *R*. The objective was reasonably successfully achieved, particularly in the case of the single processor and the BPC multiprocessor experiments. The ability of the S4 strategy to keep *R* at the value one does appear to be superior to all the other strategies that were examined.

The conclusions in this thesis are primarily based on an extensive set of experiments with five test problems. We note however that other problems were also examined during the course of the study, but for reasons of limited space, these have not all been presented. These unreported test results are entirely consistent with those that appear in this thesis.

For both single and multiprocessor variable stepsize approaches, the test results as shown in Tables 1 through 15 and Figures 1 through 43, provide adequate evidence to conclude that:

(a) Strategy S2 could be a highly recommended stepsize adjustment strategy in terms of its over all performance. It is, for example, free of failed steps and it frequently has a low METRIC value.

(b) Within any particular Q-set, none of the four stepsize adjustment strategies consistently has the smallest value for METRIC (thereby showing best performance). This applies equally to the single and multiprocessor experiments.

(c) The global solution accuracies obtained within a Q-set (i.e, the ERR values) are very similar and mostly within a factor of 2. This is true even for those cases where many failed steps occur (this frequently happens with strategy S3) which suggests that the solution methods are reasonably robust.

(d) The results obtained suggest that the five test problems do vary in difficulty from a numerical solution point of view. Problems TP1 and TP3 seem to be fairly "easy" inasmuch as they generally provided smooth R-trajectories, and relatively low NFS (number of failed steps). In contrast, problem TP4 frequently resulted in erratic R-trajectories and many failed steps.

(e) There is a difference between the behavior with high and low accuracy solutions. For example, in the high accuacy cases, the values of NRHS and NSS are higher and the R-trajectories are smoother, than in low accuracy cases. The implication here seems to be that the smaller stepsizes associated with the high accuracy solutions tend to enhance the ability of the stepsize adjustment strategies to maintain a constant value of R.

Within the context of any particular single processor solution method (i.e., RKF, RKV, PC), when the error tolerance parameter λ decreases (i.e., the solution accuracy increases), the value of METRIC tends to monotonically increase for any particular stepsize adjustment strategy. Among the three single processor methods, RKF method usually provides the highest value in NRHS, the lowest value in NFS, and the smoothest R-trajectories, for almost all the stepsize adjustment strategies.

Within the context of the BPC approach, the newly proposed stepsize adjustment strategy S4 usually performs best with a relatively low number of processors (NP) and hence for low order (because order=NP+1 for the BPC approach). Here again the implication appears to be that the lower stepsize associated with the low order method enhances the performance of the adjustment strategy. Superior behavior (lowest value of METRIC within a Q-set) is normally shared by S2 and S4. However, for the case where NP=6, strategy S2 almost always has superior behavior, since strategy S4 frequently experiences a few failed steps, which causes a higher NRHS than that with strategy S2. Notice also that the value of NFS (the number of failed stepsize) is consistently zero for strategy S2.

There is more erratic behavior in the R-trajectories with the PPC approach than with the BPC approach and this is probably due to the higher number of failed steps (NFS) which occur. Within the context of the PPC approach, strategy S2 usually provides superior behavior (lowest METRIC value within a Q-set), for cases where NP = 2, 4. Generally speaking the relative complexity of the PPC approach seems to make it more difficult to manage within a variable stepsize mode.

5.2 Future Work

The work completed in this thesis study is a contribution to the evaluation of variable stepsize techniques for both single and multiple processor approaches for ode solvers. In particular, the study demonstrates that further work on the development of efficient stepsize adjustment strategies is desirable.

The formulation of current stepsize adjustment strategies is primarily based on the solution quality measure, R, from the current step and possibly from the previous step (in the case of Watts' strategy). A possibly fruitful avenue to explore would be the incorporation of additional input to the adjustment strategy, such as first and second derivative information. Such information could assist in anticipating rapid changes in the solution trajectory.

In the multiprocessor experiments with the BPC and PPC approaches, our study examined only the cases where the order of the solution method was fixed at NP+1 (where NP=the number of processors). Further important insight could be obtained by examining the variable stepsize behavior of these multiprocessor approaches when the integration order and the number of processor is decoupled. Software for carrying out such experiments is now available. Such a study could provide valuable input into the problem of how to optimally choose integration order in multiprocessor environments.

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
RKF	1.E-4	S1	80	0.51E-4	14	0	0.25	18.7
		S2	74	0.51E-4	13	0	0.37	17.3
		S3	74	0.51E-4	13	0	0.34	17.3
		S4	68	0.51E-4	12	0	0.60	15.9
	1.E-7	S1	254	0.40E-7	43	0	0.36	34.3
		S2	224	0.66E-7	38	0	0.65	31.2
		S3	224	0.54E-7	38	0	0.65	30.8
		S4	212	0.71E-7	36	0	0.85	29.7
	1.E-9	S1	590	0.39E-9	99	0	0.42	63.7
		S2	518	0.74E-9	87	0	0.81	56.7
		S3	506	0.71E-9	85	0	0.88	55.3
		S4	500	0.77E-9	84	0	0.94	54.9
RKV	1.E-4	S1	74	0.19E-4	10	0	0.21	15.7
		S2	66	0.23E-4	9	0	0.35	14.2
		S3	73	0.19E-4	9	1	0.32	15.4
		S4	74	0.55E-4	10	0	0.42	17.4
	1.E-7	S1	210	0.22E-7	27	0	0.34	27.4
		S2	194	0.14E-7	25	0	0.52	24.7
		S3	200	0.13E-7	24	2	0.57	25.4
		S4	185	0.17E-7	23	1	0.72	23.8
	1.E-9	S1	442	0.85E-10	56	0	0.37	43.9
		S2	394	0.15E-9	50	0	0.69	40.1
		S3	378	0.17E-9	48	0	0.84	38.7
		S4	385	0.17E-9	48	1	0.87	39..4
PC	1.E-4	S1	41	0.38E-3	18	1	0.35	12.0
		S2	40	0.13E-2	18	0	0.93	13.8
		S3	54	0.37E-3	22	6	0.43	15.7
		S4	44	0.90E-3	19	2	0.70	14.4
	1.E-7	S1	140	0.65E-6	68	0	0.41	22.6
		S2	126	0.11E-5	61	0	0.78	21.2
		S3	128	0.91E-6	62	0	0.68	21.2
		S4	122	0.12E-5	59	0	0.92	20.6
	1.E-9	S1	336	0.15E-7	166	0	0.45	42.9
		S2	296	0.26E-7	146	0	0.89	39.0
		S3	297	0.24E-7	146	1	0.88	39.0
		S4	292	0.26E-7	144	0	0.96	38.5

Table 1 : Single Processor Methods with Problem TP1

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
RKF	1.E-7	S1	1052	0.13E-3	176	0	0.51	269.7
		S2	914	0.25E-3	153	0	1.01	253.7
		S3	996	0.18E-3	165	2	0.70	266.0
		S4	920	0.26E-3	154	0	1.00	256.0
	1.E-9	S1	2642	0.12E-5	441	0	0.50	445.4
		S2	2300	0.23E-5	384	0	1.00	408.4
		S3	2446	0.20E-5	395	16	0.87	429.9
		S4	2300	0.24E-5	384	0	1.00	408.8
	1.E-11	S1	6638	0.12E-7	1107	0	0.50	836.1
		S2	5780	0.23E-7	964	0	1.00	756.7
		S3	5906	0.21E-7	985	0	0.90	768.7
		S4	5780	0.23E-7	964	0	1.00	756.8
RKV	1.E-7	S1	1066	0.18E-5	134	0	0.50	185.6
		S2	954	0.37E-5	120	0	0.99	175.5
		S3	1010	0.25E-5	127	0	0.70	180.5
		S4	954	0.36E-5	120	0	0.98	175.3
	1.E-9	S1	2306	0.12E-7	289	0	0.50	291.4
		S2	2058	0.26E-7	258	0	1.00	271.1
		S3	2090	0.23E-7	262	0	0.90	273.5
		S4	2058	0.26E-7	258	0	0.99	271.0
	1.E-11	S1	4970	0.84E-10	622	0	0.50	493.3
		S2	4426	0.18E-9	554	0	1.00	453.8
		S3	4506	0.16E-9	564	0	0.90	459.7
		S4	4434	0.18E-9	555	0	1.00	454.6
PC	1.E-7	S1	634	0.81E-3	315	0	0.50	204.9
		S2	552	0.16E-2	274	0	1.00	197.4
		S3	592	0.11E-2	294	0	0.70	200.9
		S4	554	0.16E-2	275	0	1.00	198.0
	1.E-9	S1	1596	0.80E-5	796	0	0.50	313.3
		S2	1390	0.16E-4	693	0	1.00	290.2
		S3	1421	0.15E-4	708	1	0.90	293.8
		S4	1392	0.16E-4	694	0	0.99	290.5
	1.E-11	S1	4006	0.72E-7	2001	0	0.50	561.0
		S2	3488	0.15E-6	1742	0	1.00	510.8
		S3	4377	0.10E-6	1875	623	0.70	626.1
		S4	3490	0.15E-6	1743	0	1.00	511.0

Table 2 : Single Processor Methods with Problem TP2

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-8	S1	587	0.51E-2	580	0	0.48	256.0
		S2	496	0.77E-2	492	0	0.96	234.4
		S3	601	0.65E-2	571	46	0.68	274.9
		S4	491	0.77E-2	487	0	1.00	232.4
	1.E-10	S1	1815	0.20E-3	1806	0	0.49	492.0
		S2	1523	0.35E-3	1516	0	0.99	440.0
		S3	1980	0.29E-3	1847	247	0.70	559.4
		S4	1522	0.34E-3	1513	0	1.00	439.3
	1.E-13	S1	10157	0.12E-5	10140	0	0.50	1713.4
		S2	8540	0.20E-5	8526	0	1.00	1497.2
		S3	11541	0.17E-5	10698	1653	0.67	1996.9
		S4	8538	0.20E-5	8521	0	1.00	1497.0
NP=4 5th order	1.E-8	S1	147	0.50E-4	142	3	0.37	34.2
		S2	123	0.62E-4	122	0	0.89	29.2
		S3	236	0.16E-4	209	46	0.25	49.1
		S4	132	0.13E-3	126	4	0.86	33.9
	1.E-10	S1	300	0.73E-6	293	2	0.43	48.9
		S2	261	0.89E-6	257	0	0.92	43.1
		S3	458	0.13E-5	416	73	0.29	77.6
		S4	266	0.14E-5	259	3	0.95	45.5
	1.E-13	S1	916	0.64E-9	908	0	0.48	99.6
		S2	814	0.30E-9	808	1	1.00	85.5
		S3	1131	0.11E-8	1030	185	0.61	126.0
		S4	811	0.24E-8	803	0	1.00	94.0
NP=6 7th order	1.E-8	S1	91	0.50E-4	86	8	0.30	21.2
		S2	74	0.13E-3	73	0	0.83	19.1
		S3	144	0.89E-4	139	8	0.10	35.6
		S4	96	0.61E-4	89	11	0.41	22.8
	1.E-10	S1	136	0.30E-6	130	5	0.34	20.9
		S2	113	0.48E-6	112	0	2.90	17.9
		S3	242	0.77E-6	230	17	0.10	39.6
		S4	142	0.11E-5	133	12	0.58	23.8
	1.E-13	S1	288	0.50E-9	280	4	0.42	31.0
		S2	256	0.12E-7	252	1	1.70	32.3
		S3	584	0.49E-9	560	37	0.15	62.8
		S4	283	0.97E-9	271	13	0.83	31.4

Table 3 : Single Processor Methods with Problem TP3

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
RKF	1.E-4	S1	414	0.54E-3	58	14	0.43	126.7
		S2	278	0.21E-2	47	0	1.63	103.6
		S3	436	0.59E-3	60	16	0.38	135.1
		S4	394	0.79E-3	58	10	0.47	127.0
	1.E-7	S1	1366	0.78E-6	215	16	0.49	223.6
		S2	1100	0.18E-5	184	0	1.12	191.3
		S3	1341	0.11E-5	205	23	0.60	225.0
		S4	1241	0.15E-5	195	15	0.76	212.9
	1.E-9	S1	3212	0.90E-8	531	6	0.50	399.2
		S2	2768	0.18E-7	462	0	1.02	357.2
		S3	3030	0.15E-7	479	32	0.84	386.5
		S4	2887	0.19E-7	471	13	0.92	373.8
RKV	1.E-4	S1	410	0.12E-3	45	8	0.44	104.3
		S2	298	0.46E-3	38	0	1.61	89.4
		S3	483	0.87E-4	48	15	0.30	119.0
		S4	419	0.13E-3	47	7	0.41	107.9
	1.E-7	S1	1178	0.13E-6	134	16	0.46	171.2
		S2	922	0.34E-6	116	0	1.21	142.6
		S3	1181	0.19E-6	130	21	0.52	175.9
		S4	1107	0.25E-6	126	15	0.64	167.5
	1.E-9	S1	2369	0.16E-8	282	17	0.49	269.7
		S2	1986	0.36E-8	249	0	1.09	235.3
		S3	2250	0.28E-8	261	24	0.77	262.9
		S4	2178	0.33E-8	259	16	0.81	256.6
PC	1.E-4	S1	218	0.41E-2	96	22	0.45	91.5
		S2	164	0.13E-1	80	0	1.47	86.5
		S3	222	0.49E-4	99	20	0.41	96.2
		S4	202	0.67E-2	92	14	0.54	92.8
	1.E-7	S1	782	0.12E-4	382	14	0.49	158.9
		S2	662	0.23E-4	329	0	1.07	142.6
		S3	755	0.16E-4	361	29	0.65	157.3
		S4	698	0.21E-4	340	14	0.88	149.0
	1.E-9	S1	1926	0.28E-6	960	2	0.50	293.7
		S2	1676	0.49E-6	836	0	1.00	265.5
		S3	1926	0.40E-6	879	164	0.78	301.1
		S4	1688	0.50E-6	839	6	0.98	268.0

Table 4 : Single Processor Methods with Problem TP4

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
RKF	1.E-8	S1	2927	0.11E-2	481	9	0.51	992.8
		S2	2504	0.18E-2	418	0	1.07	914.4
		S3	2636	0.19E-2	435	6	0.79	972.4
		S4	2486	0.22E-2	415	0	1.00	933.0
	1.E-10	S1	7208	0.11E-4	1202	0	0.50	1457.2
		S2	6284	0.22E-4	1048	0	0.99	1349.4
		S3	6428	0.21E-4	1067	6	0.89	1372.1
		S4	6260	0.22E-4	1044	0	1.00	1346.6
	1.E-13	S1	28652	0.58E-8	4776	0	0.50	3478.1
		S2	24950	0.17E-7	4159	0	0.99	3216.2
		S3	25454	0.15E-7	4243	0	0.90	3256.1
		S4	24920	0.17E-7	4154	0	1.00	3212.6
RKV	1.E-8	S1	3365	0.13E-3	340	93	0.42	864.9
		S2	2322	0.78E-3	291	0	1.35	746.7
		S3	2523	0.67E-4	303	15	0.76	604.4
		S4	2351	0.13E-3	292	3	0.97	606.0
	1.E-10	S1	5617	0.78E-6	702	1	0.51	919.7
		S2	4994	0.19E-5	625	0	1.02	874.3
		S3	5251	0.56E-6	637	23	0.87	839.4
		S4	4978	0.53E-6	623	0	1.00	792.8
	1.E-13	S1	17697	0.29E-8	2212	1	0.50	2071.4
		S2	15762	0.28E-8	1971	0	0.99	1842.9
		S3	16074	0.35E-8	2003	8	0.90	1899.7
		S4	15730	0.35E-8	1967	0	1.00	1860.7
PC	1.E-8	S1	3176	0.35E-2	1585	2	0.50	1294.6
		S2	2764	0.83E-2	1380	0	1.00	1329.6
		S3	2904	0.32E-2	1443	14	0.80	1161.8
		S4	2763	0.43E-2	1379	1	1.00	1166.5
	1.E-10	S1	7971	0.32E-4	3983	1	0.50	1772.9
		S2	6940	0.67E-4	3468	0	1.00	1663.3
		S3	8671	0.42E-4	3725	1217	0.70	1981.3
		S4	6935	0.40E-4	3464	1	1.00	1578.0
	1.E-13	S1	31702	0.19E-7	15849	0	0.50	4105.5
		S2	27602	0.29E-7	13799	0	1.00	3662.4
		S3	34641	0.12E-7	14845	4947	0.70	4369.7
		S4	27594	0.16E-7	13795	0	1.00	3538.5

Table 5 : Single Processor Methods with Problem TP5

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-4	S1	74	0.23E-3	35	2	0.41	20.5
		S2	60	0.31E-3	30	0	1.20	17.1
		S3	84	0.30E-4	38	4	0.39	18.6
		S4	70	0.66E-4	32	3	0.77	16.8
	1.E-7	S1	336	0.96E-8	168	0	0.46	41.9
		S2	286	0.69E-8	143	0	0.89	35.0
		S3	304	0.31E-7	152	0	0.69	40.5
		S4	280	0.25E-7	140	0	0.96	36.8
	1.E-9	S1	1040	0.47E-9	520	0	0.48	111.5
		S2	876	0.47E-9	438	0	0.96	93.9
		S3	894	0.47E-9	446	1	0.89	95.8
		S4	872	0.47E-9	436	0	0.99	93.5
NP=4 5th order	1.E-4	S1	74	0.20E-4	29	6	0.48	14.9
		S2	56	0.16E-3	28	0	2.11	14.7
		S3	72	0.20E-4	30	6	0.43	15.3
		S4	64	0.40E-4	28	4	0.67	14.6
	1.E-7	S1	100	0.19E-6	50	0	0.37	14.9
		S2	88	0.31E-7	44	0	0.74	11.7
		S3	126	0.19E-7	51	12	0.60	16.3
		S4	98	0.22E-7	46	3	0.82	12.8
	1.E-9	S1	202	0.28E-10	101	0	0.43	19.1
		S2	182	0.29E-10	91	0	0.82	17.3
		S3	262	0.22E-9	98	33	0.63	27.1
		S4	176	0.95E-10	88	0	0.95	17.6
NP=6 7th order	1.E-4	S1	78	0.19E-4	32	7	0.51	16.5
		S2	64	0.35E-3	32	0	2.65	18.5
		S3	98	0.24E-4	36	13	0.41	21.2
		S4	76	0.19E-4	32	6	0.53	16.1
	1.E-7	S1	96	0.19E-7	41	7	0.45	14.2
		S2	74	0.16E-7	37	0	1.60	10.9
		S3	128	0.87E-7	41	23	0.48	18.1
		S4	82	0.27E-7	37	4	0.68	10.8
	1.E-9	S1	110	0.23E-9	49	6	0.43	11.4
		S2	92	0.10E-7	46	0	0.64	11.5
		S3	152	0.20E-9	51	25	0.54	15.7
		S4	112	0.26E-9	50	6	0.64	11.7

Table 6 : BPC Methods with Problem TP1

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-7	S1	1760	0.15E-5	880	0	0.50	301.7
		S2	1480	0.35E-5	740	0	1.00	271.2
		S3	1622	0.22E-5	810	1	0.70	287.1
		S4	1482	0.35E-5	741	0	0.99	271.4
	1.E-9	S1	5560	0.46E-8	2780	0	0.50	667.0
		S2	4678	0.11E-7	2339	0	1.00	588.0
		S3	4804	0.97E-8	2401	1	0.90	599.4
		S4	4678	0.11E-7	2339	0	1.00	588.0
	1.E-11	S1	17574	0.49E-11	8787	0	0.50	1554.3
		S2	14780	0.19E-10	7390	0	1.00	1379.8
		S3	15176	0.15E-10	7587	1	0.90	1401.1
		S4	14782	0.19E-10	7390	1	1.00	1379.8
NP=4 5th order	1.E-7	S1	350	0.47E-5	175	0	0.50	65.7
		S2	312	0.11E-4	156	0	0.99	62.7
		S3	338	0.69E-5	166	3	0.69	65.5
		S4	312	0.10E-4	156	0	0.99	62.6
	1.E-9	S1	754	0.24E-7	376	1	0.50	99.0
		S2	670	0.54E-7	335	0	0.99	92.2
		S3	954	0.31E-7	359	118	0.67	127.2
		S4	674	0.53E-7	336	1	0.98	92.6
	1.E-11	S1	1614	0.12E-9	806	1	0.50	162.5
		S2	1436	0.26E-9	718	0	1.00	149.8
		S3	2052	0.15E-9	770	256	0.67	208.9
		S4	1440	0.26E-9	719	1	0.99	150.1
NP=6 5th order	1.E-7	S1	154	0.42E-5	75	2	0.51	28.6
		S2	136	0.12E-4	68	0	1.00	27.6
		S3	190	0.53E-5	75	20	0.52	36.0
		S4	160	0.76E-5	73	7	0.74	31.3
	1.E-9	S1	270	0.57E-7	131	4	0.49	37.3
		S2	242	0.15E-6	121	0	1.82	35.5
		S3	348	0.57E-7	127	47	0.63	48.1
		S4	248	0.89E-7	121	3	0.96	35.2
	1.E-11	S1	460	0.31E-9	228	2	0.49	48.4
		S2	420	0.65E-9	210	0	1.00	45.7
		S3	592	0.39E-9	220	76	0.66	62.9
		S4	422	0.64E-9	209	2	0.98	45.9

Table 7 : BPC Methods with Problem TP2

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-8	S1	910	0.13E-5	455	0	0.48	454.5
		S2	768	0.29E-5	384	0	0.96	138.8
		S3	808	0.24E-5	402	2	0.80	143.7
		S4	766	0.47E-5	382	1	0.99	143.8
	1.E-10	S1	2854	0.88E-8	1427	0	0.49	354.3
		S2	2402	0.18E-7	1201	0	0.99	309.8
		S3	3522	0.38E-7	1322	439	0.68	474.9
		S4	2396	0.19E-7	1198	0	1.00	310.4
	1.E-13	S1	15984	0.45E-8	7992	0	0.50	1915.4
		S2	13444	0.38E-8	6722	0	1.00	1597.1
		S3	13798	0.39E-8	6898	1	0.90	1640.4
		S4	13438	0.38E-8	6719	0	1.00	1596.0
NP=4 5th order	1.E-8	S1	222	0.80E-5	109	2	0.42	42.9
		S2	196	0.18E-4	98	0	0.84	41.3
		S3	214	0.12E-4	99	8	0.73	43.4
		S4	200	0.63E-5	97	3	0.87	38.4
	1.E-10	S1	462	0.17E-7	230	1	0.46	59.4
		S2	410	0.76E-7	205	0	0.91	57.6
		S3	554	0.46E-7	215	62	0.69	75.5
		S4	412	0.11E-6	204	2	0.96	59.1
	1.E-13	S1	1426	0.70E-9	712	1	0.48	155.7
		S2	1270	0.61E-9	635	0	0.96	137.8
		S3	1656	0.59E-9	667	161	0.73	179.5
		S4	1268	0.61E-9	633	1	0.99	137.6
NP=6 7th order	1.E-8	S1	118	0.39E-5	54	5	0.37	21.8
		S2	100	0.22E-4	50	0	0.80	21.5
		S3	134	0.68E-5	51	16	0.53	25.9
		S4	114	0.12E-4	51	6	0.57	23.1
	1.E-10	S1	198	0.19E-5	94	5	0.38	34.7
		S2	172	0.12E-6	86	0	0.82	24.8
		S3	230	0.50E-7	88	27	0.64	31.5
		S4	182	0.37E-7	86	5	0.78	24.5
	1.E-13	S1	442	0.61E-9	218	3	0.43	48.0
		S2	400	0.51E-9	200	0	0.86	43.0
		S3	536	0.46E-9	204	64	0.69	57.4
		S4	404	0.44E-9	198	4	0.92	43.2

Table 8 : BPC Methods with Problem TP3

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-4	S1	412	0.96E-4	187	19	0.47	102.6
		S2	302	0.38E-3	151	0	1.30	88.3
		S3	464	0.17E-3	201	31	0.40	123.4
		S4	370	0.16E-3	169	16	0.73	97.4
	1.E-7	S1	2052	0.50E-7	1022	4	0.50	281.1
		S2	1718	0.79E-7	859	0	1.00	242.0
		S3	1944	0.51E-7	945	27	0.69	266.5
		S4	1754	0.66E-7	868	9	0.97	244.3
	1.E-9	S1	6452	0.48E-9	3226	0	0.50	692.4
		S2	5426	0.48E-9	2713	0	1.00	582.3
		S3	7564	0.48E-9	2958	824	0.72	811.7
		S4	5426	0.48E-9	2713	0	1.00	582.3
NP=4 5th order	1.E-4	S1	206	0.14E-3	87	16	0.49	53.7
		S2	168	0.35E-3	84	0	1.20	48.7
		S3	234	0.91E-4	91	26	0.39	57.9
		S4	192	0.35E-3	88	8	0.72	55.5
	1.E-7	S1	526	0.53E-7	244	19	0.48	72.3
		S2	428	0.18E-6	214	0	1.20	63.4
		S3	574	0.73E-7	239	47	0.58	80.2
		S4	484	0.97E-7	226	16	0.78	69.0
	1.E-9	S1	10[illegible]	0.31E-9	519	12	0.49	111.7
		S2	922	0.87E-9	461	0	1.00	101.8
		S3	1192	0.50E-9	486	110	0.74	128.2
		S4	976	0.75E-9	473	15	0.90	106.9
NP=6 7th order	1.E-4	S1	230	0.82E-4	94	21	0.48	56.3
		S2	186	0.28E-3	93	0	1.70	52.[illegible]
		S3	246	0.23E-3	98	25	0.43	67.8
		S4	210	0.38E-3	95	10	0.67	61.4
	1.E-7	S1	294	0.22E-6	118	29	0.47	44.2
		S2	222	0.20E-6	111	0	1.80	33.1
		S3	354	0.74E-7	120	57	0.48	49.7
		S4	268	0.86E-7	117	17	0.46	37.9
	1.E-9	S1	428	0.46E-9	194	20	0.48	45.8
		S2	352	0.14E-8	176	0	1.20	39.8
		S3	466	0.70E-9	187	46	0.65	50.9
		S4	402	0.85E-9	185	16	0.73	44.3

Table 9 : BPC Methods with Problem TP4

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-8	S1	8824	0.25E-5	4411	1	0.50	1575.6
		S2	7420	0.60E-5	3710	0	1.00	1445.2
		S3	7848	0.52E-5	3920	4	0.80	1485.3
		S4	7418	0.61E-5	3708	1	1.00	1454.3
	1.E-10	S1	27880	0.73E-8	13939	1	0.50	3427.5
		S2	23444	0.13E-7	11722	0	1.00	2974.6
		S3	34500	0.31E-7	12938	4312	0.68	4592.3
		S4	23444	0.38E-7	11720	2	1.00	3157.2
	1.E-13	S1	156716	0.89E-7	78358	0	0.50	22223.5
		S2	131784	0.75E-7	65892	0	1.00	18496.3
		S3	194000	0.82E-7	72751	24249	0.68	27382.3
		S4	131778	0.75E-7	65889	0	1.00	18493.5
NP=4 5th order	1.E-8	S1	1990	0.45E-5	984	1	0.50	351.5
		S2	1772	0.11E-5	886	0	1.00	298.0
		S3	2204	0.39E-5	923	164	0.71	408.1
		S4	1782	0.72E-5	887	4	0.99	347.6
	1.E-10	S1	4272	0.14E-7	2135	1	0.50	544.2
		S2	3806	0.16E-7	1903	0	1.00	488.6
		S3	5390	0.14E-7	2033	662	0.68	687.0
		S4	3806	0.25E-7	1901	1	1.00	500.6
	1.E-13	S1	13478	0.78E-8	6738	1	0.50	1661.9
		S2	12008	0.71E-8	6004	0	1.00	1473.1
		S3	17134	0.75E-8	6426	2141	0.67	2109.0
		S4	12004	0.70E-8	6001	1	1.00	1472.2
NP=6 7th order	1.E-8	S1	1112	0.75E-6	484	72	0.46	181.5
		S2	872	0.82E-5	436	0	1.20	171.4
		S3	1126	0.33E-6	459	104	0.66	173.7
		S4	912	0.21E-5	443	13	0.91	160.9
	1.E-10	S1	1698	0.25E-6	842	7	0.51	257.2
		S2	1544	0.67E-7	772	0	1.00	215.2
		S3	2120	0.50E-8	808	252	0.69	255.4
		S4	1558	0.36E-7	773	6	0.98	209.4
	1.E-13	S1	4026	0.18E-8	1998	15	0.50	459.8
		S2	3656	0.18E-8	1828	0	1.00	418.4
		S3	5110	0.20E-8	1922	633	0.68	587.8
		S4	3672	0.18E-8	1828	8	0.99	420.3

