1 Efficient Defect Control for IVODES based on multistep Hermite-Birkhoff interpolants

1.1 Introduction

In this chapter, we present an efficient defect control technique for use in the numerical solution of initial value ordinary differential equations (ODE). In this chapter, we will assume that the underlying numerical method is a Runge-Kutta (RK) method since the vast majority of the literature on the use of defect control for numerical solution of IVODEs focuses on RK methods. Solvers based on Runge-Kutta methods only provide a discrete numerical solution. They effectively adaptively divide the time domain into steps and return an estimate of the solution at the end of each step. To get a continuous solution approximate, the user has to fit an interpolant over the whole region.

The issue is that there is no guarantee that the interpolant will be as accurate as the discrete solution calculated by the solver. Thus if the solver returned a solution that satisfied a tolerance of 10^{-i}, there are no guarantee that in the middle of a step, the interpolant will also deliver approximate solution values whose accuracy is approximately 10^{-i}.

High quality contemporary IVODE solvers typically have a built-in interpolant that provides a continuous solution approximation. However the solvers typically do not provide any type of explicit control of the accuracy of the continuous solution approximate. We show in Section 1.1.2, that even the robust IVODE solvers in Python, that using interpolation does not guarantee an accurate solution.

There has been some work towards addressing this issue in the area of control of the defect of the continuous solution approximation. The defect is the amount by which the continuous solution approximation fails to satisfy the IVODE. There has been a number of papers on the subject of defect control of the continuous solution and we discuss them in Section 1.1.3.Typically, the interpolants employed in algorithms for defect control are based on the use of Continuous Runge-Kutta methods and the computational costs are substantial.

In this paper we will discuss an efficient method for defect control of the continuous solution using Hermite Birkhoff interpolants. We will control an estimate of the maximum defect along a step and show how that yields a continuous solution approximation over the whole time domain whose defect is typically within the tolerance.

We start by discussing the ODE problems we will use to demonstrate our technique in Section 1.1.1. We then give an overview of the issue with using error control only at the end of the step in Section 1.1.2. We discuss related work in Section 1.1.3. We give a description of the solvers that we use in Section 1.1.4.

We discuss several multistep interpolants that we have constructed based on a Runge-Kutta method of order 4 in Section 1.2 and extend them to higher order Runge-Kutta methods in Section 1.3. We then discuss possible solutions

to a fundamental issue with our approach in Section $A.1$.

1.1.1 Test Problems

In this section, we discuss the three problems that we use.

The first problem haa the ODE:

$$y'(t) = -\frac{y^3(t)}{2} \tag{1}$$

The initial condition is $y(0) = 1$ and the time domain is $[0, 10]$.

The solution to this problem is

$$y(t) = \frac{1}{\sqrt{1+t}}. \tag{2}$$

as shown in Figure 1.

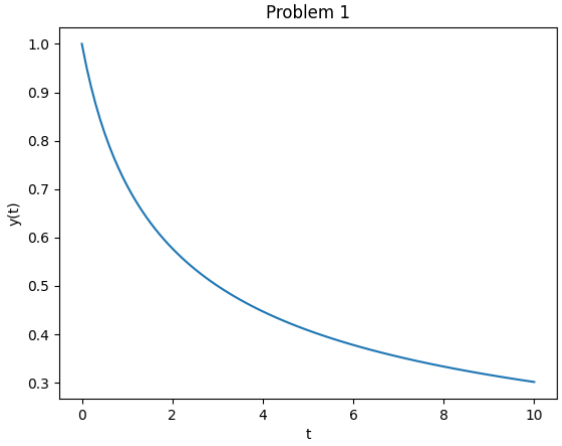

Figure 1: Solution to the first ODE problem

The second problem has the ODE:

$$y'(t) = \frac{y(t)(1 - \frac{y(t)}{20})}{4}. \tag{3}$$

The initial condition is $y(0) = 1$ and the time domain is $[0, 10]$.

The solution to this problem is

$$y(t) = \frac{20e^{\frac{t}{4}}}{e^{\frac{t}{4}} + 19}, \tag{4}$$

as shown in Figure 2.

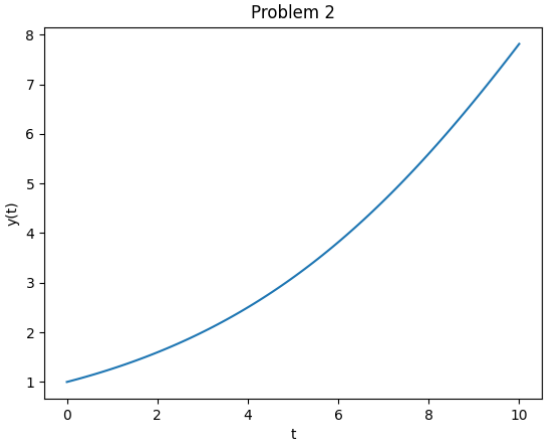

Figure 2: Solution to the second ODE problem

The third problem has the ODE:

$$y'(t, y) = -0.1y - e^{-0.1t}\sin(t) \tag{5}$$

The initial condition is $y(0) = 1$ and the time domain is $[0, 10]$.

The solution to this problem is

$$y(t) = e^{-0.1t}\cos(t), \tag{6}$$

as shown in Figure 3.

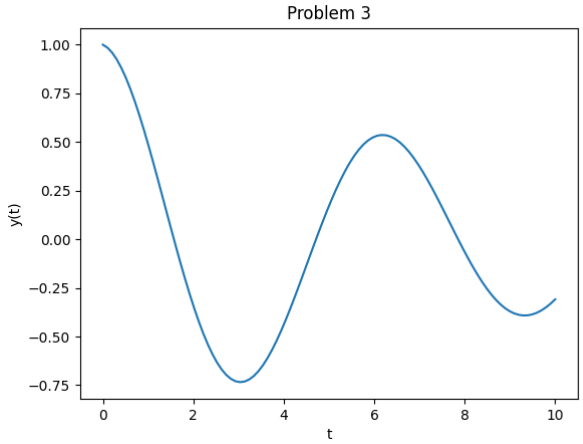

Figure 3: Solution to the third ODE problem

1.1.2 Absence of control of the continuous solution approximation for typical IVODE solvers

In this section, we discuss how conventional solvers employed in popular packages like the Scipy library solve IVODE problems. These libraries will often have the option of using an error control solver based on a Runge-Kutta pair which works as follows. The pair comes with a low order method and a high order method and will solve an ODE by taking a sequence of steps. The solver will take each step with both methods and use the difference between the higher order method and the lower order method to generate an error estimate for the discrete numerical solution at the end of the step. If the error estimate is within the user provided tolerance, the solver will accept the step and proceed to the next step. If the error estimate is not within the tolerance, the solver will reduce the step-size and attempt to take the step again. As the error of Runge-Kutta methods depends on the step-size, a smaller step-size will produce a smaller error. The solver will keep reducing the step-size until the user tolerance is satisfied and will then proceed to the next step. In an attempt to improve the efficiency, solvers will also increase the step-size when the estimated error is significantly smaller than the tolerance.

An issue with this approach is that the error estimate is computed only for the discrete numerical solution computed at the end of a step and thus the error control is only applied at the end of the step. An interpolant that provides a continuous solution approximation across the entire step is typically constructed by the solver and it is assumed that the error of the interpolant at points within the step is within the user-provided tolerance. However, as we will show below, this is sometimes not the case.

For this interpolant to provide a solution within the user provided tolerance, the interpolation error must be less than or equal to the error of the data being fitted. Thus we would ideally use an interpolant of at least order $O(h^p)$ if the discrete solution is of order $O(h^p)$. However, for high order methods, it becomes too expensive to construct an interpolant of the appropriate order. These solvers will thus usually compromise and employ a lower order interpolant that is less costly.

Constructing a continuous approximate solution that satisfies the user-provided tolerance is thus a challenge as we are not guaranteed that the continuous approximate solution is error-controlled and not guaranteed that the interpolation error will not affect our solution. Below are some results obtained when Runge-Kutta solvers are applied to the 3 test problems introduced in the previous section. The solvers apply error control to the discrete numerical solution at the end of the step but no error control of any type to the continuous numerical solution across the step.

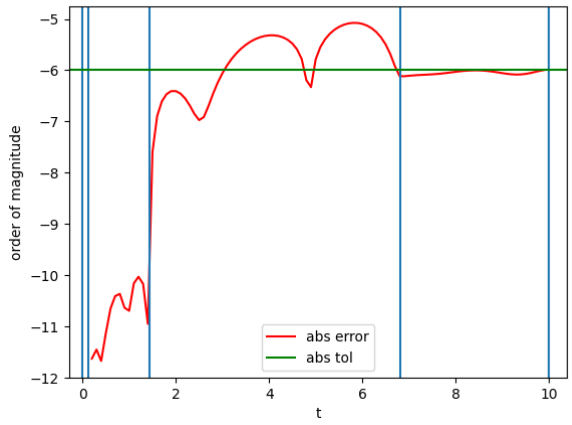

Figure 4: Scipy 'DOP853' on problem 2 with an absolute tolerance of 10^{-6} and a relative tolerance of 10^{-6}. Steps are represented by the vertical lines.

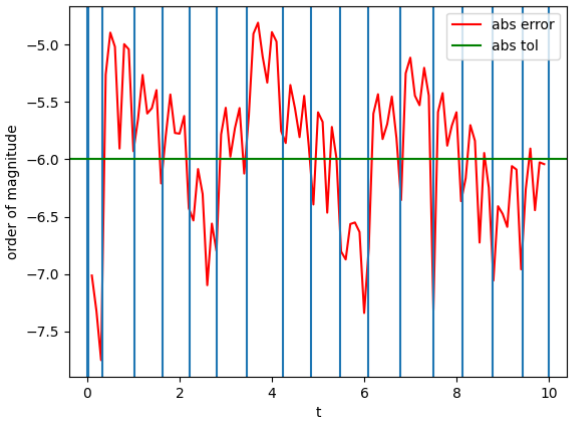

Figure 5: Scipy 'RK45' on problem 3 with an absolute tolerance of 10^{-6} and a relative tolerance of 10^{-6}. Steps are represented by the vertical lines.

Figure 4 and 5 shows the errors in the middle of each step obtained when applying 'DOP853' to problem 2 and 'RK45' to problem 3 with an absolute tolerance of 10^{-6} and a relative tolerance of 10^{-6}. We can see that the solution values at the end of each step typically satisfy the tolerance. However we can clearly see that the solution values (obtained from the built-in interpolant constructed by the solver) at points within each steps, have errors up to one order of magnitude larger than the tolerance.

We note that when a user asks for a solution whose estimated error is within a tolerance of 10^{-i}, they expect the solution to have an estimated error that is within that tolerance for the whole time domain. However, the decision to not satisfy the user provided tolerance throughout the step is made in the interest of efficiency and the loss of accuracy in the middle of the step is the a tradeoff. The goal of modern IVODE solvers is to provide a continuous solution approximation across the whole time domain. The expectation is that the solution approximations across each step is accurate as ODE solvers can be integral part of larger software packages where their approximate solutions are differentiated, integrated and manipulated in ways such that a sufficiently accurate continuous approximate solution is required.

In this chapter, we attempt to provide an efficient way of constructing interpolants that can then be used to control the defect of the continuous approximate solution across the step and thus throughout the whole time domain.

1.1.3 Defect Control and the cost of traditional Continuous Runge-Kutta method

In this section we introduce the 'defect' of a continuous approximate solution of an ODE and explain how the control of that defect provides a type of error control of the continuous numerical solution.

In the context of numerical ODEs, the defect, denoted by $\delta(t)$, is the amount by which the continuous numerical solution, $u(t)$, fails to satify the ODE. When the ODE is $y'(t) = f(t, y(t))$, the defect is

$$\delta(t) = |u'(t) - f(t, u(t))|. \tag{7}$$

Calculating the defect requires that the continuous approximate solution computed by the solver to also be differentiable. The idea of defect control is relatively new as differentiable solutions to ODEs can be expensive to calculate. One such attempt is outlined in 8888 reference to Wayne Enright CRK defect control 8888. In this paper, the defect control method employs a continuous RK method for which the number of stages grows exponentially with the order of the method as shown in Table 1. In this paper we will compute a defect controlled continuous solution with no additional cost. A typical Runge-Kutta solver will thus be able to employ the usual number of stages required for the discrete Runge-Kutta method and still produce an accurate continuous solution.

Though the defect is defined for the whole step, the estimation of the maximum defect within a step is what is important. If the maximum defect is within the tolerance, then the defect of the whole solution within the given step is within the tolerance. The key problem is to find the maximum defect within the step. An idea would be to sample the defect at several points and use the maximum value sampled. The problem with this approach is that each sampling of the defect requires an additional function evaluation. Thus we should not do too many samples. Enright's work involves constructing special interpolants that guarantee that the asymptotically maximum defect is at the same location

Table 1: Number of stages for discrete vs continuous RK method

order	discrete	continuous	asymptotically correct defect
4	4	4	8
5	5	6	12
6	6	7	15
7	7	9	20
8	8	13	27

within a step. This way only one function evaluation is required to sample the defect to obtain an estimate of the maximum defect. This is referred to as asymptotically correct defect control.

In the approach outlined in this chapter, we have observed experimentally that the maximum defect will tend to appear at one of two locations within the step. Thus in our approach, only two defect samplings must be done to get the maximum defect. Though we make an additional function evaluation compared to the asymptotically correct defect control, using less stages to construct the interpolant guarantees that our method is more efficient especially for higher orders.

1.1.4 Overview of our approach

In this chapter, we will discuss simple IVODE solvers based on discrete Runge-Kutta methods of order 4, 6 and 8 and show that we can provide accurate continuous interpolant to augment the discrete solution computed by these RK methods without having to compute any additional stages. In this section, we give an outline of the approach.

The first Runge-Kutta method upon which we build a defect control solver is the classical 4^{th} order method that uses 4 stages, the second method is a Verner 6^{th} order method, taken from his 6(5) pair 8888 need to look for a reference 8888, that uses 9 stages and the last method is a Verner 8^{th} order method from an 8(7) pair that uses 13 stages, 8888 Need a reference for that as well 8888.

The solvers that we have written use a simple step selection strategy. If the estimated maximum defect is greater than tol, the solver rejects the step and attempts to retake it again with half the step-size. If the estimated maximum defect is less than $0.1tol$, the solver accepts the step and doubles the step-size for the next step. We will elaborate on the initial step-size used by each solver later in the chapter.

We now note that the solver is not optimised. A more thorough analysis of how the solver behaves and thus a more refined step selection algorithm will produce a faster solver in practice. The code we consider in this chapter only serves as a proof of concept for a more elaborate solver.

1.2 Multistep interpolants for zero-cost defect control on a single step Runge Kutta method

In this section we will consider a multistep interpolant approach built on the classical 4^{th} order Runge Kutta method (RK4) that allows for defect control. We will augment the discrete Runge-Kutta solution with interpolants of 4^{th}, 6^{th} and 8^{th} orders respectively and explain the challenges and the gain in efficiency and accuracy of each interpolant separately.

We first note that Runge-Kutta methods are very convenient in that they are one step methods. At any point, when taking the next step, the method does not have to take into consideration the size of the previous steps and this is convenient as the solver can choose the size of the next step in the most optimal way only based on how the error estimate has satisfied the tolerance for the current step.

The first interpolant that we discuss is a single step interpolant. It is the classical Hermite cubic of order 4. We will show how this interpolant can be used to perform defect control and discuss the limitations of this interpolant.

We then show how a Hermite-Birkhoff interpolant of 6^{th} order can address the issues that we identified for the 4^{th} order interpolant. We will give an overview of how a 6^{th} order interpolant is derived and show the results of applying defect control using this interpolant to solve the three test problems.

We will then use a similar approach to derive an 8^{th} order Hermite-Birkhoff interpolant and show why the approach used for the 6^{th} order interpolant needs to be modified for the 8^{th} order. We then discuss another approach to derive an 8^{th} order interpolant that addresses the issue with the previous one and show the results of using this 8^{th} order interpolant as the basis for computing defect controlled numerical solutions of the three problems.

1.2.1 The Classical 4^{th} order RK method with a 4^{th} order Hermite Cubic Interpolant

The Hermite cubic spline is a very widely used interpolant. The basic idea is that we can use the derivative data at the data points and not just the solution values to get more data to fit an interpolant. Given points (t_i, y_i) and (t_{i+1}, y_{i+1}) with derivatives f_i and f_{i+1} (here $f_i = f(t_i, y_i)$ and $f_{i+1} = f(t_{i+1}, y_{i+1})$), respectively, the interpolant across $[t_i, t_{i+1}]$ of size h_i is defined as:

$$u(t_i + \theta h) = h_{00}(\theta)y_i + h_i h_{10}(\theta)f_i + h_{01}(\theta)y_{i+1} + h_i h_{11}(\theta)f_{i+1}, \tag{8}$$

and its derivative is:

$$u'(t_i + \theta h) = h'_{00}(\theta)y_i/h_i + h'_{10}(\theta)f_i + h'_{01}(\theta)y_{i+1}/h_i + h'_{11}(\theta)f_{i+1}. \tag{9}$$

the quantity θ is:

$$\theta = (t - t_i)/h_i. \tag{10}$$

(We note that we need to divide by h_i because of the the derivative of the cubics are defined with respect to θ and thus we need to apply the chain rule.)

The functions $h_{00}(\theta)$, $h_{01}(\theta)$, $h_{10}(\theta)$ and $h_{11}(\theta)$ are each cubics defined such that $u(t_i) = y_i$, $u'(t_i) = f_i$, $u(t_{i+1}) = y_{i+1}$ and $u'(t_{i+1}) = f_{i+1}$. When θ is 0, $t_i + \theta h_i$ is t_i and thus only $h_{00}(0)$ should be 1 and all the others cubic should evaluate to 0. Also only $h'_{10}(0)$ should be 1 and the derivatives of all the other cubics should evaluate to 0. When θ is 1, $t_i + \theta h_i$ is t_{i+1} and thus only $h_{01}(1)$ should be 1 and all the other cubics should evaluate to 0. Also only $h'_{11}(1)$ should be 1 and the derivatives of all the other cubics should be 0.

We will assume that each of the cubics has the form $a\theta^3 + b\theta^2 + c\theta + d$ and note that for each cubic we know its value for θ at 0 and 1 and the values of its derivatives for θ at 0 and 1. We thus have 4 equations for each cubic. Thus we can solve the system for each cubic to get the values of a, b, c and d for each cubic.

We know note that from equations 8 and 10, we can evaluate both the interpolant and its derivative for any θ in $[t_i, t_{i+1}]$ and therefore we can form $\delta(t + \theta h_i) = u'(t_i + \theta h_i) - f(t_i + \theta h_i, u(t_i + \theta h_i))$ which can be used to sample the defect for any θ.

We will show below experimentally that the maximum defect occurs consistently approximately either at $x_i + 0.2h_i$ or at $x_i + 0.8h_i$ so that we just need to sample the defect twice in order to obtain an estimate of the maximum defect on the step.

We now note that the interpolant comes at no additional cost. We only need (x_i, y_i, f_i) and $(x_{i+1}, y_{i+1}, f_{i+1})$ to be stored. No additional stages or function evaluations are required for the construction of the interpolant itself.

For the remainder of this chapter, we will refer to the Hermite cubic interpolant as 'HB4'.

Problem 1 results Figures 6, 7 and 8 shows the results of using the RK4 with HB4 on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect estimate within the step and this can be observed to occur at $0.2h$ and $0.8h$ for a step of size h. See Figure 8, to see the scaled defect reaching a maximum near these points. (The figure shows the shape of the defect for each of the steps that were taken to solve Problem 1 using RK4 with HB4. All the defects were scaled vertically to be in the range [0, 1] and scaled horizontally so that they map onto [0, 1]. We see that over all steps and problems, the defect has two clear peaks at $0.2h$ and $0.8h$.) We note that we are able to successfully control the defect of the continuous numerical solution using this approach; see Figure 6. To obtain these results, we sampled the defect of many points within each step.

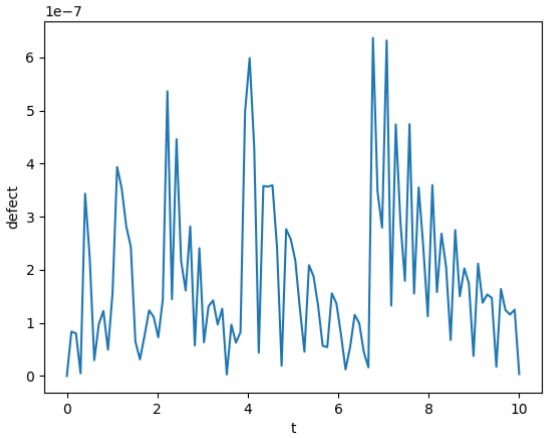

Figure 6: Defect across the entire domain of RK4 with HB4 on problem 1 at an absolute tolerance of 10^{-6}.

