Telescopic Projective Methods for Stiff Differential Equations

C. W. Gear*
Ioannis G. Kevrekidis†

April 3, 2002

Abstract

Projective methods were introduced in an earlier paper as having potential for the efficient integration of problems with a large gap between two clusters in their eigenvalue spectrum, one cluster containing eigenvalues corresponding to components that have already been damped in the numerical solution and one corresponding to components that are still active. In this paper we introduce iterated projective methods that allow for explicit integration of stiff problems that have a large spread of eigenvalues with no gaps or with several gaps in their spectrum.

Keywords Integration, stiff, explicit, stability, dimension reduction, inertial manifolds

1 Introduction

In [1] we introduced the projective method which is based on the following very simple idea: any stable method (the inner integrator) is used to integrate a problem over a number of small steps and then a *projective*[1] step uses polynomial extrapolation to compute an approximation to the solution far ahead of the inner integration steps. The first k steps of the inner integrator serve to damp the fast components in the solution. The projective step then uses the result of the last step and the results from the next q inner steps to extrapolate forward. This is shown in Figure 1 with $k = 2$ and $q = 1$. It is clear

*NEC Research Institute, retired

†Princeton University, Dept of Chemical Engineering. Supported in part by AFOSR (Dynamics and Control program, Dr. Mark Jacobs)

[1]The term *projective method* was used rather than *extrapolation method* because the latter term has already been used for another class of methods. *Projective* should be associated with *projectile* and not be confused with the mathematical term *projection*.

that the slope of the chord through y_2 and y_3 in Figure 1 is a first-order approximation to the derivative so we call this example a Projective Forward Euler (PFE) method.

The combination of the inner steps and the projective step effectively constitutes another integrator which we call the *outer* integrator. It was shown that this integrator could be constructed so that it was absolutely stable (hereafter referred to as "stable") if the eigenvalues of the problem Jacobian matrix were in one of two regions, one corresponding to rapidly decaying components handled very stably by the inner step and one corresponding to an approximation to the stability region of the outer integrator (the Forward Euler in the Figure 1 example.) Thus, for problems with a large gap between the time constants of the fast inactive components and the time constants of the slow active components (those still causing changes in the solution) one is able to project forward over large steps commensurate with the slow components and gain speed (i.e., increase average step size). In the example shown in Figure 1 we use three inner integrations steps of length h_0 to cover a distance $h_1 = (3 + M)h_0$. If we assume that the inner integration step represents the bulk of the work (because, for example, evaluations of the derivatives are very expensive) and that it is not possible to use a larger step size in the inner integrator, then we can define the speedup of the projective method as *the number of inner integration steps needed to integrate over the interval directly divided by the number used when combined with the projective step.* Thus the PFE method in Figure 1 has a speedup of

$$S = (3 + M)/3$$

In this paper we consider an obvious extension of the projective method: Since the outer, or projective integrator can be viewed as just another integrator, why not use it as the inner integrator in yet a further projective integrator, and so on, *ad infinitum*? This is illustrated in Figure 2 which has two projective levels. Note that in this illustration the speedup of the 2nd level method is $[(3 + M)/3]^2$ since we cover a distance $(3 + M)^2 h$ with 9 inner integrations.

We will show that the methods resulting from this iteration of the projective step can have two quite different set of properties. In one case they can handle problems that have multiple gaps - that is, whose eigenvalues lie in one of a number of well-separated regions of the complex plane, as shown in Figure 3. In the second case we can chose the method parameters so that its stability region includes a large section of the negative real axis and neighboring points in the complex plane - methods we will call "[0,1] stable" for reasons that will be apparent later.

In the next section we will briefly review the stability analysis of Projective methods and their important properties so that we can discuss [0,1] stable methods in the third section and multiple gaps in the fourth section.