Table 10 : BPC Methods with Problem TP5

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-4	S1	46	0.50E-4	44	0	0.25	10.7
		S2	41	0.89E-4	39	0	0.49	10.1
		S3	64	0.44E-4	59	6	0.25	14.7
		S4	41	0.11E-3	38	2	0.74	10.4
	1.E-7	S1	212	0.23E-6	210	0	0.43	32.0
		S2	179	0.40E-6	177	0	0.84	28.0
		S3	209	0.31E-6	202	11	0.59	32.1
		S4	173	0.41E-6	171	0	0.95	27.1
	1.E-9	S1	661	0.71E-8	654	0	0.47	81.1
		S2	553	0.12E-7	549	0	0.94	69.9
		S3	662	0.10E-7	627	56	0.74	82.9
		S4	550	0.12E-7	543	0	0.99	69.0
NP=4 5th order	1.E-4	S1	58	0.17E-4	54	6	0.37	12.2
		S2	47	0.13E-3	46	0	1.60	12.1
		S3	61	0.23E-4	57	5	0.19	13.1
		S4	55	0.51E-4	51	6	0.47	12.8
	1.E-7	S1	81	0.30E-7	74	11	0.30	10.8
		S2	50	0.13E-5	49	0	2.20	8.5
		S3	77	0.26E-7	77	10	0.40	10.9
		S4	72	0.15E-6	67	7	0.58	10.5
	1.E-9	S1	122	0.26E-9	120	2	0.37	12.7
		S2	105	0.40E-8	104	0	0.85	12.5
		S3	134	0.25E-9	126	13	0.63	13.9
		S4	115	0.14E-8	112	4	0.82	13.0
NP=6 7th order	1.E-4	S1	62	0.11E-4	57	7	0.42	12.5
		S2	49	0.52E-3	48	0	3.80	14.9
		S3	59	0.24E-4	56	4	0.21	12.8
		S4	63	0.29E-4	58	7	0.28	13.9
	1.E-7	S1	70	0.12E-7	65	8	0.38	8.8
		S2	58	0.39E-6	57	0	3.20	9.1
		S3	88	0.30E-7	83	7	0.67	11.7
		S4	72	0.36E-7	67	7	0.06	9.7
	1.E-9	S1	80	0.12E-9	75	8	0.35	8.1
		S2	65	0.65E-9	64	0	1.00	7.1
		S3	114	0.15E-9	108	10	0.12	11.6
		S4	82	0.62E-9	76	10	0.33	8.9

Table 11 : PPC Methods with Problem TP1

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-7	S1	1125	0.86E-3	1118	0	0.50	366.7
		S2	942	0.15E-2	938	0	1.00	333.0
		S3	1598	0.96E-3	1426	331	0.43	529.8
		S4	949	0.14E-2	942	0	1.00	333.3
	1.E-9	S1	3538	0.28E-4	3526	0	0.50	776.3
		S2	2973	0.47E-4	2964	0	1.00	686.4
		S3	4234	0.38E-4	3972	701	0.63	958.9
		S4	2979	0.46E-4	2967	0	1.00	687.5
	1.E-11	S1	11156	0.88E-6	11139	0	0.50	1840.5
		S2	9380	0.15E-5	9366	0	1.00	1609.1
		S3	13144	0.12E-5	12082	2091	0.64	2224.8
		S4	9381	0.15E-5	9367	0	1.00	1609.3
NP=4 5th order	1.E-7	S1	251	0.15E-4	247	0	0.51	52.1
		S2	223	0.68E-4	222	0	1.40	53.5
		S3	277	0.41E-5	271	4	0.42	51.4
		S4	288	0.21E-4	272	25	0.61	61.5
	1.E-9	S1	501	0.19E-7	495	0	0.50	64.8
		S2	446	0.17E-6	442	0	1.20	65.9
		S3	590	0.21E-6	578	12	0.38	88.3
		S4	497	0.75E-7	481	20	0.89	69.7
	1.E-11	S1	1050	0.18E-8	1042	0	0.50	120.0
		S2	933	0.31E-8	927	0	1.10	109.7
		S3	1280	0.50E-8	1247	49	0.38	154.1
		S4	939	0.28E-8	931	0	1.00	109.9
NP=6 7th order	1.E-7	S1	151	0.23E-4	143	9	0.49	32.5
		S2	126	0.10E-3	125	0	1.70	31.6
		S3	155	0.16E-4	149	6	0.27	32.2
		S4	178	0.22E-4	167	20	0.42	38.3
	1.E-9	S1	232	0.22E-6	221	16	0.49	34.8
		S2	193	0.19E-5	191	1	2.95	33.7
		S3	243	0.12E-6	235	9	0.25	35.1
		S4	243	0.30E-6	230	20	0.59	37.3
	1.E-11	S1	336	0.16E-8	330	0	0.51	38.2
		S2	307	0.11E-7	304	0	1.80	38.6
		S3	402	0.60E-9	392	9	0.21	43.5
		S4	409	0.17E-8	383	40	0.48	46.6

Table 12 : PPC Methods with Problem TP2

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-8	S1	587	0.51E-2	580	0	0.48	256.0
		S2	496	0.77E-2	492	0	0.96	234.4
		S3	601	0.65E-2	571	46	0.68	274.9
		S4	491	0.77E-2	487	0	1.00	232.4
	1.E-10	S1	1815	0.20E-3	1806	0	0.49	492.0
		S2	1523	0.35E-3	1516	0	0.99	440.0
		S3	1980	0.29E-3	1847	247	0.70	559.4
		S4	1522	0.34E-3	1513	0	1.00	439.3
	1.E-13	S1	10157	0.12E-5	10140	0	0.50	1713 [illegible]
		S2	8540	0.20E-5	8526	0	1.00	1497.2
		S3	11541	0.17E-5	10698	1653	0.67	1996.9
		S4	8538	0.20E-5	8521	0	1.00	1497.0
NP=4 5th order	1.E-8	S1	147	0.50E-4	142	3	0.37	34.2
		S2	123	0.62E-4	122	0	0.89	29.2
		S3	236	0.16E-4	209	46	0.25	49.1
		S4	132	0.13E-3	126	4	0.86	33.9
	1.E-10	S1	300	0.73E-6	293	2	0.43	48.9
		S2	261	0.89E-6	257	0	0.92	43.1
		S3	458	0.13E-5	416	73	0.29	77.6
		S4	266	0.14E-5	259	3	0.95	45.5
	1.E-13	S1	916	0.64E-9	908	0	0.48	99.6
		S2	814	0.30E-9	808	1	1.00	85.5
		S3	1131	0.11E-8	1030	185	0.61	126.0
		S4	811	0.24E-8	803	0	1.00	94.0
NP=6 7th order	1.E-8	S1	91	0.50E-4	86	8	0.30	21.2
		S2	74	0.13E-3	73	0	0.83	19.1
		S3	144	0.89E-4	139	8	0.10	35.6
		S4	96	0.61E-4	89	11	0.41	22.8
	1.E-10	S1	136	0.30E-6	130	5	0.34	20.9
		S2	113	0.48E-6	112	0	2.90	17.9
		S3	242	0.77E-6	230	17	0.10	39.6
		S4	142	0.11E-5	133	12	0.58	23.8
	1.E-13	S1	288	0.50E-9	280	4	0.42	31.0
		S2	256	0.12E-7	252	1	1.70	32.3
		S3	584	0.49E-9	560	37	0.15	62.8
		S4	283	0.97E-9	271	13	0.83	31.4

Table 13 : PPC Methods with Problem TP3

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-4	S1	308	0.71E-3	285	42	0.38	97.8
		S2	194	0.21E-2	192	0	1.50	72.6
		S3	358	0.70E-3	326	60	0.31	113.4
		S4	275	0.11E-2	259	29	0.54	92.4
	1.E-7	S1	1317	0.38E-5	1305	15	0.49	242.7
		S2	1081	0.64E-5	1079	0	1.10	208.3
		S3	1795	0.44E-5	1623	335	0.43	335.3
		S4	1128	0.64E-5	1116	16	0.95	217.1
	1.E-9	S1	4076	0.12E-6	4067	0	0.50	587.9
		S2	3425	0.20E-6	3418	0	1.00	510.7
		S3	4697	0.16E-6	4339	698	0.66	692.2
		S4	3431	0.20E-6	3421	1	1.00	511.8
NP=4 5th order	1.E-4	S1	175	0.19E-3	157	33	0.37	47.2
		S2	123	0.17E-2	122	0	2.30	44.4
		S3	211	0.30E-4	206	7	0.02	46.7
		S4	158	0.28E-3	149	16	0.47	44.4
	1.E-7	S1	391	0.24E-6	365	50	0.39	59.1
		S2	270	0.10E-5	269	0	1.70	45.0
		S3	439	0.11E-6	434	7	0.14	63.1
		S4	354	0.48E-6	338	30	0.59	56.0
	1.E-9	S1	721	0.28E-8	699	36	0.46	84.3
		S2	584	0.71E-8	583	0	1.20	71.7
		S3	992	0.32E-8	913	151	0.31	116.7
		S4	675	0.55E-8	655	32	0.78	81.7
NP=6 7th order	1.E-4	S1	187	0.70E-4	166	40	0.38	45.0
		S2	128	0.59E-2	126	1	3.50	57.4
		S3	148	0.36E-4	144	5	0.04	33.3
		S4	165	0.26E-3	154	20	0.30	46.0
	1.E-7	S1	264	0.20E-6	230	66	0.31	39.4
		S2	155	0.11E-5	154	0	2.30	26.0
		S3	248	0.38E-7	241	11	0.05	33.4
		S4	233	0.26E-6	218	27	0.36	35.4
	1.E-9	S1	383	0.22E-8	342	80	0.35	44.2
		S2	235	0.15E-7	234	0	2.10	30.0
		S3	420	0.10E-8	413	11	0.06	46.7
		S4	350	0.29E-8	328	41	0.40	41.0

Table 14 : PPC Methods with Problem TP4

METHOD	λ	STRATEGY	NRHS	ERR	NSS	NFS	Avg R	METRIC
NP=2 3rd order	1.E-8	S1	5552	0.90E-2	5550	0	0.50	2713.0
		S2	4670	0.15E-1	4668	0	1.00	2558.1
		S3	5755	0.13E-1	5508	490	0.68	3030.2
		S4	4664	0.15E-1	4662	0	1.00	2576.9
	1.E-10	S1	17533	0.29E-3	17528	0	0.50	4959.2
		S2	14742	0.48E-3	14740	0	1.00	4447.5
		S3	20518	0.42E-3	18912	3204	0.65	6075.2
		S4	14739	0.49E-3	14734	0	1.00	4458.1
	1.E-13	S1	98540	0.18E-5	98531	0	0.50	17126.0
		S2	82860	0.28E-5	82853	0	1.00	14938.1
		S3	115066	0.24E-5	106120	17874	0.65	20504.1
		S4	82859	0.29E-5	82850	0	1.00	14952.6
NP=4 5th order	1.E-8	S1	1271	0.46E-4	1266	8	0.51	293.0
		S2	1121	0.11E-3	1120	0	1.00	283.0
		S3	1931	0.76E-4	1872	115	0.17	468.8
		S4	1140	0.62E-4	1134	10	0.98	270.9
	1.E-10	S1	2703	0.77E-6	2701	2	0.50	442.1
		S2	2406	0.16E-5	2405	0	1.00	415.9
		S3	4016	0.52E-6	3950	130	0.17	639.0
		S4	2403	0.13E-5	2401	1	1.00	407.3
	1.E-13	S1	8520	0.12E-7	8516	0	0.50	1077.8
		S2	7587	0.14E-7	7586	0	1.00	964.1
		S3	12193	0.16E-7	12095	188	0.19	1566.6
		S4	7585	0.13E-7	7582	0	1.00	962.2
NP=6 7th order	1.E-8	S1	890	0.24E-4	802	174	0.34	192.4
		S2	557	0.73E-4	556	0	1.60	134.6
		S3	1164	0.15E-4	1099	127	0.11	238.7
		S4	677	0.53E-4	650	51	0.67	158.4
	1.E-10	S1	1192	0.82E-7	1154	74	0.47	168.2
		S2	985	0.10E-5	984	0	1.20	164.2
		S3	1969	0.12E-7	1908	120	0.11	268.5
		S4	1127	0.16E-6	1093	65	0.79	165.6
	1.E-13	S1	2536	0.30E-8	2534	2	0.50	297.4
		S2	2323	0.26E-8	2322	0	1.00	270.7
		S3	7325	0.71E-8	7127	394	0.15	898.9
		S4	2734	0.30E-8	2640	186	0.74	321.0

Table 15 : PPC Methods with Problem TP5

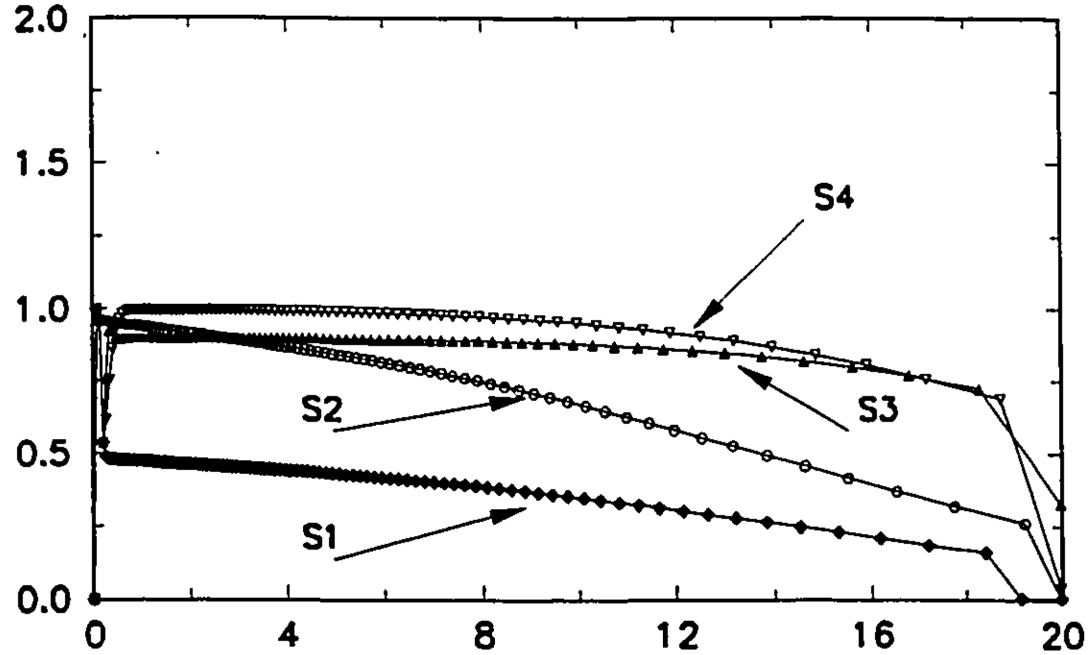


Figure 1 : R-trajectories for RKF case (Problem TP1, $\lambda = 10^{-9}$)

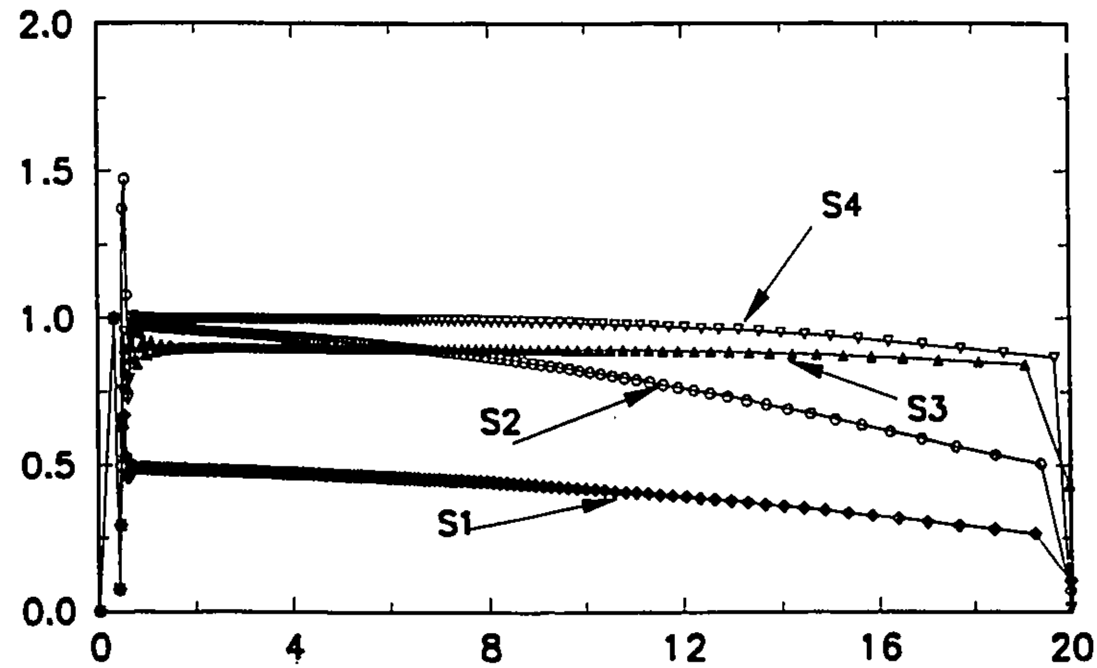


Figure 2 : R-trajectories for PC case (Problem TP1, $\lambda = 10^{-9}$)

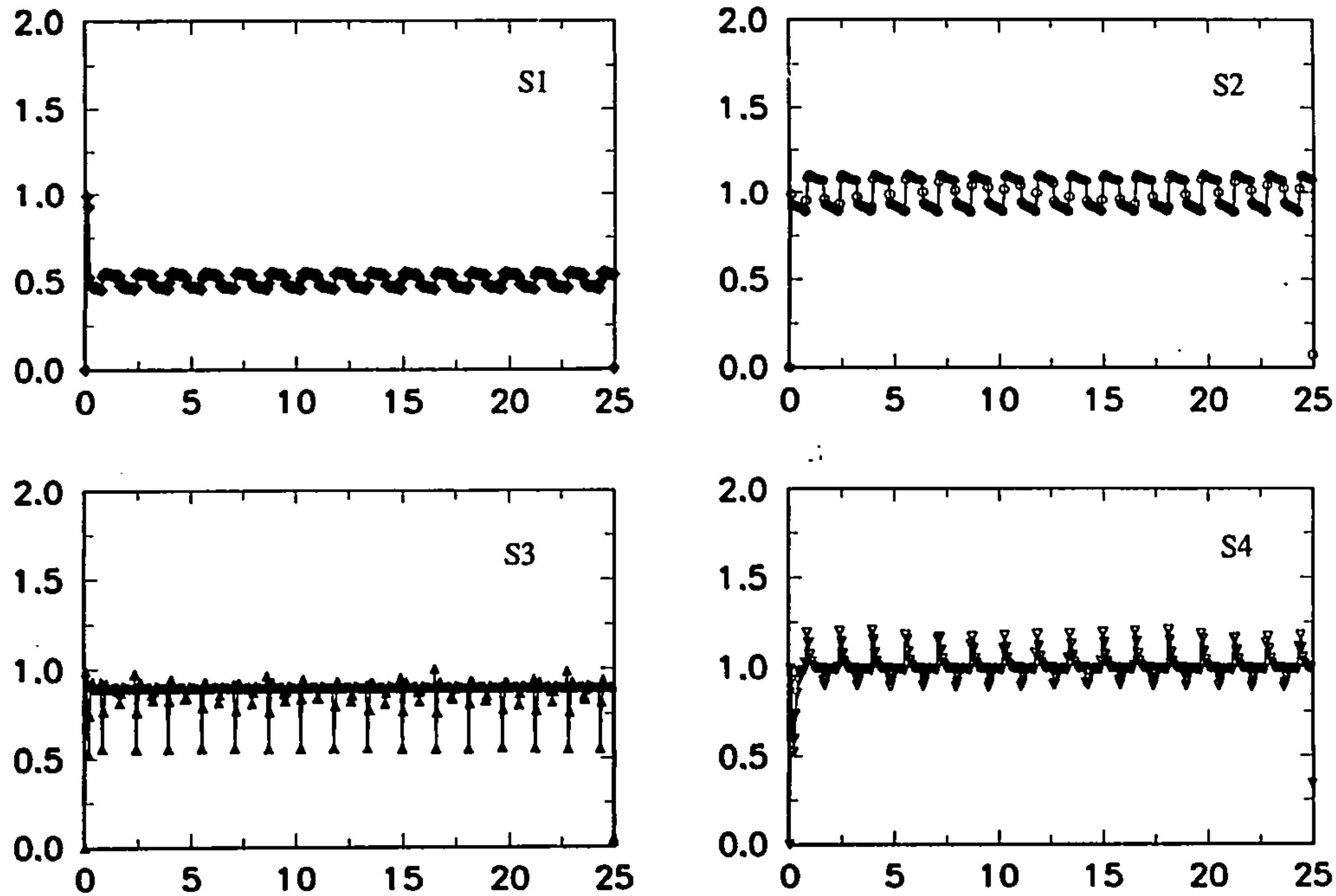


Figure 3 : R-trajectories for RKF case (Problem TP2, $\lambda = 10^{-9}$)

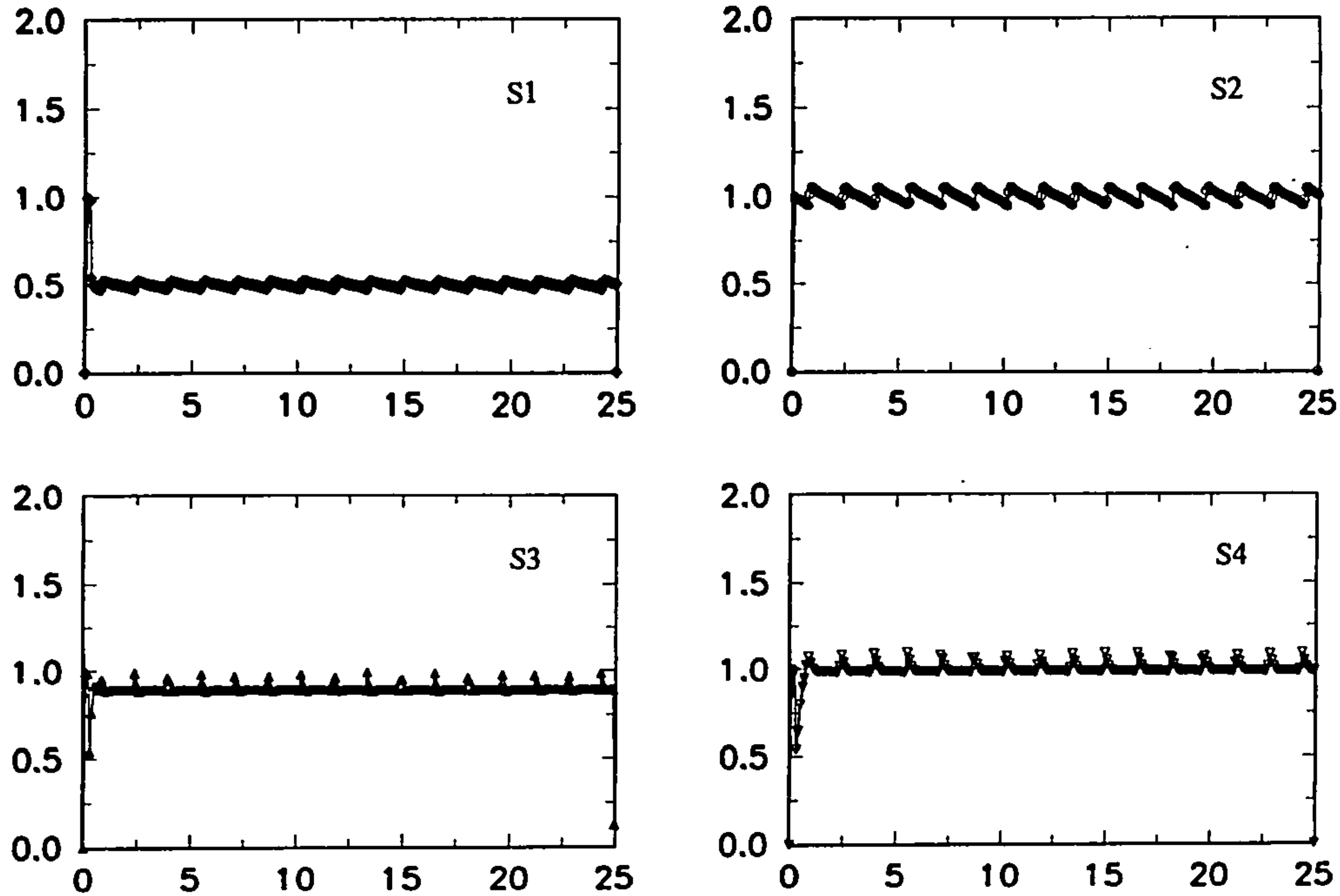


Figure 4 : R-trajectories for RKV case (Problem TP2, $\lambda = 10^{-9}$)

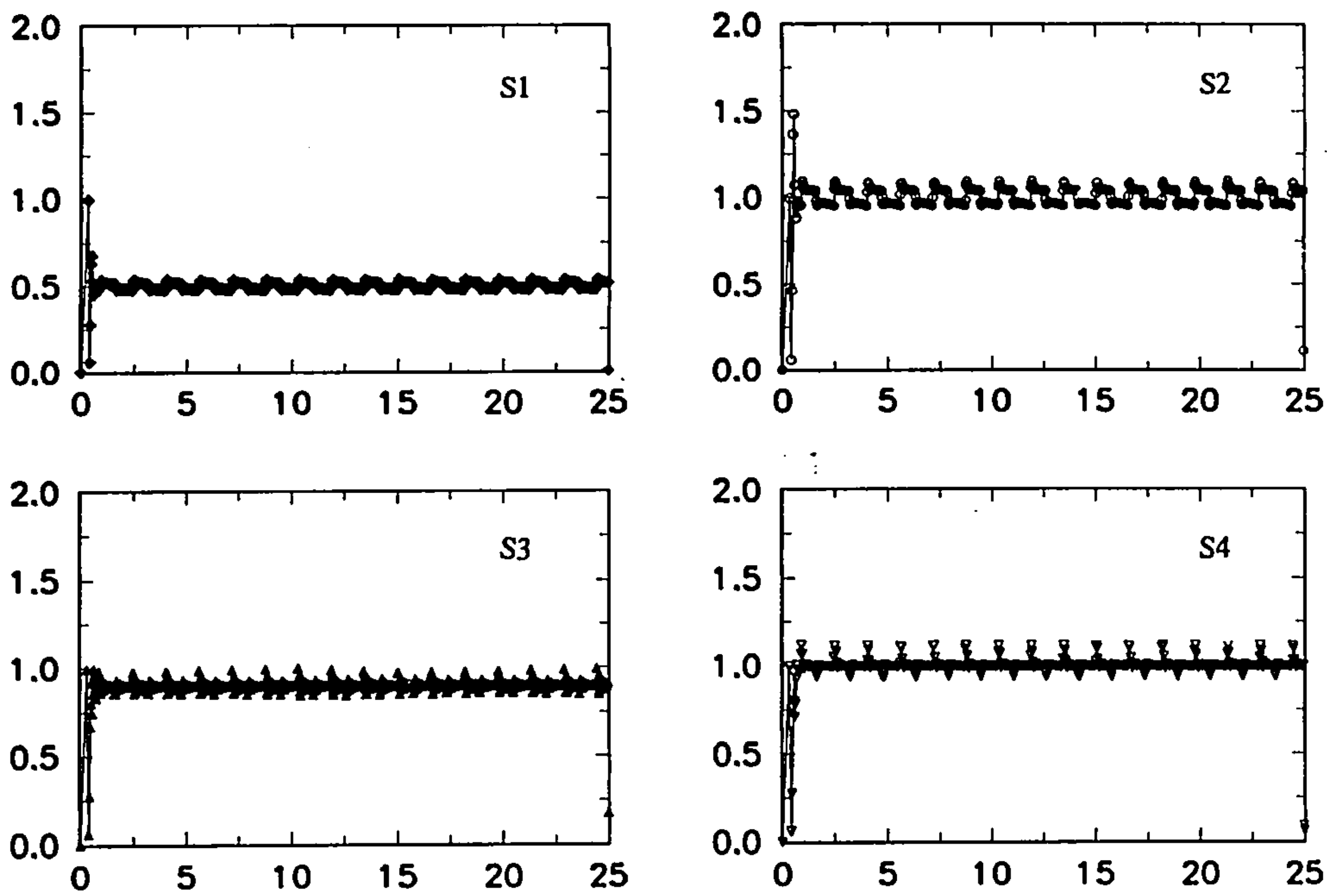


Figure 5 : R-trajectories for PC case (Problem TP2, $\lambda = 10^{-9}$)

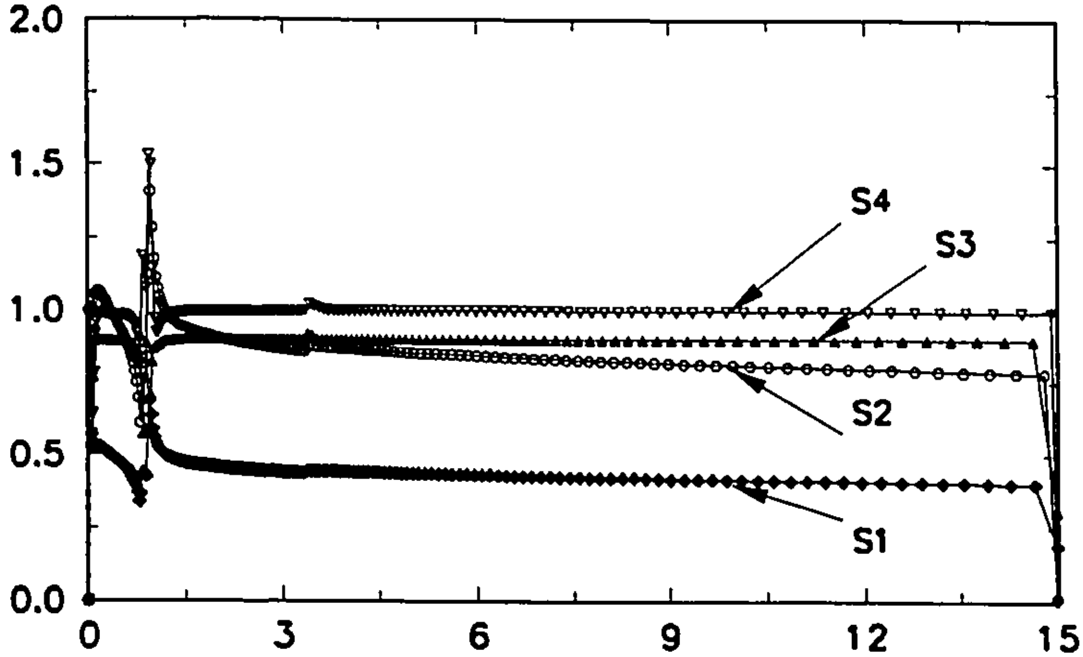


Figure 6 : R-trajectories for RKF case (Problem TP3, $\lambda = 10^{-10}$)