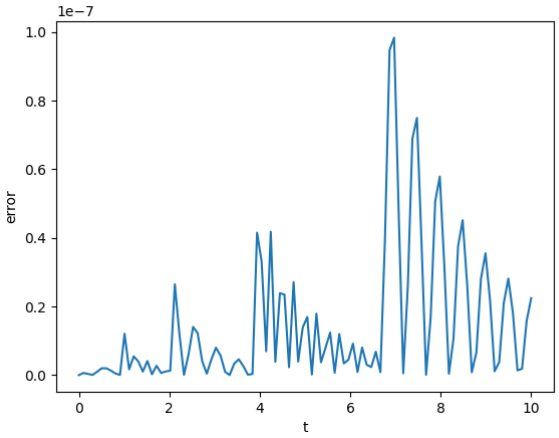

Figure 7: Global Error of RK4 with HB4 on problem 1 at an absolute tolerance of 10^{-6}.

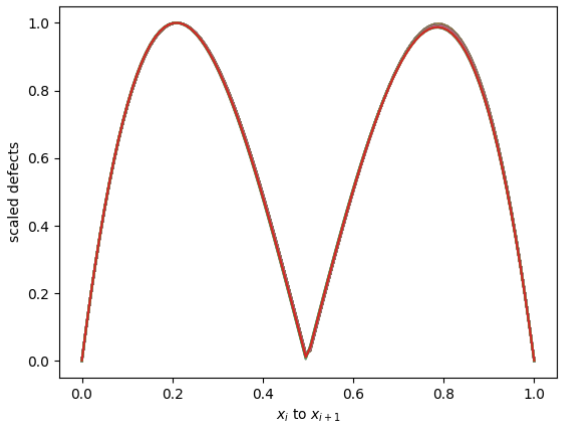

Figure 8: Scaled defects over all steps taken RK4 with HB4 on problem 1 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 2 results Figures 9, 10 and 11 shows the results of using RK4 with HB4 on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect estimate within the step and this can be observed to occur at $0.2h$ and $0.8h$ for a step of size, h. See Figure 11, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach; see Figure 9. For Problem 2, the defect gets noisy on small steps and we do not get two clean peaks. However, we note that we quite consistently get the maximum defects at $0.2h$ and $0.8h$ and thus we only require two defect samplings.

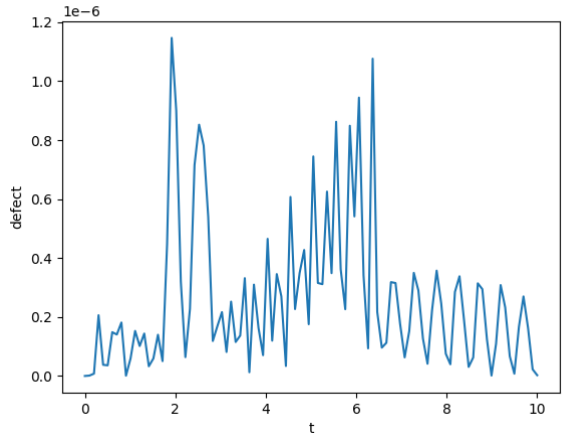

Figure 9: Defect across the entire domain of RK4 with HB4 on problem 2 at an absolute tolerance of 10^{-6}.

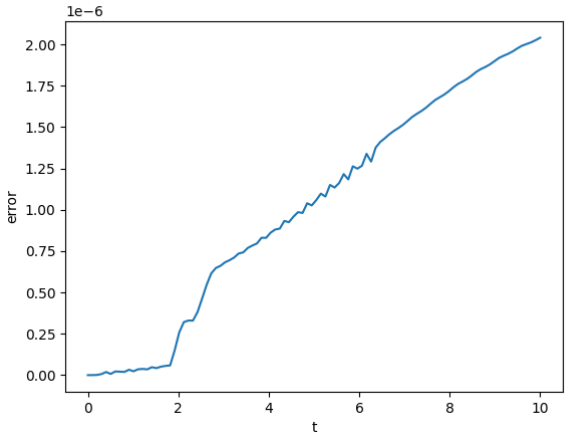

Figure 10: Global Error of RK4 with HB4 on problem 2 at an absolute tolerance of 10^{-6}.

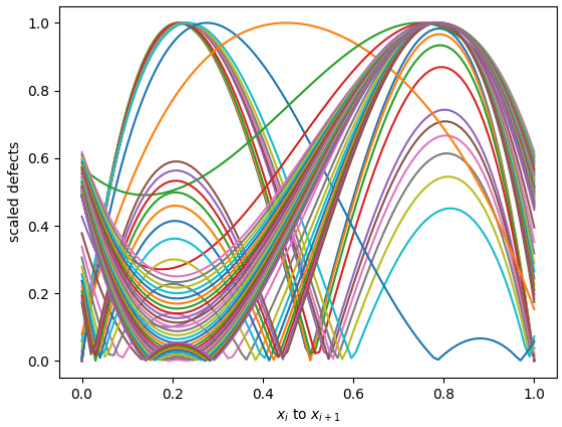

Figure 11: Scaled defects over all steps taken RK4 with HB4 on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

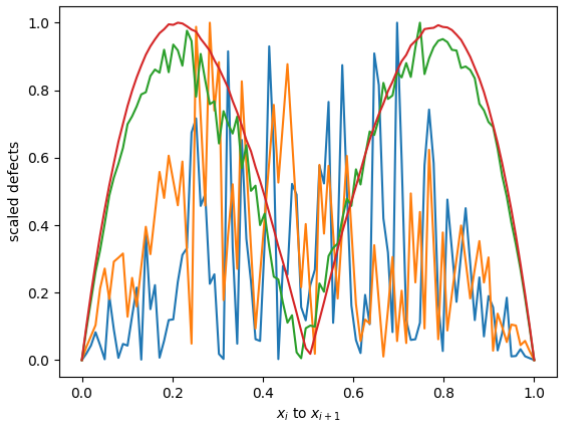

Figure 12: Scaled defects small steps taken RK4 with HB4 on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 3 results Figures 13, 14 and 15 shows the results of using the modification of RK4 with HB4 on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.2h$ and $0.4h$ along a step of size, h. See Figure 15, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 13.

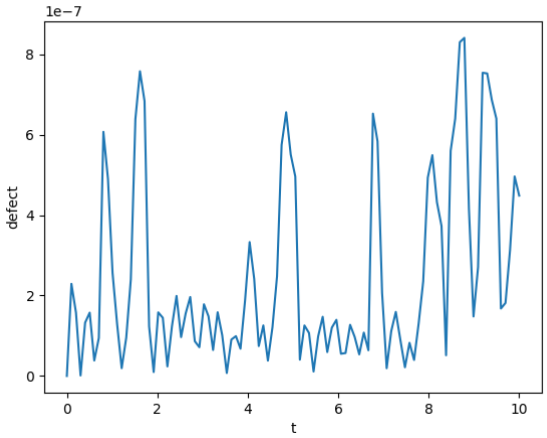

Figure 13: Defect across the entire domain of RK4 with HB4 on problem 3 at an absolute tolerance of 10^{-6}.

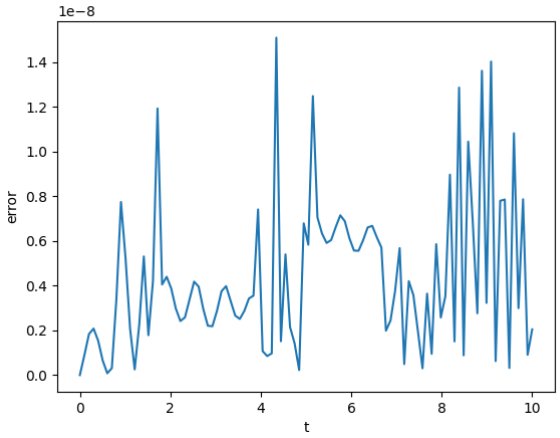

Figure 14: Global Error of RK4 with HB4 on problem 3 at an absolute tolerance of 10^{-6}.

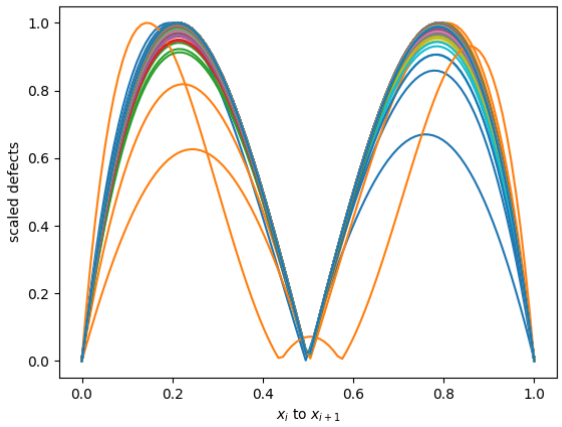

Figure 15: Scaled defects over all steps taken RK4 with HB4 on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Efficiency data and discussion of the interpolation error We now present the number of steps that were taken by the solver to solve each problem along with the number of successful steps.

Table 2: Number of steps taken by RK4 using defect control with HB4

Problem	successful steps	total steps
1	88	88
2	59	62
3	225	232

The Hermite cubic, HB4, is an interpolant of 4^{th} order. However, to perform defect control we need to calculate the derivative of this interpolant. Since we are differentiating the interpolant, the order of the derivative is 3. The numerical solution is of order 4 as we are using the classical rk4 and thus the derivative of the interpolant is less accurate than the ODE solution. We need a way to get an interpolant whose derivative is at least of order 4 so that the error of the derivative of the interpolant is not larger than the error of the discrete numerical solution obtained from the RK4 method.

To do that, in the next section, we will introduce a new interpolation scheme based on a Hermite-Birkhoff interpolant which is of 6^{th} order and will thus have a derivative of order 5.

1.2.2 RK4 with a 6^{th} order Hermite-Birkhoff interpolant

We first start by noting that this interpolant is a multistep interpolant as it depends on the previous step.

Suppose that the step taken by a solver to go from t_i to t_{i+1} is of size h. We can define the size of the step from t_{i-1} to t_i by using a weight α such that the size of the step from t_{i-1} to t_i is αh. Then given the solution values and the derivative values at all the three points, i.e, $(t_{i-1}, y_{i-1}, f_{i-1})$, (t_i, y_i, f_i) and $(t_{i+1}, y_{i+1}, f_{i+1})$, we can fit a two-step quintic interpolant of order 6 defined as such:

$$\begin{aligned} u(t_i + \theta h) = d_0(\theta)y_{i-1} + h_i d_1(\theta)f_{i-1} \\ +d_2(\theta)y_i + h_i d_3(\theta)f_i + d_4(\theta)y_{i+1} + h_i d_5(\theta)f_{i+1}, \end{aligned} \tag{11}$$

and its derivative is

$$\begin{aligned} u'(t_i + \theta h) = d_0'(\theta)y_{i-1}/h_i + d_1'(\theta)f_{i-1} \\ +d_2'(\theta)y_i/h_i + d_3'(\theta)f_i + d_4'(\theta)y_{i+1}/h_i + d_5'(\theta)f_{i+1}. \end{aligned} \tag{12}$$

As before, θ is:

$$\theta = (t - t_i)/h_i. \tag{13}$$

This time θ is allowed to vary between -α and 1 such that $t_i + \theta$h is t_{i-1} when θ is $-\alpha$, t_i when θ is 0 and t_{i+1} when θ is 1.

Each of $d_0(\theta)$, $d_1(\theta)$, $d_2(\theta)$, $d_3(\theta)$, $d_4(\theta)$, and $d_5(\theta)$ is a quintic of the form $a\theta^5 + b\theta^4 + c\theta^3 + d\theta^2 + e\theta + f$ where the six coefficients for each can be found involving α by solving a linear system of 6 equations in terms of α. The six equations are obtained as follows. First for $\theta = -\alpha$, only $d_0(\theta)$ evaluates to 1 and all the other quintic polynomials evaluate to 0 as $u(t_i - \alpha h) = u(t_{i-1}) = y_{i-1}$. Also at this θ, only the derivative of $d_1(\theta)$ evaluates to 1 and all the other quintic polynomials' derivatives evaluate to 0 as $u'(t_i - \alpha h) = u'(t_{i-1}) = f_{i-1}$. When θ is 0, only $d_2(\theta)$ evaluates to 1 and all the other polynomials evaluate to 0 as $u(t_i - 0(h)) = u(t_i) = y_i$. Also at this θ value, only the derivative of $d_3(\theta)$ evaluates to 1 and all the other quintic polynomial derivatives evaluate to 0 as $u'(t_i - 0(h)) = u'(t_i) = f_i$. When θ is 1, only $d_4(\theta)$ evaluates to 1 and all the other polynomials evaluate to 0 as $u(t_i + 1(h)) = u(t_{i+1}) = y_{i+1}$. Also at this θ, only the derivative of $d_5(\theta)$ evaluates to 1 and all the other quintic polynomial derivatives evaluate to 0 as $u'(t_i - 1(h)) = u'(t_{i+1}) = f_{i+1}$. With these six conditions, we can get six equations for each quintic in terms of α and, using a symbolic management package, we can solve all of these to find the six quintic polynomials. (Their coefficients will be given in terms of α)

We again note that as the solver is stepping across the problem domain, these interpolants are constructed for no additional cost in terms of evaluation of $f(t, y(t))$. We just need to store the 3 data points $(t_{i-1}, y_{i-1}, f_{i-1})$, (t_i, y_i, f_i) and $(t_{i+1}, y_{i+1}, f_{i+1})$. We will also observe that the defect peaks at two positions within the new step, $[t_i, t_{i+1}]$, and thus, we can find the maximum defect by only sampling twice. This technique essentially provides an efficient defect control of a continuous approximate solution.

The interpolant defined as above will now be referred to as 'HB6' for the remainder of this chapter. We note that it is of order 6 and its derivative is of order 5.

We now note that for RK4 as the solution values are only accurate to 4^{th} order, and we would want the derivative of the interpolant to be of order 4 or higher so that interpolation error is relatively negligible. This scheme satisfies this condition and we will see below how this allows us to take fewer time steps to solve a given problem.

Problem 1 results Figures 16, 17 and 18 shows the results of using RK4 with HB6 on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 18, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach; see Figure 16.

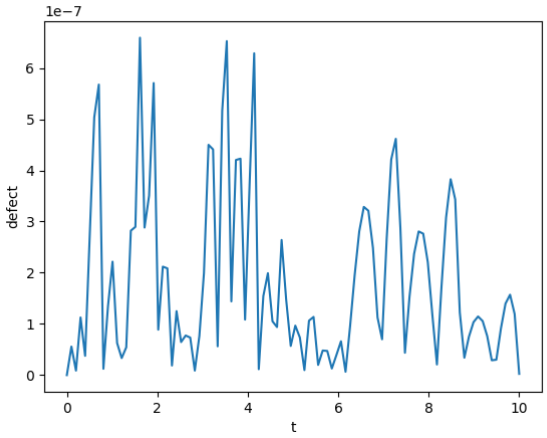

Figure 16: Defect across the entire domain of RK4 with HB6 on problem 1 at an absolute tolerance of 10^{-6}.

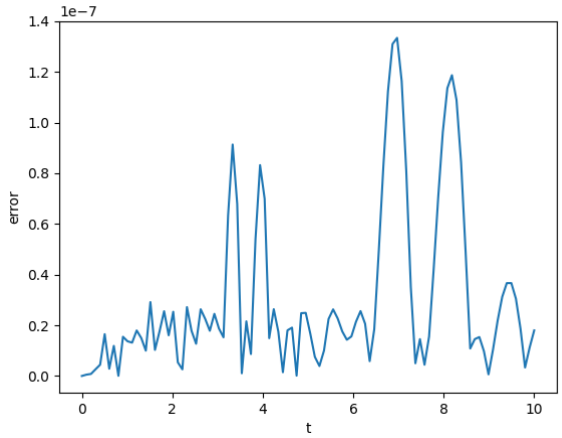

Figure 17: Global Error of RK4 with HB6 on problem 1 at an absolute tolerance of 10^{-6}.

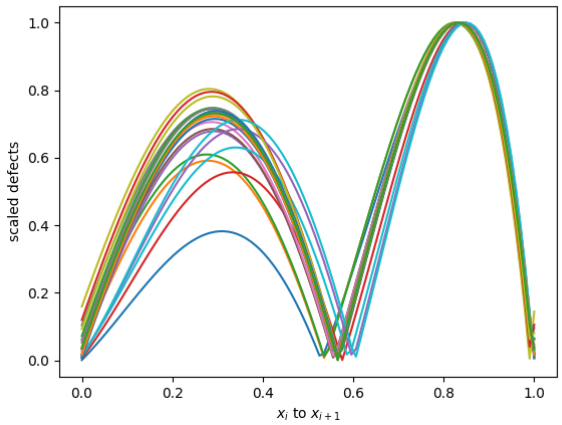

Figure 18: Scaled defects of RK4 with HB6 on problem 1 at an absolute tolerance of 10^{-6} mapped into $[0, 1]$.

Problem 2 results Figures 19, 20 and 21 shows the results of using the modification of RK4 with HB6 on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 21, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 19. For Problem 2, the defect gets noisy on small steps and

we do not get two clean peaks. However, we note that we quite consistently get the maximum defects at $0.4h$ and $0.8h$ and thus we only require two samplings.

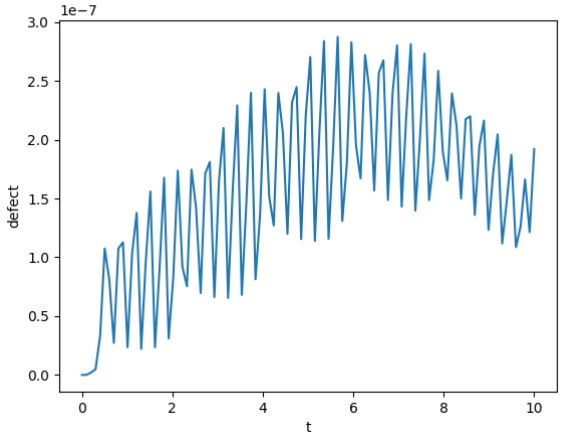

Figure 19: Defect across the entire domain of RK4 with HB6 on problem 2 at an absolute tolerance of 10^{-6}.

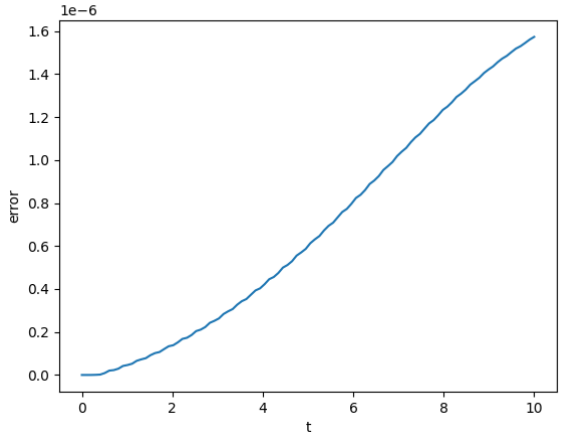

Figure 20: Global Error of RK4 with HB6 on problem 2 at an absolute tolerance of 10^{-6}.

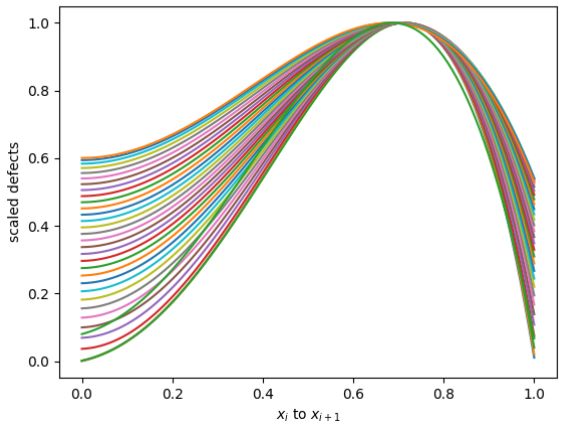

Figure 21: Scaled defects of RK4 with HB6 on problem 2 at an absolute tolerance of 10^{-6} mapped into $[0, 1]$.