2 Stability Analysis

The usual linear stability analysis of time-stepping methods discusses stability in the $h\lambda$-plane, where h is the time step and λ is an eigenvalue of a local linearization of the problem. We are applying a projective step to any inner integrator (we may have, for example, a legacy code that performs one time step in a manner we do not fully understand). Since the nature of the inner integrator affect the stability and we wish to analyze the stability of the projective process independently of the nature of the inner integrator, we analyze stability as a function of the error amplification, $\rho(h_0\lambda)$, of the inner integrator, where h_0 is the time step of the inner integrator. For example,

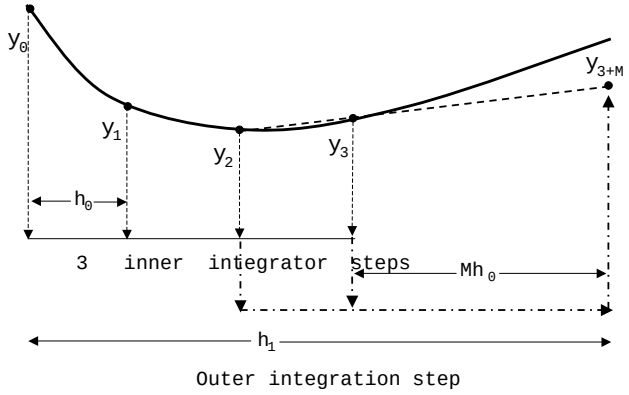

Figure 1: Projective Forward Euler method with $k = 2$

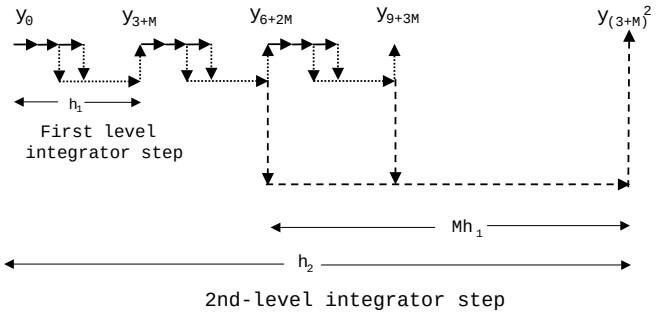

Figure 2: A Two-level Projective Integrator

if the inner integrator were *exact* then $\rho = \exp(h_0\lambda)$, while if the inner integrator were Forward Euler, $\rho = (1 + h_0\lambda)$. (Of course, we cannot have an exact numerical integrator unless the problem is particularly simple, but it is possible that a legacy code or a more detailed simulation model provides an almost exact integration over one step for the range of problems considered, albeit at a heavy computational cost). Note that if the inner integrator is the forward Euler (FE) method, then the ρ-plane is simply a unit translation of the more conventional $h\lambda$-plane since in that case $\rho = 1 + h_0\lambda$.

The projective method consists of two processes: an inner integrator and a projective step. For example, the PFE method completes one outer step over a distance $h_1 = (M + k + 1)h_0$ from y_n as follows:

1. Form y_{n+i} for $i = 1, 2, \cdots, k + 1$ starting from y_n using an inner integrator.

2. Form
$$y_{n+k+1+M} = (M + 1)y_{n+k+1} - My_{n+k} \tag{1}$$

If the inner integrator has a one step error amplification of ρ, then, as discussed in [1], the error amplification of one outer step of the PFE is

$$\sigma = [(M + 1)\rho - M]\rho^k \tag{2}$$

It was shown in [1] that the stability region of this method breaks into two separate pieces whenever M is more than about three times k. In other words, for larger M the method is only valuable for problems with a gap in their spectrum. This is illustrated in Figure 4 which shows the absolute stability region in the ρ-plane for PFE method with $k = 2$ and $M = 9$, that is, the method consists of three inner steps of size h_0 followed by a linear projective step over a distance of $9h_0$ from the last two computed values. The above example covers twelve steps for each three derivative evaluations, so the speedup is four but there has to be a gap in the problem spectrum for this parameter choice to be stable for the problem.

In contrast, if M is smaller a single region of absolute stability is obtained, as illustrated in Figure 5, which is the stability region for the same method but with $M = 5$. However, this method has a speedup of only 8/3.