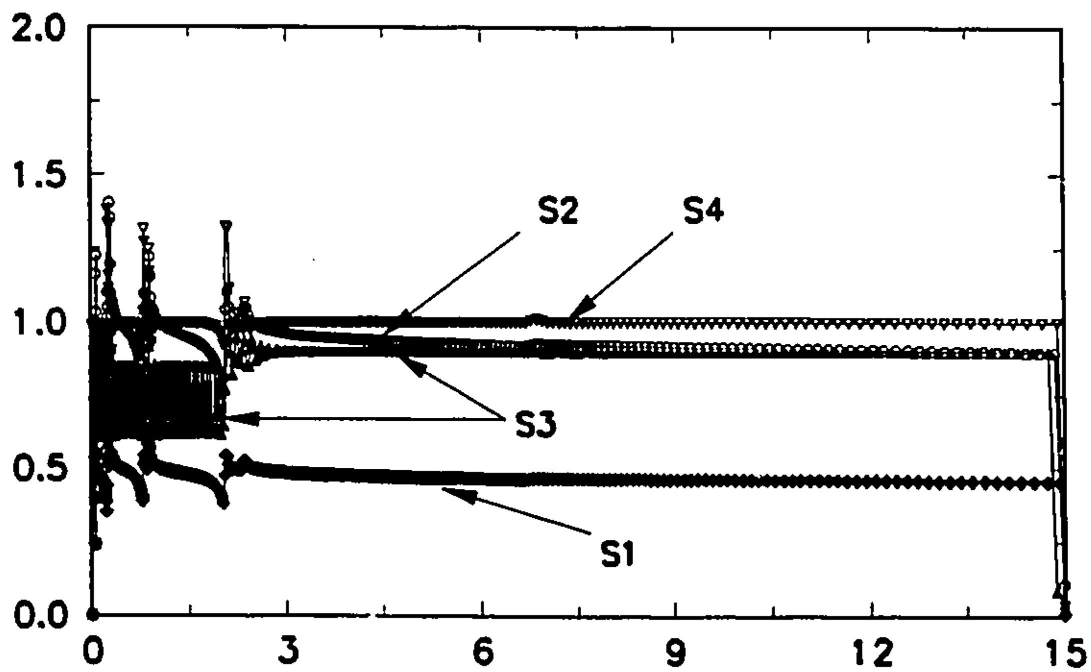


Figure 7 : R-trajectories for PC case (Problem TP3, $\lambda = 10^{-10}$)

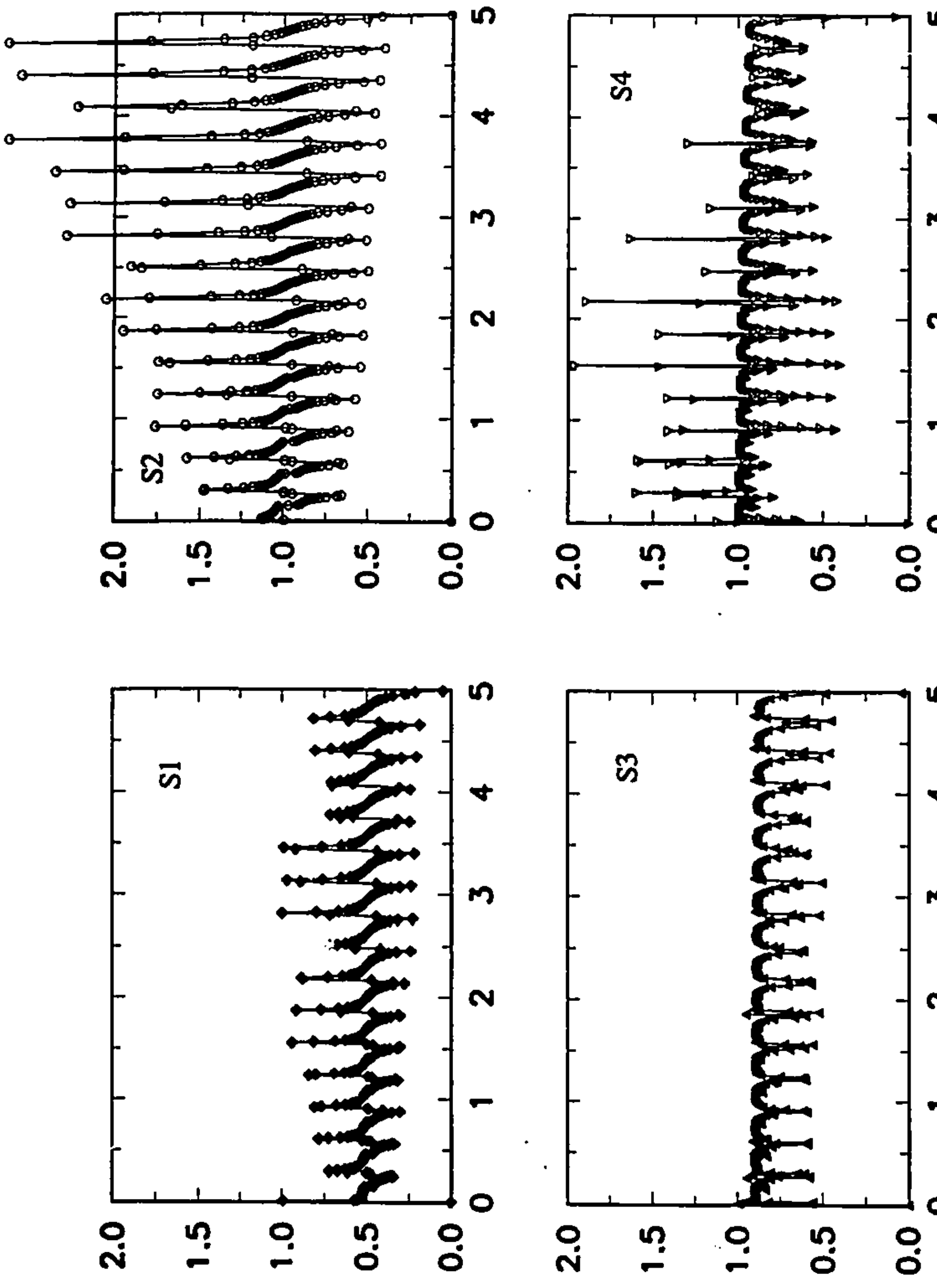


Figure 8 : R-trajectories for RKF case (Problem TP4, $\lambda = 10^{-9}$)

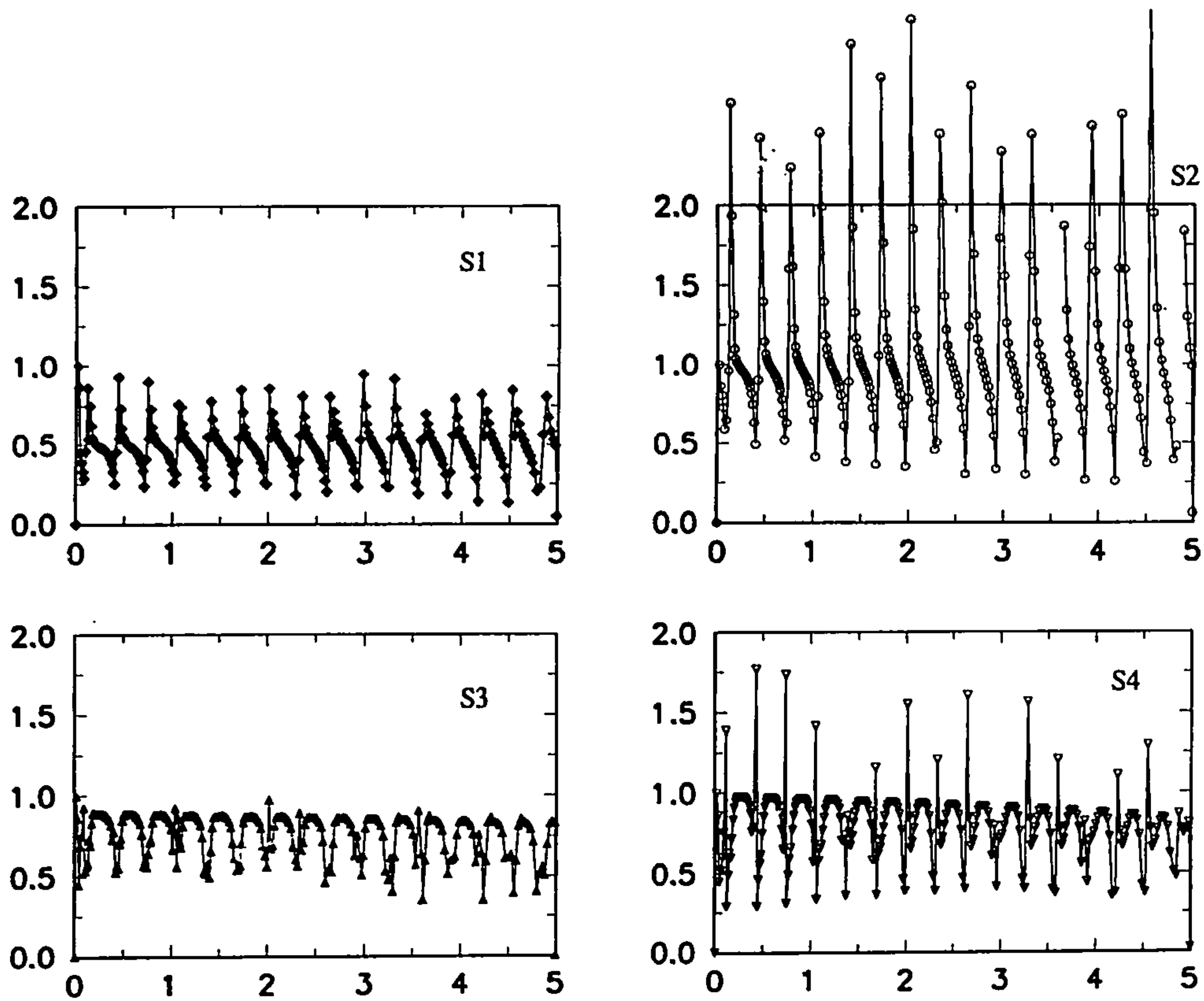


Figure 9 : R-trajectories for RKV case (Problem TP4, $\lambda = 10^{-9}$)

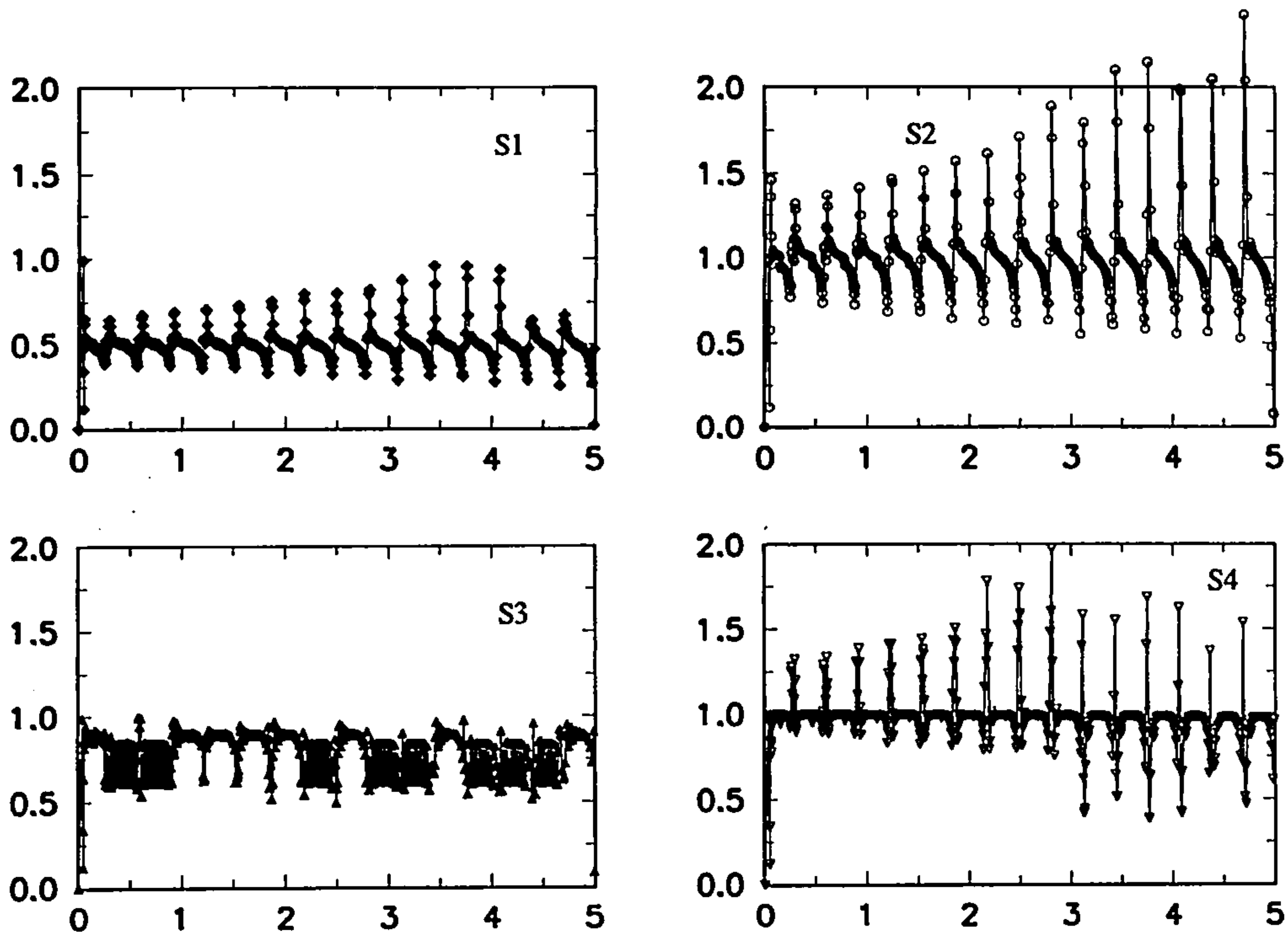


Figure 10 : R-trajectories for PC case (Problem TP4, $\lambda = 10^{-9}$)

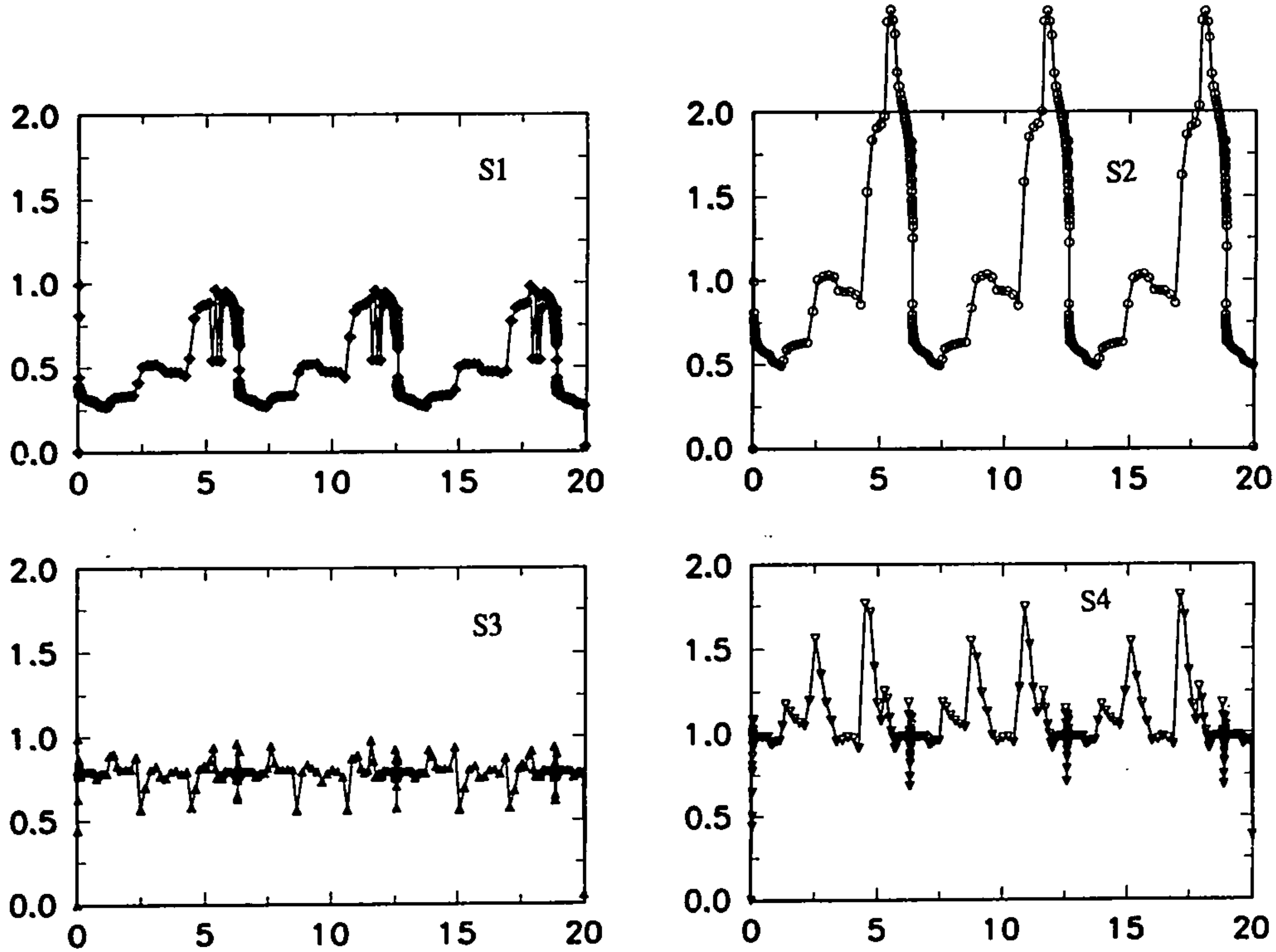


Figure 11 : R-trajectories for RKF case (Problem TP5, $\lambda = 10^{-8}$)

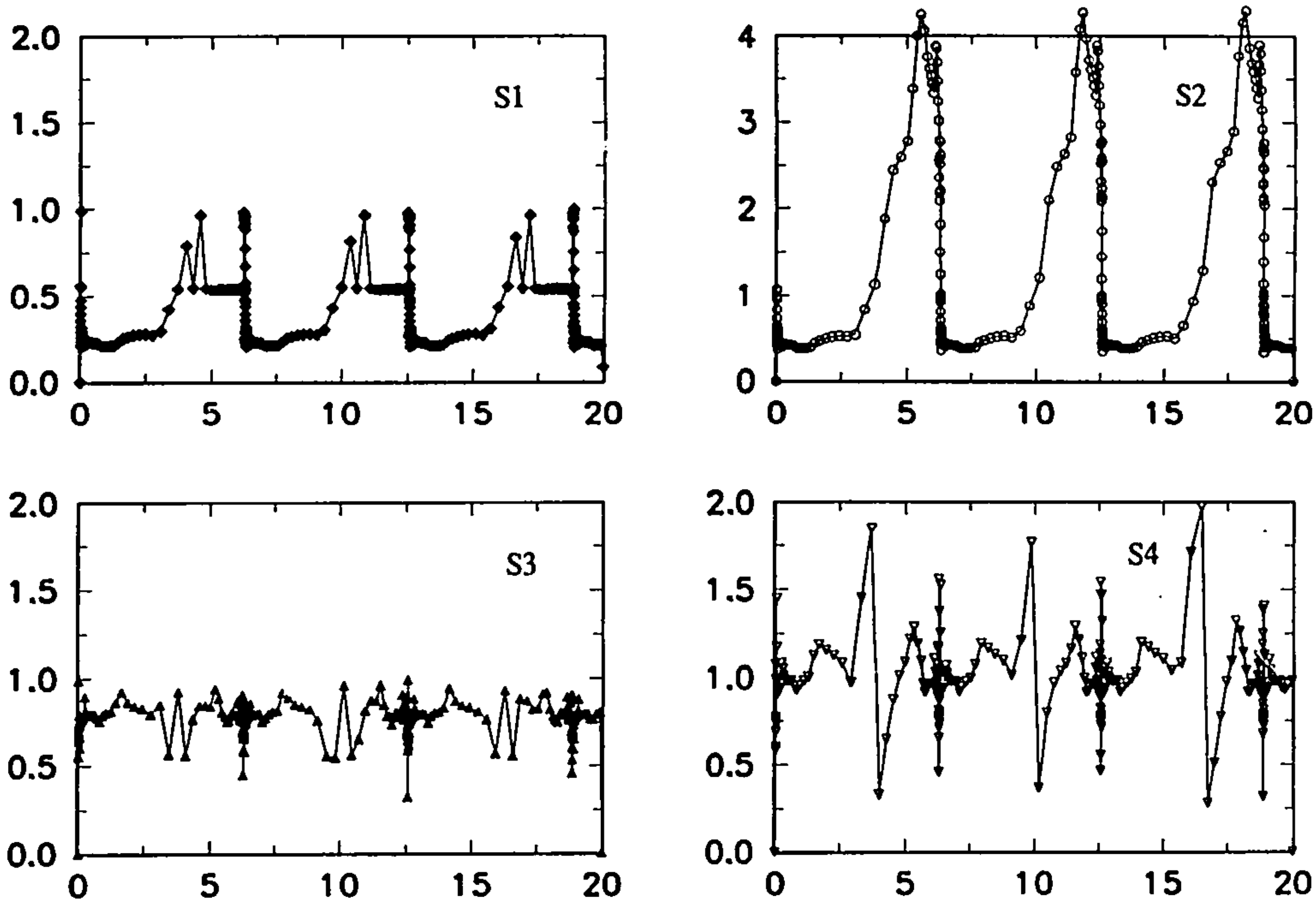


Figure 12 : R-trajectories for RKV case (Problem TP5, $\lambda = 10^{-8}$)

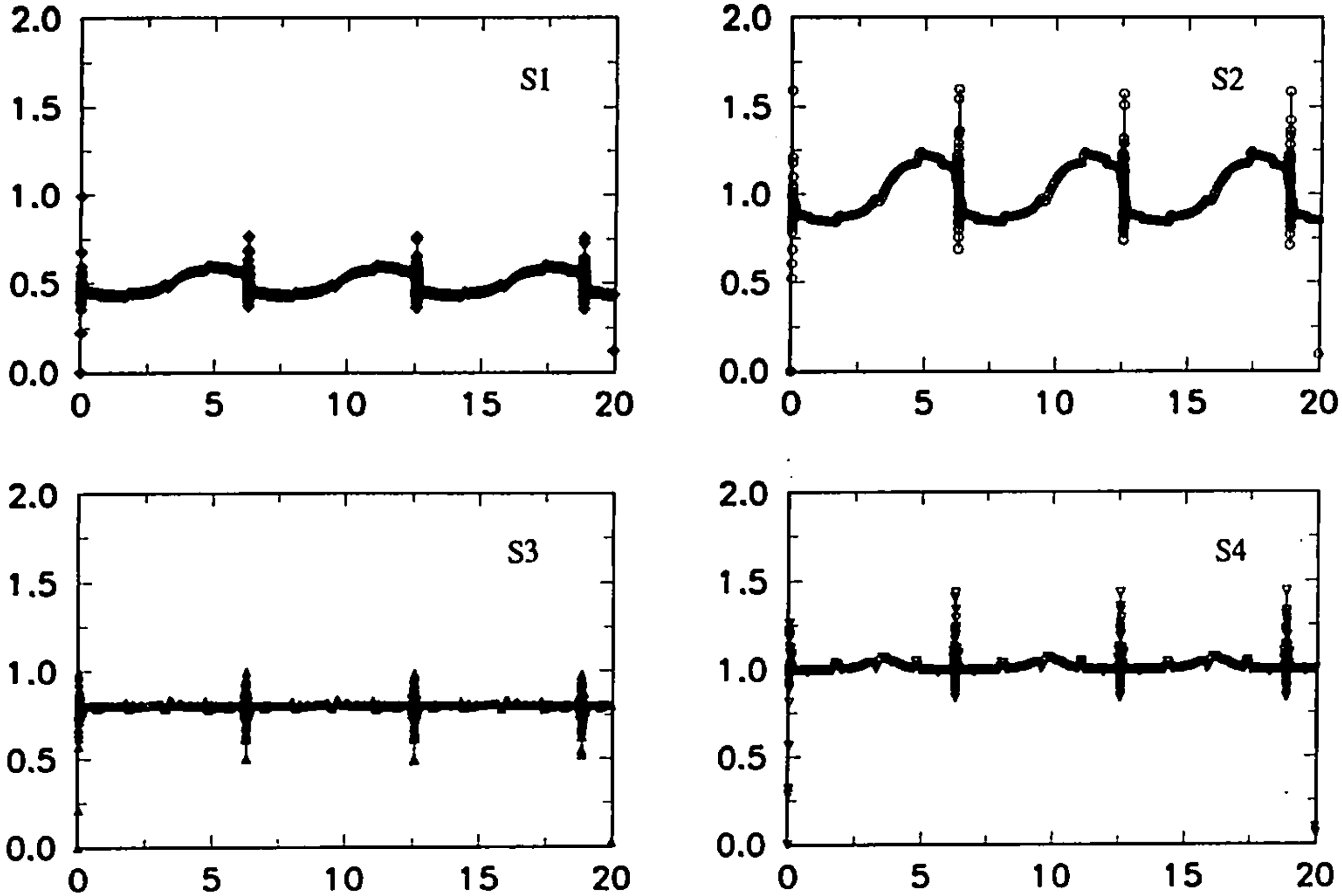


Figure 13 : R-trajectories for PC case (Problem TP5, $\lambda = 10^{-8}$)

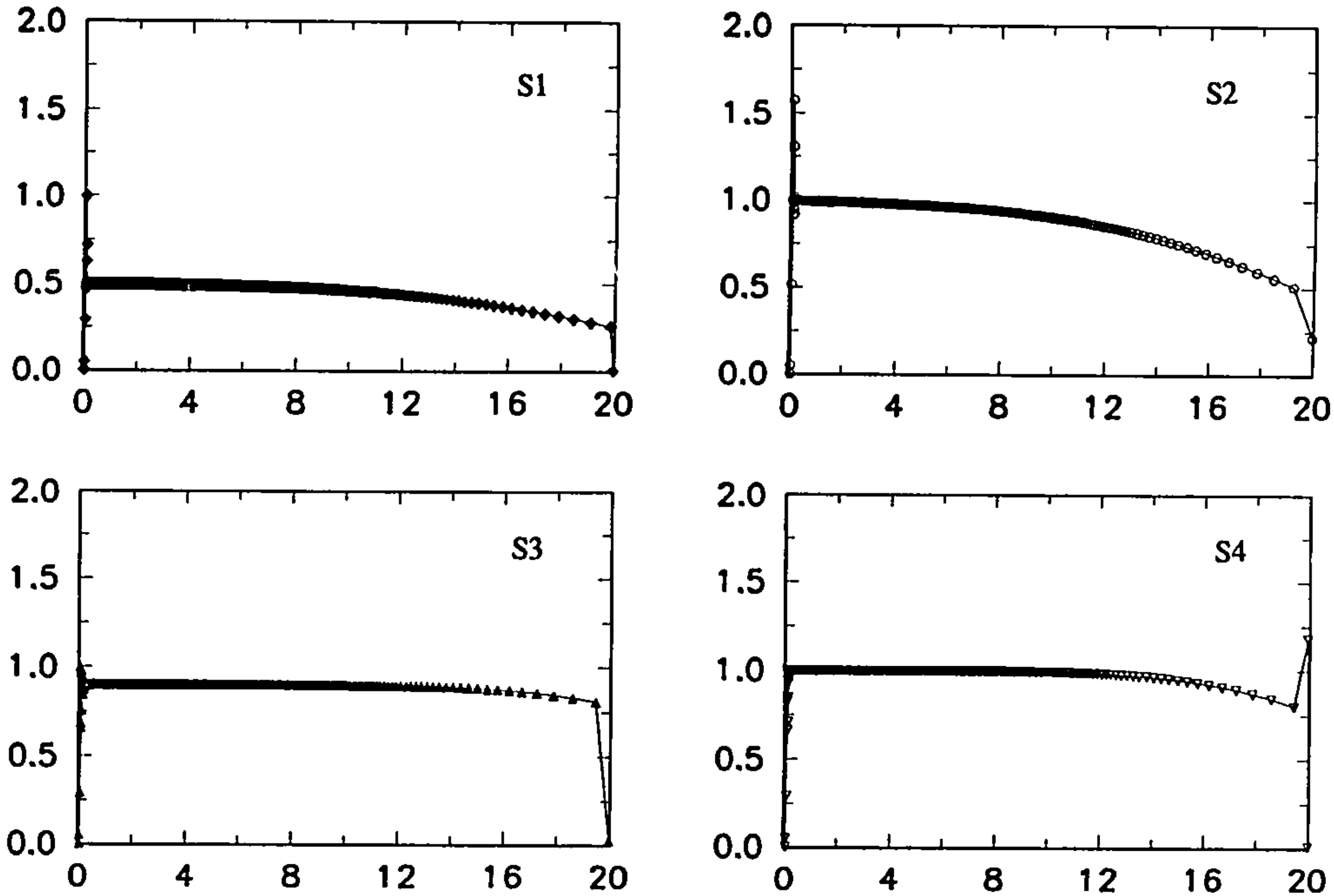


Figure 14 : R-trajectories for BPC case (Problem TP1, $\lambda = 10^{-9}$, NP=2, 3rd order)