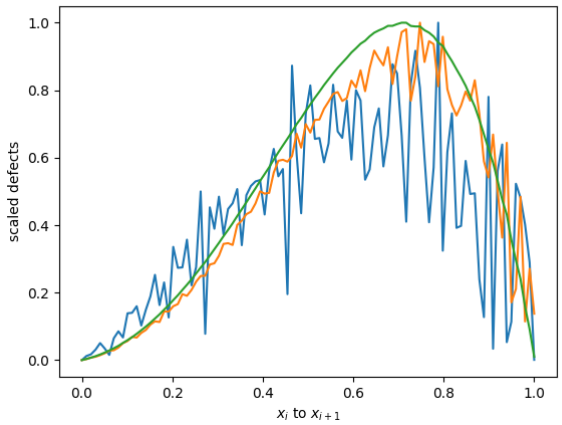

Figure 22: Scaled defects of RK4 with HB6 on small steps on problem 2 at an absolute tolerance of 10^{-6} mapped into $[0, 1]$.

Problem 3 results Figures 23, 24 and 25 shows the results of using the modification of RK4 with HB6 on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 25, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 23.

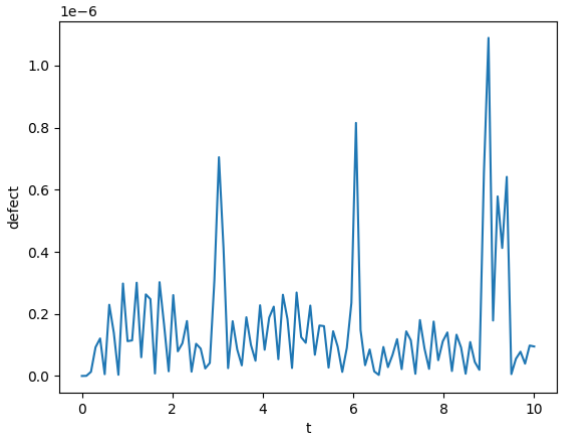

Figure 23: Defect across the entire domain of RK4 with HB6 on problem 3 at an absolute tolerance of 10^{-6}.

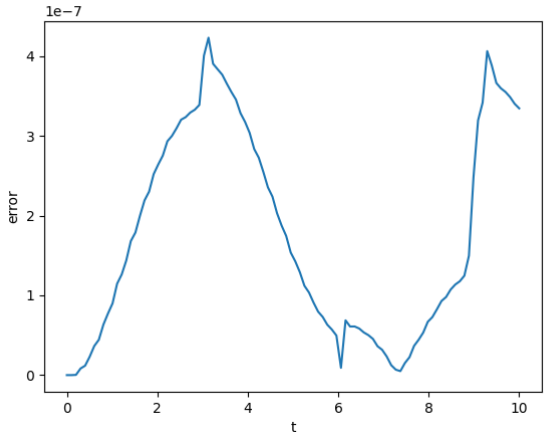

Figure 24: Global Error of RK4 with HB6 on problem 3 at an absolute tolerance of 10^{-6}.

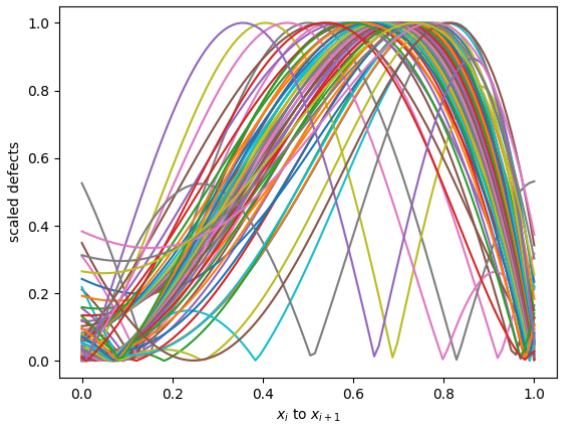

Figure 25: Scaled defects of RK4 with HB6 on problem 3 at an absolute tolerance of 10^{-6} mapped into $[0, 1]$.

We note that the defects are not as clean they were in the case with HB4. There are two peaks most of the time at around $0.4h$ and $0.8h$ but as was the case in the third problem, the peak sometimes appears at $0.6h$. However, we can see that the defect is still being controlled. We will also see that it is twice as fast as it uses around half the number of steps as HB4.

Table 3: Number of steps taken by RK4 when modified to do defect control with HB6 vs when modified with HB4

Problem	succ. steps HB4	succ. steps HB6	nsteps HB4	nsteps HB6
1	88	27	88	27
2	59	36	62	40
3	225	62	232	73

Table 3 shows how the number of steps is less than half when we use HB6 as opposed to HB4. This is entirely because the interpolation error is less than the actual ODE error especially for the derivative.

Issues with α values The issue with this scheme is that the interpolant is a multistep interpolant while the Runge-Kutta method is a one step method. The Hermite Birkhoff interpolant, HB6, is based on two steps and the parameter α defines how big the previous step is compared to the actual step. The error term in the Hermite-Birkhoff interpolant is minimised when the size of the two steps are the same size, i.e, when α is 1. The error term is proportional to $(t + \alpha)^2 t^2 (t + 1)^2$. When α differs from 1, the accuracy of the interpolant is

reduced.

In Figure 26, we perform a simple experiment to show how this scheme depends on the value of α. We place 3 data points along the t-axis such that their distance apart are αh and h for several values of α and we vary h from 1 to 10^{-7}; we then report on the order of the maximum defect at each h for these values.

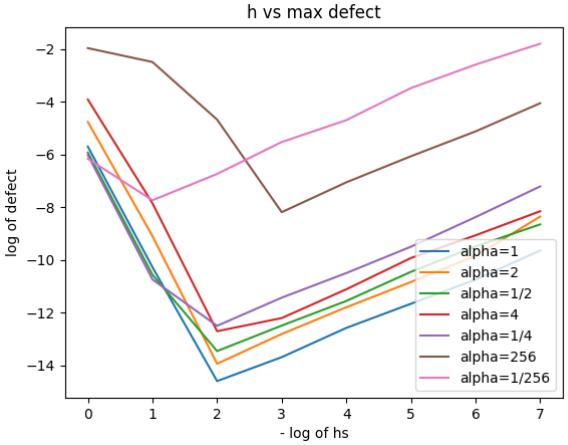

Figure 26: HB6 maximum order of accuracy based on different values of alpha

Figure 26 shows that at $\alpha = 2$, $\frac{1}{2}$, 4 and $\frac{1}{4}$, the errors are comparable to those that we get when α is at 1. However, we see that $\alpha = 256$ and $\frac{1}{256}$, we cannot get high accuracy.

We note that in solving the 3 problems above, α is very rarely bigger than 4 or smaller than $\frac{1}{4}$ and thus, we can be satisfied with the approach that we have considered in this section. We discuss the situation further in Section 1.4. We note that in order for the results given in Figure 26 to be relevant, we would have to be considering a very sharp tolerance since for reasonable values of α, e.g, 2, $\frac{1}{2}$, 4 and $\frac{1}{4}$, the interpolants all deliver very small defects of 10^{-12} to 10^{-14}.

Another idea would be to use an even higher order interpolant so as to reduce the interpolation error more. We note that with the Hermite-Birkhoff scheme there is no additional cost to get use higher order interpolants. In the next section, we discuss an 8^{th} order interpolant and show how to derive such an interpolant.

1.2.3 RK4 with an 8^{th} order Hermite-Birkhoff interpolant

Derivation of HB8 In this section, we discuss a derivation of an 8^{th} order interpolant. To derive an 8 order interpolant, we need 4 data points over 3 steps. We need the data points to be (t_i, y_i, f_i) as well as the two previous steps

$(t_{i-1}, y_{i-1}, f_{i-1})$ and $(t_{i-2}, y_{i-2}, f_{i-2})$ and the next step, $(t_{i+1}, y_{i+1}, f_{i+1})$. We present two schemes:

- The first scheme when the parameters α and β are associated with the previous two steps, so the three steps are of the size βh, αh and h respectively. Thus the scheme establishes the base step-size based on the third step.
- The second scheme when the parameters α are associated with the next step and the parameter β is associated with the previous step and the middle step is the base step. Thus the sizes for the steps are αh, h and βh.

First Scheme In the first scheme, the step sizes are βh, αh and h respectively. The interpolant defined on $(t_{i-2}, y_{i-2}, f_{i-2})$, $(t_{i-1}, y_{i-1}, f_{i-1})$, (t_i, y_i, f_i) and $(t_{i+1}, y_{i+1}, f_{i+1})$ are defined as such:

$$\begin{aligned} u(t_i + \theta h) = d_0(\theta)y_{i-2} + h_i d_1(\theta)f_{i-2} + d_2(\theta)y_{i-1} + h_i d_3(\theta)f_{i-1} \\ + d_4(\theta)y_i + h_i d_5(\theta)f_i + d_6(\theta)y_{i+1} + h_i d_7(\theta)f_{i+1}, \end{aligned} \tag{14}$$

and the derivative is:

$$\begin{aligned} u'(t_i + \theta h) = d_0'(\theta)y_{i-2}/h_i + d_1'(\theta)f_{i-2} + d_2'(\theta)y_{i-1}/h_i + d_3'(\theta)f_{i-1} \\ + d_4'(\theta)y_i/h_i + d_5'(\theta)f_i + d_6'(\theta)y_{i+1}/h_i + d_7'(\theta)f_{i+1}. \end{aligned} \tag{15}$$

Again θ is:

$$\theta = (t - t_i)/h_i. \tag{16}$$

This time θ is allowed to vary between $-\alpha - \beta$ and 1 such that $t_i + \theta$h is t_{i-2} when θ is $-\alpha - \beta$, t_{i-1} when θ is $-\alpha$, t_i when θ is 0 and t_{i+1} when θ is 1. Also $d_0(\theta)$, $d_1(\theta)$, $d_2(\theta)$, $d_3(\theta)$, $d_4(\theta)$, $d_5(\theta)$, $d_6(\theta)$ and $d_7(\theta)$ are all septic polynomials that will each have 8 coefficients.

Each septic polynomial is assumed to have the form $a\theta^7 + b\theta^6 + c\theta^5 + d\theta^4 + e\theta^3 + f\theta^2 + g\theta + h$ where the eight coefficients for each can be found in terms of α and β by solving a linear system of 8 equations in terms of α and β. First at $\theta = -\alpha - \beta$, only $d_0(\theta)$ evaluates to 1 and all the other septic polynomials evaluate to 0 as $u(t_i - (\alpha + \beta)h) = u(t_{i-2}) = y_{i-2}$. Also at this θ value, only the derivative of $d_1(\theta)$ evaluates to 1 and all the other septic polynomial derivatives evaluate to 0 as $u'(t_i - (\alpha + \beta)h) = u'(t_{i-2}) = f_{i-2}$. When $\theta = -\alpha$, only $d_2(\theta)$ evaluates to 1 and all the other septic polynomials evaluate to 0 as $u(t_i - \alpha h) = u(t_{i-1}) = y_{i-1}$. Also at this θ value, only the derivative of $d_3(\theta)$ evaluates to 1 and all the other septic polynomial derivatives evaluate to 0 as $u'(t_i - \alpha h) = u'(t_{i-1}) = f_{i-1}$. When θ is 0, only $d_4(\theta)$ evaluates to 1 and all the other polynomials evaluate to 0 as $u(t_i - 0(h)) = u(t_i) = y_i$. Also at this θ value, only the derivative of $d_5(\theta)$ evaluates to 1 and all the other septic polynomial derivatives evaluate to 0 as $u'(t_i - 0(h)) = u'(t_i) = f_i$. When θ is 1, only $d_6(\theta)$ evaluates to 1 and all the other polynomials evaluate to 0 as $u(t_i + 1(h)) = u(t_{i+1}) = y_{i+1}$. Also at

this θ value, only the derivative of $d_7(\theta)$ evaluates to 1 and all the other septic polynomial derivatives evaluate to 0 as $u'(t_i - 1(h)) = u'(t_{i+1}) = f_{i+1}$. With these eight conditions, we can get eight equations for each polynomial in terms of α and β and, using a symbolic management package, we can solve all of these to find the eight septic polynomials.

We again note that as the solver is stepping through the problem, these interpolants can be obtained without the need for any extra evaluations of f. We just need to store the 4 data points $(t_{i-1}, y_{i-1}, f_{i-1})$, $(t_{i-1}, y_{i-1}, f_{i-1})$, (t_i, y_i, f_i) and $(t_{i+1}, y_{i+1}, f_{i+1})$.

The interpolant defined as above will now be referred to as 'HB8' for the remainder of this chapter. We note that it is of order 8 and its derivative is of order 7.

Unfortunately, this scheme has a serious issue. The accuracy of the interpolant is very sensitive to a slight change in α and/or β. This is because the error term is now proportional to $(t+\alpha+\beta)^2(t+\alpha)^2(t)^2(t+1)^2$. We note that the first term depends on both α and β and thus very small deviations of these values from 1 will result in reduced accuracy.

In Figure 27, we perform a simple experiment to show this issue. We place 4 data points along the t-axis such that their distances apart are βh, αh and h for several values of α and β and we vary h from 1 to 10^{-10}; we then report on the order of the maximum defect at each h for these values.

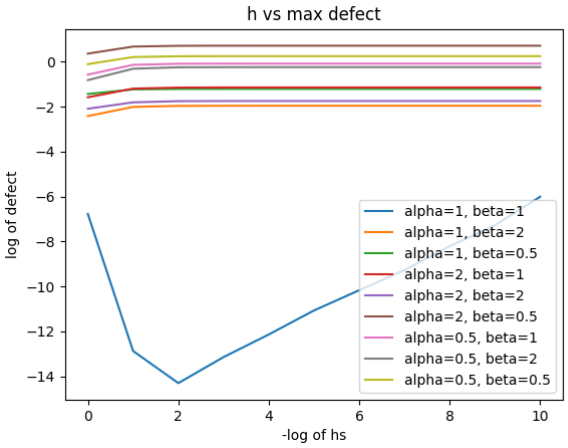

Figure 27: HB8 First Scheme - maximum order of accuracy based on different values of α and β

From Figure 27, we can see the issue with this scheme. It only works if both α and β are 1 and even small deviations from 1 of either α or β drastically reduces the accuracy. More importantly, we can never halve the step with this method as if either α or β is 2, the interpolant is not accurate to even one order of magnitude. We will now consider a second scheme which is more stable with

respect to changes in α and β.

Second Scheme In this second scheme, we still use a 3 step interpolant with 4 data points $(t_{i-1}, y_{i-1}, f_{i-1})$, $(t_{i-2}, y_{i-2}, f_{i-2})$, (t_i, y_i, f_i) and $(t_{i+1}, y_{i+1}, f_{i+1})$ but now the distance between the points are αh, h and then βh. The middle step is the base step. This way the error term is approximately proportional to $(t-(1+\alpha))^2(t+1)^2t^2(t+\beta)^2$. We avoid the $(t-(\alpha+\beta))^2$ factor. We will show that this scheme is more resilient to changes in α and β.

Its derivation is very similar to the first scheme. The equation for the interpolant is:

$$\begin{aligned} u(t_i+\theta h) = d_0(\theta)y_{i-2} + h_i d_1(\theta)f_{i-2} + d_2(\theta)y_{i-1} + h_i d_3(\theta)f_{i-1} \\ +d_4(\theta)y_i + h_i d_5(\theta)f_i + d_6(\theta)y_{i+1} + h_i d_7(\theta)f_{i+1}, \end{aligned} \tag{17}$$

and its derivative is:

$$\begin{aligned} u'(t_i+\theta h) = d_0'(\theta)y_{i-2}/h_i + d_1'(\theta)f_{i-2} + d_2'(\theta)y_{i-1}/h_i + d_3'(\theta)f_{i-1} \\ +d_4'(\theta)y_i/h_i + d_5'(\theta)f_i + d_6'(\theta)y_{i+1}/h_i + d_7'(\theta)f_{i+1}. \end{aligned} \tag{18}$$

Again θ is:

$$\theta = (t-t_i)/h_i \tag{19}$$

This time θ is allowed to vary between $-1-\alpha$ and β such that $t_i+\theta h$ is t_{i-2} when θ is $-1-\alpha$, t_{i-1} when θ is -1, t_i when θ is 0 and t_{i+1} when θ is β. Also $d_0(\theta)$, $d_1(\theta)$, $d_2(\theta)$, $d_3(\theta)$, $d_4(\theta)$, $d_5(\theta)$, $d_6(\theta)$ and $d_7(\theta)$ are all septic polynomials and will each have 8 conditions from which we can build a system to find their coefficients in terms of α and β.

Each is a septic of the form $a\theta^7 + b\theta^6 + c\theta^5 + d\theta^4 + e\theta^3 + f\theta^2 + g\theta + h$ where the eight coefficients for each can be found in terms of α and β by solving a linear system of 8 equations in terms of α and β. First at $\theta = -1-\alpha$, only $d_0(\theta)$ evaluates to 1 and all the other septic polynomials evaluate to 0 as $u(t_i-(1+\alpha)h) = u(t_{i-2}) = y_{i-2}$. Also at this θ value, only the derivative of $d_1(\theta)$ evaluates to 1 and all the other septic polynomial derivatives evaluate to 0 as $u'(t_i-(1+\alpha)h) = u'(t_{i-2}) = f_{i-2}$. When $\theta = -1$, only $d_2(\theta)$ evaluates to 1 and all the other septic polynomials evaluate to 0 as $u(t_i - 1(h)) = u(t_{i-1}) = y_{i-1}$. Also at this θ value, only the derivative of $d_3(\theta)$ evaluates to 1 and all the other septic polynomials' derivatives evaluate to 0 as $u'(t_i - 1(h)) = u'(t_{i-1}) = f_{i-1}$. When θ is 0, only $d_4(\theta)$ evaluates to 1 and all the other polynomials evaluate to 0 as $u(t_i - 0(h)) = u(t_i) = y_i$. Also at this θ value, only the derivative of $d_5(\theta)$ evaluates to 1 and all the other septic polynomials' derivatives evaluate to 0 as $u'(t_i - 0(h)) = u'(t_i) = f_i$. When θ is β, only $d_6(\theta)$ evaluates to 1 and all the other polynomials evaluate to 0 as $u(t_i + \beta h) = u(t_{i+1}) = y_{i+1}$. Also at this θ value, only the derivative of $d_7(\theta)$ evaluates to 1 and all the other septic polynomial derivatives evaluate to 0 as $u'(t_i - \beta h) = u'(t_{i+1}) = f_{i+1}$. With these eight conditions, we can get eight equations for each septic in terms of α and β and using a symbolic management package, we can solve all of these to find the 8 coefficients for each septic polynomial.

We now perform a simple experiment to show that the resultant interpolant is more resilient to changes in α and β. We place 4 data points along the x-axis such that their distance apart are αh, h and βh for several values of α and β and we vary h from 1 to 10^{-10}, we then report on the order of the maximum defect at each h for these values.

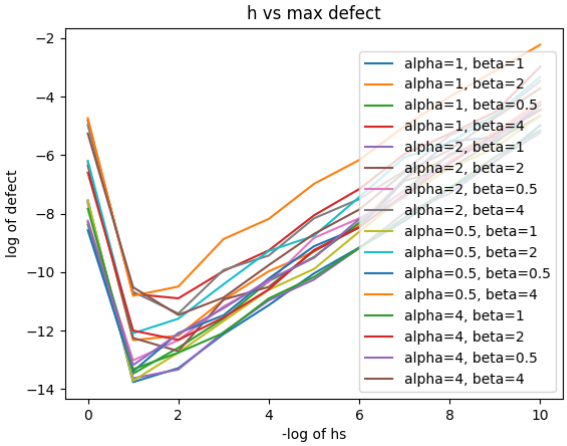

Figure 28: HB8 Second Scheme - maximum order of accuracy based on different values of alpha and beta

From Figure 28, we can see how this scheme is better than the first scheme. We can use α and β equal to 2 and to 1/2 and still get a maximum defect of around 10^{-12} and we can even be accurate to 10^{-10} for both α and β equal to 4.

For the remainder of this chapter, we will denote this 8^{th} order interpolant by HB8. Its derivative has order 7 and any subsequent higher derivative will have one less order.

Results We will now use RK4 with the second HB8 scheme and use the new defect control solver to solve the three test problems. We will show that we need to sample the defect only twice to estimate the maximum defect and that this scheme can provide good quality defect control.

Problem 1 results Figures 29, 30 and 31 shows the results of using the modification of RK4 with HB8 on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 31, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 29.

