

1 Low Cost Defect Control with Hermite-Birkhoff interpolants

1.1 Introduction

why defect control. applications. Say that we often need continuous solution, especially to find derivatives/integrate/in bigger software. application to BVPs...

1.1.1 Problems used

Give a small description of the problems used - pick the non-anomalous data. One with growth. one with decay and one which oscillates (I think there is one which oscillates that does well.)

1.1.2 Lack of Defect Control of end of step solvers

Show that results are encouraging but still not as good a control of the defect as we might like it.

1.1.3 Cost of Traditional CRK-based Defect Control

Discussion of Enright's work <https://www.cs.toronto.edu/~enright/recentMSc.html>
Say that we will examine a proof of concept of a new low cost interpolant

1.1.4 Small explanation of our basic Runge Kutta

Need to say that for all the codes below the Runge Kutta methods we use are basic solvers. they so the step and do step resizing based on $[0.2\text{tol}, 0.8\text{tol}]$ and start with an initial step-size of $\sqrt{\text{tol}}$. (Basically, need to point out that we just built an 'inefficient' proof of concept).

Say that we are halving the steps when $\geq 0.8*\text{tol}$ and doubling the steps when error $\geq 0.2 * \text{tol}$...

Small description of classical Butcher 4. rk6 from Jim Verner and your rk8. Say their number of stages and their order. explain that though they come as pair.

1.2 Where we got the idea = Defect control with Hermite interpolant. rk4 with hb4

1.2.1 Discussion of HB4

cubic spline. Free interpolant. Free defect control. CAN SAMPLE ONLY TWICE as we know where the maximum is going to be.

1.2.2 results

Show that we can keep the defect under control =, satisfy a tolerance of $1e-6$. show nsteps but say we can do better. Point to explanation in next section.

Show a plot of some of the defects to show that we need to only sample TWICE

1.2.3 problems with interpolation error

interpolation error is too big $O(h^4)$ interpolation error vs $O(h^4)$ rk4 error. differentiating cubic spline gives an $O(h^3)$ interpolation error for the derivative. Interpolation error is thus interfering. If we could get a more accurate interpolant we would guarantee that interpolation error, especially for the derivative interpolant, is lower.

To do that we need to talk about HB6.

1.3 Defect control with HB6. rk4 with HB6

1.3.1 Discussion of HB6

Derivation. order. order of derivative. Because the interpolant has order 6. and its derivative have order 5, numerical errors now dominate.

1.3.2 results

Keeps defect under control. uses less steps.

will look to make the results better with HB8. Show a plot of some of the defects to show that we need to only sample TWICE

1.3.3 problems with alpha

Show problems with alpha. small deviation of alpha from 1 creates a problem. Show v-shaped graph at several alpha Discuss that alpha does not get TOO BAD. rarely 4. rarely $1/4$. Usually 0.5, 1, 2

1.4 Defect control with HB8 - rk4 with HB8

1.4.1 Derivation of HB8

First Scheme Show derivation with βh , αh , h Show problem. If α , β is not 1. this is useless. Say that there is a better derivation.

Second Scheme Show derivation with αh , h , βh show that more resilient to α , β not being 1.

1.4.2 results

Show that there isn't a big improvement. Discuss how this may help with higher order Runge Kutta methods. Show a plot of some of the defects to show that we need to only sample TWICE

1.4.3 problems

reliance of alpha, beta being 1 though better than first scheme.

1.5 Higher Order Runge Kuttas

Discuss how CRK is exponential with the order. Will require 27 function evaluations with 8th order but we get it for free. (2 function evaluations for sampling only.)

1.5.1 rk6

Discuss rk6 with HB6 and rk6 with *HB8_second_scheme*. show uses less steps as expected

1.5.2 rk8

Discuss rk8 with *HB8_second_scheme*. Say that this is way cheaper. propose that we may need to go to HB10.

1.6 the Vshaped graph

1.6.1 HB6

show that several problems with the interpolant run on random data points) have the same optimal h. Possibility to derive optimal h. This way a solver based on this approach will try BIG steps. if failed we reduce the step to that optimal h. If failed still, we cannot solve the problem.

1.6.2 HB8

show that several problems with the interpolant run on random data points) have the same optimal h. Possibility to derive optimal h. This way a solver based on this approach will try BIG steps. if failed we reduce the step to that optimal h. If failed still, we cannot solve the problem.

1.7 Static alpha

Because we have interpolants on all previous steps. We can use these interpolants to look up: basically for HB6 if going from x_i to $x_{i_{plus_1}}$ with a step of size h. we can look up $x_{i_{minus_1}}$ using alpha=1 with `previousInterps.eval(x_i - h)` for $y_{i_{minus_1}}$ and `previousInterps.prime(x_i - h)` for $f_{i_{minus_1}}$

Show results of rk4 with hb6 static alpha to show that we can get defect control with that idea.

basically for HB8 if going from x_i to x_{i+1} with a step of size h . we can look up $x_{i_m \text{inus}_1}$ with `previousInterps.eval($x_i - h$)` for $y_{i_m \text{inus}_1}$ and `previousInterps.prime($x_i - h$)` for $f_{i_m \text{inus}_1}$ we can look up $x_{i_m \text{inus}_2}$ with `previousInterps.eval($x_{i_m \text{inus}_1} - h$)` for $y_{i_m \text{inus}_2}$ and `previousInterps.prime($x_{i_m \text{inus}_1} - h$)` for $f_{i_m \text{inus}_2}$.

show results of rk4 with hb8 static alpha to show that we get defect control

1.7.1 pros

NO COST. same cost as variable alpha (only two function evaluations for the sampling). Keep using optimal h throughout the whole process.

1.7.2 problems

the first step. we may have to artificially restrict the GROWTH in the step-size. 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 $2*h$ as $t_0 + h - 2*h = t_0 - h$ but we dont have an interpolant in the region $j t_0$.

1.8 Future work

1.8.1 additional problems to solve to get this to work

the first/first and second step take an error control step with a stricter tolerance than the user provided tolerance. Take a CRK interpolant step.

V-shaped error Derive optimal h . keep solver using BIG steps. reduce to optimal h .

static alpha use variable alpha for the first few steps. Usually not a problem as in all our experiments alpha is at a good values. If alpha goes to 8, use static alpha.