Note that the stability region in Figure 5 includes all of the real axis in the ρ-plane from 0 to 1. We will call this a [0,1] stability region and call such methods *[0,1] stable*. If the inner integrator is the Forward Euler method, $\rho = 1 + h_0\lambda$ so any real $\lambda \in [-1/h_0, 0]$ maps into $\rho \in [0, 1]$. Similar results apply to most explicit methods. Hence [0,1]-stable methods are potentially useful when problems have a spread of eigenvalues along the negative real axis.

The disadvantage of [0,1]-stable methods is that we have shown that their speedup can not be much larger than three. In the next section we show how the projective methods can be applied recursively to achieve [0,1] stability regions with greater speedup.

In the generalized projective method we iteratively apply a new outer projective step to the results of the previous outer step. To express the general form, it is convenient to change notation slightly. Write one step of the inner integrator as

$$y_{s+1} = \Phi_0(y_s)$$

so that one step of the outer integrator (for the PFE method) starting from a value z_r is:

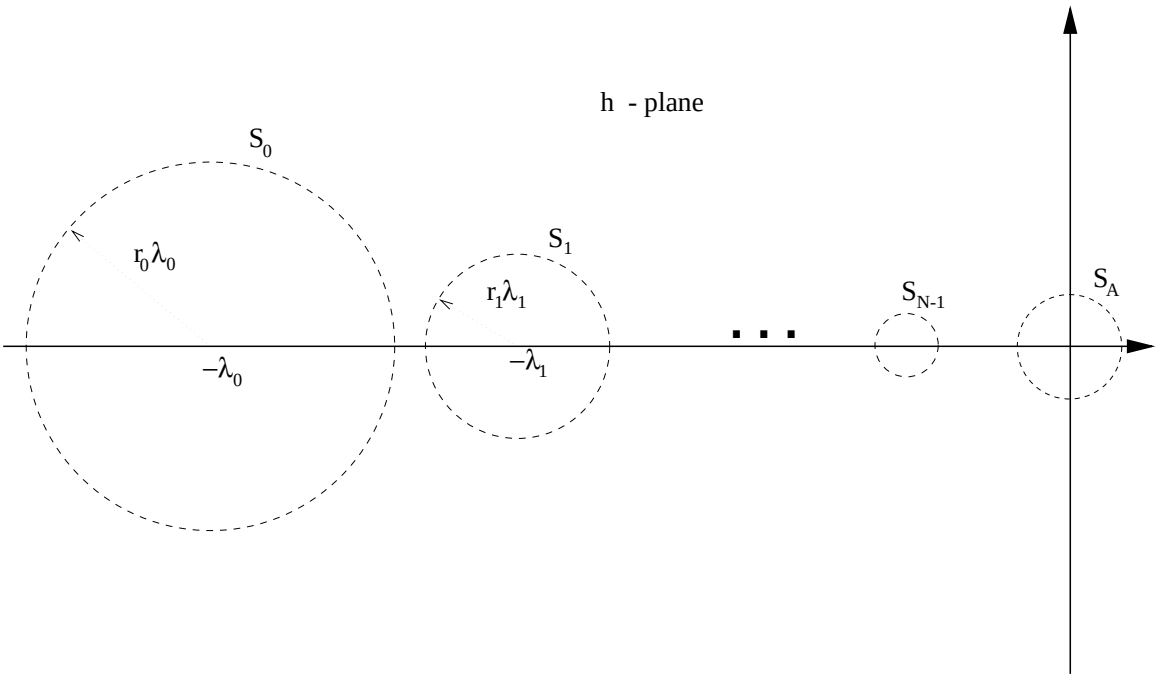

Figure 3: Eigenvalue Clusters

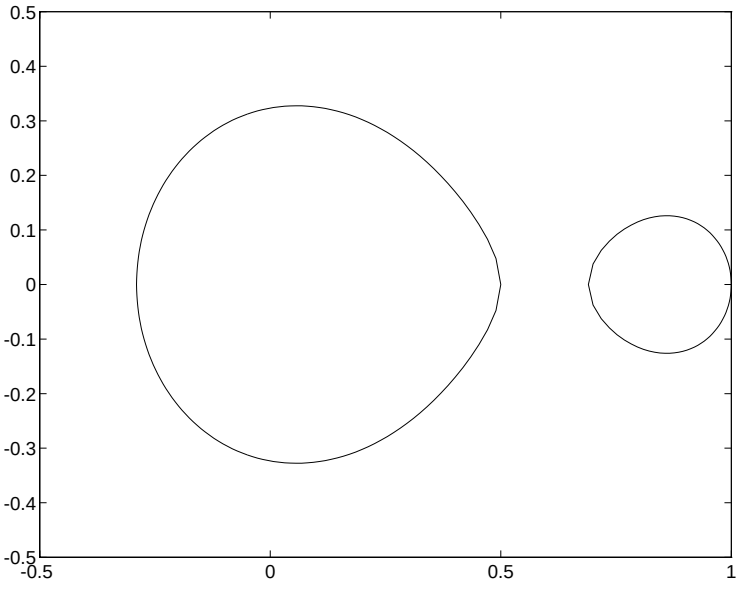

Figure 4: Complex ρ-plane stability for P2-1-9 Method

1. Set $y_0 = z_r$
2. Form $y_i = \Phi_0(y_{i-1})$ for $i = 1, 2, \cdots, k+1$
3. Form $z_{r+1} = (M+1)y_{k+1} - My_k$

We now refer to this as the *first level* outer integrator, and write it as:

$$z_{r+1} = \Phi_1(z_r)$$

The m-th level outer integrator, Φ_m, is defined recursively by:

1. if $m = 0$, Φ_m is the inner integrator
2. if $m > 0$ then $\Phi_m(z)$ is defined by the process
   (a) Set $y_0 = z$
   (b) Form $y_i = \Phi_{m-1}(y_{i-1})$ for $i = 1, 2, \cdots, k+1$
   (c) Form $\Phi_m(z) = (M+1)y_{k+1} - My_k$