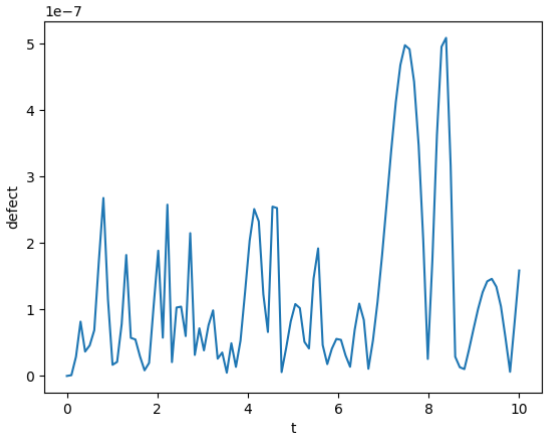

Figure 29: Defect across the entire domain of RK4 with HB8 on problem 1 at an absolute tolerance of 10^{-6}

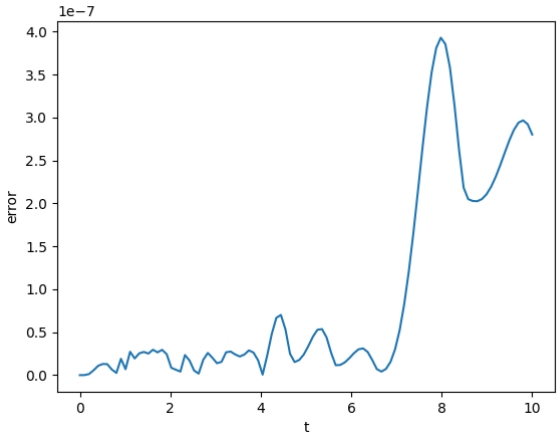

Figure 30: Global Error of RK4 with HB8 on problem 1 at an absolute tolerance of 10^{-6}

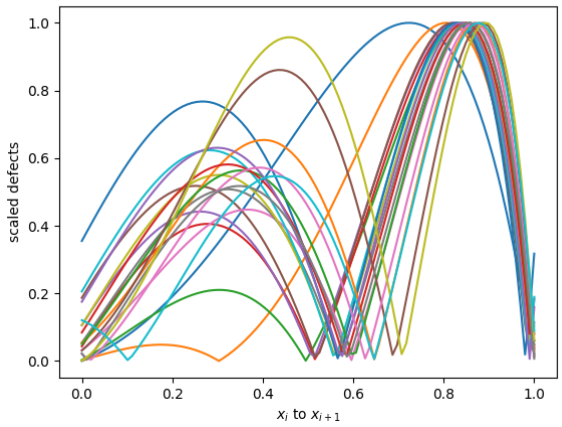

Figure 31: Scaled defects of RK4 with HB8 on problem 1 at an absolute tolerance of 10^{-6} mapped into $[0, 1]$.

Problem 2 results Figures 32, 33 and 34 shows the results of using the modification of RK4 with HB6 on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 34, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 32.

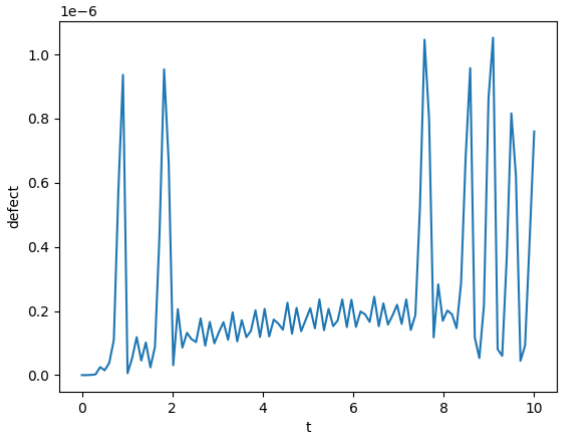

Figure 32: Defect across the entire domain of RK4 with HB8 on problem 2 at an absolute tolerance of 10^{-6}

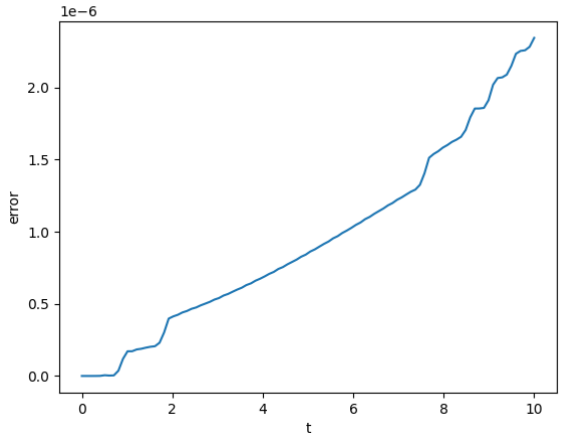

Figure 33: Global Error of RK4 with HB8 on problem 2 at an absolute tolerance of 10^{-6}

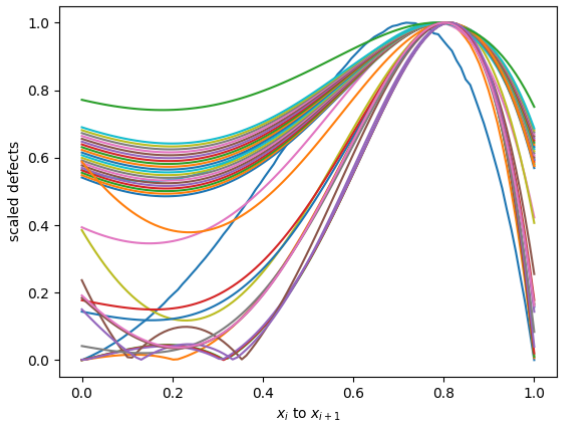

Figure 34: Scaled defects of RK4 with HB8 on problem 2 at an absolute tolerance of 10^{-6} mapped into $[0, 1]$.

Problem 3 results Figures 35, 36 and 37 shows the results of using the modification of RK4 with HB6 on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 37, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 35.

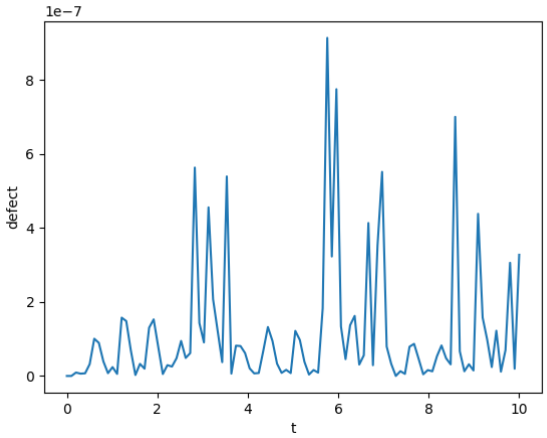

Figure 35: Defect across the entire domain of RK4 with HB8 on problem 3 at an absolute tolerance of 10^{-6}

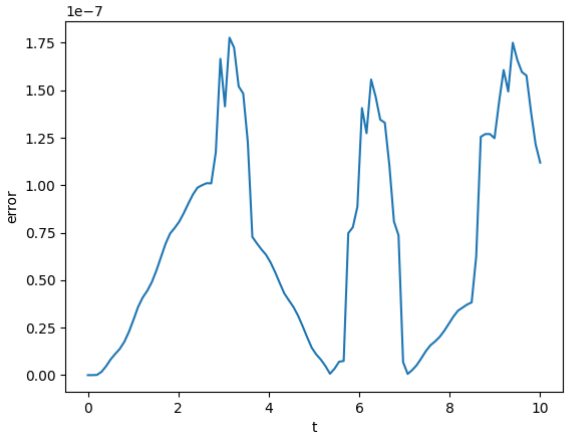

Figure 36: Global Error of RK4 with HB8 on problem 3 at an absolute tolerance of 10^{-6}

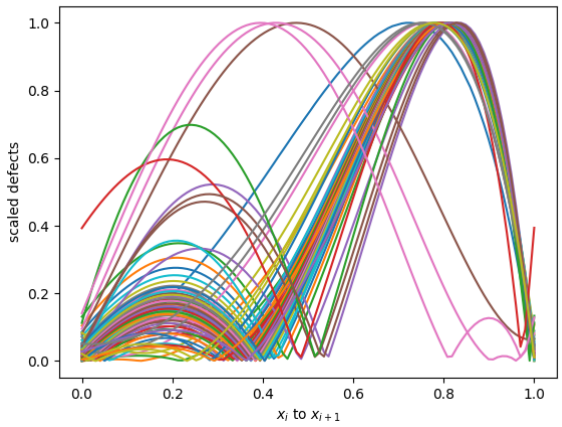

Figure 37: Scaled defects of RK4 with HB8 on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

We note that the defects are not as clean they were in the case with HB4. There are two peaks most of the time at around $0.4h$ and $0.8h$ but as was the case for the third problem in the previous tests, the peak sometimes comes at $0.6h$. However, we can see that the defect is still being controlled and thus that the error is still being controlled. We will also see that it is twice as fast as it uses around half the number of steps as HB4.

Table 4: Number of steps taken by RK4 when modified to do defect control with HB8 vs when modified with HB6

Problem	succ. steps HB8	succ. steps HB6	nsteps HB8	nsteps HB6
1	20	27	20	27
2	37	36	60	40
3	69	62	89	73

From Table 4, we can see that the number of steps with HB6 and with HB8 are relatively similar. This indicates that the interpolation error is no longer the limiting factor, even in HB6. The limiting factor is the ODE solution which is as required. Thus though we can use RK4 with HB8 at the same cost as modifying RK4 with HB6, using HB8 does not improve the efficiency. Furthermore, HB8 is less stable to changes in α and β than HB6 is to changes to α. Thus RK4 is best augmented with HB6. However HB8 provides a new opportunity, we can now augment a 6^{th} order Runge Kutta method and possibly an 8^{th} order Runge-Kutta method, but in the latter case, the interpolation error will still affect the accuracy. Augmenting higher order methods with our HB6 and HB8

schemes is significant because as we have discussed in Section 1.1.3 when using continuous Runge-Kutta solvers to obtain the interpolants, the number of stages grows exponentially with the order of the method. Our scheme is zero-cost and thus effective defect control using this approach will be relatively even more efficient.

1.3 Higher Order Runge Kutta Methods

In this section, we attempt to perform defect control based on efficient multistep interpolants for higher order Runge-Kutta methods. We recall that related previous work used significantly more stages to obtain a continuous 6^{th} order Runge-Kutta method and a continuous 8^{th} order Runge-Kutta method.

In this section, we will first augment the RK6 method (See Section 1.1.4 for details about the discrete method) with HB6 and then with HB8. We hope to perform defect control and to find that the use of HB8 allows significantly fewer steps. We will then augment the RK8 method (See Section 1.1.4 for more details) with HB8 to show that though interpolation error is present, the scheme does allow defect control of a continuous 8^{th} order solution.

For both method, we will sample the defect only twice in a step, at $0.4h$ and $0.8h$ in a step of size h, as the previous experiments appears to indicate that the maximum defects tend to occur at these locations within each step.

1.3.1 RK6 with HB6

Problem 1 results Figures 38, 39 and 40 shows the results of using the RK6 with HB6 on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 40, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 38.

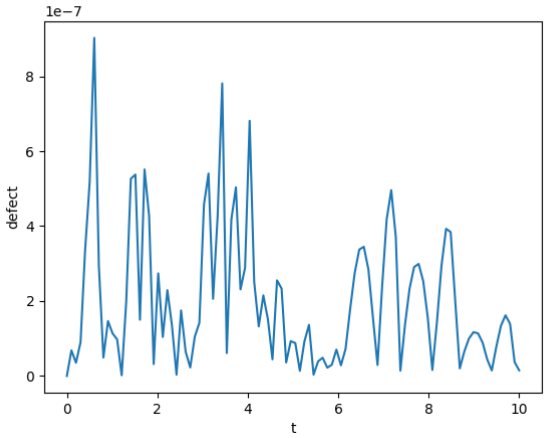

Figure 38: Defect across the entire domain of RK6 with HB6 on problem 1 at an absolute tolerance of 10^{-6}

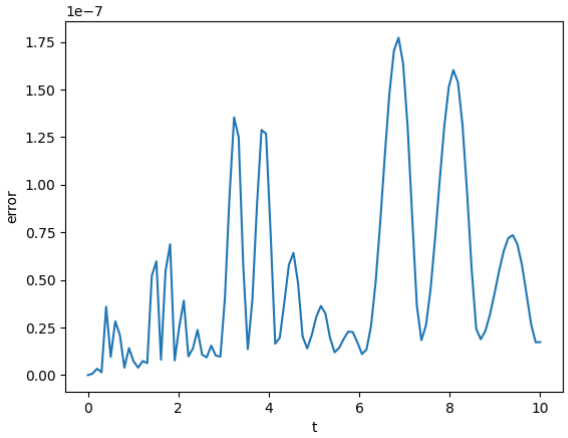

Figure 39: Global Error of RK6 with HB6 on problem 1 at an absolute tolerance of 10^{-6}

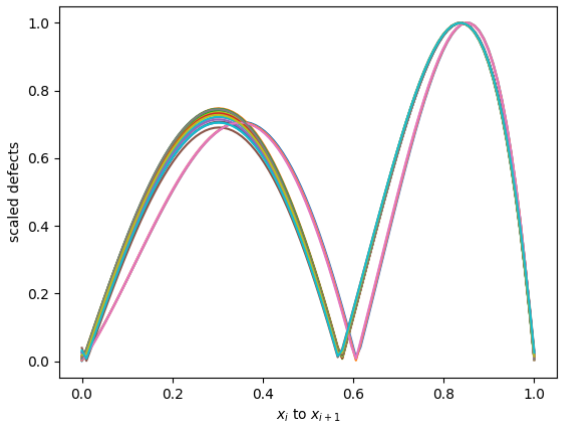

Figure 40: Scaled defects of RK6 with HB6 on problem 1 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 2 results Figures 41, 42 and 43 shows the results of using the modification of RK6 with HB6 on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 43, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 41.

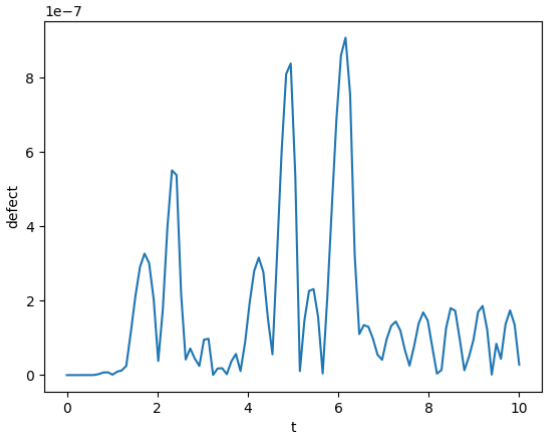

Figure 41: Defect across the entire domain of RK6 with HB6 on problem 2 at an absolute tolerance of 10^{-6}

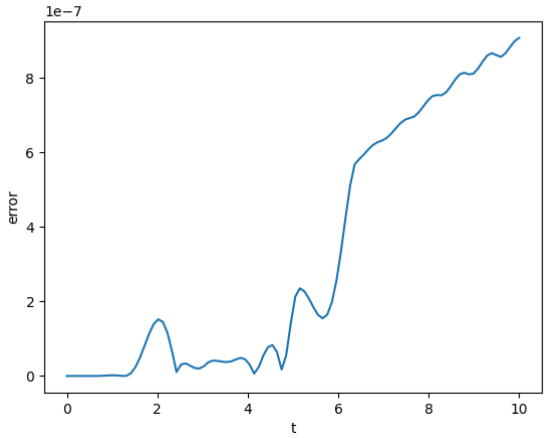

Figure 42: Global Error of RK6 with HB6 on problem 2 at an absolute tolerance of 10^{-6}

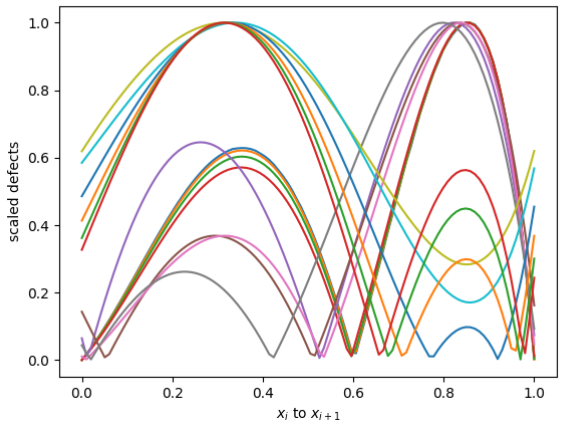

Figure 43: Scaled defects of RK6 with HB6 on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

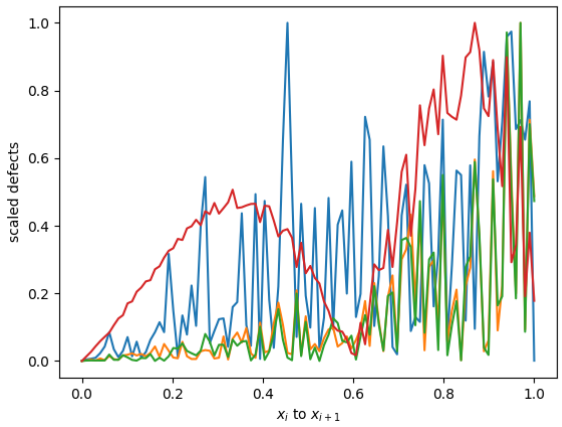

Figure 44: Scaled defects of RK6 with HB6 on small steps on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 3 results Figures 45, 46 and 47 shows the results of using the modification of RK6 with HB6 on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 47, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 45.

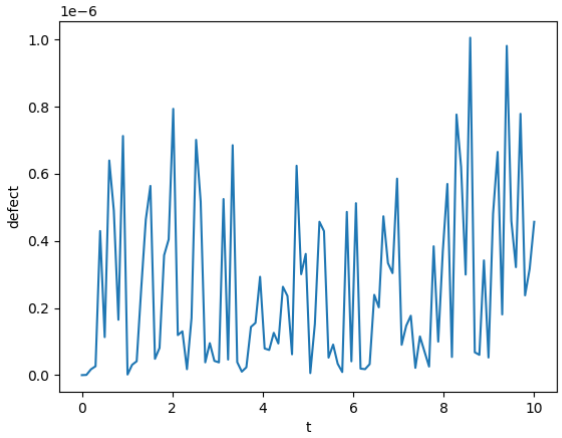

Figure 45: Defect across the entire domain of RK6 with HB6 on problem 3 at an absolute tolerance of 10^{-6}

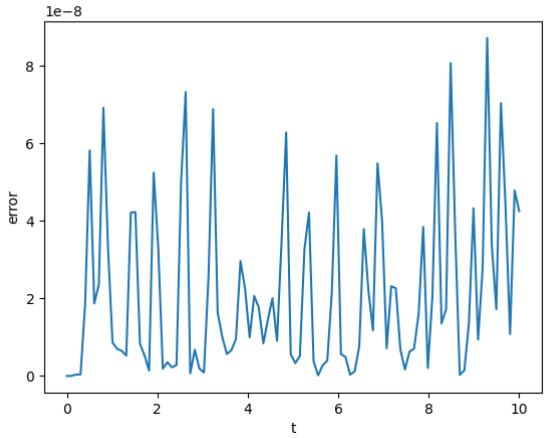

Figure 46: Global Error of RK6 with HB6 on problem 3 at an absolute tolerance of 10^{-6}

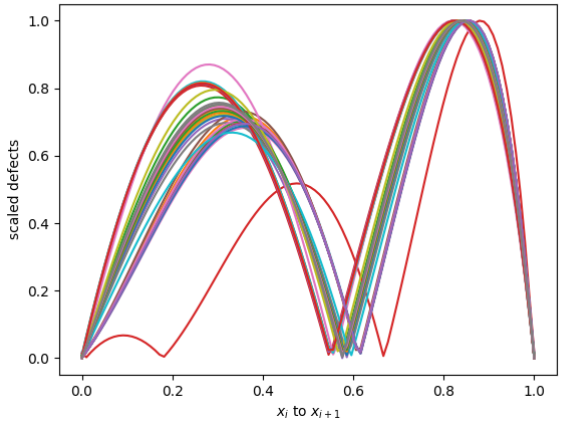

Figure 47: Scaled defects of RK6 with HB6 on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

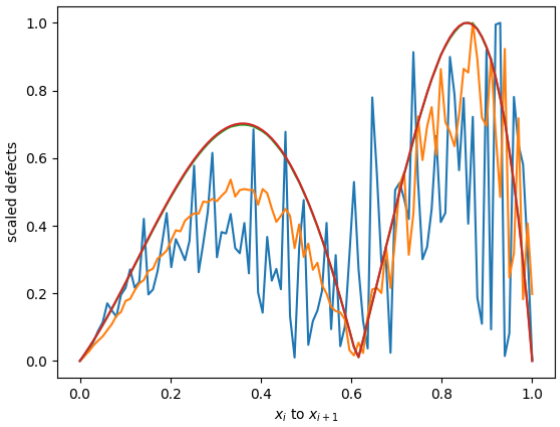

Figure 48: Scaled defects of RK6 with HB6 on small steps on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

1.3.2 RK6 with HB8

Problem 1 results Figures 49, 50 and 51 shows the results of using the modification of RK6 with HB8 on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 51, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 49.