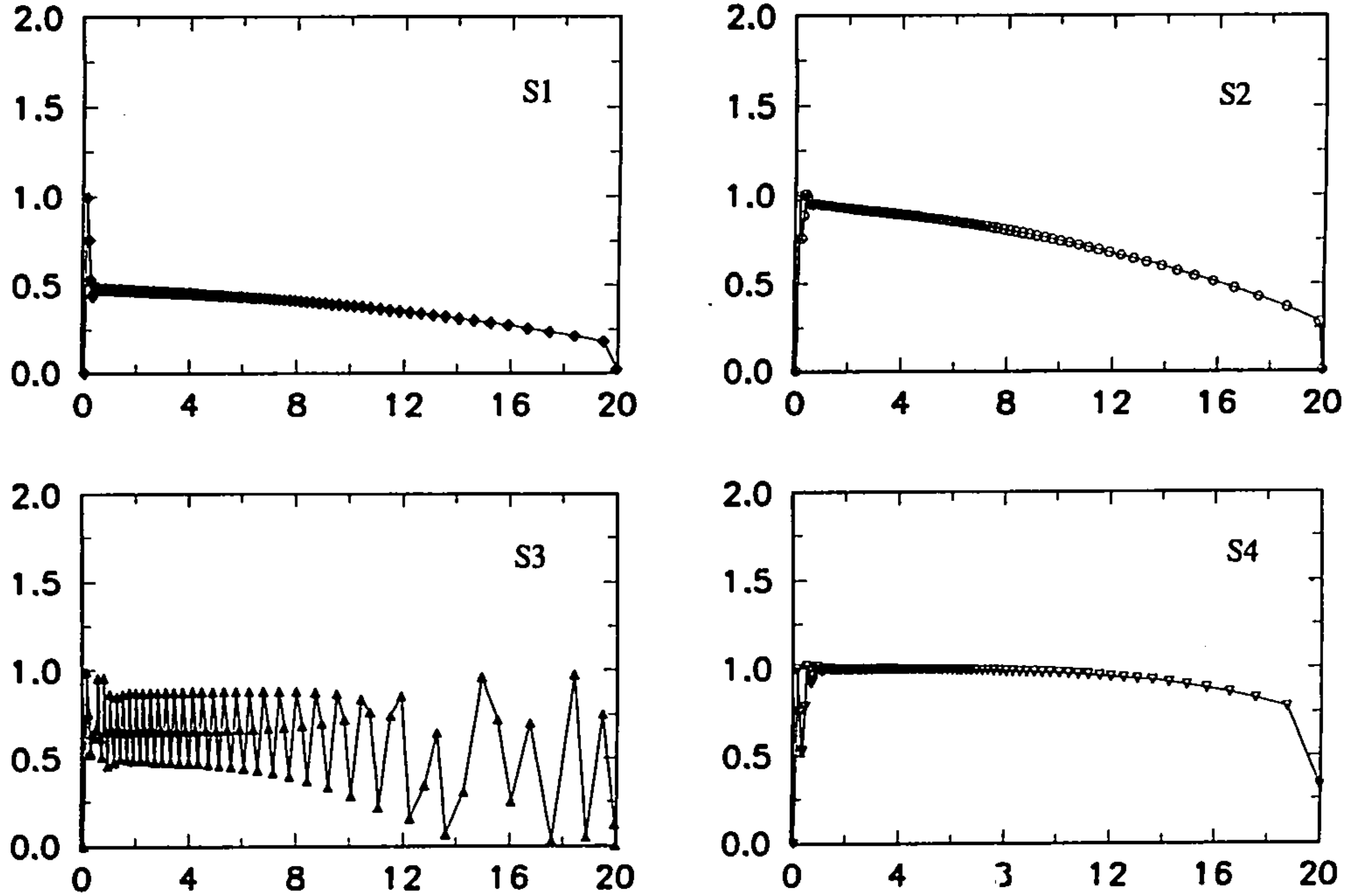


Figure 15 : R-trajectories for BPC case (Problem TP1, $\lambda = 10^{-9}$ NP=4, 5th order)

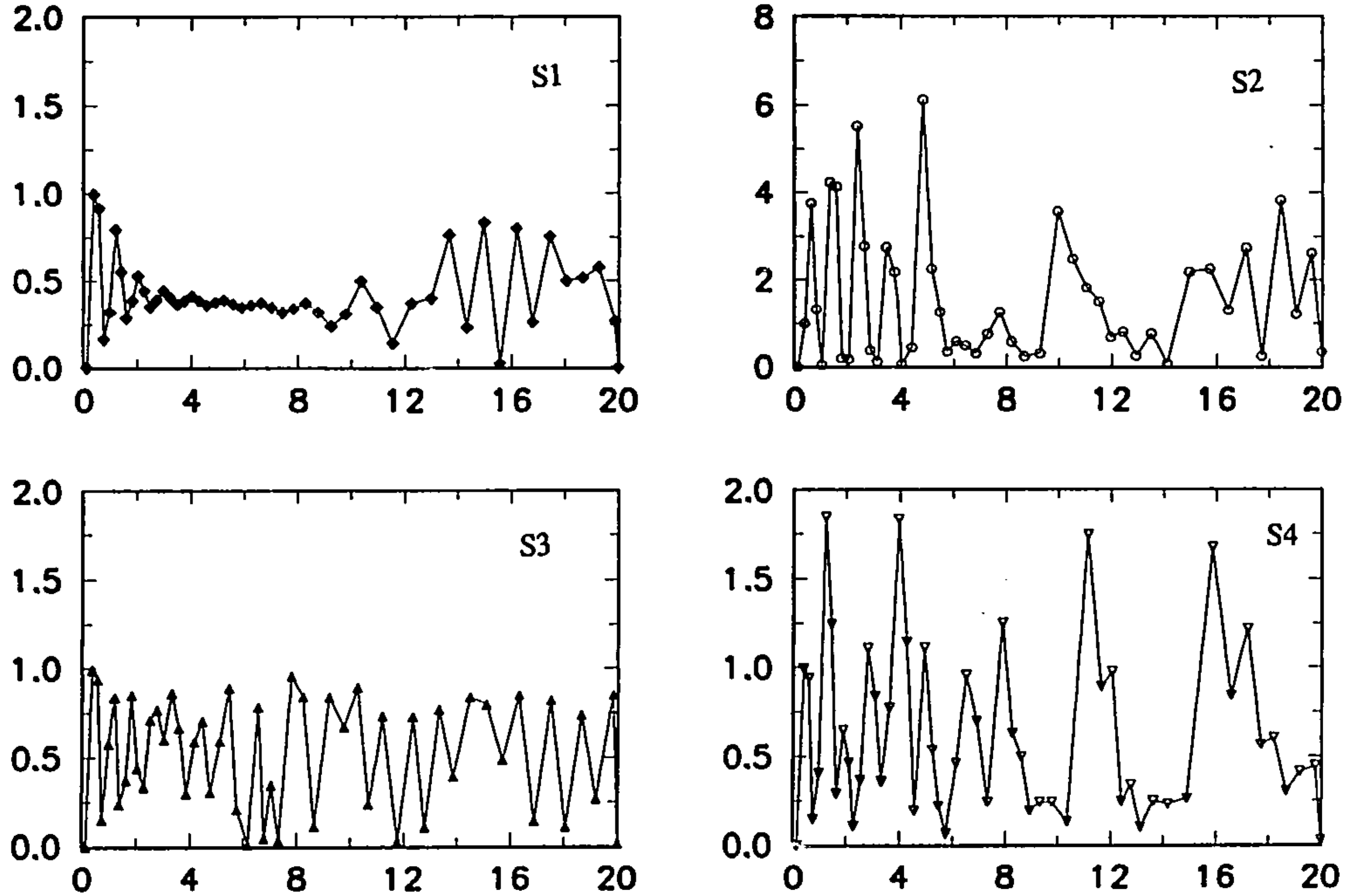


Figure 16 : R-trajectories for BPC case (Problem TP1, $\lambda = 10^{-9}$ NP=6, 7th order)

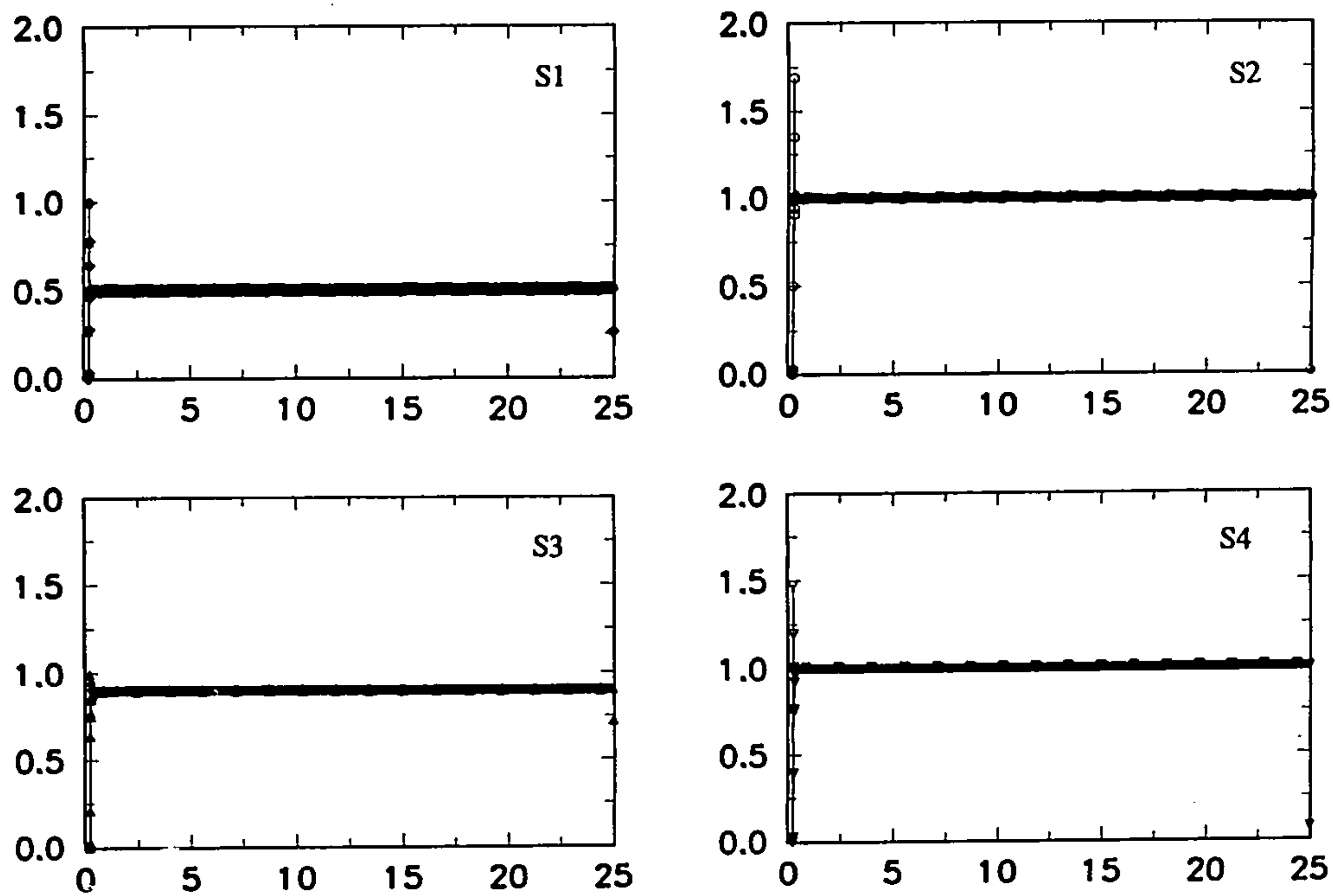


Figure 17 : R-trajectories for BPC case (Problem TP2, $\lambda = 10^{-9}$, NP=2, 3rd order)

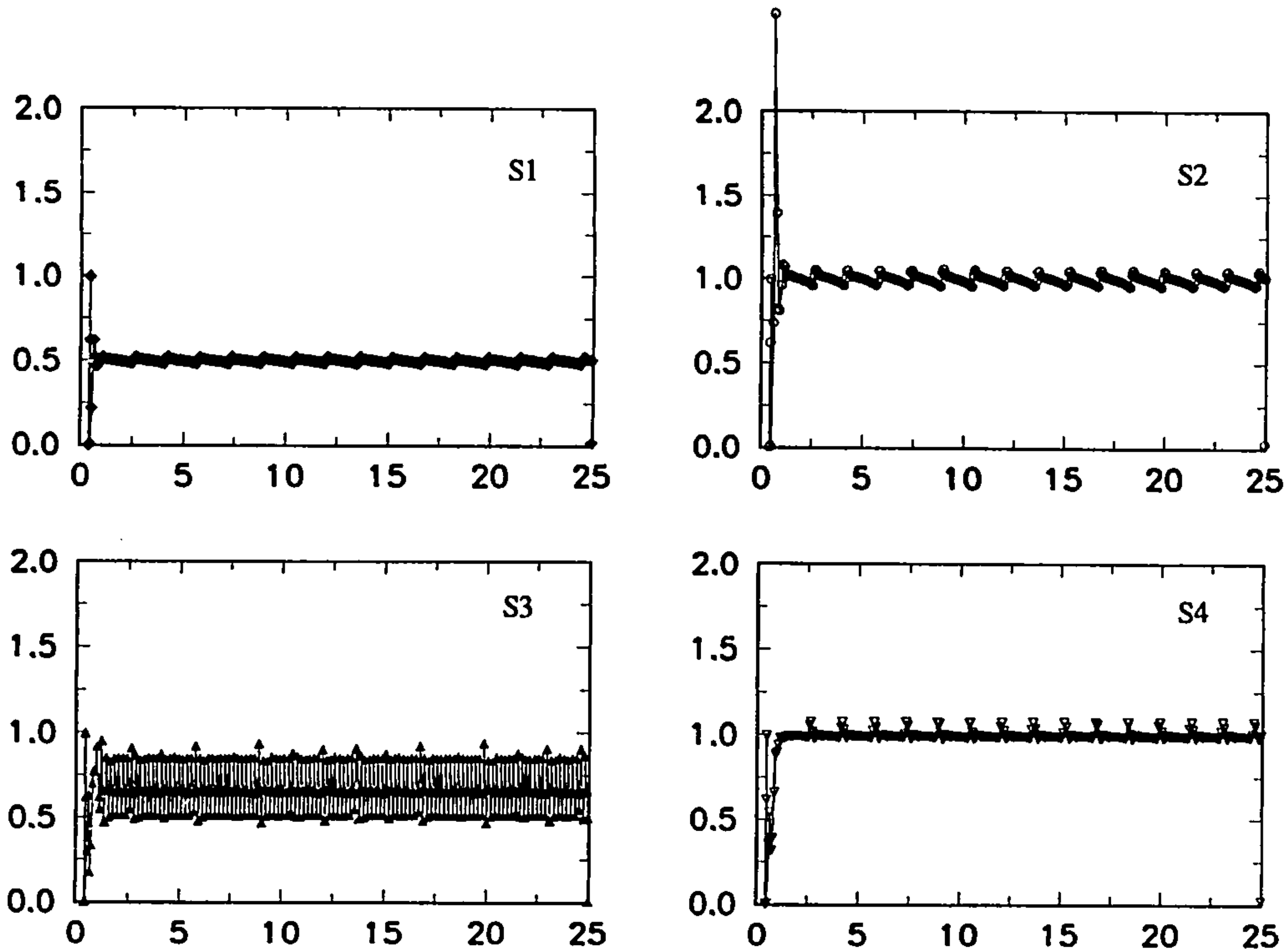


Figure 18 : R-trajectories for BPC case (Problem TP2, $\lambda = 10^{-9}$ NP=4, 5th order)

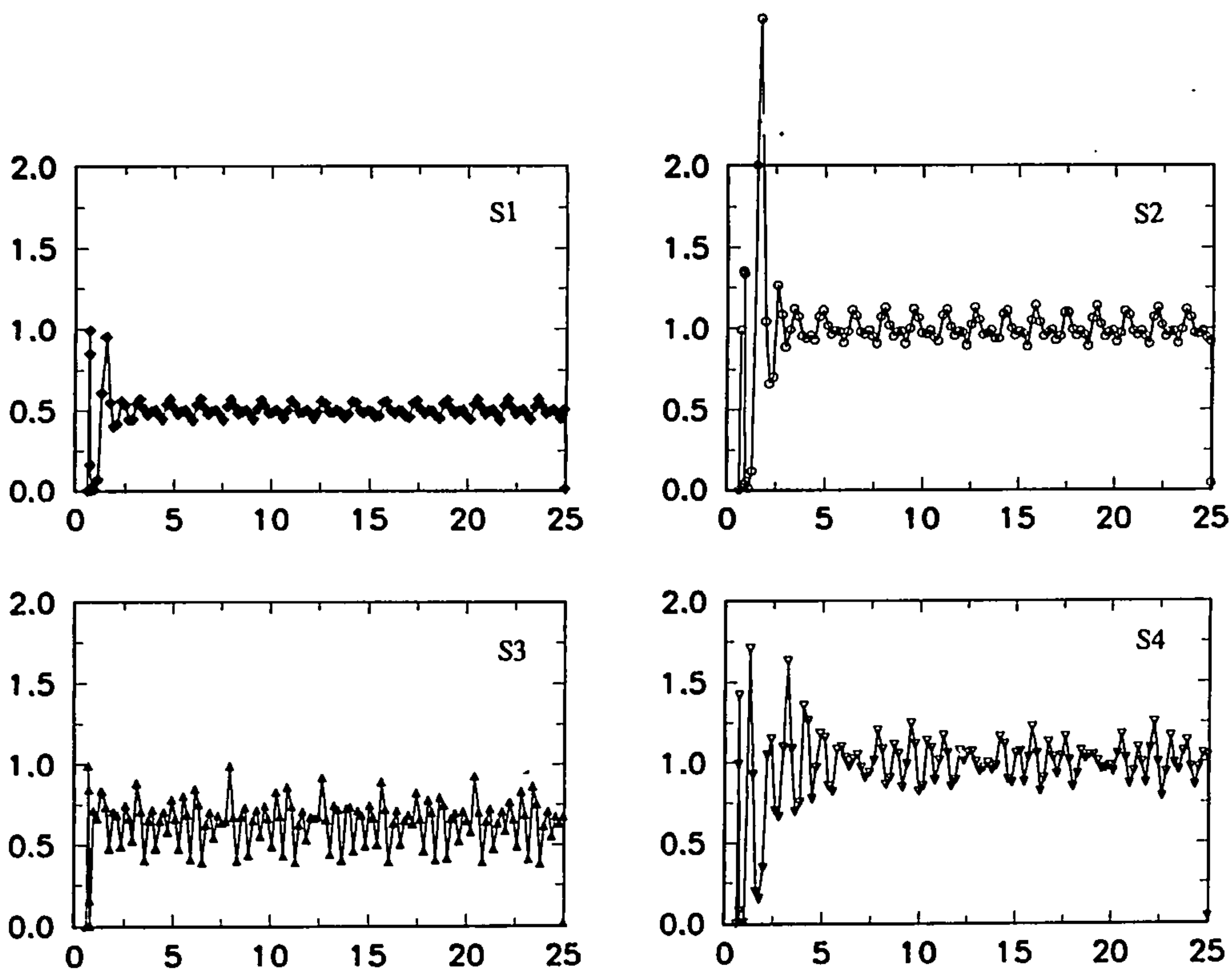


Figure 19 : R-trajectories for BPC case (Problem TP2, $\lambda = 10^{-9}$ NP=6, 7th order)

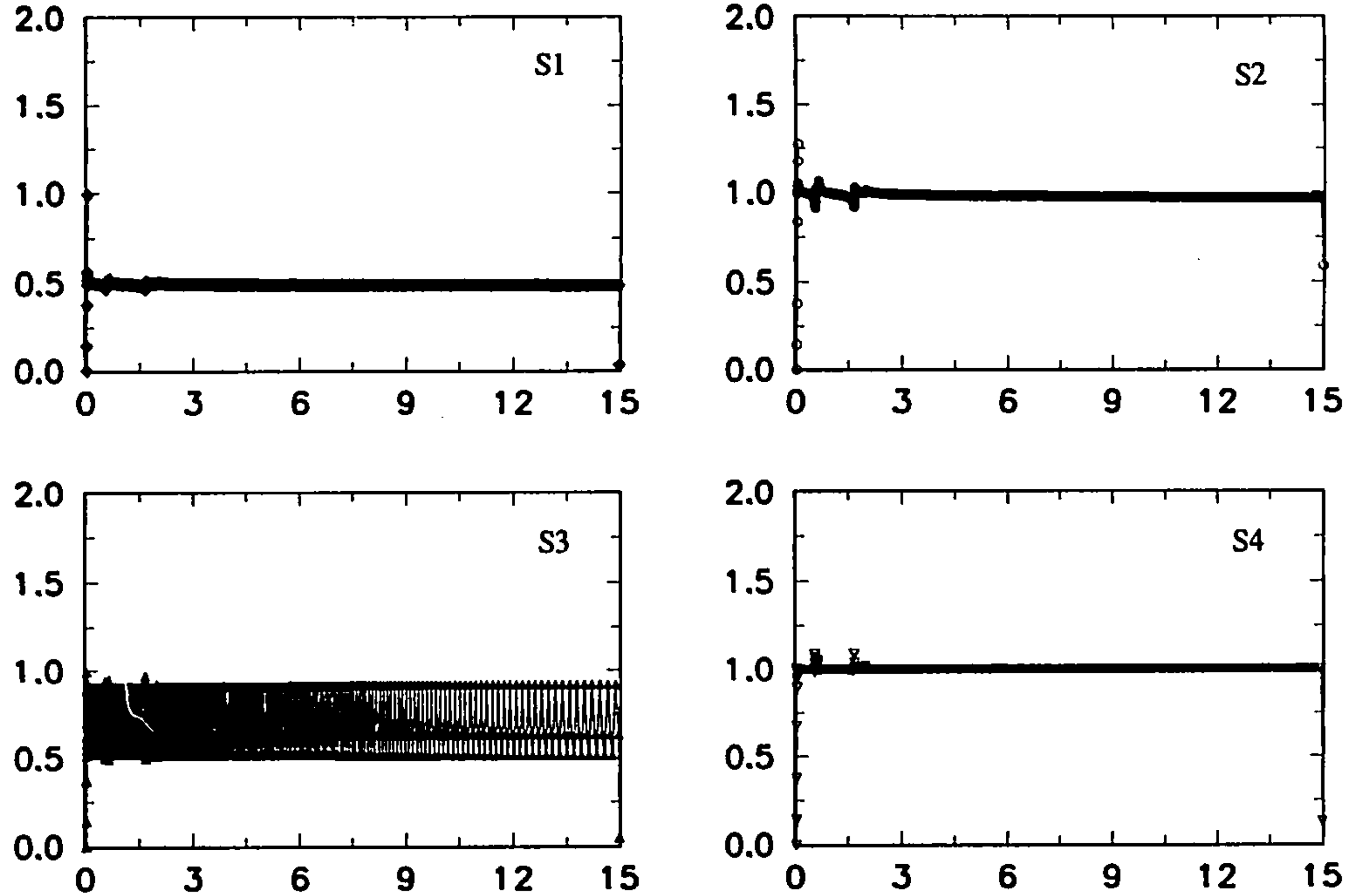


Figure 20 : R-trajectories for BPC case (Problem TP3, $\lambda = 10^{-10}$ NP=2, 3rd order)

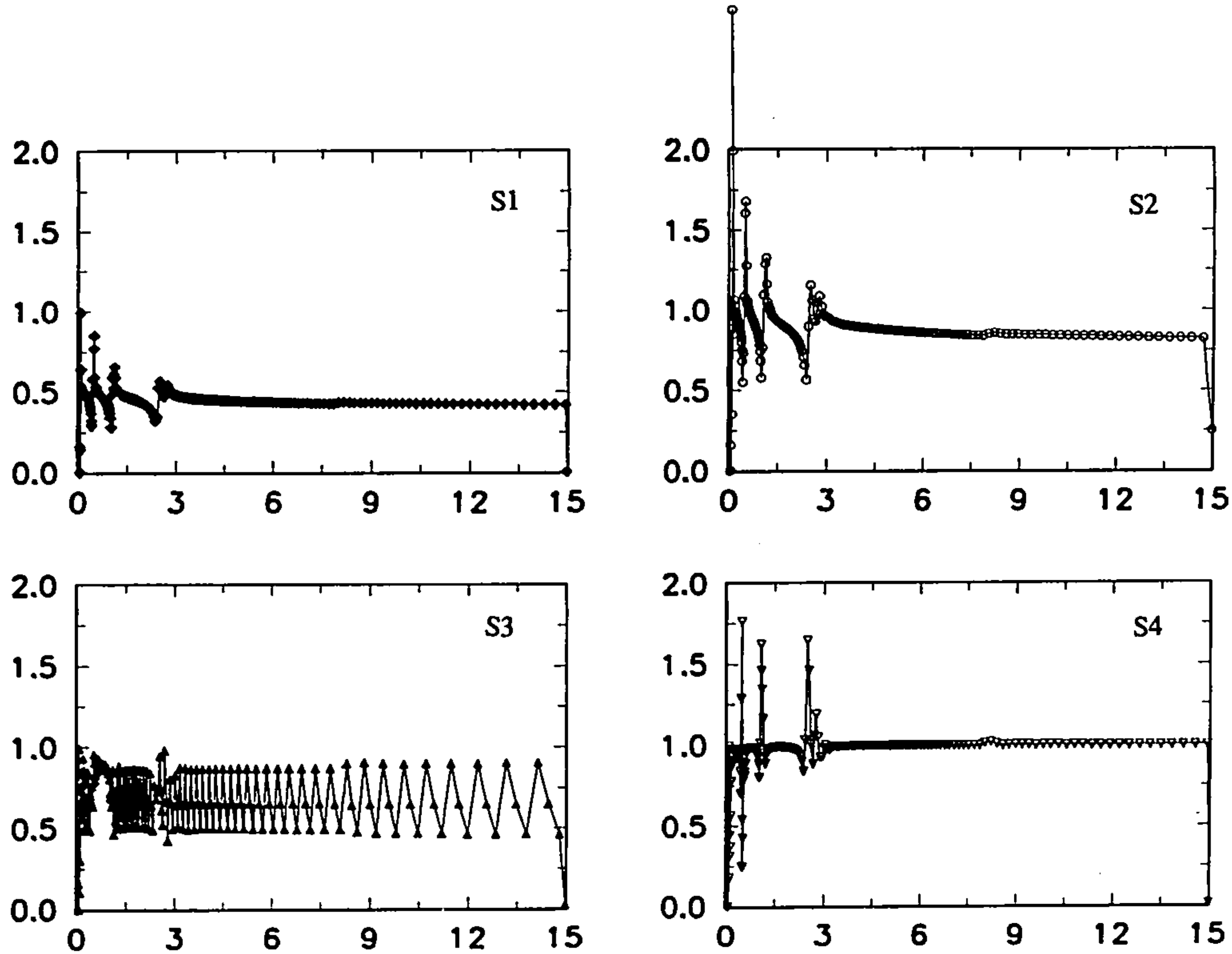


Figure 21 : R-trajectories for BPC case (Problem TP3, $\lambda = 10^{-10}$ NP=4, 5th order)