Although we have used recursion for its definition, the method is applied *iteratively*, that is, from the bottom up. We start with an inner integrator and, after sufficient steps ($k+1$ in the discussion above) have been taken, we take a level-1 outer step. After sufficient level-1 steps (each of which involves multiple level-0, or inner, steps) have been taken, we take a level-2 outer step, and so on. Each successive outer level "looks forward" or *telescopes* over many lower levels. Hence we call it a *Telescopic Projective*, or TP, method.

For now, we will consider the stability of the fixed step, constant k-M TP method. We will assume that ρ for the inner method is constant. (This is equivalent to the standard linear, constant coefficient analysis assumption. Showing that a method is stable for such problems is not sufficient to show stability for more general problems without significant additional assumptions, but methods that are not linearly stable are seldom worth using for any problems!)

We define the stability region of the TP method to be the set of ρ such that all outer integrators are stable. Suppose the amplification of one step of the m-th outer integrator is σ_m. Then we have

$$\sigma_{m+1} = [(M+1)\sigma_m - M]\sigma_m^k, \qquad m = 0, 1, \cdots \tag{3}$$

where $\sigma_0 = \rho$. Hence the stability region of the TP method is the set of ρ such that the iteration

$$\rho \leftarrow [(M+1)\rho - M]\rho^k \tag{4}$$

remains in the unit disk[2]. This is the same as ρ remaining bounded since, if $|\sigma_m| > 1$ eq. (3) implies

$$|\sigma_{m+1}| - 1 > (M+1)(|\sigma_m| - 1)$$

from which it follows that σ_m diverges for $M > 0$.

[2]Formally this will be a fractal set, but all that is important for our purposes is that the set contains one (or more) connected regions of stability

In practice, of course, we do not use an infinite number of iterations of the projective operation. If we stop after any given number of iterations, the resulting stability region will contain the stability region of the infinitely iterated method as defined above. Hence, if a linear problem's eigenvalues are in the stability region as defined above, the use of any finite number of iterations will result in a linearly stable method.