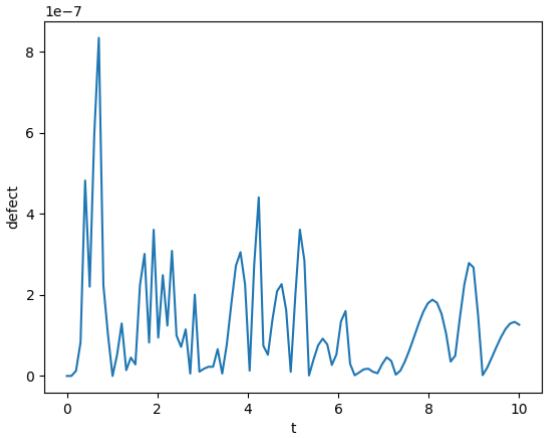

Figure 49: Defect across the entire domain of RK6 with HB8 on problem 1 at an absolute tolerance of 10^{-6}

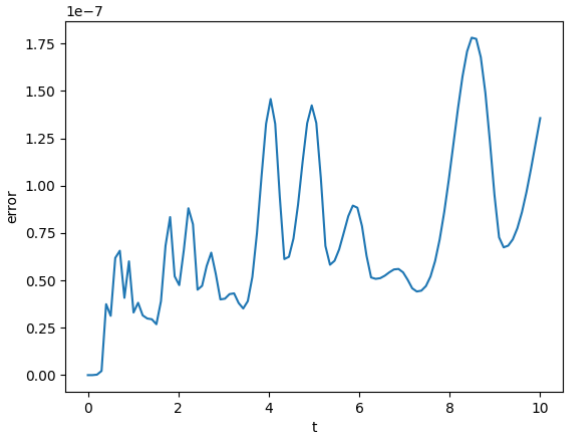

Figure 50: Global Error of RK6 with HB8 on problem 1 at an absolute tolerance of 10^{-6}

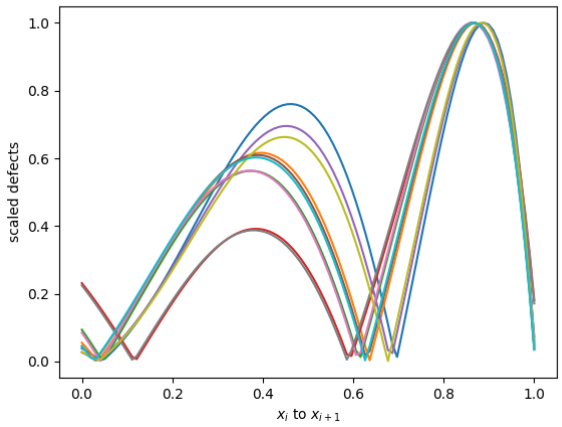

Figure 51: Scaled defects of RK6 with HB8 on problem 1 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 2 results Figures 52, 53 and 54 shows the results of using the modification of RK6 with HB8 on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 54, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 52.

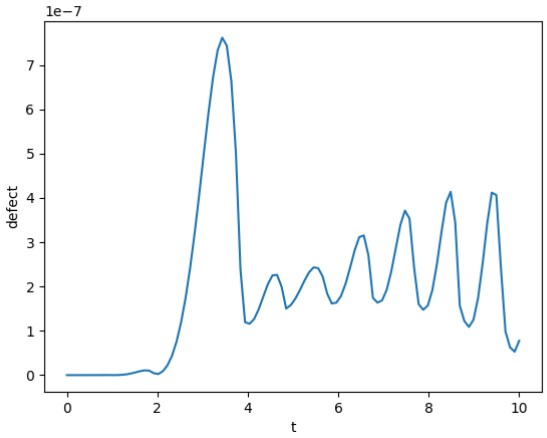

Figure 52: Defect across the entire domain of RK6 with HB8 on problem 2 at an absolute tolerance of 10^{-6}

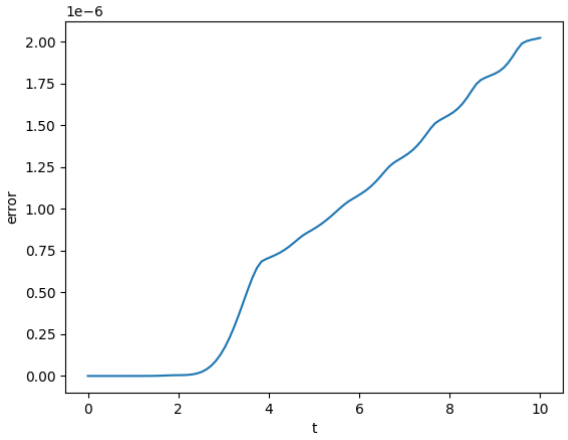

Figure 53: Global Error of RK6 with HB8 on problem 2 at an absolute tolerance of 10^{-6}

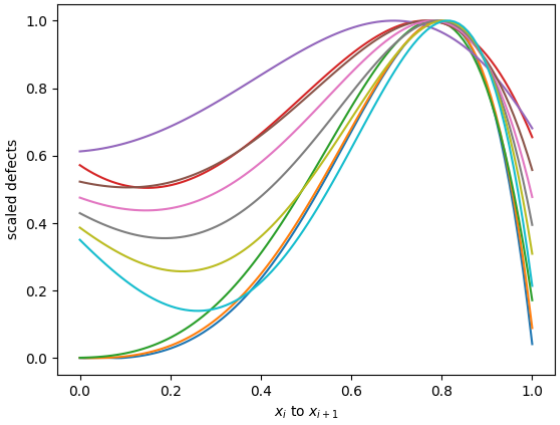

Figure 54: Scaled defects of RK6 with HB8 on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

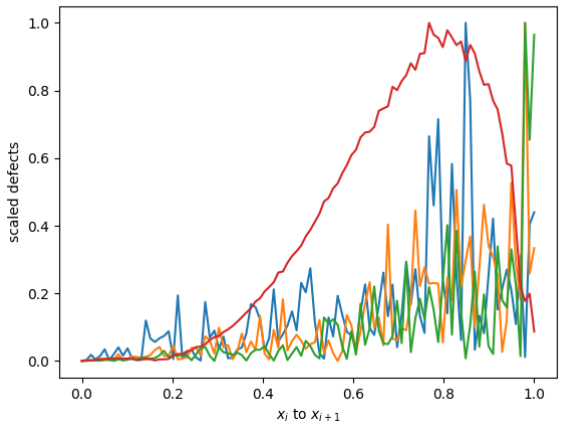

Figure 55: Scaled defects of RK6 with HB8 on small steps on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 3 results Figures 56, 57 and 58 shows the results of using the modification of RK6 with HB8 on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at either $0.4h$ or $0.8h$ along a step of size, h. See Figure 58, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 56.

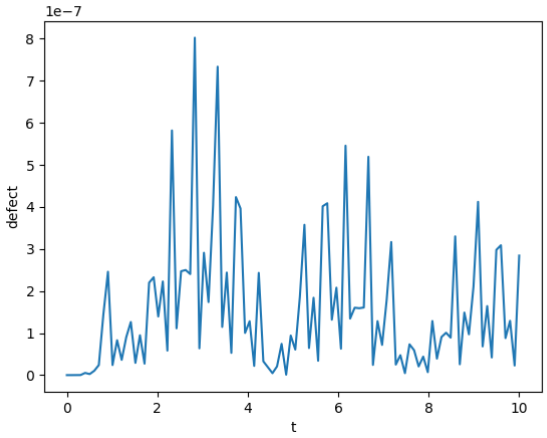

Figure 56: Defect across the entire domain of RK6 with HB8 on problem 3 at an absolute tolerance of 10^{-6}

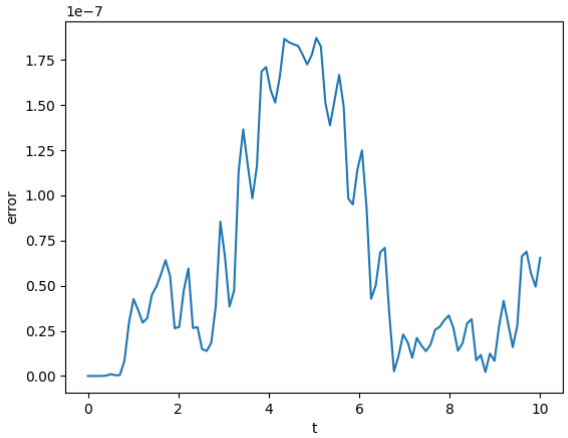

Figure 57: Global Error of RK6 with HB8 on problem 3 at an absolute tolerance of 10^{-6}

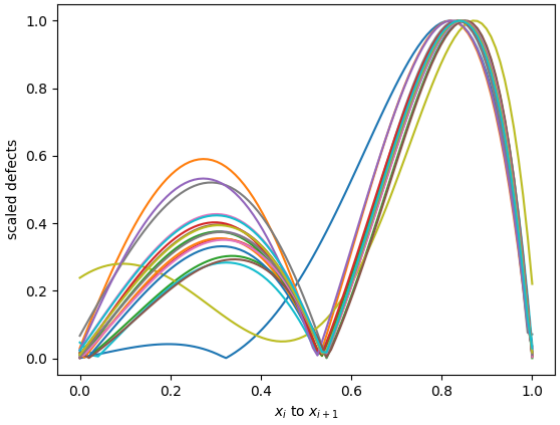

Figure 58: Scaled defects of RK6 with HB8 on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

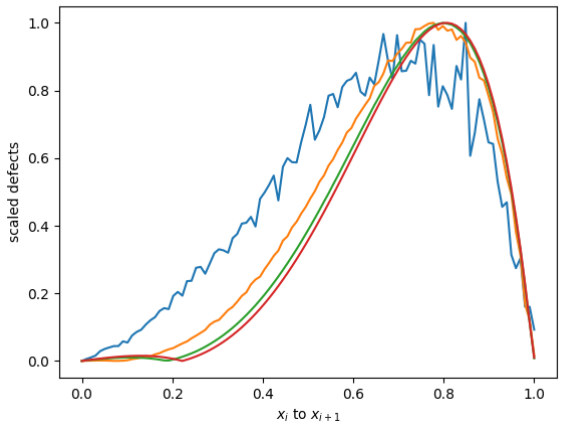

Figure 59: Scaled defects of RK6 with HB8 on small steps on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Table 5: Number of steps taken by RK6 when modified to do defect control with HB8 vs when modified with HB6

Problem	succ. steps HB8	succ. steps HB6	nsteps HB8	nsteps HB6
1	18	26	18	27
2	14	18	17	23
3	24	42	26	51

From Table 5, we can see that again, using an interpolant whose interpolation error and especially the interpolation error of its derivative is of higher order than the ODE solution drastically reduces the number of steps. The code becomes more efficient as a result. Since rk6 with HB6 and with HB8 is behaving similarly to RK4 with HB4 and with HB6, we expect that using a 10^{th} order method would not improve the situation more than HB8 has over HB6. Though fitting RK6 with HB10 will be as efficient as fitting it with HB8, the fact that the interpolation error is no longer the limiting factor means that HB10 will not improve the situation. For RK6, HB8 is the preferred interpolant.

We also note that during the solving of the 3 problems, the value of α with HB6 rarely was bigger than 4 or smaller than $\frac{1}{4}$. The values of α and β with HB8 also rarely were bigger than 4 or smaller than $\frac{1}{4}$.

1.3.3 RK8 with HB8

In this section, we fit the RK8 method, described in Section 1.1.4, with HB8. Though we expect the interpolation error to reduce the efficiency of this mod-

ification, we provide a proof of concept that an RK8 can have defect control with the scheme presented in this chapter. We will look into the challenges and possibilities of deriving an HB10 scheme in the Future Works section.

Problem 1 results Figures 60, 61 and 62 shows the results of using the modification of RK8 with HB8 on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 62, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 60.

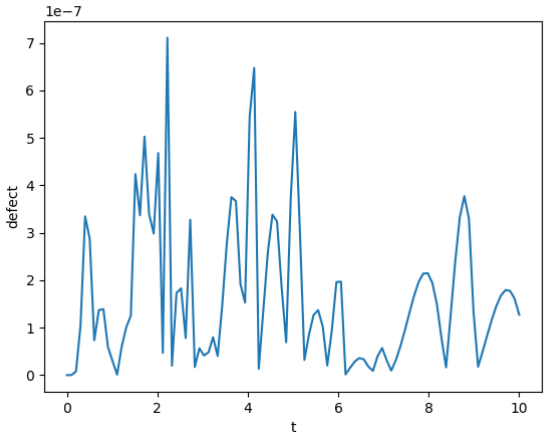

Figure 60: Defect across the entire domain of RK8 with HB8 on problem 1 at an absolute tolerance of 10^{-6}

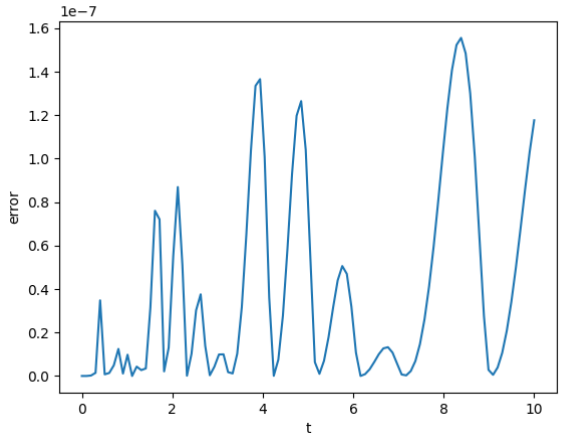

Figure 61: Global Error of RK8 with HB8 on problem 1 at an absolute tolerance of 10^{-6}

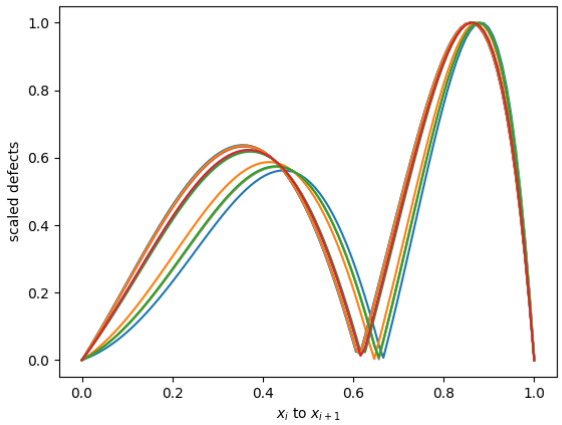

Figure 62: Scaled defects of RK8 with HB8 on problem 1 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 2 results Figures 63, 64 and 65 shows the results of using the modification of RK8 with HB8 on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 65, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 63.

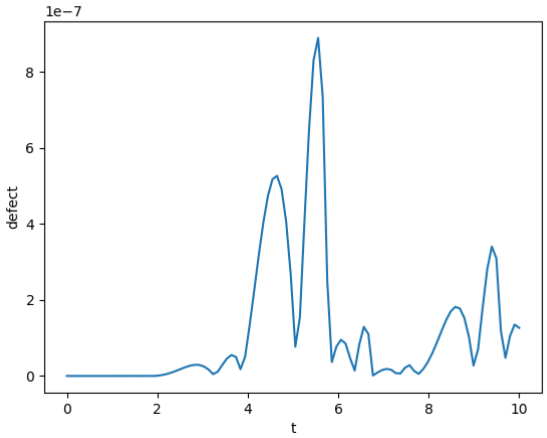

Figure 63: Defect across the entire domain of RK8 with HB8 on problem 2 at an absolute tolerance of 10^{-6}

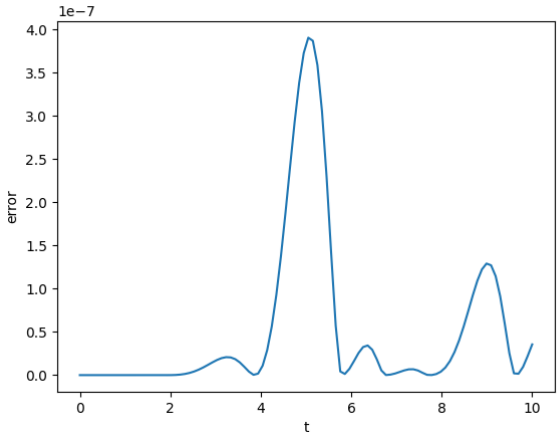

Figure 64: Global Error of RK8 with HB8 on problem 2 at an absolute tolerance of 10^{-6}

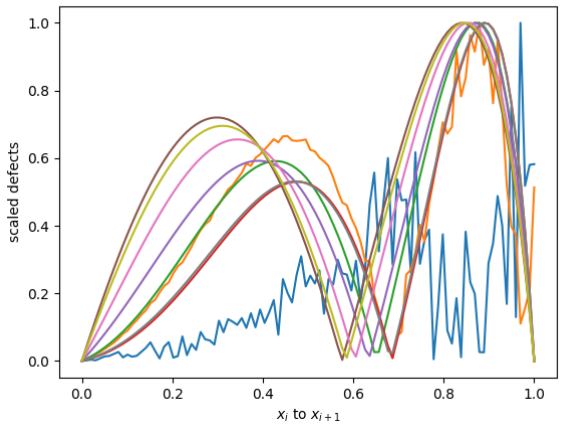

Figure 65: Scaled defects of RK8 with HB8 on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

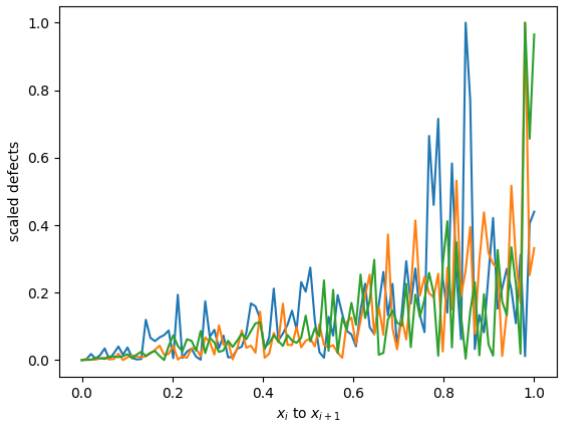

Figure 66: Scaled defects of RK8 with HB8 on small steps on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 3 results Figures 67, 68 and 69 shows the results of using the modification of RK8 with HB8 on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 69, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 67.