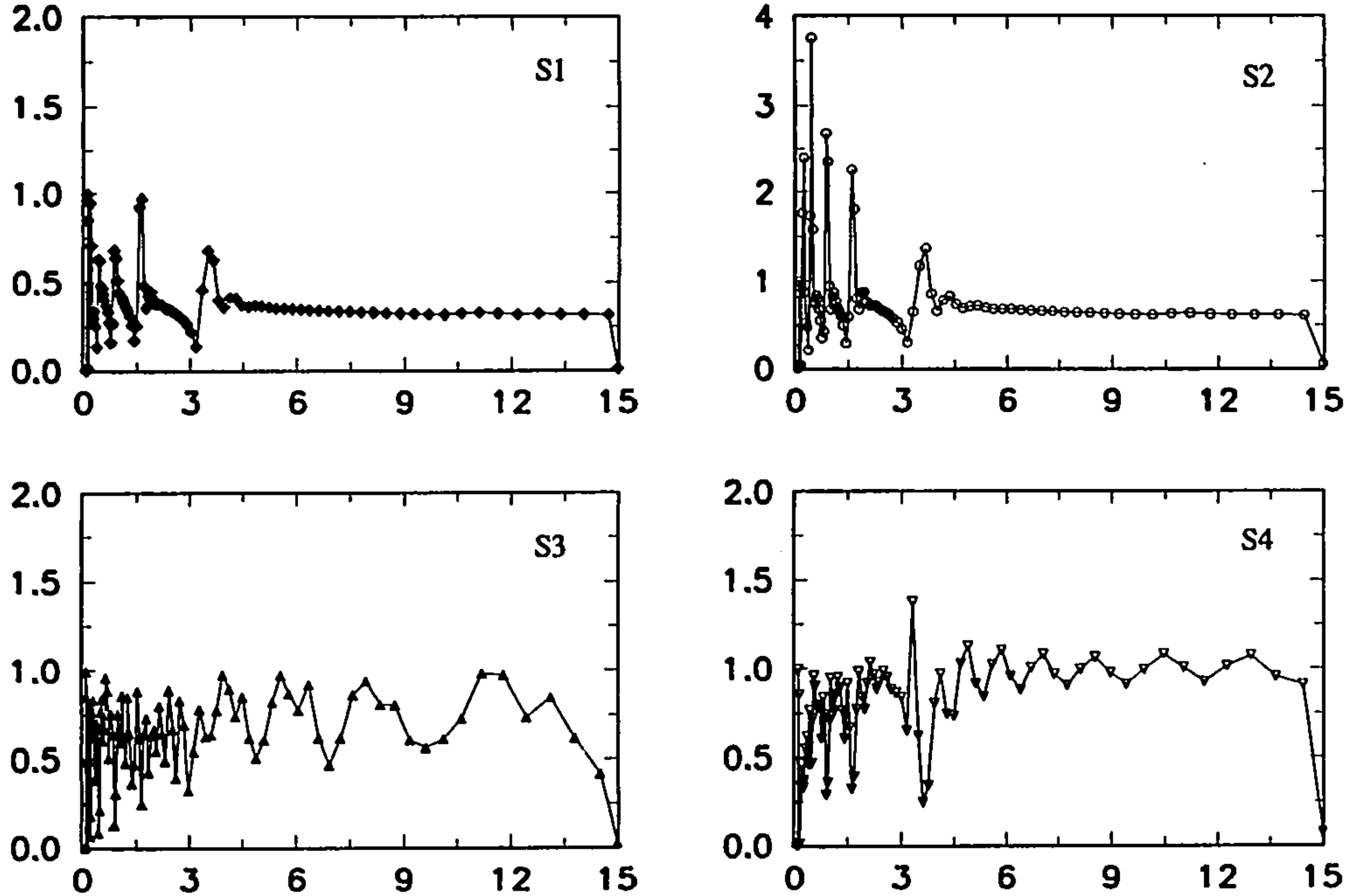


Figure 22 : R-trajectories for BPC case (Problem TP3, $\lambda = 10^{-10}$ NP=6, 7th order)

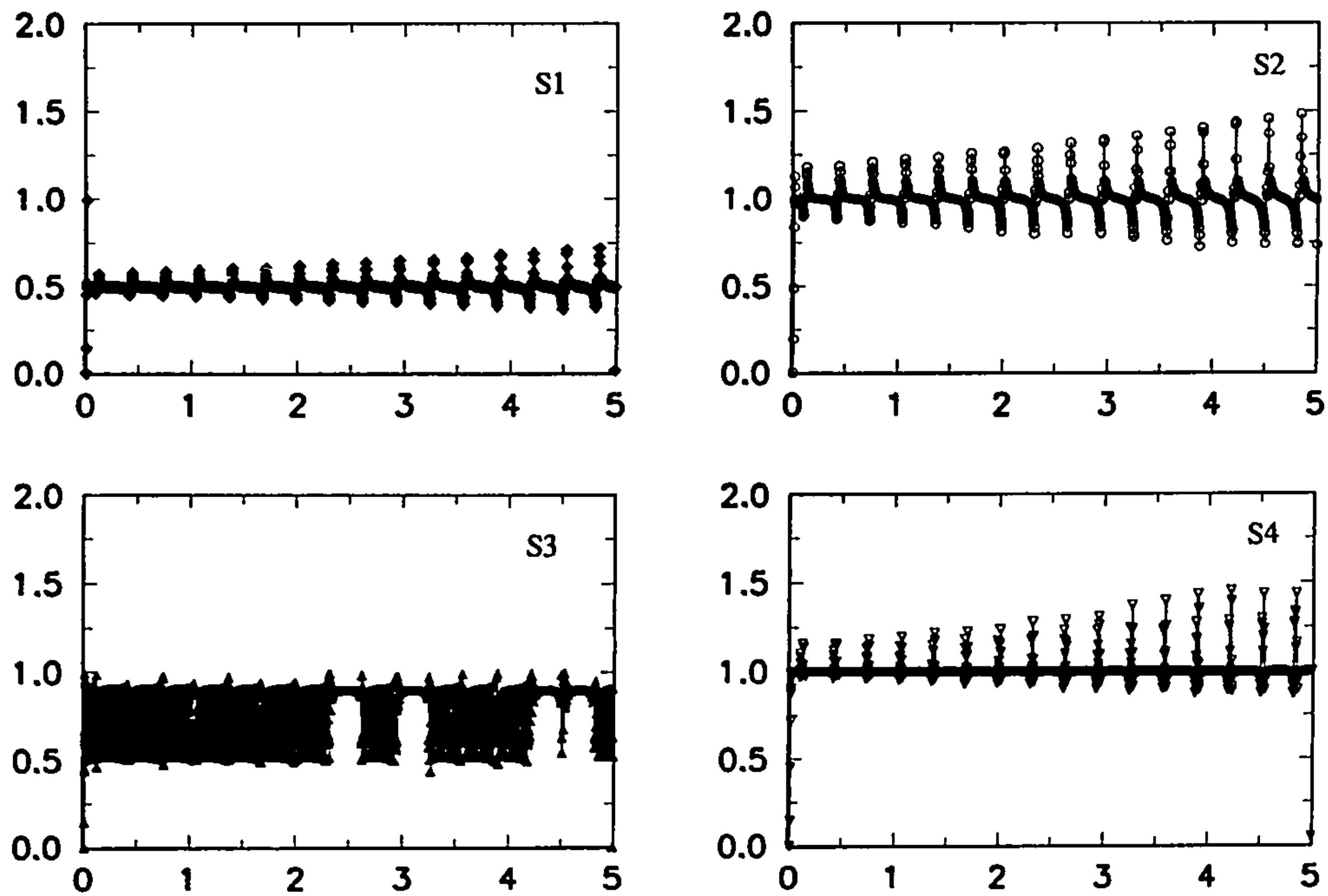


Figure 23 : R-trajectories for BPC case (Problem TP4, $\lambda = 10^{-9}$ NP=2, 3rd order)

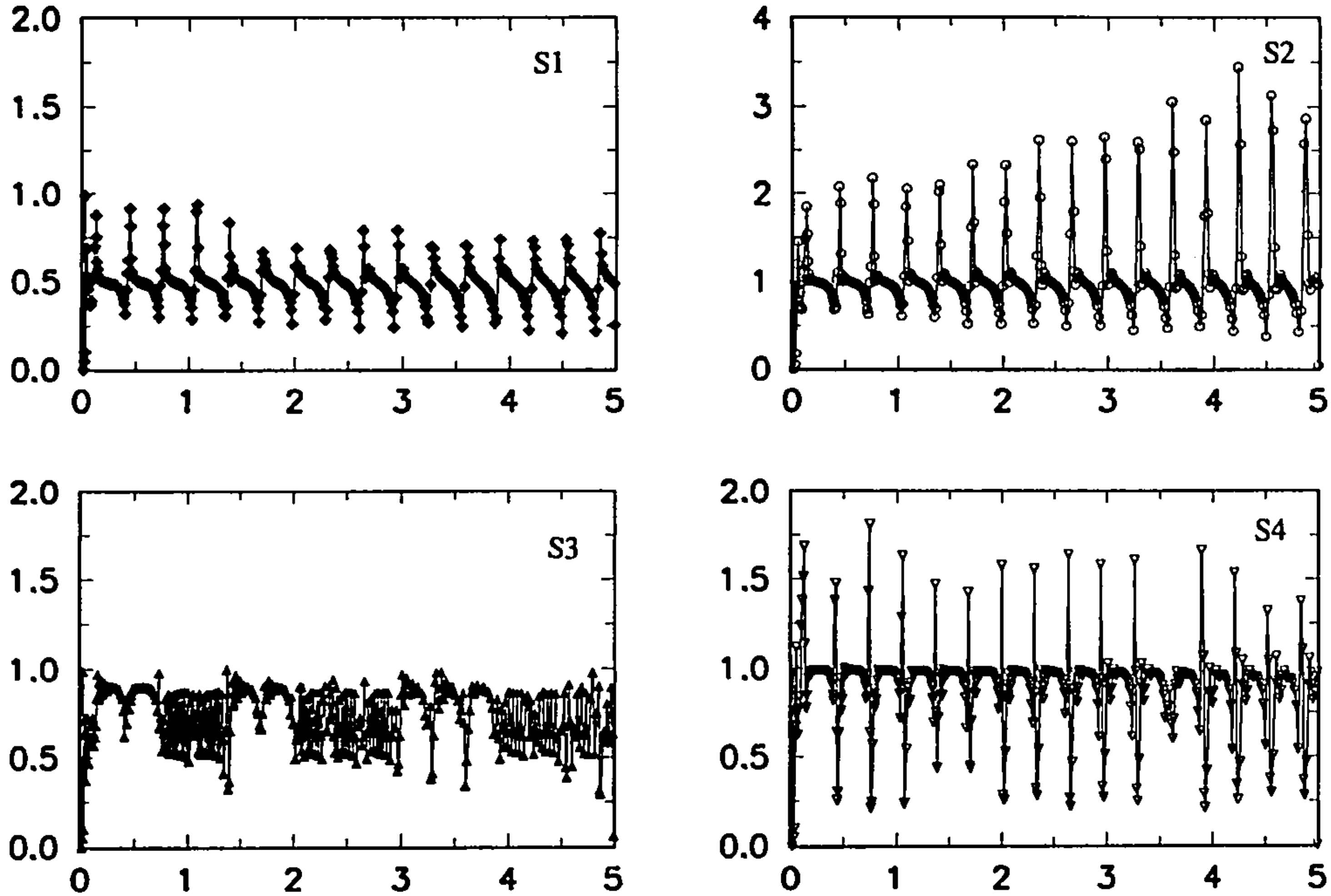


Figure 24 : R-trajectories for BPC case (Problem TP4, $\lambda = 10^{-9}$ NP=4, 5th order)

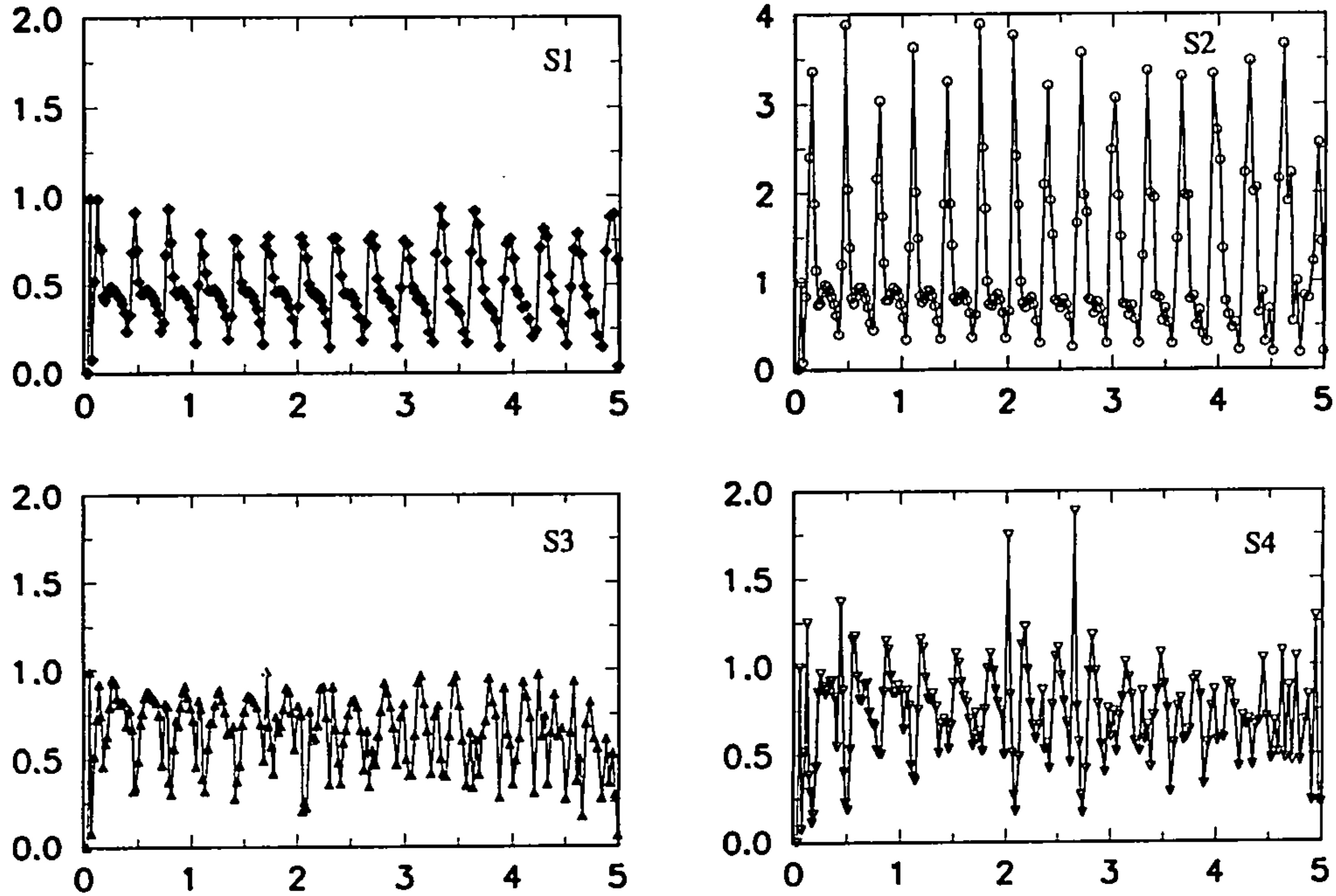


Figure 25 : R-trajectories for BPC case (Problem TP4, $\lambda = 10^{-9}$ NP=6, 7th order)

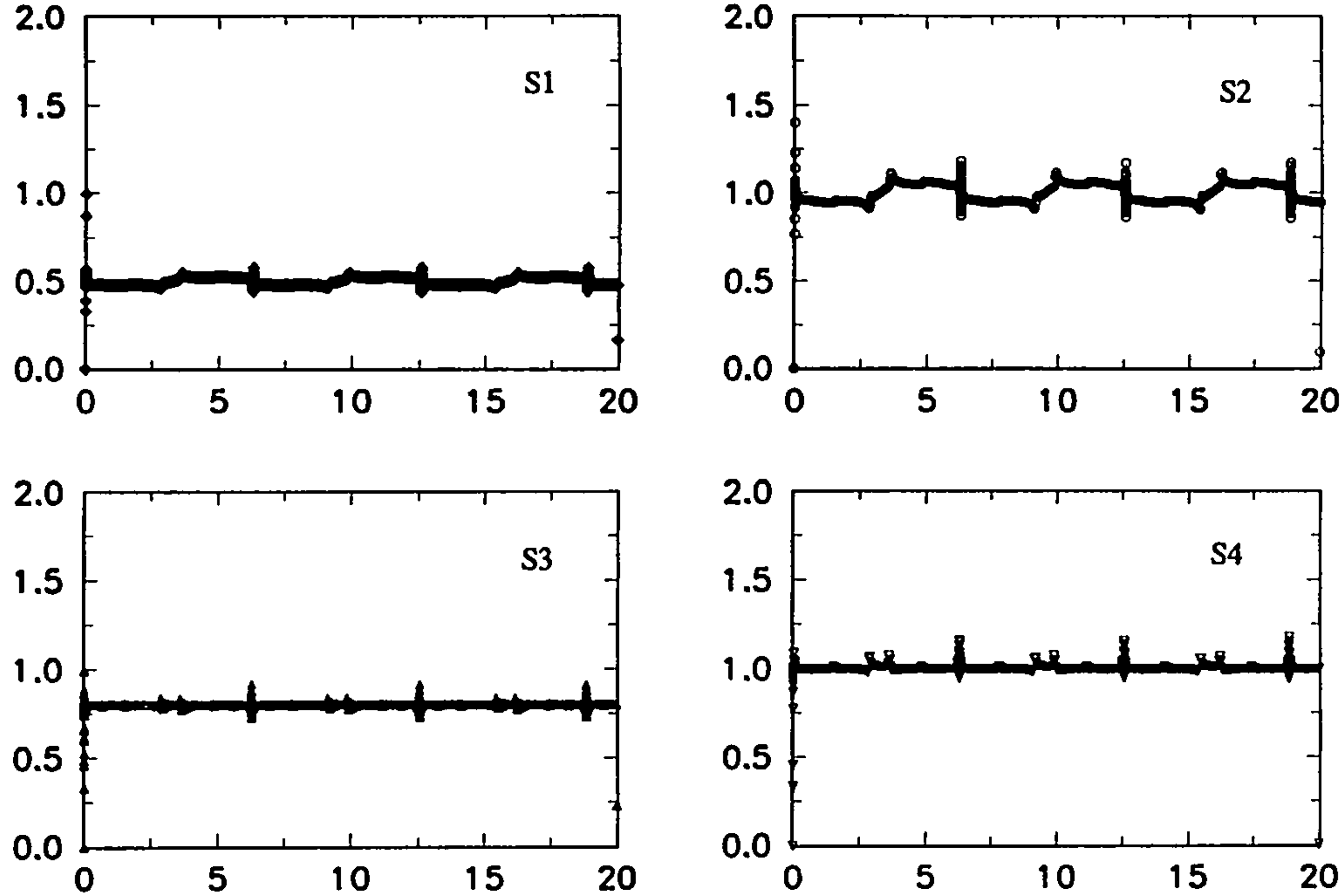


Figure 26 : R-trajectories for BPC case (Problem TP5, $\lambda = 10^{-8}$ NP=2, 3rd order)

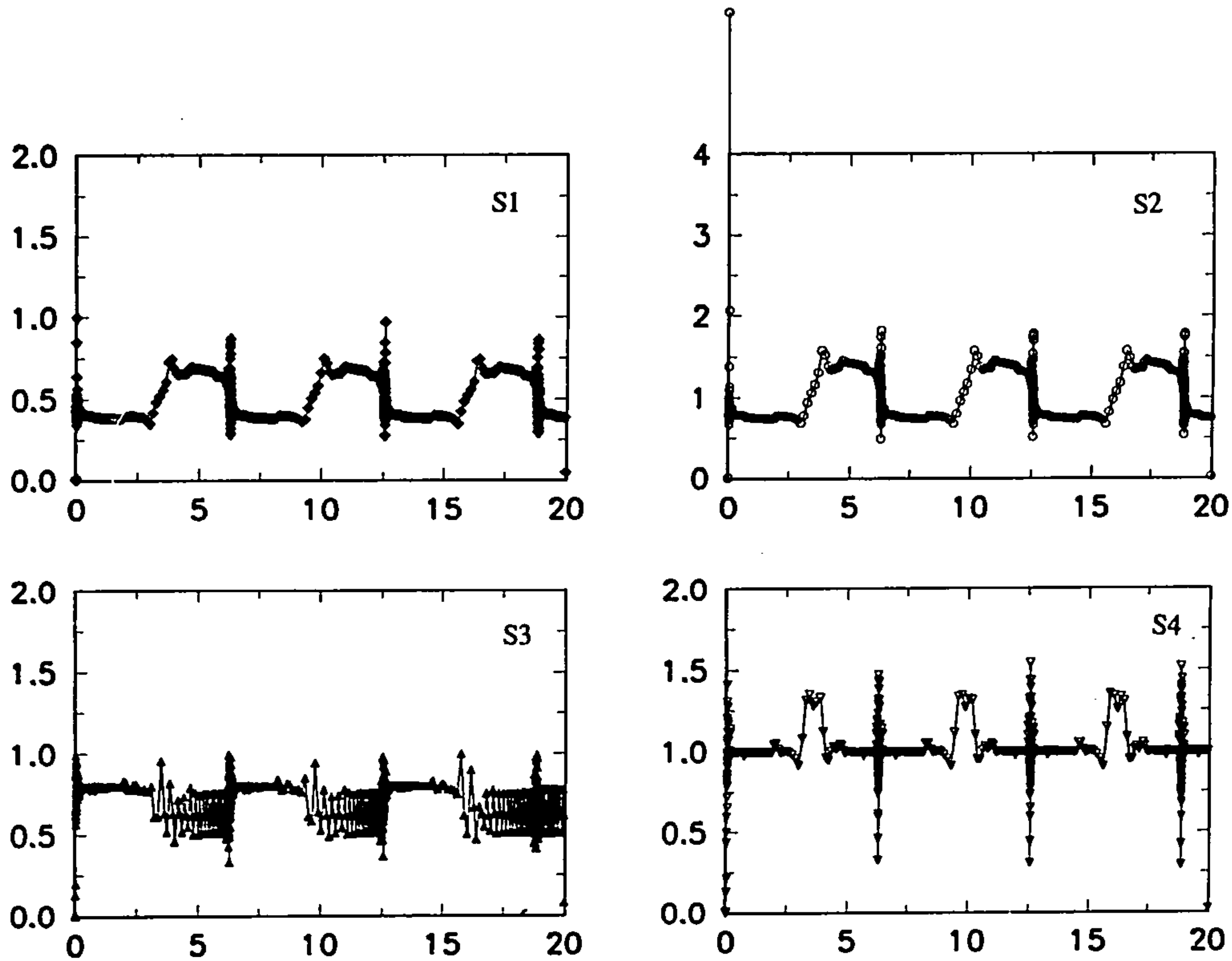


Figure 27 : R-trajectories for BPC case (Problem TP5, $\lambda = 10^{-8}$ NP=4, 5th order)

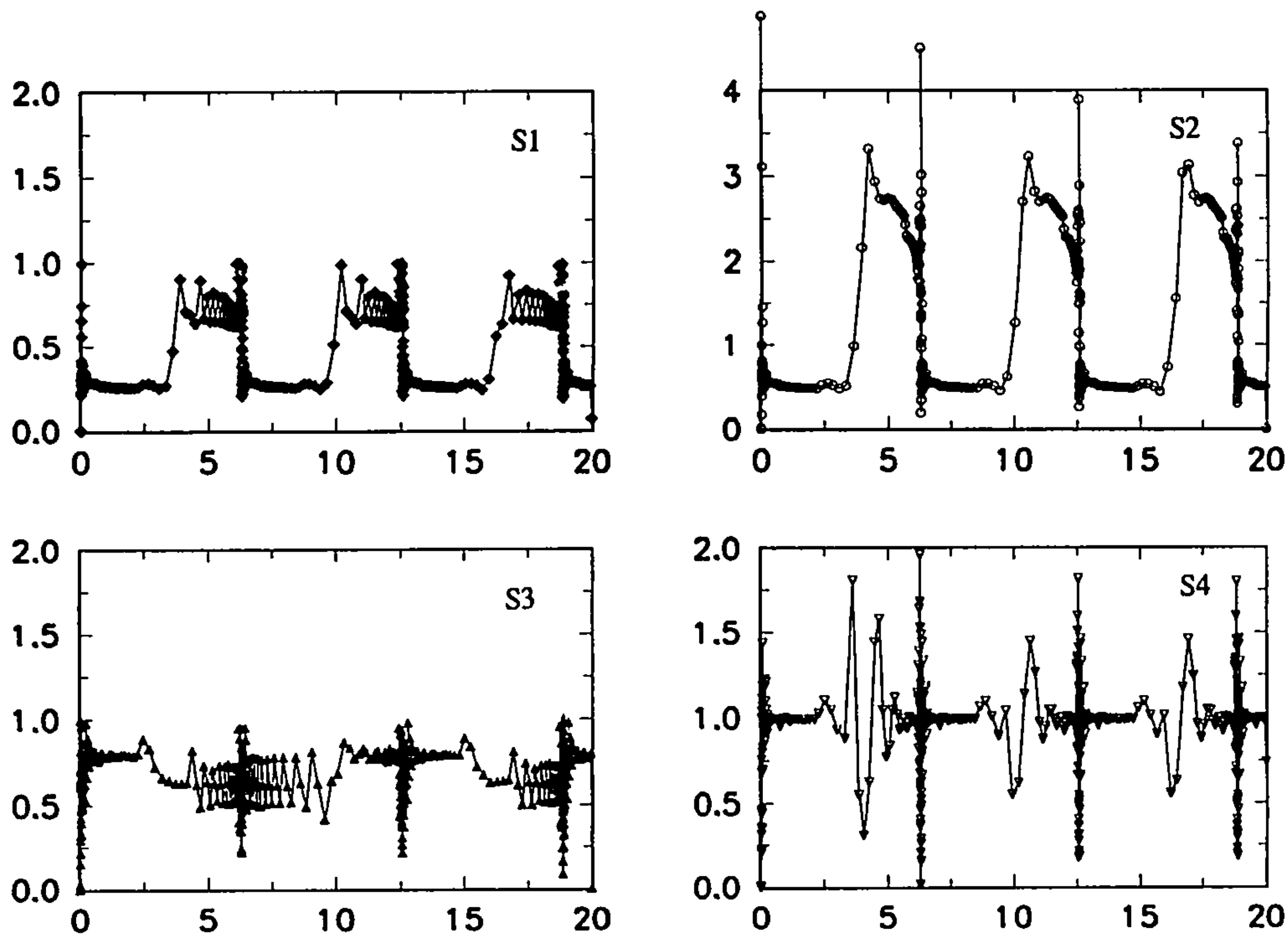


Figure 28 : R-trajectories for BPC case (Problem TP5, $\lambda = 10^{-8}$ NP=6, 7th order)

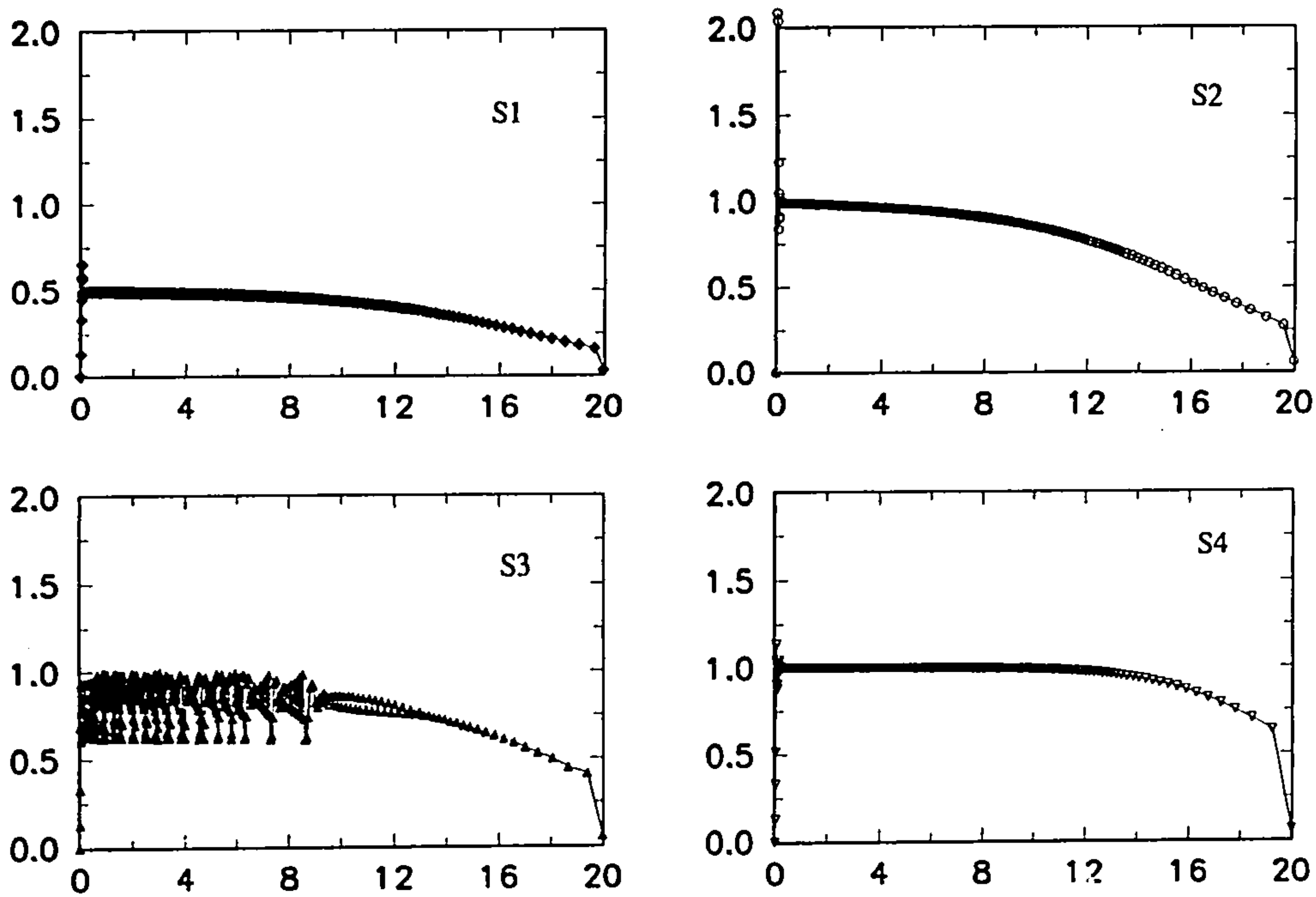


Figure 29 : R-trajectories for PPC case (Problem TP1, $\lambda = 10^{-9}$, NP=2, 3rd order)