In the above discussion, we have assumed the same value of M and k for each level of recursion and a constant inner step size. That does not have to be the case and in a practical code we would probably dynamically change the method parameters based on estimates of equation parameters. However, that complicates any analysis of stability.

In many ways, the proposed methods are similar to the explicit Runge-Kutta methods with extended ranges of stability (see [2], [3], and [4]). Indeed, the iterated projective methods can be expressed in the Runge-Kutta formalism so it is appropriate to compare them with the Runge-Kutta methods. However, the methods presented here are a collection of relatively simple steps. It is possible to modify the steps size of inner integrators at any level "on the fly" as estimates of the eigenvalues may change. Using the Runge-Kutta formalism makes it difficult to change *internal* "step sizes" (i.e. the placement of the intermediate points in a single RK step) since it would require a dynamic recomputation the interior points during the integration - something that might be worth exploring for Runge-Kutta methods but a process that seems to present some challenging analysis problems.

3 Stability and Speedup of TP Methods

One way to compute the stability region is to map the unit circle under the inverse of the mapping (4) for a number of iterations. Actually, then we have the boundary of the stability region of that many applications of the TP method. Since the region shrinks at each iteration (and starts from the finite unit disk) it must converge and in practice we quickly get a reasonable approximation to the infinitely iterated region. Figure 6 shows the stability region for 10 iterations of the P2-1-3 method[3]. With $k = 2$ three inner integrations are used to cover one first-level outer integration step of length $6h_0$ so its speedup is 2. Each successive outer integration provides an additional doubling of the speedup, so after ten iterations we have an speedup of $2^{10} = 1024$. However, by this time, the outer step size is $6^{10}h_0$ which may well too large for accurate integration of the active components! Thus, the speedup will be limited by the relative speed of the fastest components compared to the active components. This will be discussed in the analysis below.

Figure 6 suggests that the real axis from about -0.25 to +1.0 is inside the stability region - although we have not shown that errors in calculation of the figure or further iterates do not change that conclusion. Can we always guarantee that a section of the real axis of the ρ-plane including [0,1] is always in the stability region so this method is [0,1] stable? The answer is that for any $k \geq 1$ and order extrapolation $q \geq 1$ (using the last $q+1$ values) there exists an $M_{k,q}$ such that all $\rho \in [0,1]$ are stable for any $M \leq M_{k,q}$. We will prove that statement at the end of this section.

The M chosen for Figure 6 is the largest consistent with [0,1] stability for $k = 2, q = 1$. Figure 6 suggests that the boundary of the stability region touches the real axis in the

[3]The designation Pk-q-M method means that k initial steps are used, a further q steps are used to generate a q-th order extrapolate through the last $q+1$ points, and the extrapolation is over M steps.