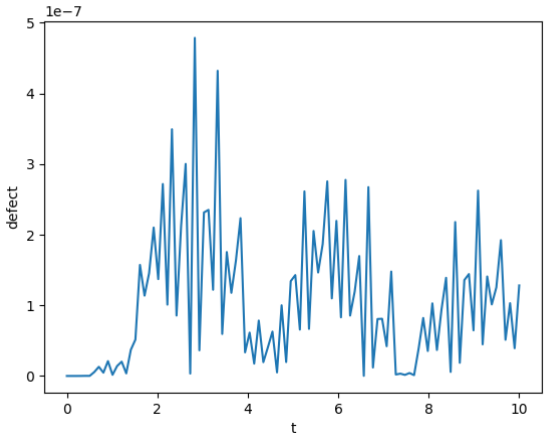

Figure 67: Defect across the entire domain of RK8 with HB8 on problem 3 at an absolute tolerance of 10^{-6}

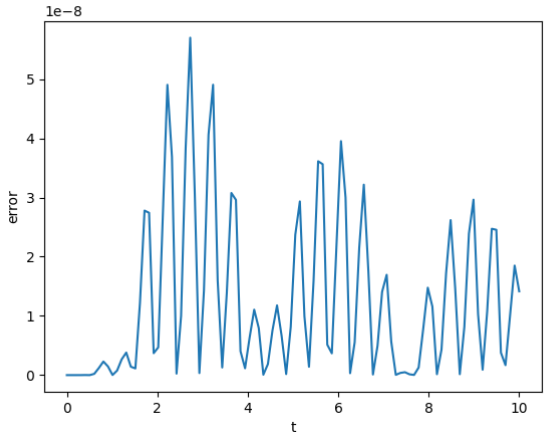

Figure 68: Global Error of RK8 with HB8 on problem 3 at an absolute tolerance of 10^{-6}

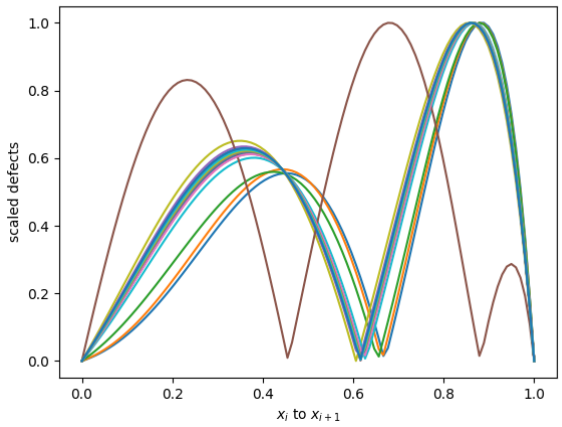

Figure 69: Scaled defects of RK8 with HB8 on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

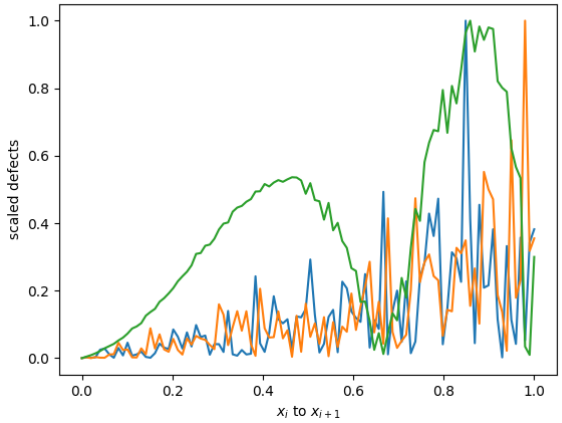

Figure 70: Scaled Defects of RK8 with HB8 on small steps on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

1.4 Using the previous interpolants to keep the weight parameters at 1

One idea to solve the problem of the loss in accuracy due to deviation of the α and β parameters from 1 is to construct the interpolant using a value of 1 for the parameters. We note that the best accuracy we can hope to get is with

Table 6: Number of steps taken by RK8 when modified to do defect control with HB8

Problem	succ. steps	nsteps
1	18	19
2	12	15
3	24	29

a value of 1 and thus employ data values situated at uniform distances apart guarantees the minimum error.

HB6 For the HB6 case, a simple way to guarantee that the value of α is 1 is by using the previous interpolants to get the required values at $x_{i-1} = x_i - h$. Say we are at a value x_i and we took a step of size h to get to the value x_{i+1} where the function evaluation was f_{i+1} and the solution was y_{i+1}. We could get the values of the solution and the derivative by using the previous interpolants defined on the range $[0, x_i]$ to get a value at $x_i - h$ for the solution approximation, y_{i-1}, and then we can use this y_{i-1} value to compute $f(t_{i-1}, y_{i-1})$ to get the derivative f_{i-1}. We could thus create the new interpolant using these data points to guarantee that α stays at 1. We note that we have built interpolants for the solution approximation and the derivative that can interpolate up to x_i for all cases except for the case $x_i = 0$ or for the first few steps. This technique costs one extra function evaluation to obtain f_{i-1}. We also note that we cannot use the interpolant of the derivative on the previous step to get f_{i-1} without making any function call as the derivative will be of lower order and thus the interpolant will not be of higher order than the numerical solution.

This technique works (See Figures 71 to 80 to see the defect being controlled) but will require that the step-size is artificially limited on the first few steps so that we can interpolant back $x_i - h$ and still be in a range where our interpolants are correctly defined. For example, If we go from t_0 to $t_0 + h$ and the error estimate is much lower than the tolerance, we cannot use a step size of $2h$ as $t_0 + h - 2h = t_0 - h$ because we do not have an interpolant in the region $\leq t_0$. However this is not an issue as we can perform the first few steps with a CRK scheme for example.

Problem 1 results Figures 71, 72 and 73 shows the results of using the modification of RK4 with HB6 and static parameters on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.3h$ and $0.8h$ along a step of size, h. See Figure 73, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 71.

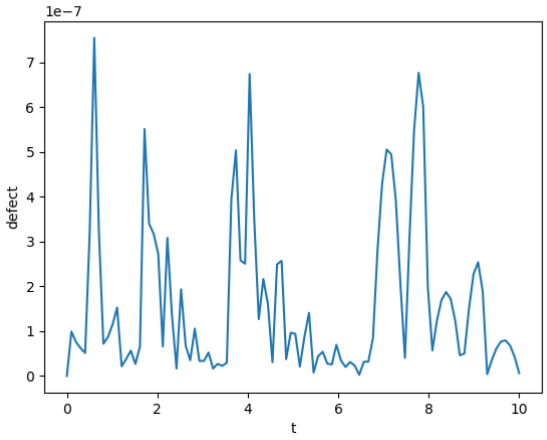

Figure 71: Defect across the entire domain of RK4 with HB6 using $\alpha = 1$ on problem 1 at an absolute tolerance of 10^{-6}

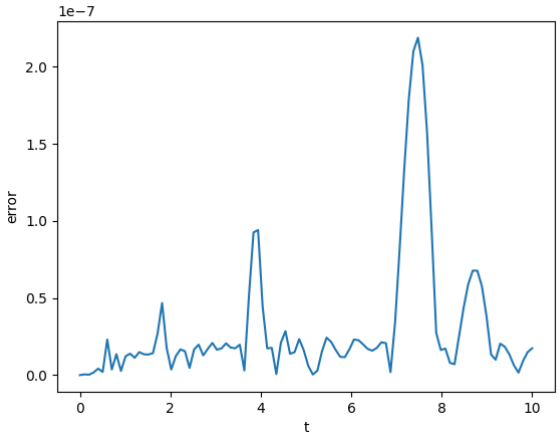

Figure 72: Global Error of RK4 with HB6 using $\alpha = 1$ on problem 1 at an absolute tolerance of 10^{-6}

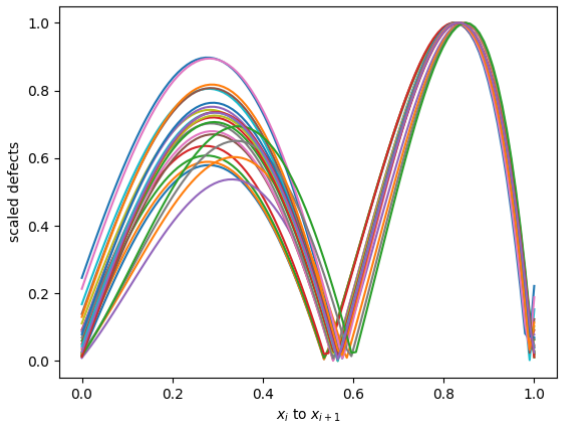

Figure 73: Scaled defects of RK4 with HB6 using $\alpha = 1$ on problem 1 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Problem 2 results Figures 74, 75 and 76 shows the results of using the modification of RK4 with HB6 and static parameters on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 76, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 74.

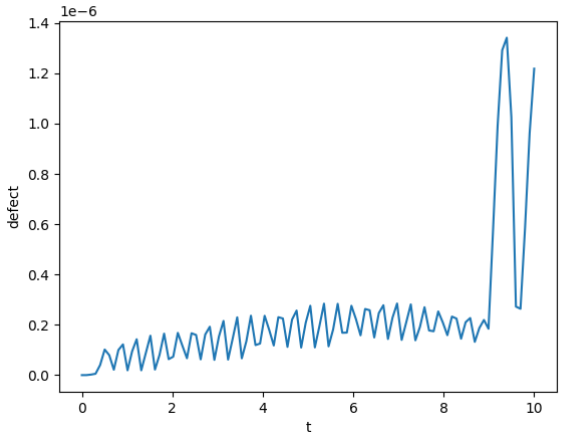

Figure 74: Defect across the entire domain of RK4 with HB6 using $\alpha = 1$ on problem 2 at an absolute tolerance of 10^{-6}

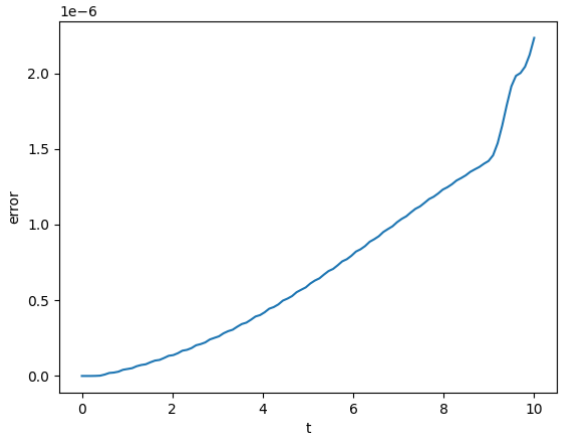

Figure 75: Global Error of RK4 with HB6 using $\alpha = 1$ on problem 2 at an absolute tolerance of 10^{-6}

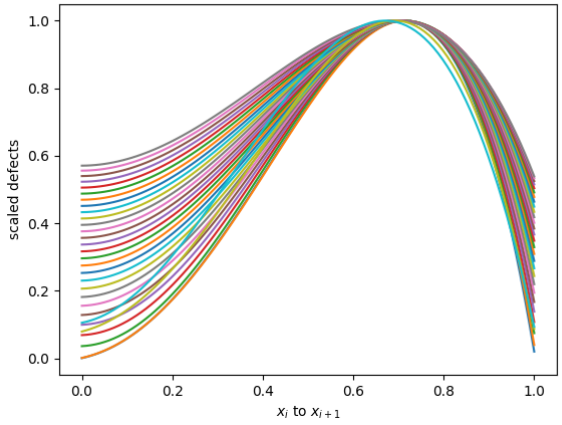

Figure 76: Scaled defects of RK4 with HB6 using $\alpha = 1$ on problem 2 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

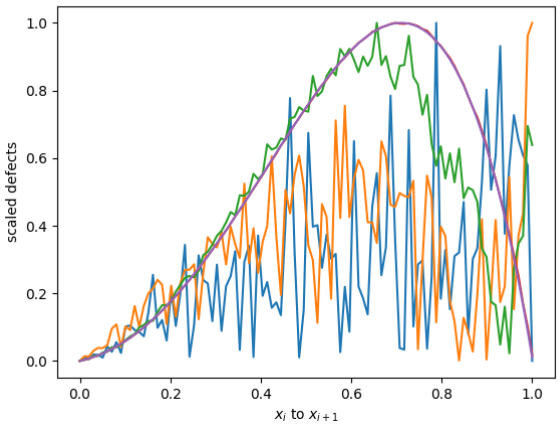

Figure 77: Scaled defects of RK4 with HB6 using $\alpha = 1$ on small steps on problem 2 at an absolute tolerance of 10^{-6}

Problem 3 results Figures 78, 79 and 80 shows the results of using the modification of RK4 with HB6 and static parameters on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.3h$ and $0.8h$ along a step of size, h. See Figure 80, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 78.

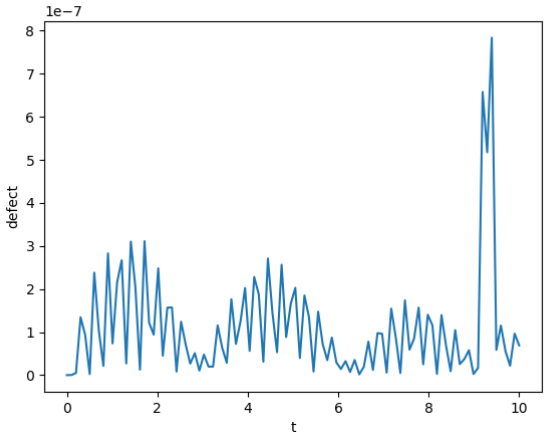

Figure 78: Defect across the entire domain of RK4 with HB6 using $\alpha = 1$ on problem 3 at an absolute tolerance of 10^{-6}

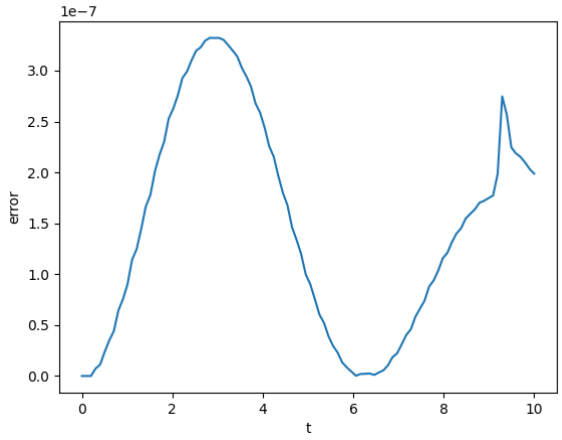

Figure 79: Global Error of RK4 with HB6 using $\alpha = 1$ on problem 3 at an absolute tolerance of 10^{-6}

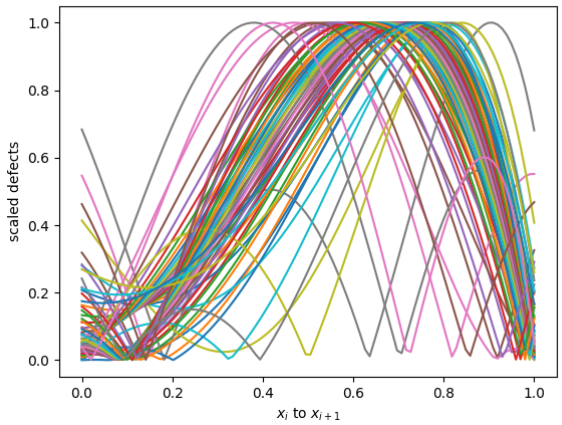

Figure 80: Scaled defects of RK4 with HB6 using $\alpha = 1$ on problem 3 at an absolute tolerance of 10^{-6} mapped onto $[0, 1]$.

Table 7 shows how the solver with α fixed at 1 and the solver with α allowed to vary differ in the number of steps that they take. The results are very similar because, as we noted before, α tends to stay close to 1 and rarely deviates to a value smaller than $\frac{1}{4}$ or larger than 4.

HB8 For the HB8 case, a simple way to guarantee that the values of α and β are 1 is by using the previous interpolants to get the correct values at $x_{i-1} =$

Table 7: Number of steps taken by RK4 when modified to do defect control with HB6 when we fix α at 1 vs when we allow it to fluctuate during the integration

Problem	succ. steps	α=1 succ.	nsteps	α=1 nsteps
1	27	29	27	33
2	36	34	40	36
3	62	65	73	82

$x_i - h$ and $x_{i-2} = x_i - 2h$. Suppose that we are at x_i and we took a step of size h to get to the value x_{i+1} where the function evaluation was f_{i+1} and the solution was y_{i+1}. We could get the values of the solution and the derivative by using the previous interpolants defined on the range $[0, x_i]$ to get the value at exactly $x_i - h$ for the solution, y_{i-1}, and then evaluate $f(t_{i-1}, y_{i-1})$ to get the derivative f_{i-1}. We can also use the previous interpolants on $[0, x_i]$ to get the values of the solution, y_{i-2} and use it to evaluate $f(t_{i-2}, y_{i-2})$ to get the values of the derivative, f_{i-2}, at $x_{i-2} = x_i - 2h$. We could thus create create the new interpolant using the data values defined as such to guarantee that α and β equal to 1. We note that we have built interpolants for both the solution and the derivative that can interpolate up to x_i for all cases except for the case $x_i = 0$ or for the first few steps. This scheme uses two more function evaluations.

This technique works (See Figures 81 to 91 to see the defect being controlled) but will require that the step-size is artificially limited on the first few steps so that we can interpolant back $x_i - h$ and/or $x_i - 2h$ and still be in a range where our interpolants are correctly defined. This is not an issue as a CRK scheme could be used for the first few steps.

Problem 1 results Figures 81, 82 and 83 shows the results of using the modification of RK6 with HB8 and static parameters on Problem 1. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.4h$ and $0.8h$ along a step of size, h. See Figure 83, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 81.

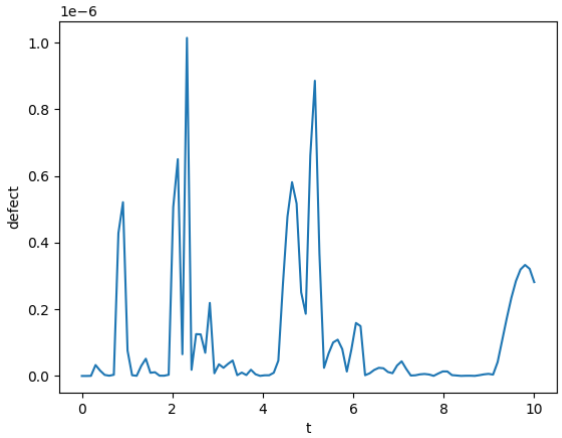

Figure 81: Global Defect of RK6 with HB8 using α and $\beta = 1$ on problem 1 at an absolute tolerance of 10^{-6}

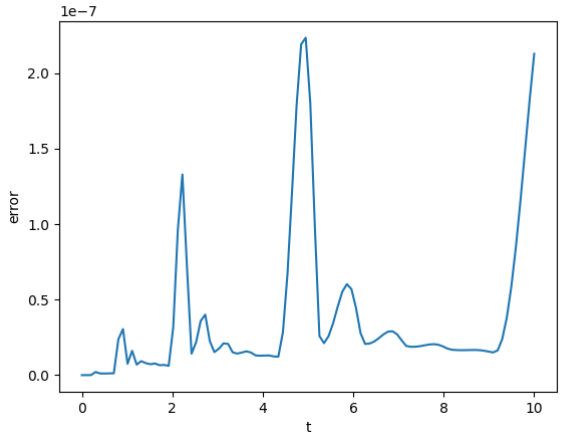

Figure 82: Global Error of RK6 with HB8 using α and $\beta = 1$ on problem 1 at an absolute tolerance of 10^{-6}

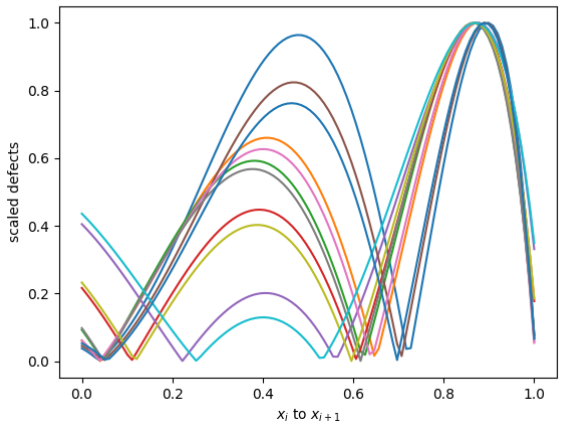

Figure 83: Scaled Defects of RK6 with HB8 using α and $\beta = 1$ on problem 1 at an absolute tolerance of 10^{-6}

Problem 2 results Figures 84, 85 and 86 shows the results of using the modification of RK6 with HB8 and static parameters on Problem 2. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.8h$ along a step of size, h. See Figure 86, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 84.