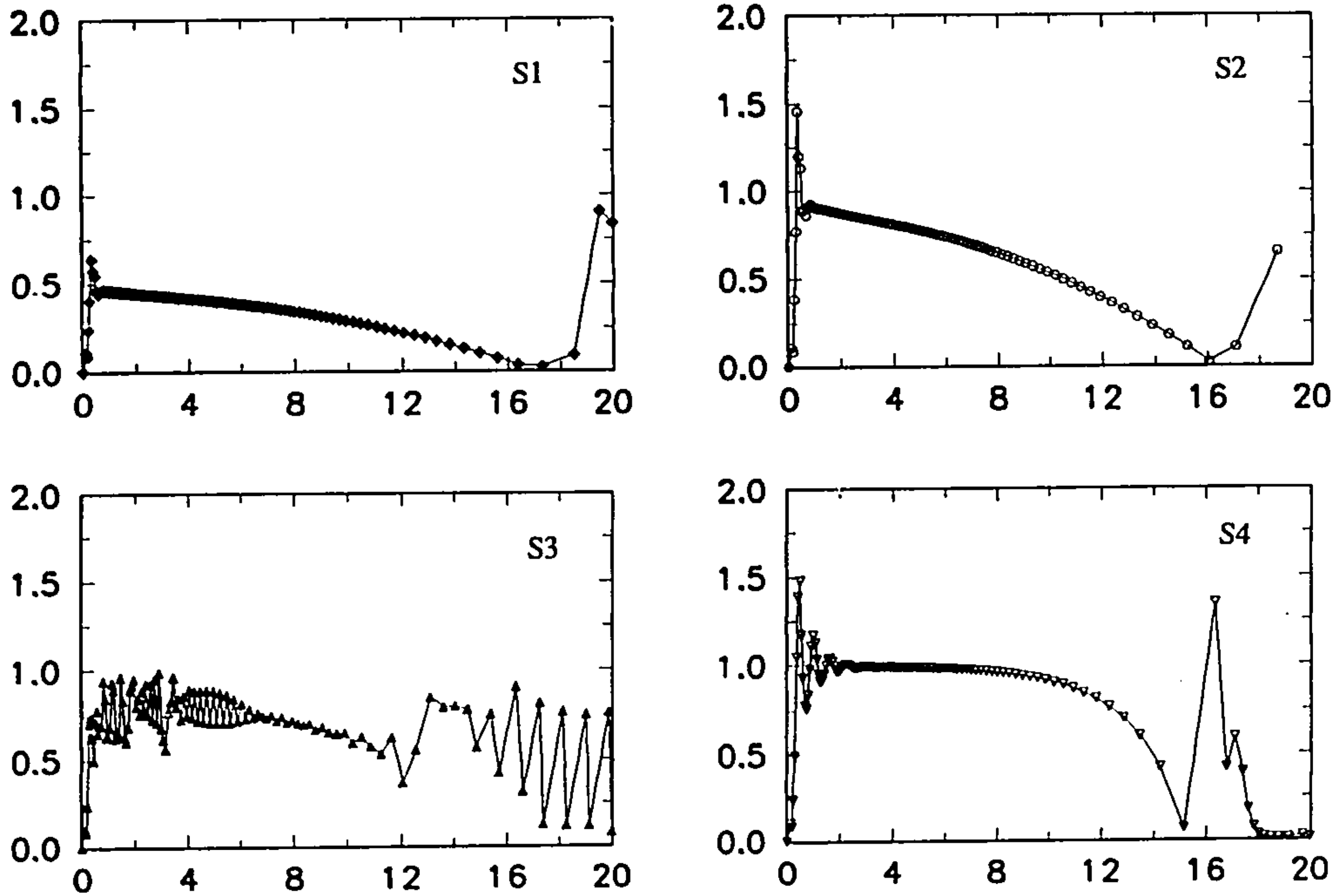


Figure 30 : R-trajectories for PPC case (Problem TP1, $\lambda = 10^{-9}$ NP=4, 5th order)

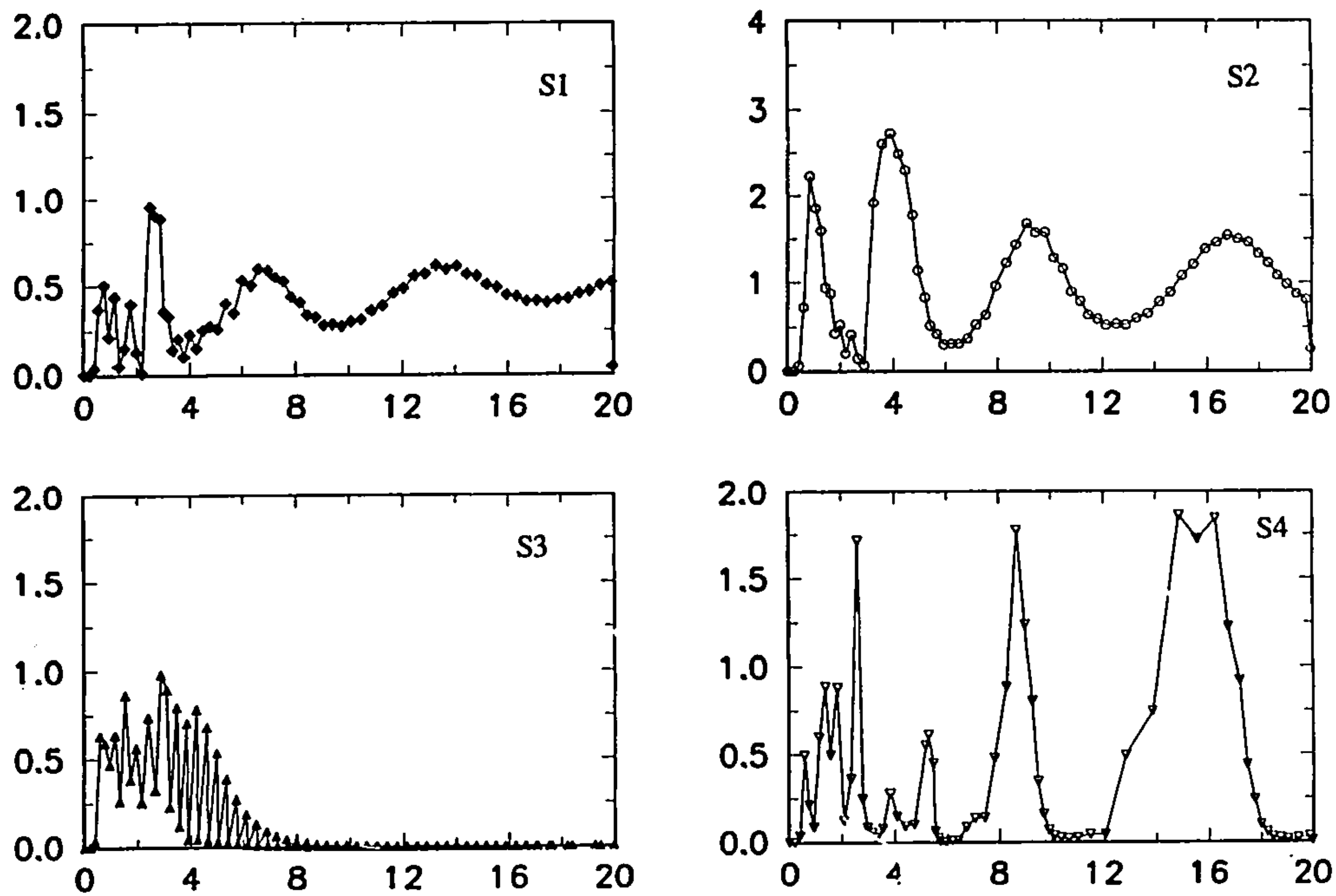


Figure 31 : R-trajectories for PPC case (Problem TP1, $\lambda = 10^{-9}$ NP=6, 7th order)

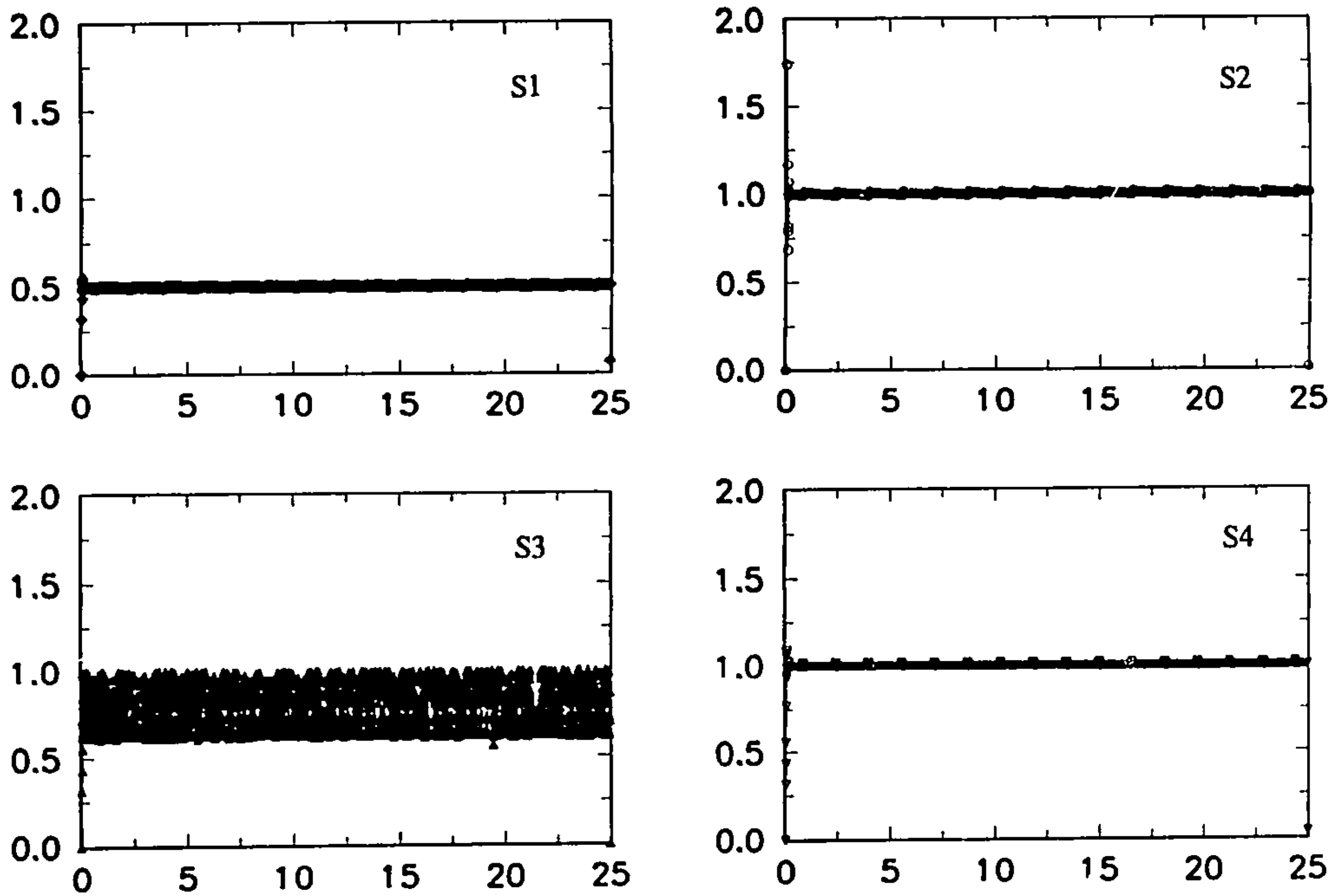


Figure 32 : R-trajectories for PPC case (Problem TP2, $\lambda = 10^{-9}$, NP=2, 3rd order)

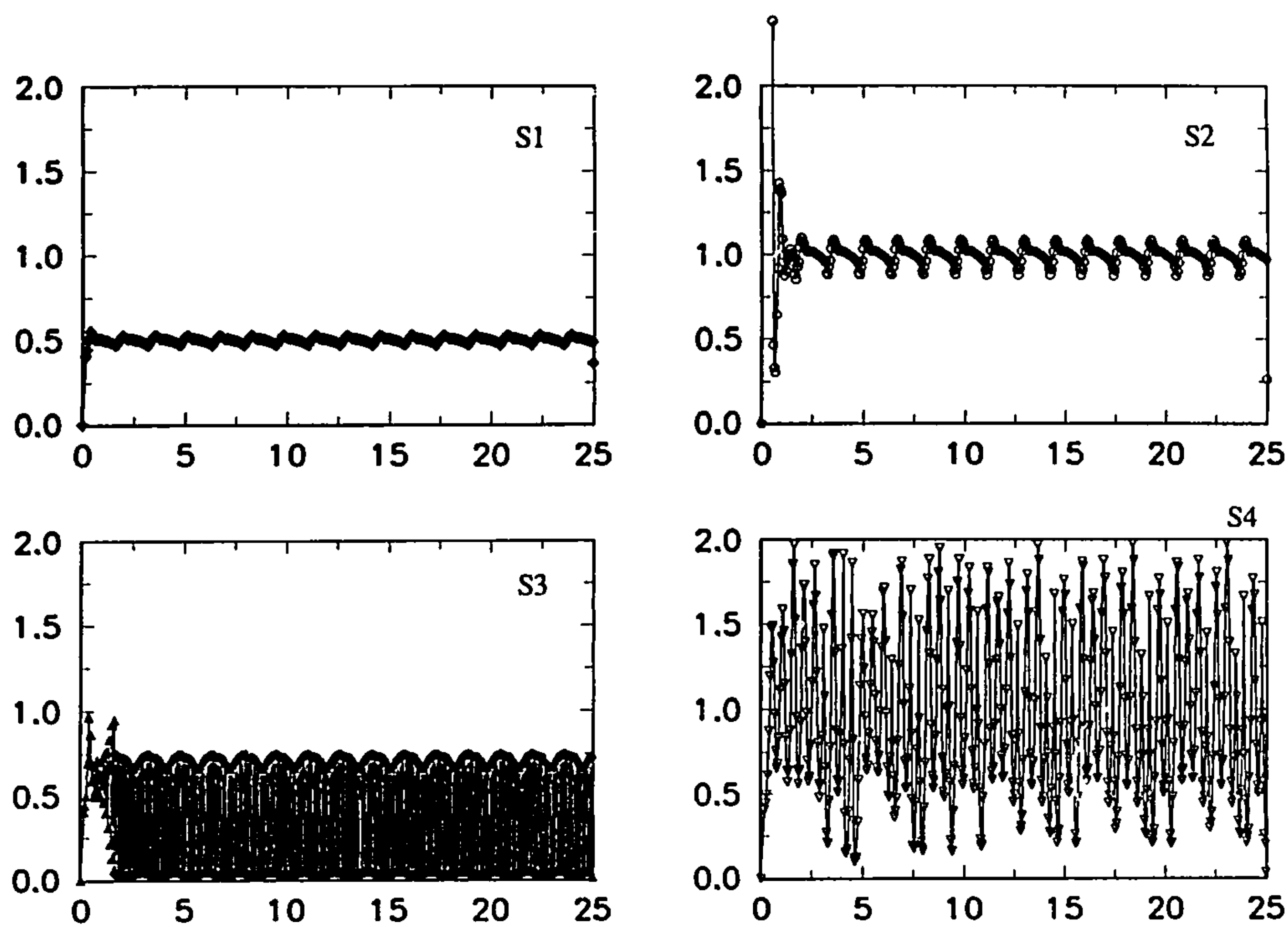


Figure 33 : R-trajectories for PPC case (Problem TP2, $\lambda = 10^{-9}$ NP=4, 5th order)

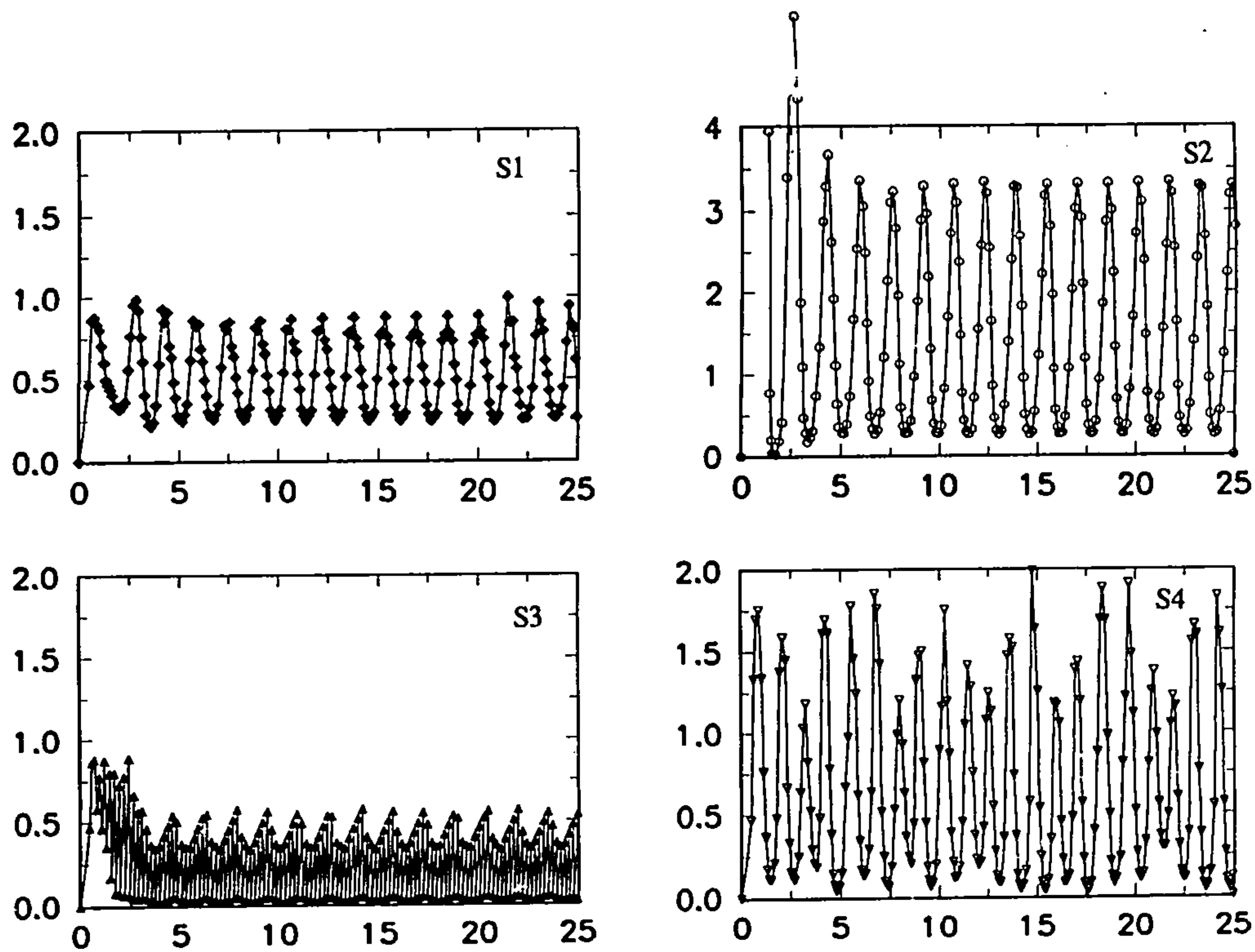


Figure 34 : R-trajectories for PPC case (Problem TP2, $\lambda = 10^{-9}$ NP=6, 7th order)

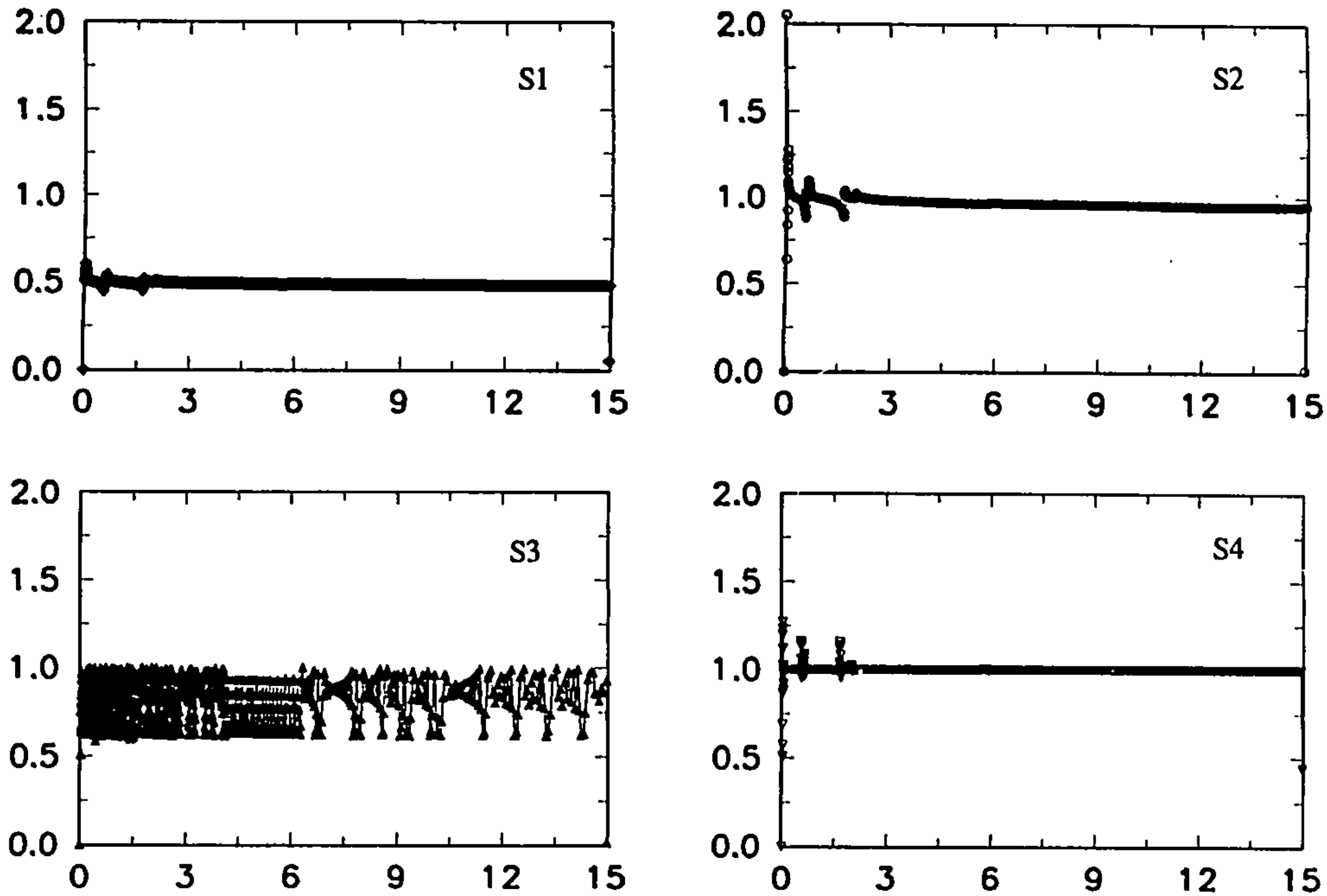


Figure 35 : R-trajectories for PPC case (Problem TP3, $\lambda = 10^{-10}$ NP=2, 3rd order)

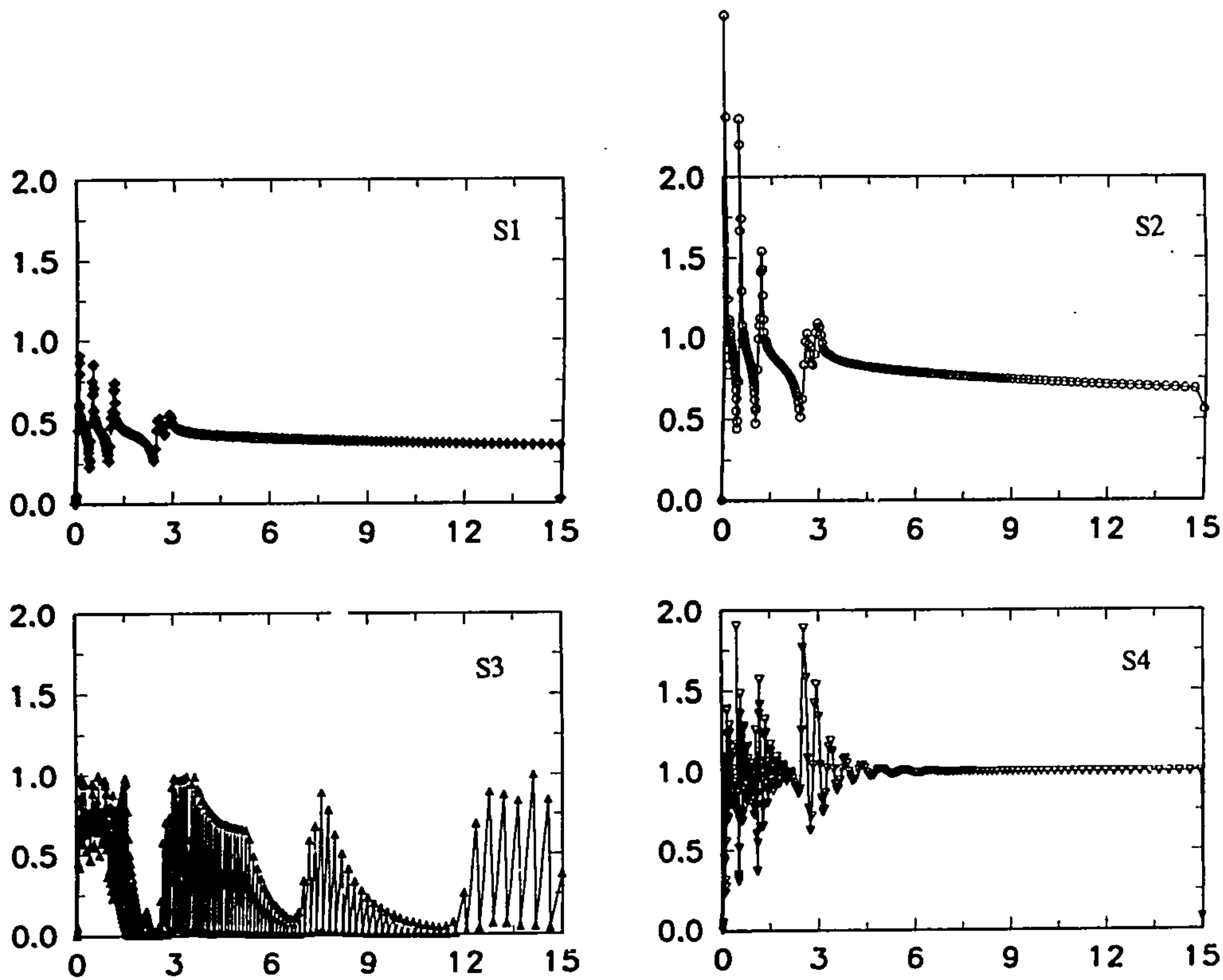


Figure 36 : R-trajectories for PPC case (Problem TP3, $\lambda = 10^{-10}$ NP=4, 5th order)

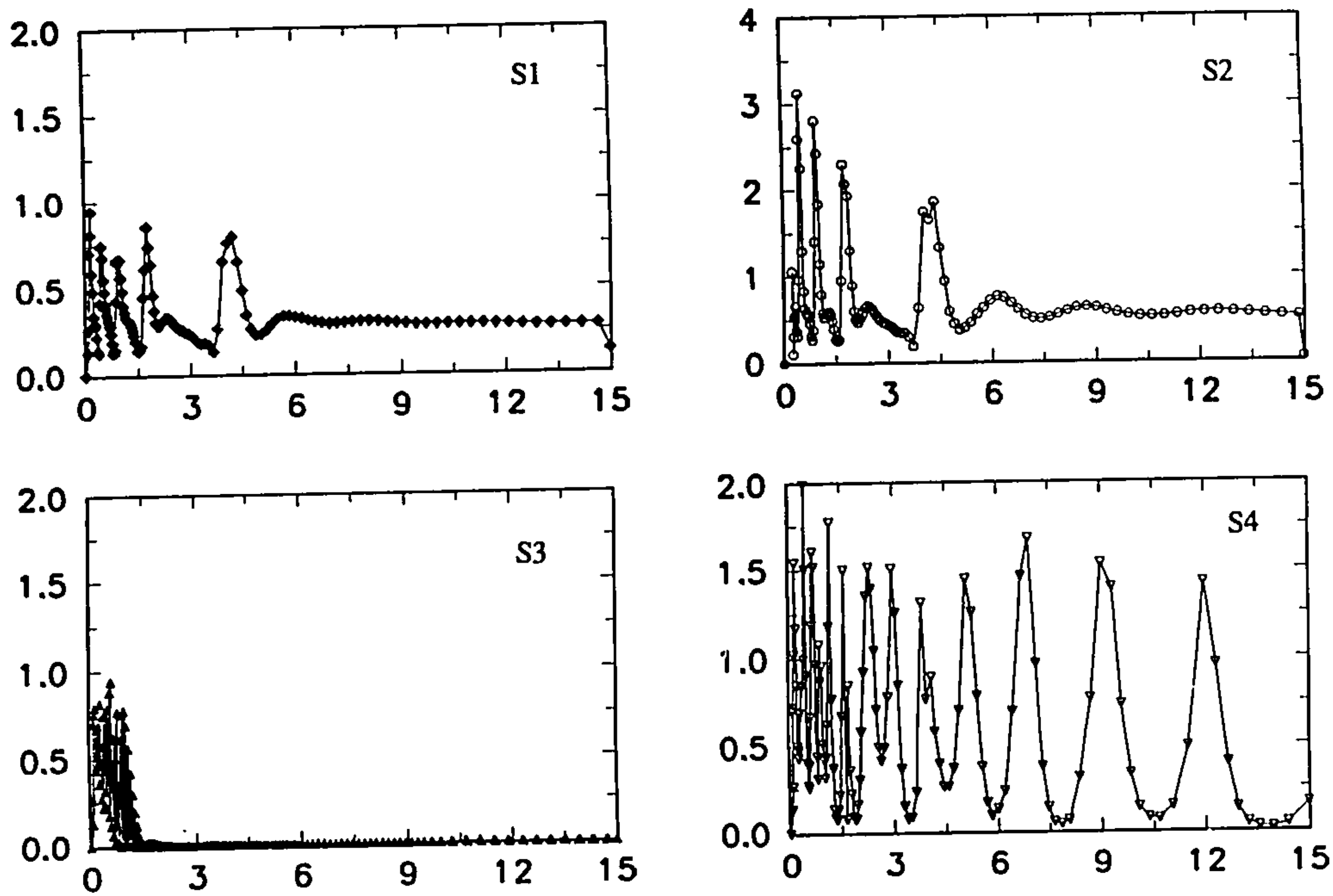


Figure 37 : R-trajectories for PPC case (Problem TP3, $\lambda = 10^{-10}$ NP=6, 7th order)

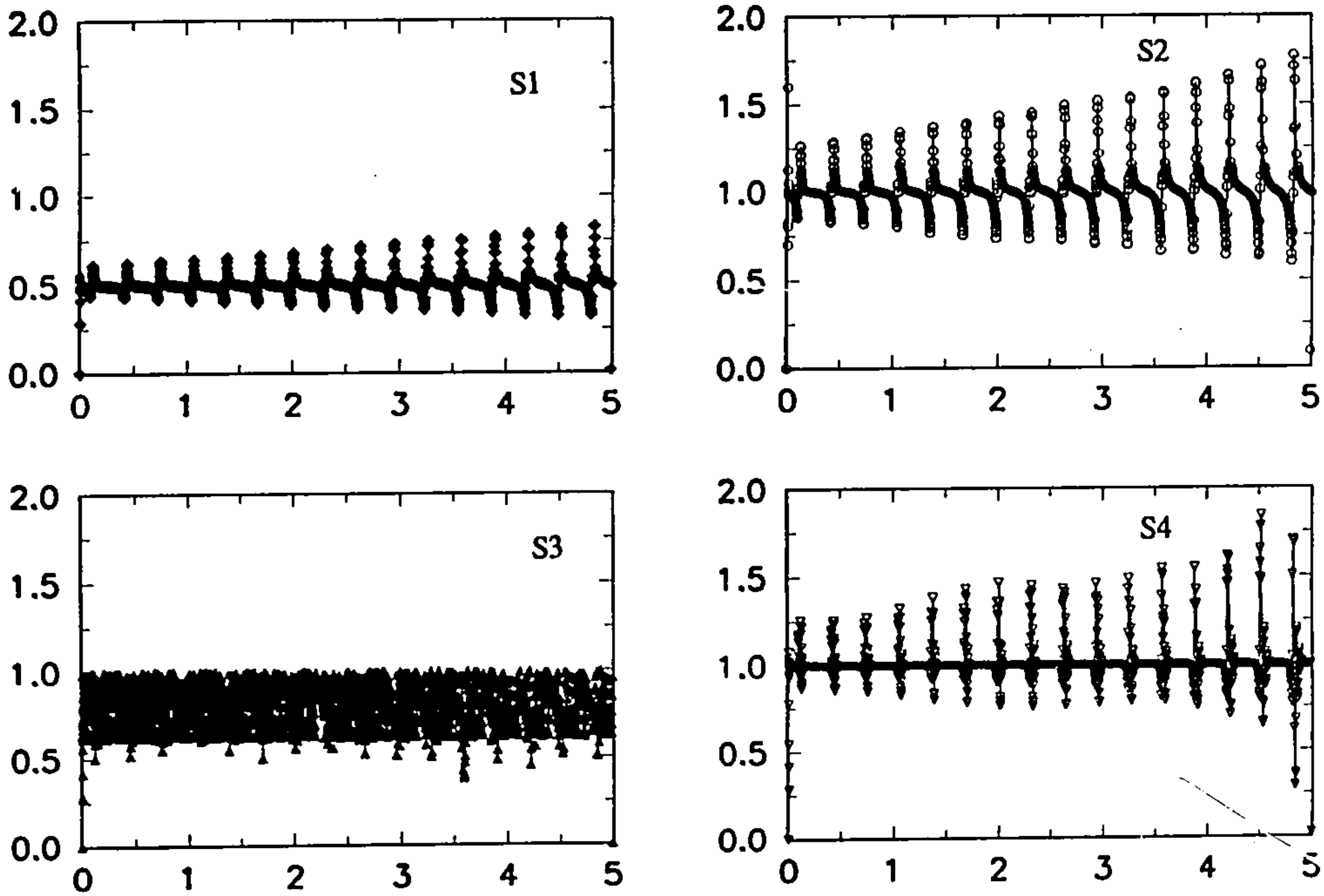


Figure 38 : R-trajectories for PPC case (Problem TP4, $\lambda = 10^{-9}$ NP=2, 3rd order)

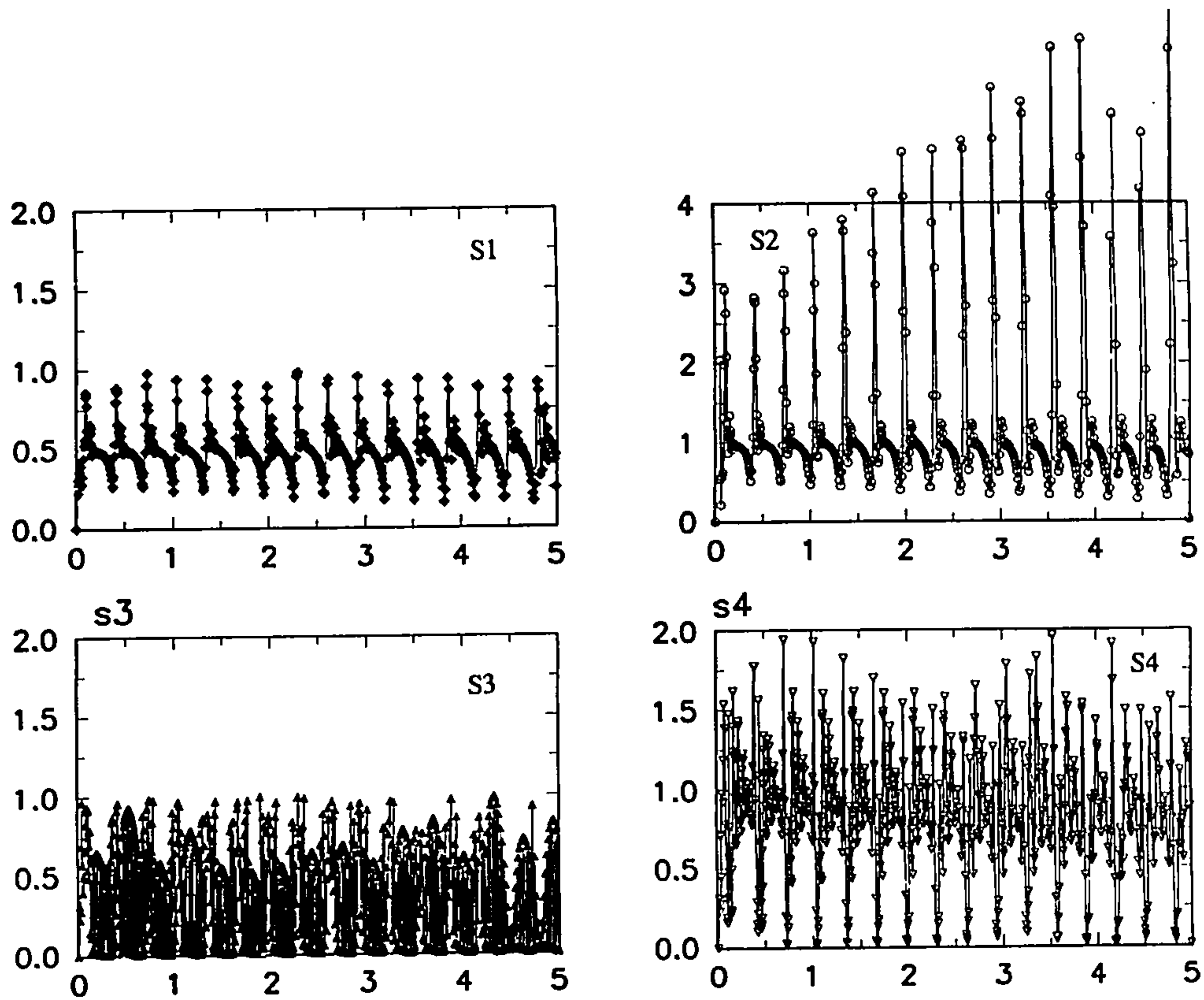


Figure 39 : R-trajectories for PPC case (Problem TP4, $\lambda = 10^{-9}$ NP=4, 5th order)

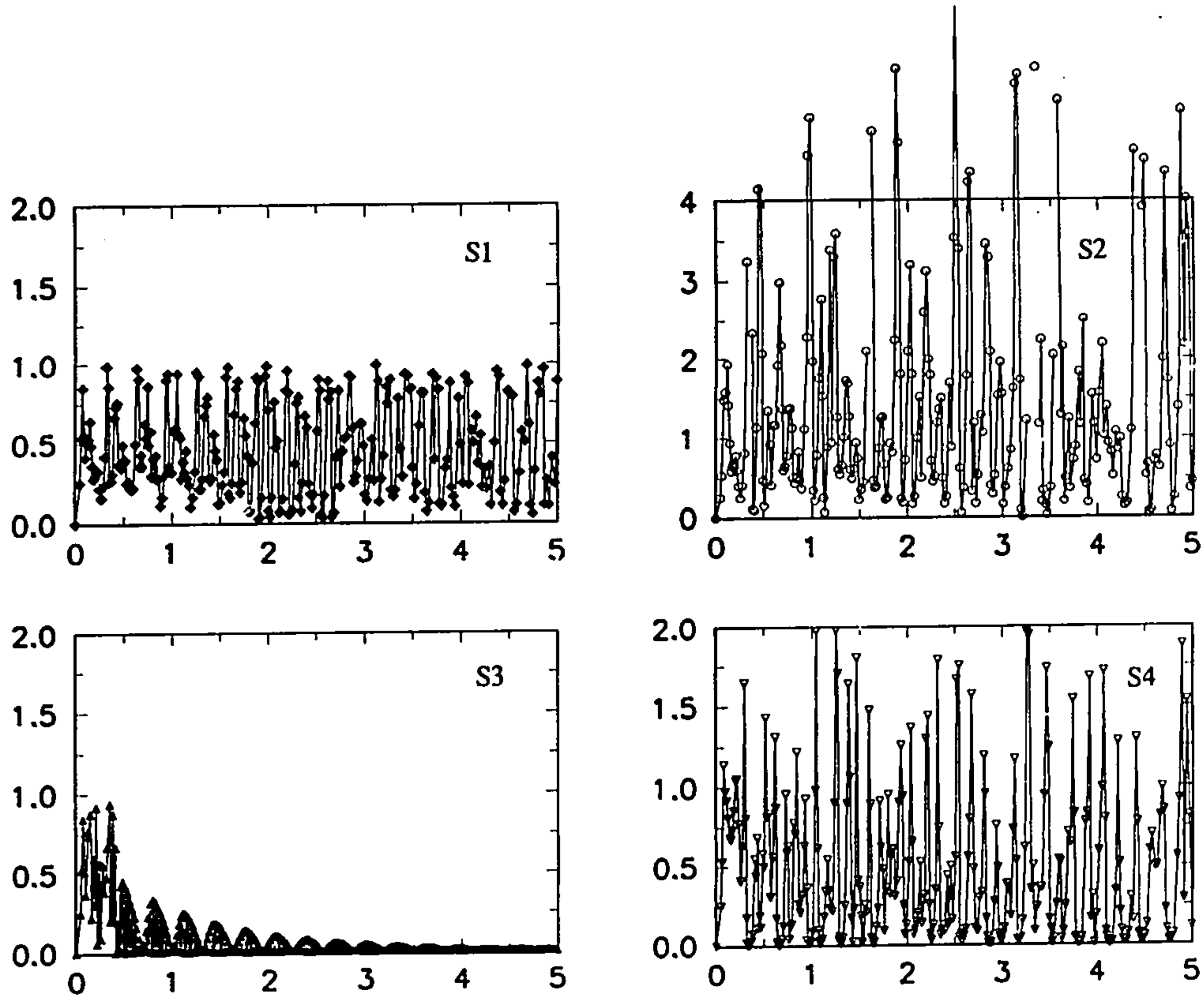


Figure 40 : R-trajectories for PPC case (Problem TP4, $\lambda = 10^{-9}$ NP=6, 7th order)

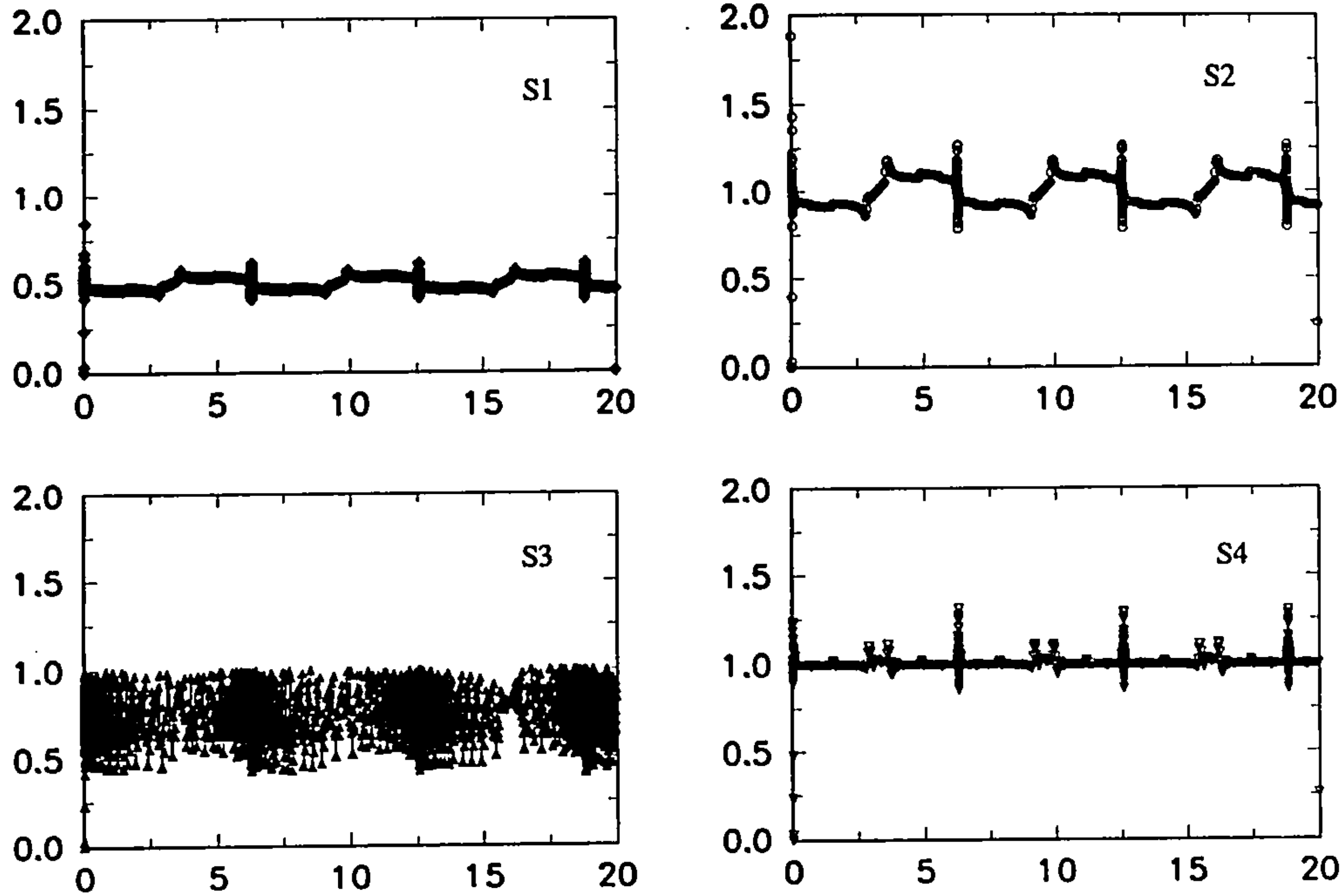


Figure 41 : R-trajectories for PPC case (Problem TP5, $\lambda = 10^{-8}$ NP=2, 3rd order)

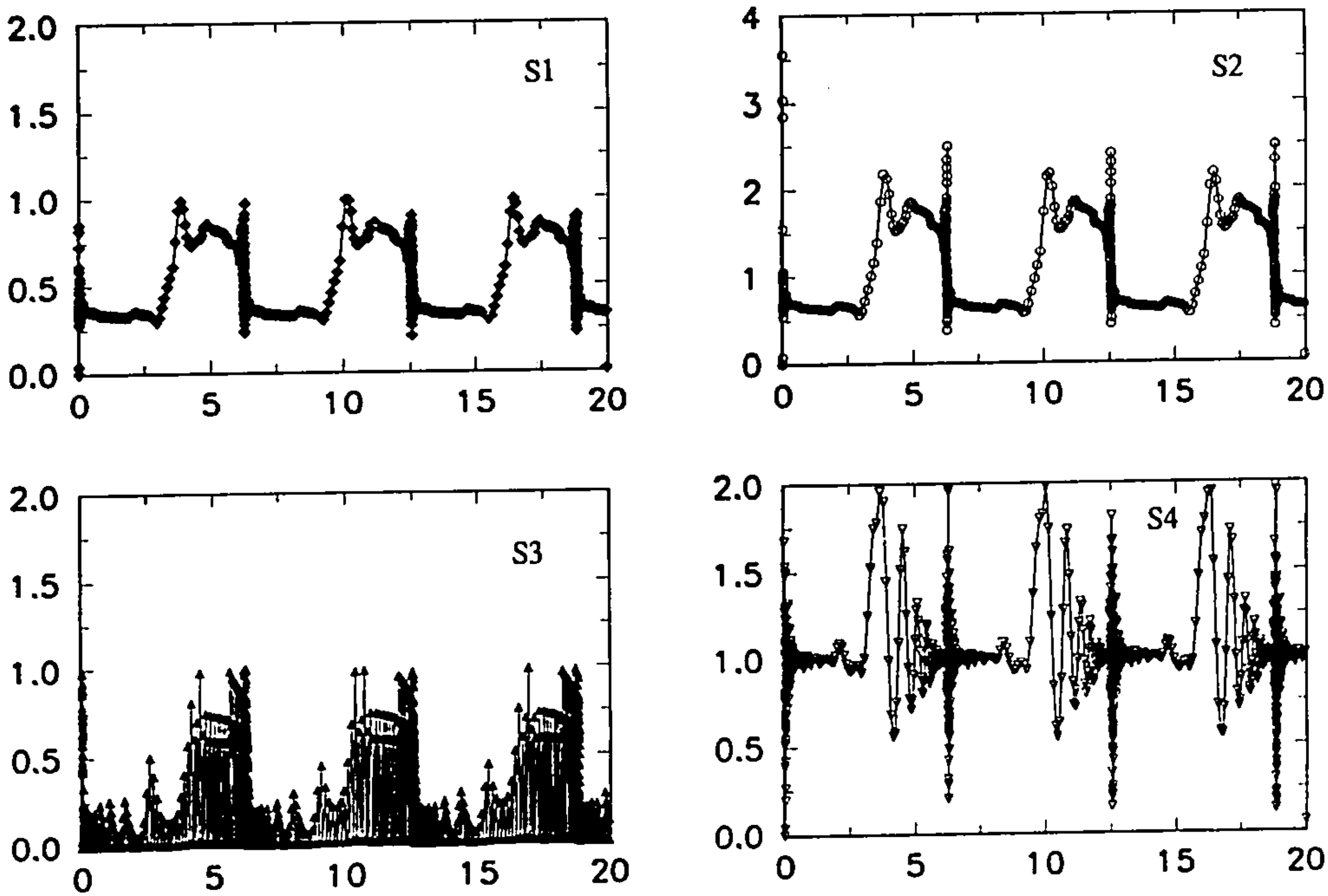


Figure 42 : R-trajectories for PPC case (Problem TP5, $\lambda = 10^{-8}$ NP=4, 5th order)

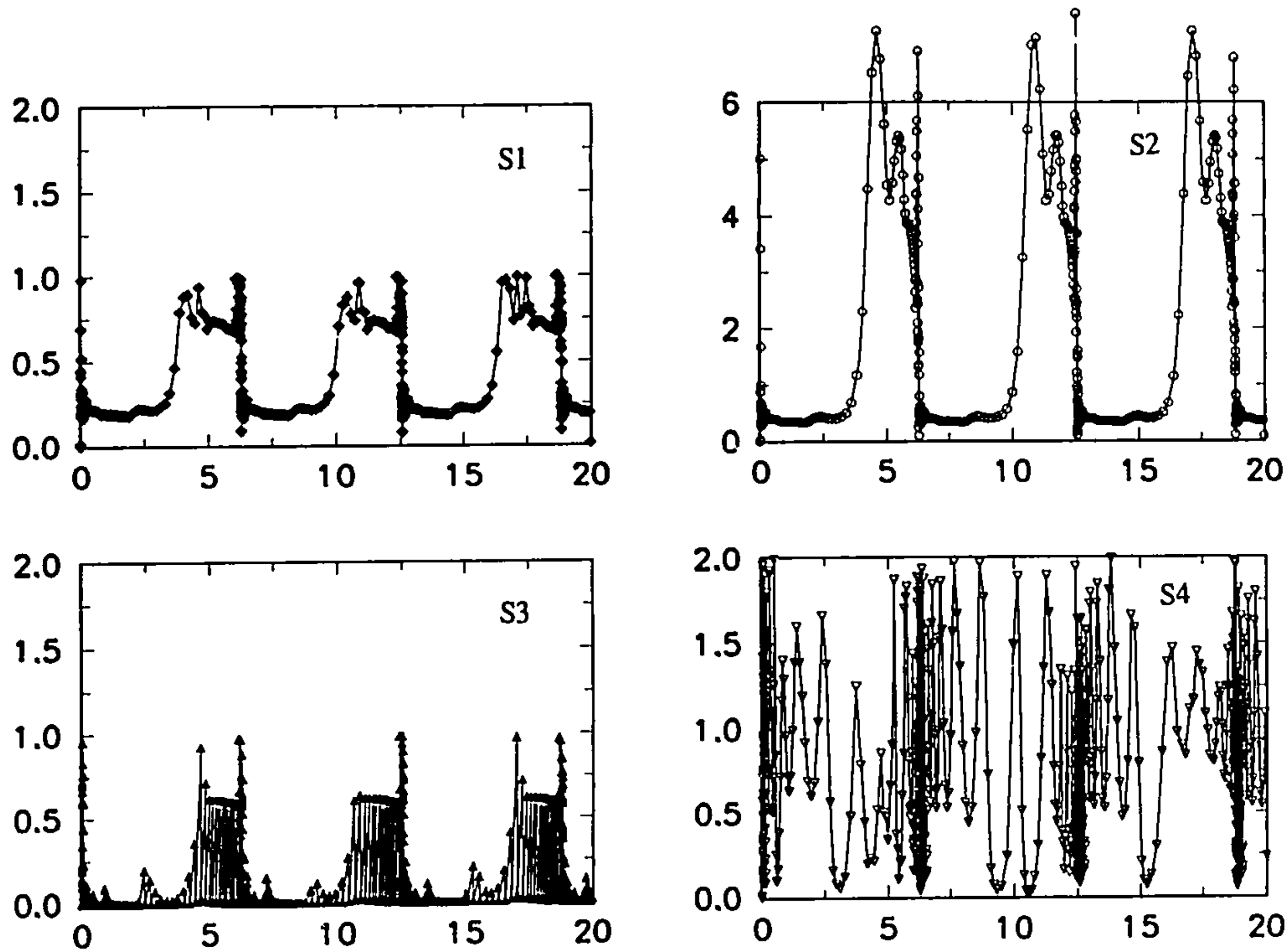


Figure 43 : R-trajectories for PPC case (Problem TP5, $\lambda = 10^{-8}$ NP=6, 7th order)

Appendix 1
Summary of Test Problems

In this Appendix we provide a summary of the five test problems used in the experimental evaluations carried out in this study.

Problem TP1:

$$\dot{y} = -y \; ; \qquad \begin{aligned} y(0) &= 1 \\ t_f &= 20. \end{aligned}$$

Analytic solution

$$y = e^{-t}.$$

The corresponding solution trajectory is given in Figure A1.1.

Problem TP2:

$$\begin{aligned} \dot{y}_1 &= y_2 \; ; & y_1(0) &= 1 \\ \dot{y}_2 &= -y_1/r^3 \; ; & y_2(0) &= 0 \\ \dot{y}_3 &= y_4 \; ; & y_3(0) &= 0 \\ \dot{y}_4 &= -y_3/r^3 \; ; & y_4(0) &= 1 \end{aligned}$$

with

$$r = (y_1^2 + y_3^2)^{1/2} \; ; \qquad t_f = 25.$$

Analytic solution

$$\begin{aligned} y_1 &= \cos(t) \\ y_2 &= -\sin(t) \\ y_3 &= \sin(t) \\ y_4 &= \cos(t). \end{aligned}$$

The corresponding solution trajectory is given in Figure A1.2.

Problem TP3:

$$\dot{y}_1 = y_2 \; ; \qquad y_1(0) = 1$$

$$\dot{y}_2 = -2y_1^2(1-4t^2y_1) \; ; \qquad y_2(0) = 0$$

$$t_f = 15.$$

Analytic solution

$$\dot{y}_1 = \frac{1}{1+t^2}$$

$$\dot{y}_2 = \frac{-2t}{(1+t^2)^2} \; .$$

The corresponding solution trajectory is given in Figure A1.3.

Problem TP4:

$$\dot{y}_1 = y_2 \; ; \qquad y_1(0) = 0$$

$$\dot{y}_2 = -2y_2 - 101y_1 \; ; \qquad y_2(0) = 1$$

$$\dot{y}_3 = y_4 \; ; \qquad y_3(0) = 0$$

$$\dot{y}_4 = y_1 - 4y_4 - 29y_3 \; ; \qquad y_4(0) = 0$$

$$t_f = 5.$$

Analytic solution

$$y_1 = 0.1e^{-t}\sin(10t)$$

$$y_2 = (1.01)^{1/2}e^{-t}\cos(10t+\theta_1)$$

$$y_3 = (5e^{-t}\sin(10t-\psi)+10e^{-2t}\sin(5t+\phi))/Q$$

$$y_4 = (5e^{-t}(101)^{1/2}\cos(10t-(\psi-\theta_1)) + 10e^{-2t}(29)^{1/2}\cos(5t-(\phi-\theta_2)))/Q$$

with

$$Q = 50(6476)^{1/2}$$
$$\psi = \tan^{-1}(-20/74)$$
$$\phi = \tan^{-1}(-10/76)$$
$$\theta_1 = \tan^{-1}(0.1)$$
$$\theta_2 = \tan^{-1}(0.4).$$

The corresponding solution trajectory is given in Figure A1.4.

Problem TP5:

$$\dot{y}_1 = y_3 \; ; \qquad y_1(0) = 1-\varepsilon$$
$$\dot{y}_2 = y_4 \; ; \qquad y_2(0) = 0$$
$$\dot{y}_3 = \frac{-y_1}{(y_1^2+y_2^2)^{3/2}} \; ; \qquad y_3(0) = 0$$
$$\dot{y}_4 = \frac{-y_2}{(y_1^2+y_2^2)^{3/2}} \; ; \qquad y_4(0) = [\frac{1+\varepsilon}{1-\varepsilon}]^{1/2}$$

$$\varepsilon = 0.9; \qquad t_f = 20.$$

Analytic solution

$$y_1 = \cos(\mu) - \varepsilon$$
$$y_2 = (1-\varepsilon^2)^{1/2} \sin(\mu)$$
$$y_3 = \frac{-\sin(\mu)}{1-\varepsilon\cos(\mu)}$$
$$y_4 = \frac{(1-\varepsilon^2)^{1/2}\cos(\mu)}{1-\varepsilon\cos(\mu)}$$

where

$$\mu - \varepsilon \sin(\mu) - t = 0.$$

The corresponding solution trajectory is given in Figure A1.5.