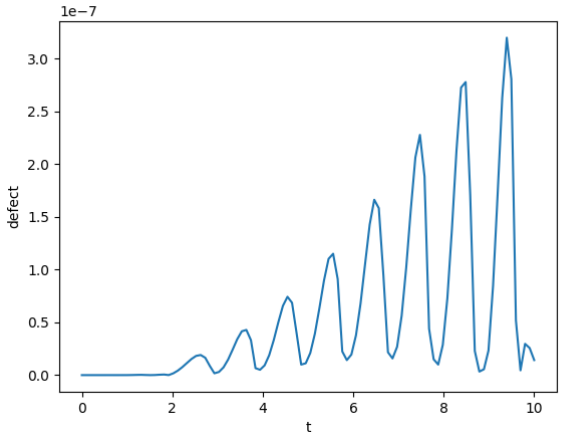

Figure 84: Global Defect of RK6 with HB8 using α and $\beta = 1$ on problem 2 at an absolute tolerance of 10^{-6}

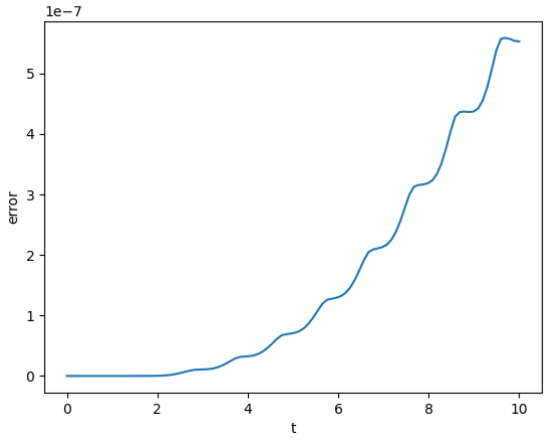

Figure 85: Global Error of RK6 with HB8 using α and $\beta = 1$ on problem 2 at an absolute tolerance of 10^{-6}

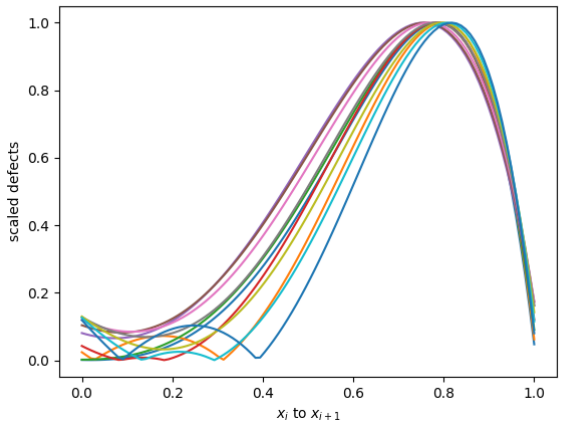

Figure 86: Scaled Defects of RK6 with HB8 using α and $\beta = 1$ on problem 2 at an absolute tolerance of 10^{-6}

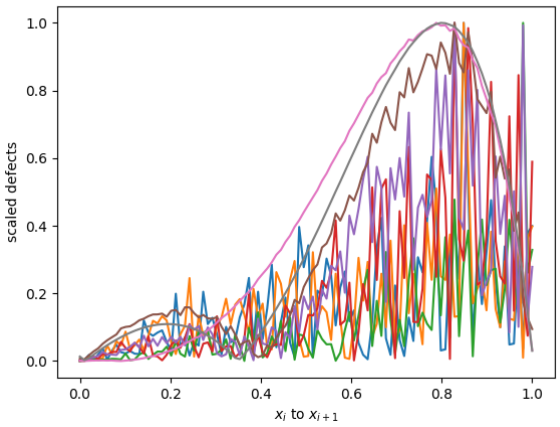

Figure 87: Scaled Defects of RK6 with HB8 using α and $\beta = 1$ on small steps on problem 2 at an absolute tolerance of 10^{-6}

Problem 3 results Figures 88, 89 and 90 shows the results of using the modification of RK6 with HB8 and static parameters on Problem 3. We note that an absolute tolerance of 10^{-6} is applied on the maximum defect within the step and this can be shown to occur at $0.3h$ and $0.8h$ along a step of size, h. See Figure 90, to see the scaled defect reaching a maximum near these points. We note that we are able to successfully control the defect of the continuous numerical solution using this approach, see Figure 88.

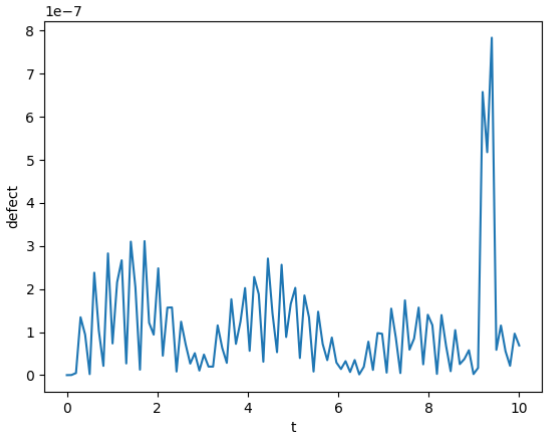

Figure 88: Global Defect of RK6 with HB8 using α and $\beta = 1$ on problem 3 at an absolute tolerance of 10^{-6}

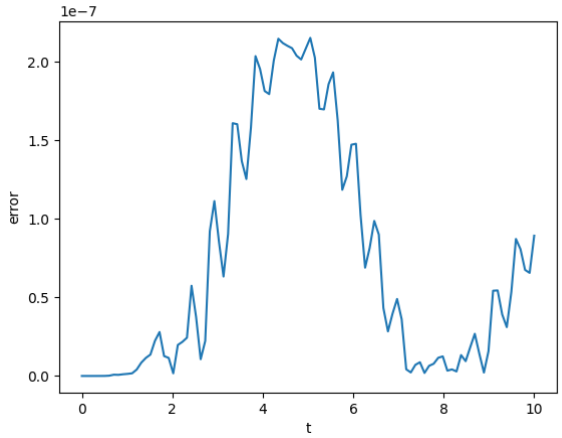

Figure 89: Global Error of RK6 with HB8 using α and $\beta = 1$ on problem 3 at an absolute tolerance of 10^{-6}

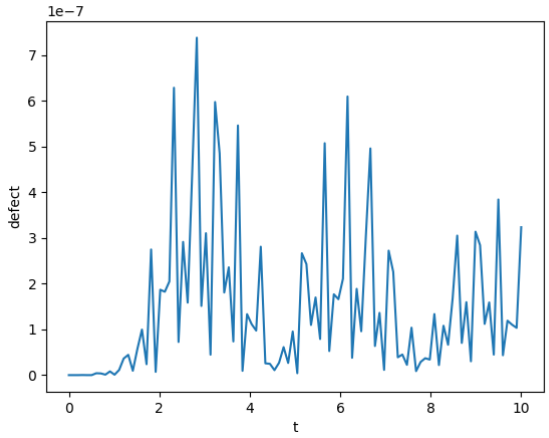

Figure 90: Scaled Defects of RK6 with HB8 using α and $\beta = 1$ on problem 3 at an absolute tolerance of 10^{-6}

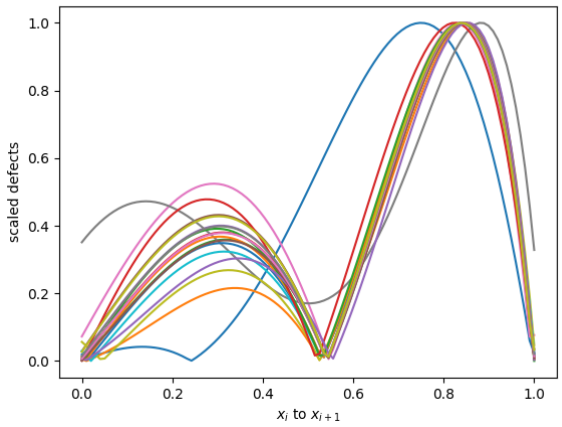

Figure 91: Scaled Defects of RK6 with HB8 using α and $\beta = 1$ on small steps on problem 3 at an absolute tolerance of 10^{-6}

Table 8: Number of steps taken by RK6 when modified to do defect control with HB8 with α and β forcibly at 1 and variable α and β

Problem	succ.	params=1 succ. steps	nsteps	params=1 nsteps
1	18	23	18	31
2	14	18	17	21
3	24	26	26	27

Table 8 shows how the solver with α and β fixed at 1 and the solver with α and β allowed to vary differs in the number of steps that they take. The results are somewhat similar because, as we noted before, α and β tends to stay close to 1 and rarely deviates to a value smaller than $\frac{1}{4}$ or larger than 4.

1.5 Final recommendations

As we have noted before, all the interpolants have a V-shaped defect. We should note that experimentally, the trough for these V-shapes seem to be problem-independent. We thus used the optimal h value, that is the h value where the minimum error is found, as the initial h value for each solver. We need this, especially in the variable parameter case, so that the first few steps are accepted. These optimal values are as follows. For HB4, the optimal h value seems to be at about 10^{-3}, for HB6, the optimal h value seems to be at around 10^{-2} and for HB8, the optimal value seems to be at around 10^{-1} and 10^{-2}. See Appendix $A.1$ for more details.

We also experimented with using representations of the polynomials other than the monomial form to reduce the effect of the rounding-off error. See Appendix $A.2$ for more details.

Some recommendations for a final solver will be to start with the optimal h for the respective interpolant as the first step size so that the parameters do not get too bad for the first steps. In an ideal case, we would like to keep accepting steps at the start and allow the parameters to be $\frac{1}{2}$ for as long as possible.

We should solve with a solver with variable α at the start and then if the solver fails too many step repeatedly, that is, the parameters get too far from 1, we should use the technique of forcing the parameters to be 1 and using the previous interpolants.

The first recommendation guarantees that the first few steps are taken at the minimum error possible and thus that they succeed. The second recommendation guarantees that if we meet a tough problem in the middle, we would be able to step through it with static parameters at the cost of additional function evaluations.

We can also look to use the Barycentric or Horner form of the polynomials for some more accuracy.

We note that because of the V-shape, there is a high likelihood that we cannot solve any problem at very sharp tolerances (as sharper as 10^{-12}). However, as have shown in Appendix $A.3$, the solvers that were created were able to solve for tolerances of 2.5×10^{-12}. We should thus cap the tolerances and refuse to integrate if the user asks for sharper tolerances.

2 Future Works

2.1 Future work

The first few steps Throughout this chapter we have used the exact solution values for the first few steps in order to allow us to create the first interpolant. Another important research project in this area is to try different techniques including but not limited to the use of CRK methods, error control with a sharper tolerance than the user provided tolerance, and possibly other methods to perform the first few steps.

asymptotically correct defect control with multistep interpolants We can also look into developing interpolants that could lead to asymptotically correct defect control. This would guarantee that the maximum defect is always at the same relative location within each step and would thus only require one function evaluation to sample the defect.

HB10 An idea for future work is to derive a 10^{th} order interpolant. Such an interpolant will be forced to use 3 step-size parameters but an idea is to fix one or more of the parameters at 1. This can be done by using the technique that we employed in Section 8888 reference to section about static alpha 88888 or by

using another technique such as computing a solution value in the middle of the step $[x_{i-1}, x_i]$ using the interpolant from that step and performing an additional function evaluation at that data point to obtain the two values required to build an interpolant. Thus we get to use just two parameters α and β for the step from x_i to x_{i+1} and the step from x_{i-2} to x_{i-1}. This will give the required 10 data points to produce such an interpolant which could then be used to augment RK8 to provide a more efficient defect control scheme for the 8^{th} order case.

Early explorations into creating an HB10 interpolants seem to be promising. See Figure 92 to see how an HB10 derived by 'breaking the middle step' is resilient to changes to its parameters α and β. We note that the interpolant was built with step-size $[\alpha * h, h, h, \beta * h]$ and thus α and β is usually 2 when there are no step-size changes. We also note that θ was allowed to vary between $-2 - \alpha$ to β.

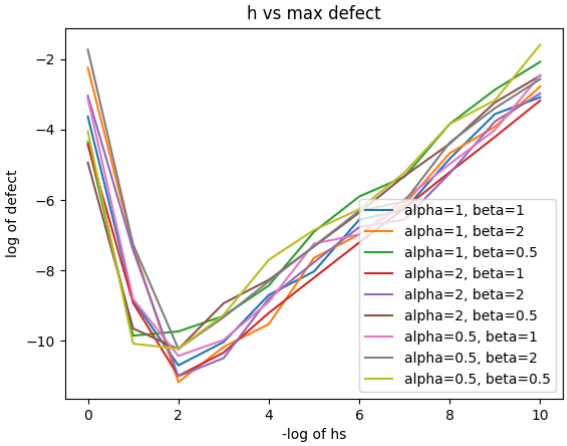

Figure 92: V-shape of HB10 created by 'breaking the middle step'

2.1.1 Doing error control instead of Defect control

Another idea is to consider error control instead of defect control. We would thus need a way to create two interpolants, one of a higher order and one of a lower order and sample the difference between these two interpolants to estimate the error of the continuous solution approximation. A step-size selection algorithm based on that error estimate could provide an effective error controlled solution.

The main problem we face with defect control is the V-shape of the accuracy. We know that this is entirely because of the $\frac{1}{h}$ in the derivative definition of the Hermite-Birkhoff interpolants as the interpolant itself does not suffer from rounding-off error but its derivative does.

For all the schemes, the defect is V-shaped but the error itself is not. This is because the Hermite-Birkhoff interpolant does not contain a term in $\frac{1}{h}$ whereas

its derivative does contain such a term. Figure 93 and 94 shows this phenomenon for HB6 but the same can be see for HB4 and HB8.

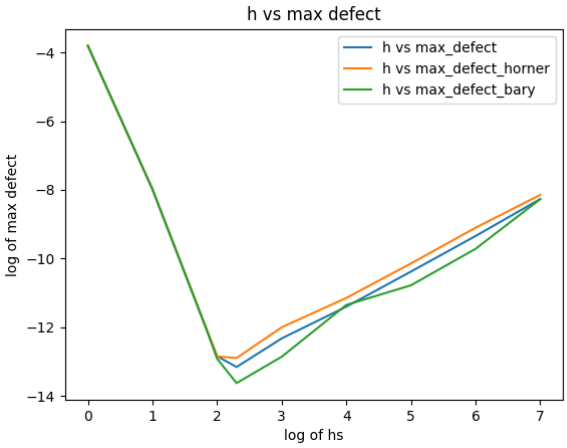

Figure 93: Defect is V-shape

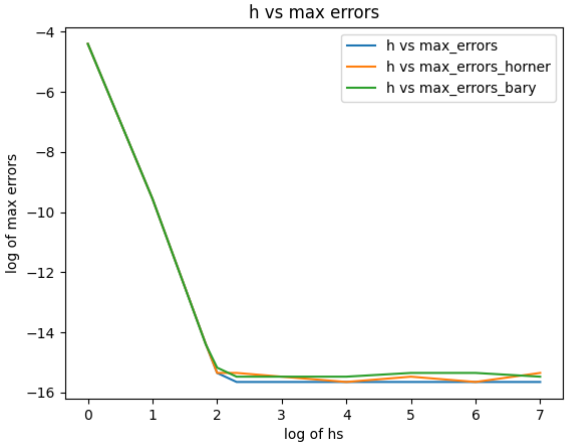

Figure 94: Error is not V-shape

A Final recommendation to be able to solve at sharp tolerances

A.1 The Vshaped graph and dependence on unit weight parameters

As we have shown several times in this report, the maximum order of accuracy of an interpolant depends on the value of the parameter α for the HB6 case and α and β in the HB8 case. Furthermore, the accuracy also follows a V-shape as the interpolation error for sufficiently small h reaches the round-off error. This is because the derivative of the interpolant uses a term in $\mathrm{O}(\frac{1}{h})$ where h is the step-size. In this section we will look into how much these issues affect our interpolation.

HB4 In HB4, the error across several problems, across several sampling points, and at different step-sizes is as shown in Figure 95. We note that this is without any parameters and at each h and each point x, we sampled at x and $x + h$ to create the interpolant. The scheme suffers from an $O(\frac{1}{h})$ rounding off error inherently.

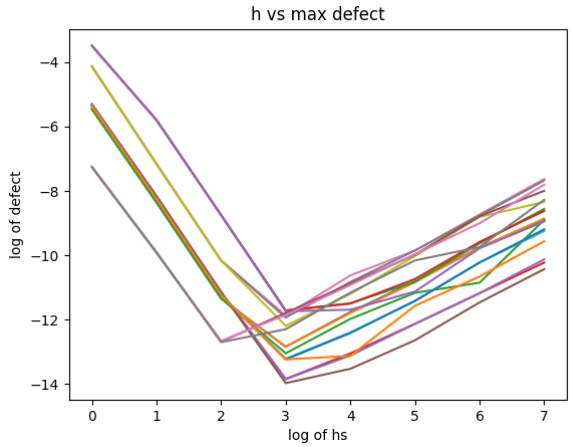

Figure 95: V-shape of HB4 across problems at several sampling points and with h

We note that there is a clear optimal h at about 10^{-3} and that we are able to achieve a tolerance of 10^{-14} with some problems and 10^{-12} with almost every other problem. This gives us hope that this scheme can be used at very sharp tolerances. In this section, we will try several techniques to improve the situation.

HB6 In HB6, the error across several problems, across several sampling points, and at different step-sizes is as shown in Figure **??**. We note that that the parameter α is constant at 1 throughout the whole process as we sample at the point, x and the points $x+h$ and $x-h$. This is the ideal case as the interpolation error is thus minimised.

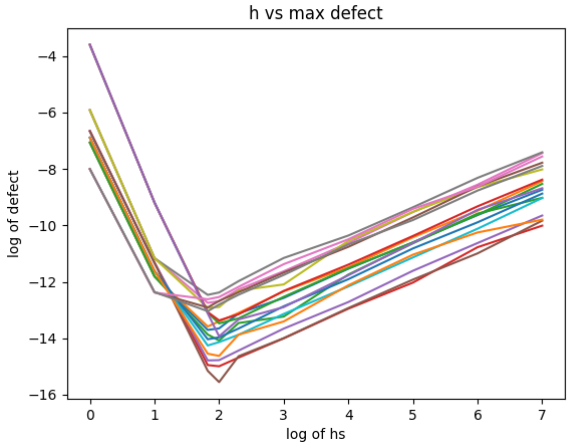

Figure 96: V-shape of HB6 across problems at several sampling points and with h

We note that α was kept at 1 in Figure 96. To see how the parameter α reduces the accuracy as it deviates from 1, see Figure 26.

We can see that the optimal h is at around 10^{-2}. We note that in most cases we were able to reach an error of 10^{-14} but in some cases we were only able to solve at 10^{-12}.

HB8 In HB8, the error across several problems, across several sampling points, and at different step-sizes is as shown in Figure 97. We note that that both parameters α and β were constant at 1 throughout the whole process as we sample at the point, x and the points $x+h$, $x-h$ and $x-2h$. This is the ideal case as the interpolation error is thus minimised.

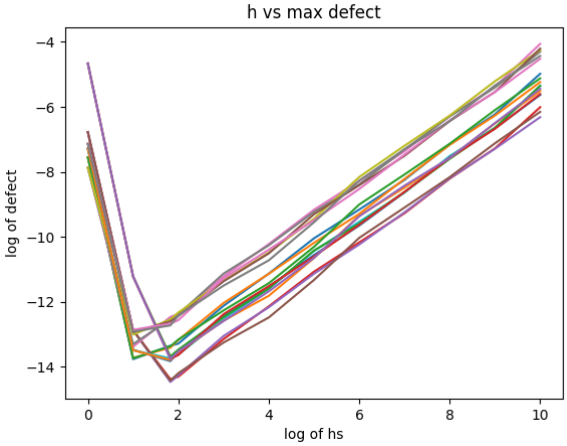

Figure 97: V-shape of HB8 across problems at several sampling points and with h

We note that both α and β were kept at 1 in Figure 97. To see how the parameters α and β reduces the accuracy as it deviates from 1, see Figure **??**.

We can see that the optimal h is at around 10^{-1} and 10^{-2}. We note that in most cases we were able to reach an error of 10^{-14} but in some cases we were only able to solve at 10^{-13}.

A.2 Horner's and Barycentric interpolants to get to lower tolerances.

We note that up to this form, we have been using the monomial form of each of the cubic, quintic and septic polynomials that we have derived. In the previous section, we have shown how these were suffering from being eventually dominated by the rounding-off error as we use smaller and smaller step-sizes. One idea to deal with the loss of accuracy due to the interference from the rounding-off error is to use a different representation of the polynomials.

One idea, borrowed from 8888 Give reference to your paper in the Summer... 8888 is to use the Horner's form of the polynomials. The Horner's form of a polynomial minimises the number of multiplications that need to be undertaken during the evaluation. It is faster and also a less prone to rounding-off errors since fewer arithmetic operations are involved.

Another idea is to use a Barycentric interpolant. At the time of the creation of the interpolant, if it is of order n over the interval $[a, b]$, we can find n Chebyshev points in the interval and then sample the interpolant at these n points and then fit a Barycentric interpolant to these data points. A Barycentric interpolant using n data points is of order n and if such an interpolant is used to interpolate a polynomial of order n, it perfectly matches the polynomial that

is, this process gives a different but exact representation of the original polynomial. The Chebyshev points are used to guarantee the minimum interpolation error. Additionally, as shown in 8888 Reference your paper in the Summer 88888, it reduces the 'noise' in the interpolant and this reduces interference from rounding-off errors.

We will use these two interpolation schemes to supplement the Hermite Birkhoff interpolants HB4, HB6 and HB8 at a few different sampling points at different problems and report on the improvements that they give.

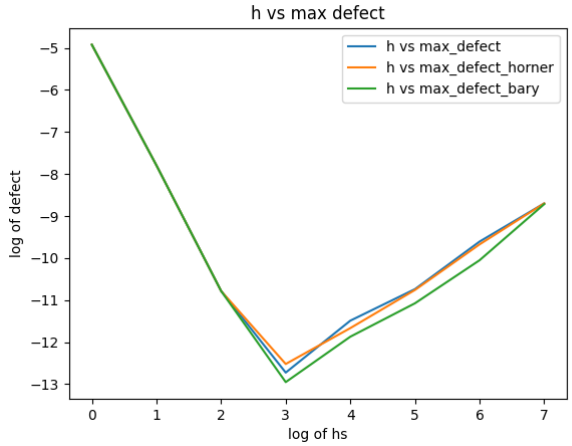

Figure 98: Typical plot of the HB4 interpolant monomial form vs the Horner form vs the Barycentric form

Figure 98 shows a typical plot of HB4 in the monomial form alongside its Horner form and Barycentric interpolation forms. We can report that the Horner form almost always matches the accuracy of the interpolant but that the Barycentric form slightly improves the accuracy. The V-shape is inevitable as both forms either themselves suffer from $O(\frac{1}{h})$ rounding-off error or interpolate over an interpolant that suffers from such a rounding-off error.

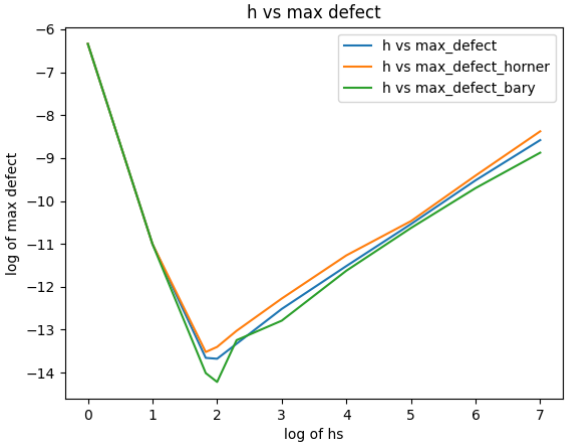

Figure 99: Typical plot of the HB6 interpolant monomial form vs the Horner form vs the Barycentric form

Figure 99 shows a typical plot of HB6 in the monomial form alongside its Horner form and Barycentric interpolation forms. We note that the parameter α is kept at 1 in the above plot. We can report that the Horner form almost always matches the accuracy of the interpolant but that the Barycentric form slightly improves the accuracy. The V-shape is inevitable as both forms either themselves suffer from $O(\frac{1}{h})$ rounding-off error or interpolate over an interpolant that suffers from such a rounding-off error.

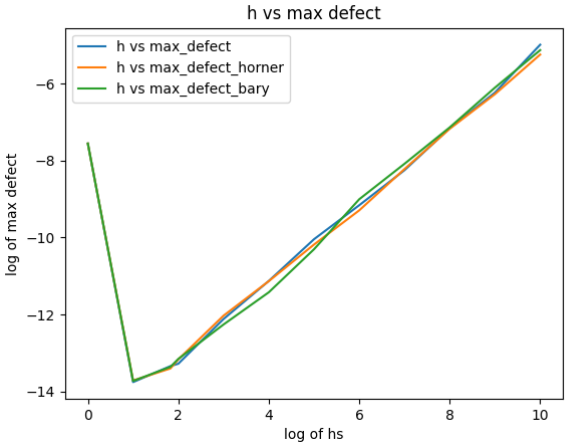

Figure 100: Typical plot of the HB8 interpolant monomial form vs the Horner form vs the Barycentric form

Figure 100 shows the typical plot of HB6 in the monomial form alongside its Horner form and Barycentric interpolation forms. We note that the parameters α and β are kept at 1 in the above plot. For the case of HB8, the use of the Horner method or even the Barycentric method does not improve the situation. The monomial form is as accurate as we can get. The V-shape is inevitable as both forms either themselves suffer from $O(\frac{1}{h})$ rounding-off error or interpolate over an interpolant that suffers from such a rounding-off error.

A.3 Solving at sharp tolerances

Using the recommendations, we show that the scheme can be used at very sharp tolerances for RK4 with HB6 and RK6 with HB8. In the plots below, we do not employ the switch from variable parameters to static parameters as they are not needed. The first few steps are all necessarily small and thus do not go out of range. The initial h value is the experimental optimal h value and we use the static parameters solvers.

RK4 with HB6

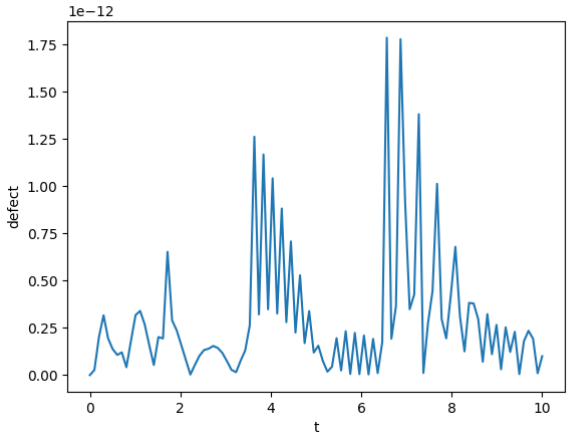

Figure 101: Global Defect of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 1 at an absolute tolerance of 2.5×10^{-12}

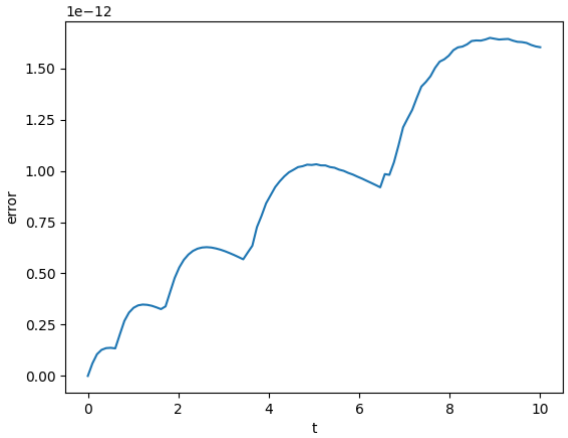

Figure 102: Global Error of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 1 at an absolute tolerance of 2.5×10^{-12}

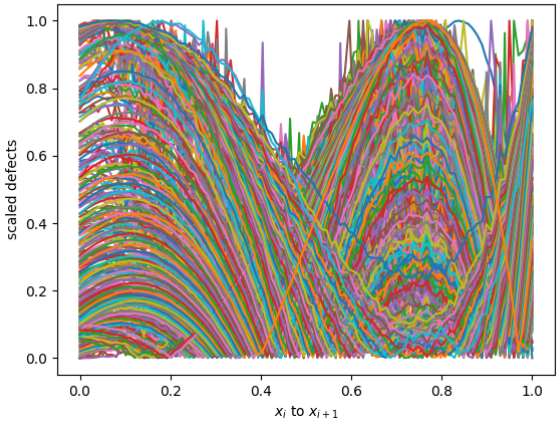

Figure 103: Scaled Defects of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 1 at an absolute tolerance of 2.5×10^{-12}

Problem 1 results

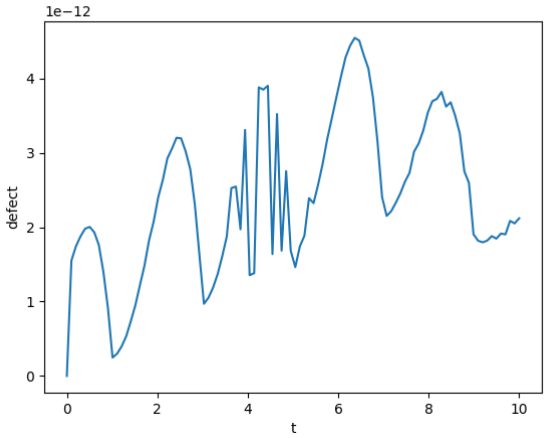

Figure 104: Global Defect of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 2 at an absolute tolerance of 2.5×10^{-12}

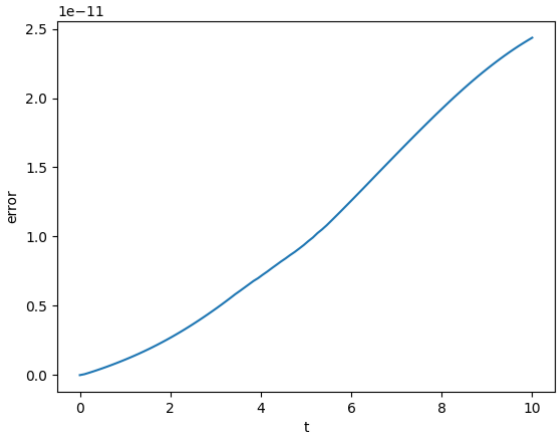

Figure 105: Global Error of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 2 at an absolute tolerance of 2.5×10^{-12}

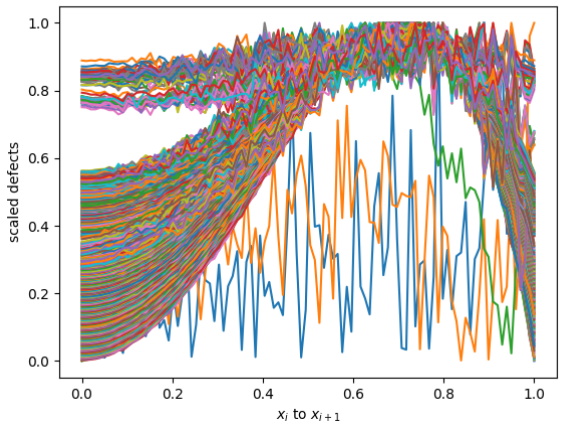

Figure 106: Scaled Defects of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 2 at an absolute tolerance of 2.5×10^{-12}

Problem 2 results

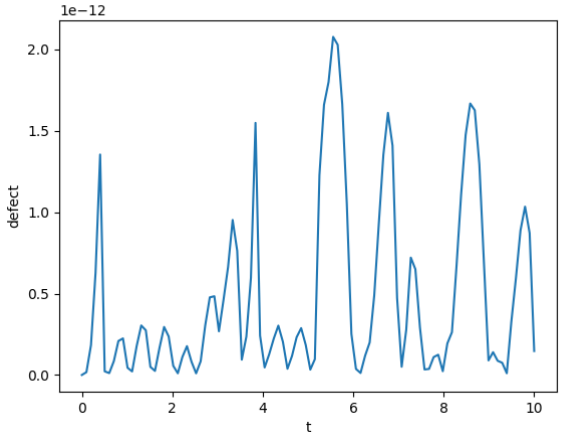

Figure 107: Global Defect of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 3 at an absolute tolerance of 2.5×10^{-12}

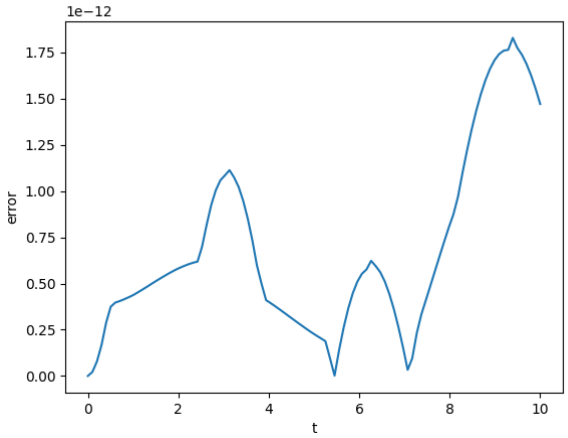

Figure 108: Global Error of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 3 at an absolute tolerance of 2.5×10^{-12}

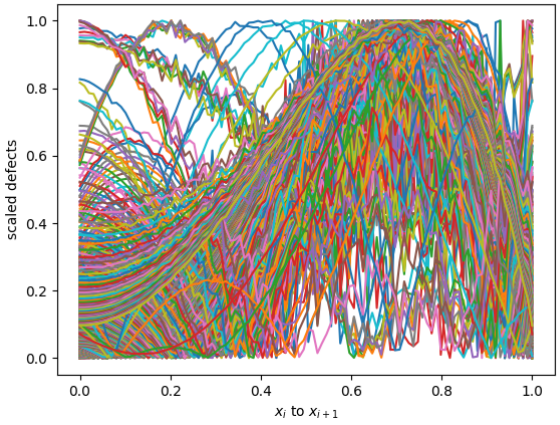

Figure 109: Scaled Defects of RK4 with HB6 using alpha = 1 and optimal h as the starting h value on problem 3 at an absolute tolerance of 2.5×10^{-12}

Problem 3 results

RK6 with HB8

Table 9: RK4 with HB6 using static alpha and optimal h as the starting h value at sharp tolerance

Problem	n successful steps	nsteps
1	443	523
2	534	568
3	1378	1731

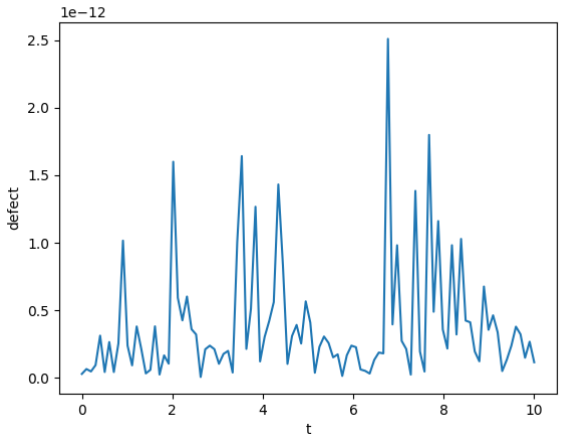

Figure 110: Global Defect of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 1 at an absolute tolerance of 2.5×10^{-12}

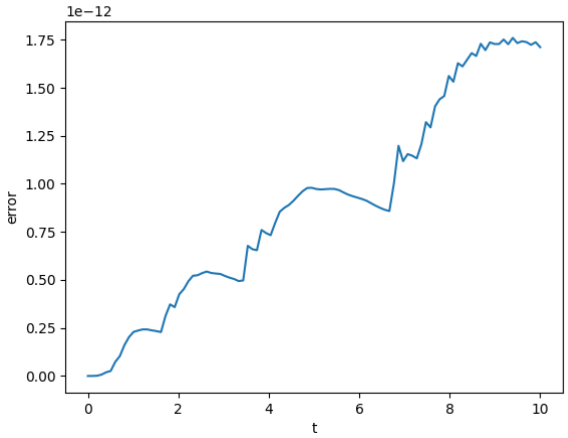

Figure 111: Global Error of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 1 at an absolute tolerance of 2.5×10^{-12}

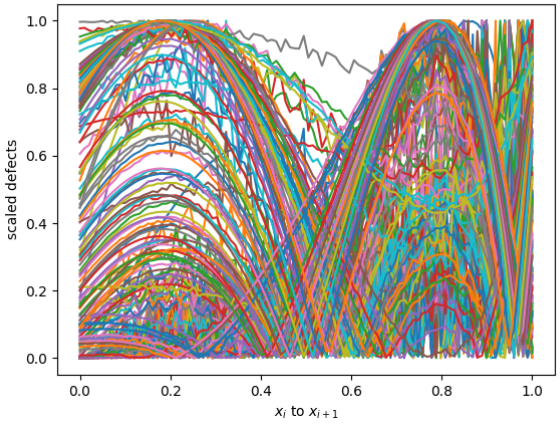

Figure 112: Scaled Defects of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 1 at an absolute tolerance of 2.5×10^{-12}

Problem 1 results

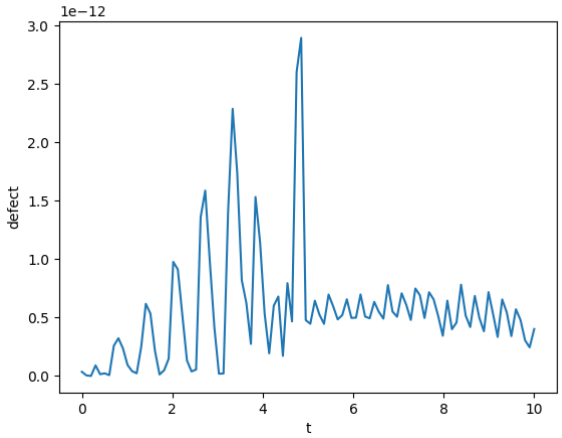

Figure 113: Global Defect of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 2 at an absolute tolerance of 2.5×10^{-12}

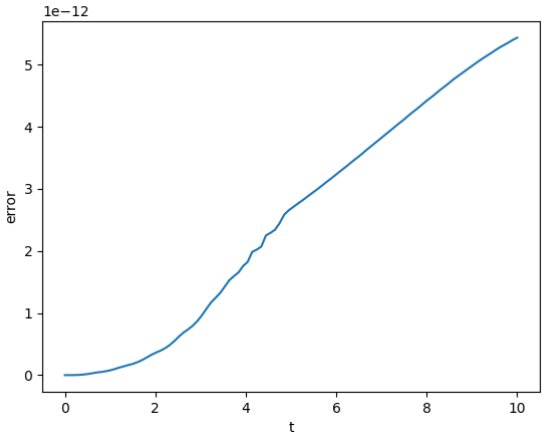

Figure 114: Global Error of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 2 at an absolute tolerance of 2.5×10^{-12}

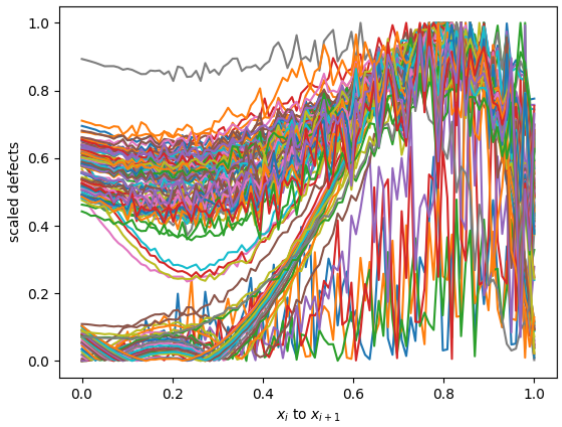

Figure 115: Scaled Defects of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 2 at an absolute tolerance of 2.5×10^{-12}

Problem 2 results

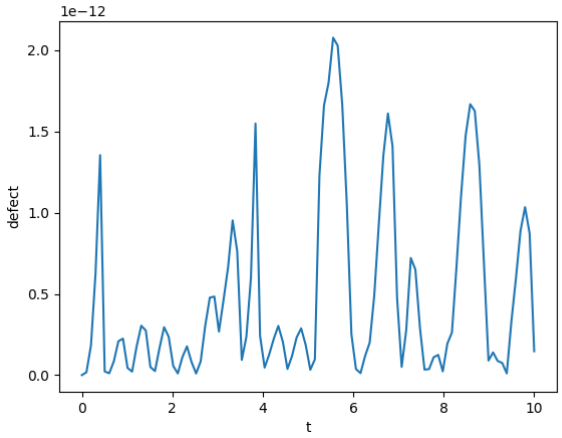

Figure 116: Global Defect of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 3 at an absolute tolerance of 2.5×10^{-12}

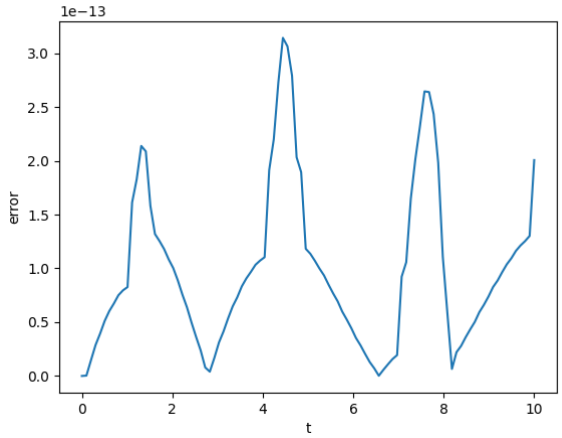

Figure 117: Global Error of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 3 at an absolute tolerance of 2.5×10^{-12}

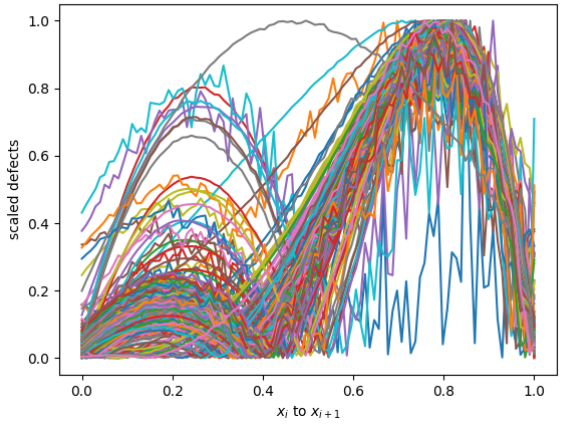

Figure 118: Scaled Defects of RK6 with HB8 using alpha and beta = 1 and optimal h as the starting h value on problem 3 at an absolute tolerance of 2.5×10^{-12}

Problem 3 results

Table 10: rk6 with hb8 using static alpha and beta and optimal h as the starting h value at sharp tolerance

Problem	n successful steps	nsteps
1	261	290
2	134	196
3	297	466