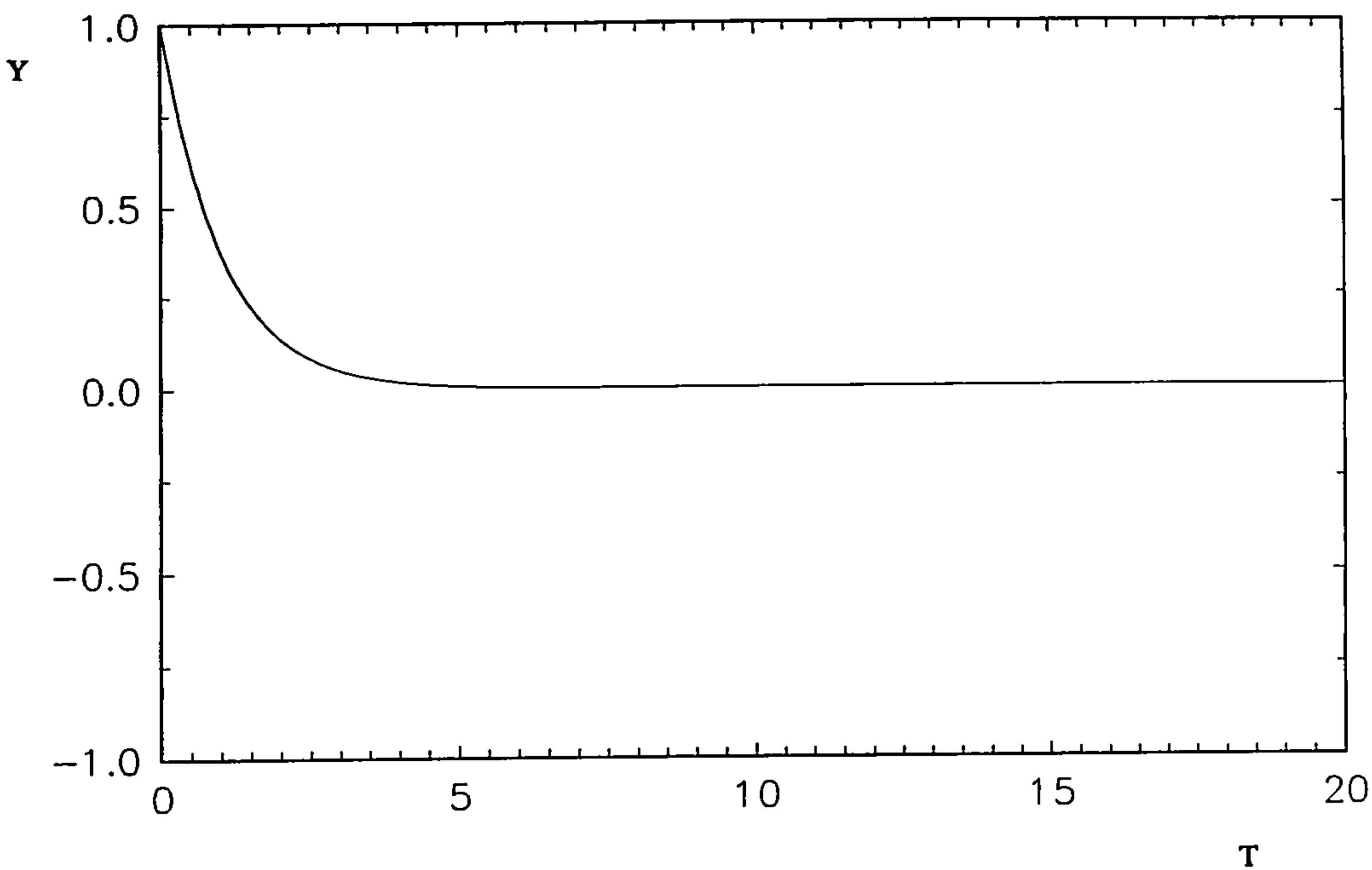


Figure A1.1 : Solution trajectory for problem TP1

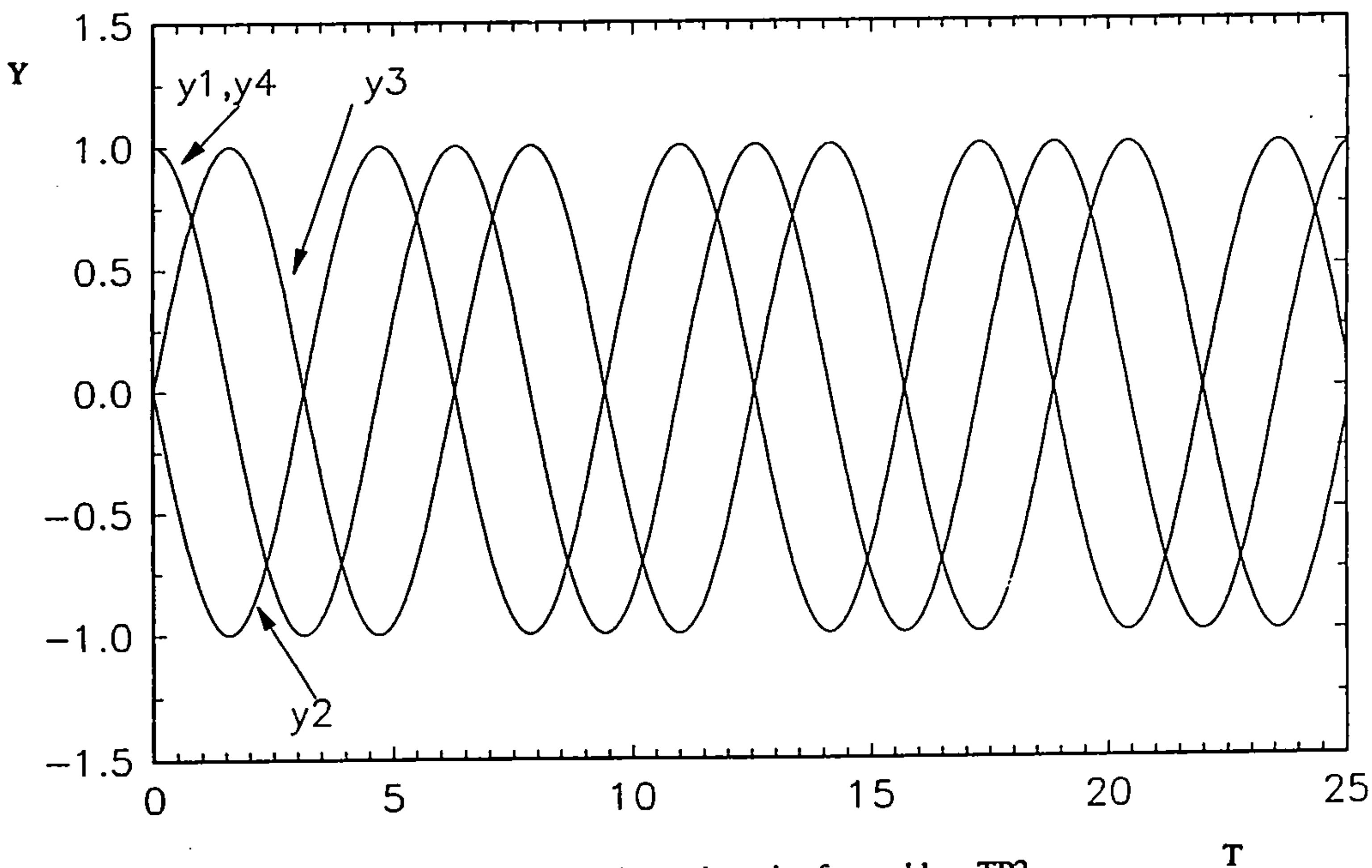


Figure A1.2 : Solution trajectories for problem TP2

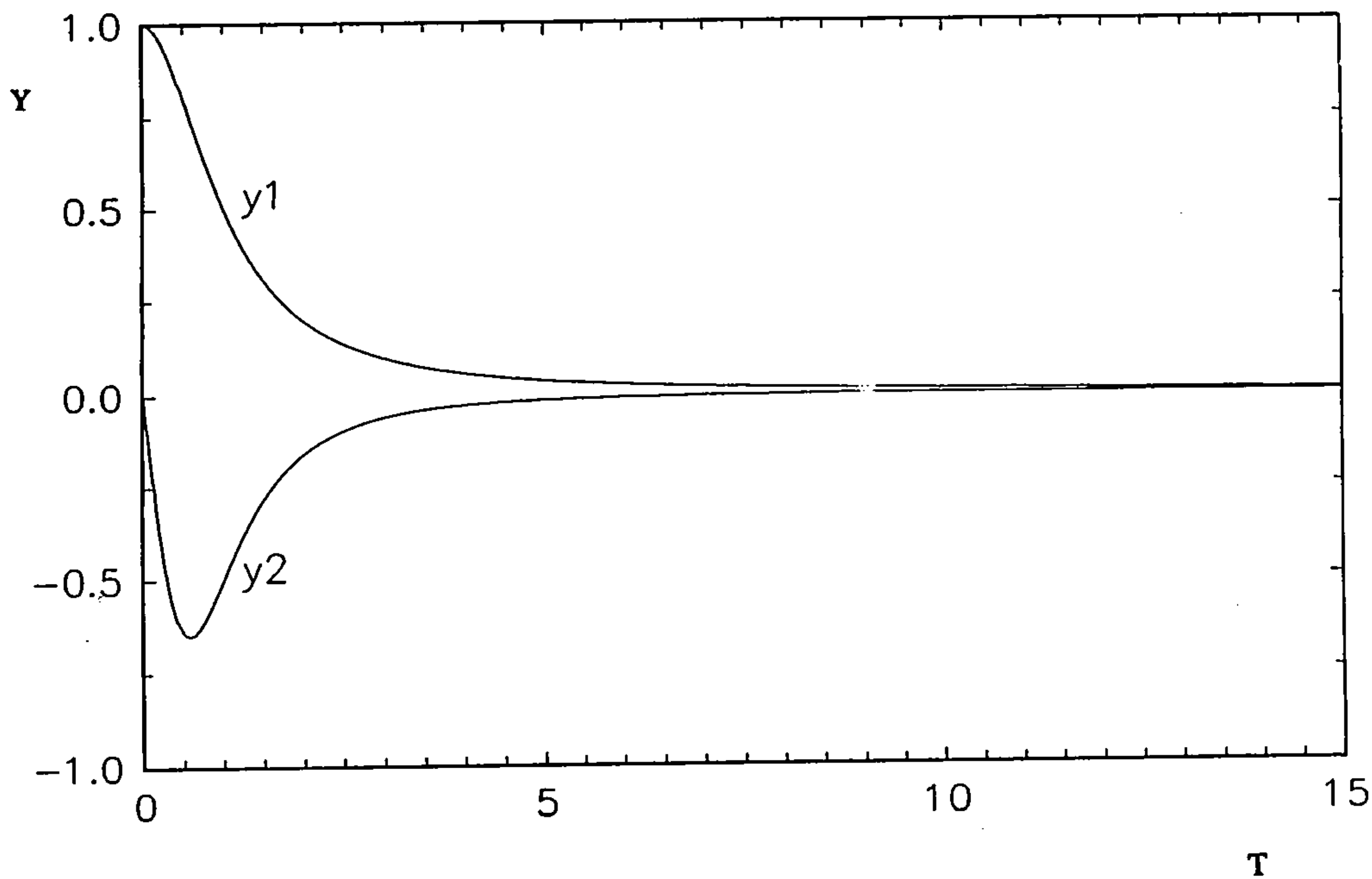


Figure A1.3 : Solution trajectories for problem TP3

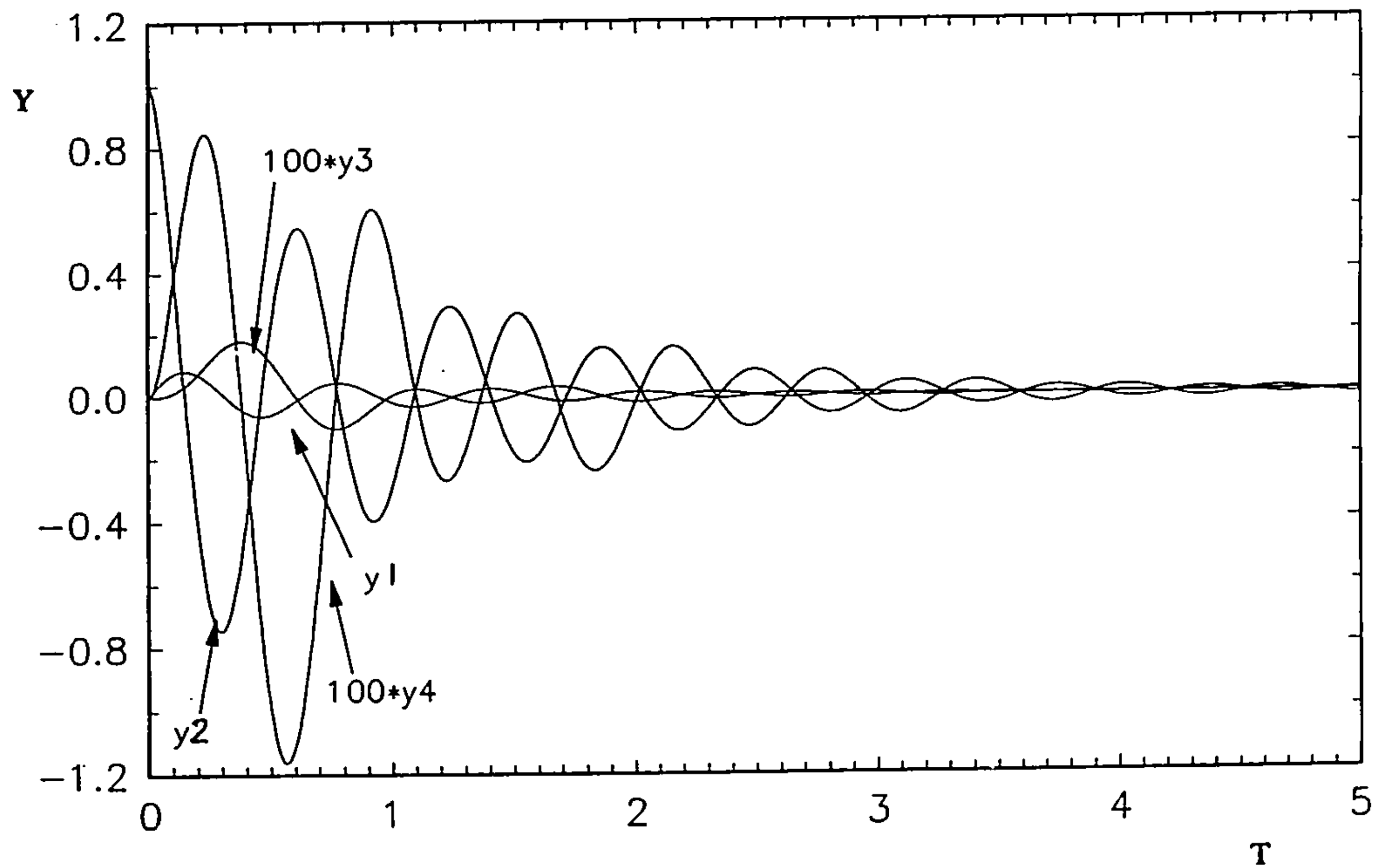


Figure A1.4: Solution trajectories for problem TP4

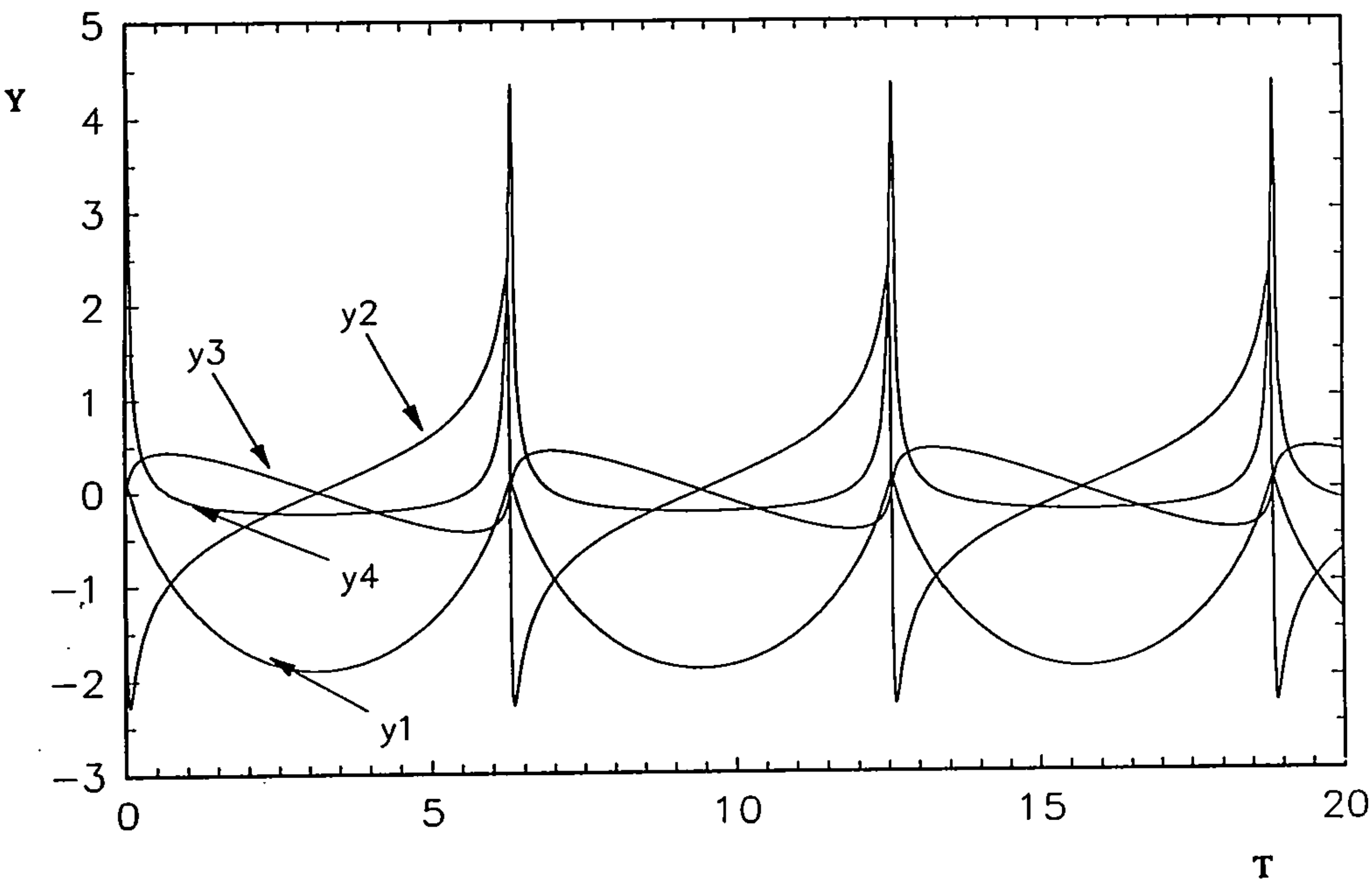
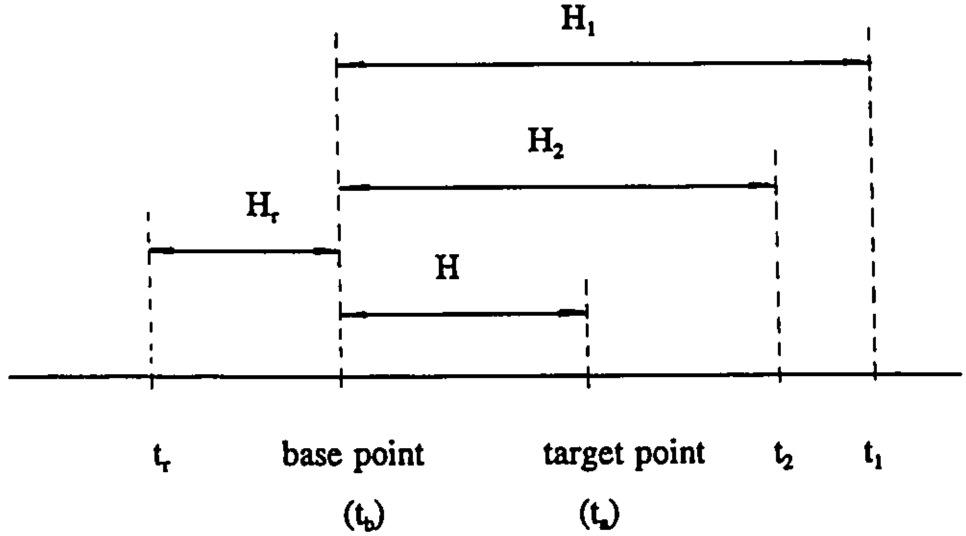


Figure A1.5 : Solution trajectories for problem TP5

Appendix 2

General Formulation of Linear Multistep Formulas

In this Appendix we present a general formulation of a particular form of the linear multistep formula that is of special relevance to the BPC and PPC approaches.



General form:

$$y_a = y_b + H\sum_{j=1}^{r} \beta_j f(t_b + H_j, y(t_b + H_j)) \qquad (*)$$

with $H_j = \sigma_j H$ ($\sigma_j < \sigma_i$ for $j > i$) and $y'(t) = f(t, y(t))$

For (*) to be r^{th} order, it must be exactly satisfied for:

(i) $y(t) = 1 \rightarrow y'(t) = 0$

(ii) $y(t) = t^{(m+1)} \rightarrow y'(t) = (m+1)t^m$ for $m = 0, 1, 2, \ldots, (r-1)$

This requirement corresponds to the requirement that

$$y(t_a) = y(t_b + H) = y(t_b) + H\sum_{j=1}^{r} \beta_j y'(t_b + H_j) \qquad (**)$$

be satisfied when y takes on the values specified by (i) and (ii). Note that $y(t) = 1$ does satisfy (**), hence (i) introduces no constraint whatsoever.

From (ii) we get:

$$(t_b + H)^{m+1} = t_b^{m+1} + H\sum_{j=1}^{r} \beta_j (m+1)(t_b + H_j)^m \quad , \quad m = 0,1,2,\ldots(r-1)$$

without loss in generality, set $t_b = 0$

$$H^{m+1} = H\sum_{j=1}^{r} \beta_j (m+1) H_j^m = H\sum_{j=1}^{r} \beta_j (m+1)(\sigma_j H)^m \ , \ m = 0,1,2,\ldots(r-1)$$

or

$$H^{m+1} = H^{m+1}\sum_{j=1}^{r} \beta_j (m+1)\, \sigma_j^m$$

or

$$1 = \sum_{j=1}^{r} \beta_j (m+1)\, \sigma_j^m \quad , \quad m = 0,1,2,\ldots(r-1)$$

This system of r equations can be written as:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 2\sigma_1 & 2\sigma_2 & 2\sigma_3 & \cdots & 2\sigma_r \\ 3\sigma_1^2 & 3\sigma_2^2 & 3\sigma_3^2 & \cdots & 3\sigma_r^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r\sigma_1^{r-1} & r\sigma_2^{r-1} & r\sigma_3^{r-1} & \cdots & r\sigma_r^{r-1} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_r \end{bmatrix}$$

or alternately:

$$\begin{bmatrix} 1 \\ 1/2 \\ 1/3 \\ \vdots \\ 1/r \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \sigma_1 & \sigma_2 & \sigma_3 & \cdots & \sigma_r \\ \sigma_1^2 & \sigma_2^2 & \sigma_3^2 & \cdots & \sigma_r^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_1^{r-1} & \sigma_2^{r-1} & \sigma_3^{r-1} & \cdots & \sigma_r^{r-1} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_r \end{bmatrix}$$

Some typical formulas for both BPC and PPC approaches are provided below.

BPC (NP=2, order=5)

Corrector:

$$y_{n-1} = y_{n-2} + h/720(11f_{n-4} - 74f_{n-3} + 456f_{n-2} + 346f_{n-1}^p - 19f_n^p)$$
$$y_n = y_{n-2} + h/90(-f_{n-4} + 4f_{n-3} + 24f_{n-2} + 124f_{n-1}^p + 29f_n^p)$$

Predictor:

$$y_{n+1} = y_n + h/720(251f_{n-4} - 1274f_{n-3} + 2616f_{n-2} - 2774f_{n-1} + 1901f_n)$$
$$y_{n+2} = y_n + h/90(269f_{n-4} - 1316f_{n-3} + 2544f_{n-2} - 2396f_{n-1} + 1079f_n)$$

BPC (NP=4, order=3)

Corrector:

$$y_{n-3} = y_{n-4} + h/12(53f_{n-2}^p - 64f_{n-1}^p + 23f_n^p)$$
$$y_{n-2} = y_{n-4} + h/3(19f_{n-2}^p - 20f_{n-1}^p + 7f_n^p)$$
$$y_{n-1} = y_{n-4} + h/4(27f_{n-2}^p - 24f_{n-1}^p + 9f_n^p)$$
$$y_n = y_{n-4} + h/3(20f_{n-2}^p - 16f_{n-1}^p + 8f_n^p)$$

Predictor:

$$y_{n+1} = y_n + h/12(5f_{n-2} - 16f_{n-1} + 23f_n)$$
$$y_{n+2} = y_n + h/3(7f_{n-2} - 20f_{n-1} + 19f_n)$$
$$y_{n+3} = y_n + h/4(27f_{n-2} - 72f_{n-1} + 57f_n)$$
$$y_{n+4} = y_n + h/3(44f_{n-2} - 112f_{n-1} + 80f_n)$$

PPC (NP=4, order=5)

Corrector:

$$y_{n-1} = y_{n-2} + h/1440(-515f_{n-5} + 1152f_{n-4} - 972f_{n-3} + 1268f_{n-2} + 507f^P_{n-1})$$
$$y_n = y_{n-2} + h/90(-f_{n-4} + 4f_{n-3} + 24f_{n-2} + 124f^P_{n-1} + 29f^P_n)$$

Predictor:

$$y_{n+1} = y_{n-2} + h/80(27f_{n-4} - 183f_{n-3} + 312f_{n-2} - 198f^P_{n-1} + 237f^P_n)$$
$$y_{n+2} = y_{n-2} + h/45(-1178f_{n-4} + 656f_{n-3} + 1284f_{n-2} - 1136f^P_{n-1} + 554f^P_n)$$

PPC (NP=8, order=3)

Corrector:

$$y_{n-3} = y_{n-4} + h/12(-f_{n-5} + 8f_{n-4} + 5f^P_{n-3})$$
$$y_{n-2} = y_{n-4} + h/3(f_{n-4} + 4f^P_{n-3} + f^P_{n-2})$$
$$y_{n-1} = y_{n-4} + 3h/4(3f^P_{n-2} + f^P_{n-1})$$
$$y_n = y_{n-4} + h/3(20f^P_{n-2} - 16f^P_{n-1} + 8f^P_n)$$

Predictor:

$$y_{n+1} = y_{n-4} + h/12(85f^P_{n-2} - 80f^P_{n-1} + 55f^P_n)$$
$$y_{n+2} = y_{n-4} + 3h(3f^P_{n-2} - 4f^P_{n-1} + 3f^P_n)$$
$$y_{n+3} = y_{n-4} + h/12(329f^P_{n-2} - 523f^P_{n-1} + 329f^P_n)$$
$$y_{n+4} = y_{n-4} + h/3(64f^P_{n-2} - 128f^P_{n-1} + 88f^P_n)$$

Bibliography

[1] Birta, L. G. and O. Abou-Rabia, "Parallel Block Predictor-Corrector Methods for ODE's", *IEEE Trans. on Computers*, Vol.C-36, No. 3, 1987, pp. 299-311.

[2] Abou-Rabia, O., L. G. Birta and M. Chen, "A Comparative Evaluation of the BPC and PPC Methods for the Parallel Solution of ode's", *Transations of SCS*, Vol. 6, No. 4, 1990, pp. 265-290.

[3] Bellen, A. and M. Zennaro, "Parallel Algorithms for Initial-Value Problems for Difference and Differential Equations", *J. Comput. Appl. Math.*, 25 (1989), pp. 341-350.

[4] Chu, M.T. and H. Hamilton, "Parallel Solutions of ODE's by Multi-Block Methods", *Siam J. Sci. Stat. Comp.*, Vol. 8, No. 3, May 1987, pp. 342-353.

[5] Tam, H. W., "Parallel Methods for the Numerical Solution of Ordinary Differential Equations", *Report No. UIUCDCS-R-89-1516*, Dept. of Computer Science, University of Illinois (Urbana-Champaign), 1989.

[6] Bellen, A., R. Vermiglio and M. Zennaro, "Parallel ODE-Solvers with Stepsize Control", *J. Comput. Appl. Math.*, 31 (1990), pp. 277-293.

[7] Watts, H. A., "Step Size Control in Ordinary Differential Equation Solvers", *Trans. Soc. for Computer Simulation*, Vol. 1, No. 1, 1986, pp. 15-25.

[8] Abou-Rabia, O. and L. G. Birta, "Some Variation on the BPC Parallel Integration Method", *Proc. 1987 Summer Computer Simulation Conference*, Montreal, PQ, July 27-30, 1987, pp. 37-42.

[9] Birta, L. G., M. Yang and O. Abou-Rabia, "An Adaptive Approach to Stepsize Control in ODE Solvers", *Applied Numerical Math.*, (accepted for publication).

[10] Shampine, L. F. and M.K. Gordon, "Computer Solution of Ordinary Differential Equations: The Initial Value Problem", W. H. Freeman, 1975.

[11] Franklin, M. A., "Parallel Solution of Ordinary Differential Equations", *IEEE Trans. Comput.*, Vol.C-27, 1978, pp. 413-420.

[12] Shampine, L.F. and H.A. Watts, "Block Implicit One-step Methods" *Math. Comput.*, Vol. 23, 1969, pp. 731-740.

[13] Miranker, W. L. and W. M. Liniger, "Parallel Methods for the Numerical Integration of Ordinary Differential Equations", *Math. Comput.*, Vol. 21, 1967, pp. 303-320.

[14] Abou-Rabia, O., "Multiprocessing in Continuous System Simulation", Ph.D. thesis, Department of Electrical Engineering, University of Ottawa, 1985.