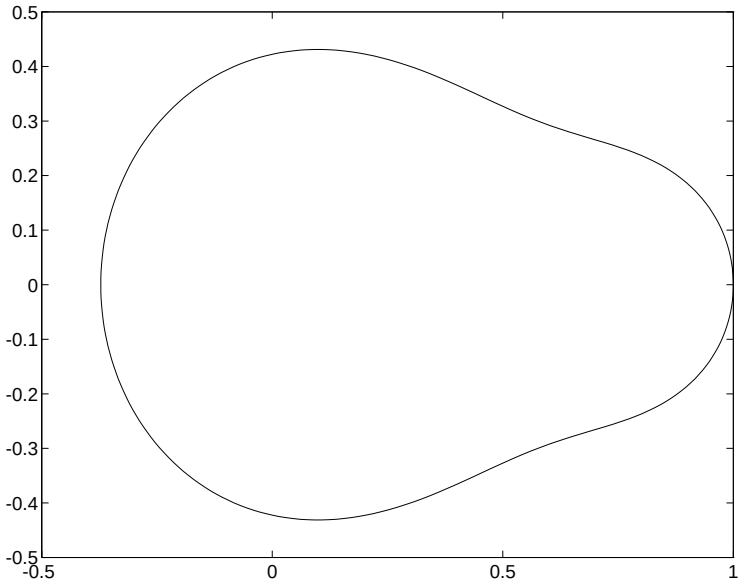

Figure 5: Complex ρ-plane stability for P2-1-5 Method

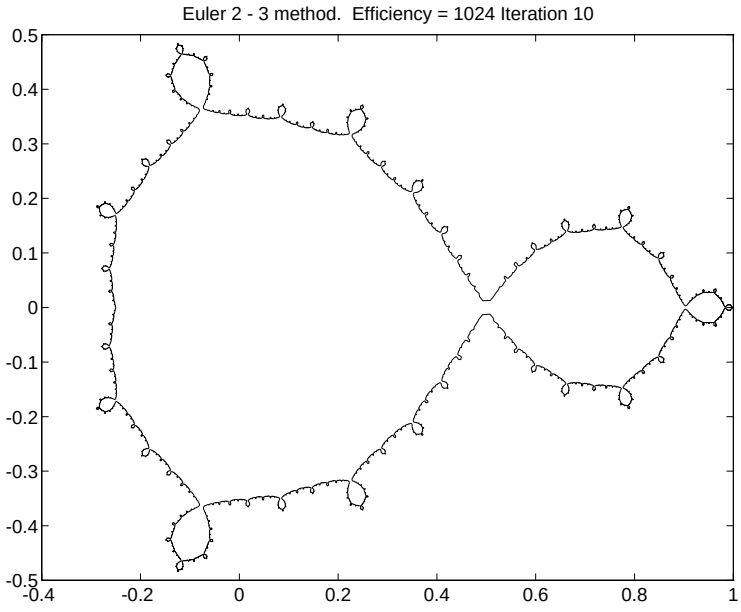

Figure 6: Stability Region for P2-1-3 Method after 10 Iterations

Table 1: Values of $M_{k,q}$

	q				
k	1	2	3	4	5
1	2.00	3.56	1.57	2.94	1.50
2	3.00	5.92	2.25	4.68	2.14
3	6.66	8.27	4.34	6.40	3.92
4	8.32	10.60	5.35	8.11	4.82
5	12.21	12.93	7.47	9.82	6.59
6	14.24	15.27	8.66	11.52	7.62
7	18.22	17.60	10.78	13.23	9.37
8	20.48	19.93	12.07	14.93	10.48
9	24.48	22.25	14.18	16.63	12.21
10	26.91	24.58	15.55	18.37	13.38

interval [0,1]. That is true, because M has its maximum value. Inevitably, when ones chooses the "best" value of one parameter, the limit is pushed on other criteria. We can keep the interior of [0,1] in the interior of the stability region by choosing a value of M smaller than the maximum allowed. For example, Figure 7 shows the stability region for ten iterations of the P2-1-2 method. However, the speedup of each projective step is 5/3 rather than the two of the P2-1-3 method.

Table 1 gives the values of $M_{k,q}$ for $1 \leq k \leq 10$ and $1 \leq q \leq 5$. (The way these can be calculated is indicated at the end of this section.)

Clearly, the larger M the greater the speedup of the method because it integrates over a greater distance for a given number of inner integration steps. The speedup of the first level of the projective step using the maximum value $M = M_{k,q}$ consistent with [0,1] stability is shown in Table 2.

Table 3 shows the speedup of the method after five iterations. It also shows the step size ratio - that is, the size of the 5th-level outer step as a multiple of h_0.

Although the speedups appear to increase significantly as k increases, in practice the largest outer step size will be limited, and the size of the outer step also increases with k. The effect is to make smaller values of k more efficient as shown below.

After the m-th level of iteration we have a speedup of

$$S_m = {S_1}^m = [(k + q + M)/(k + q)]^m$$

By this time the outermost step size is

$$h_m = (k + q + M)^m h_0$$

The largest outer step size that can be used will be limited by the time constant of the active components in the solution - the outer integrator must limit its step size for their accurate integration. Let us suppose that this time scale is Dh_0 - that is, we can use an

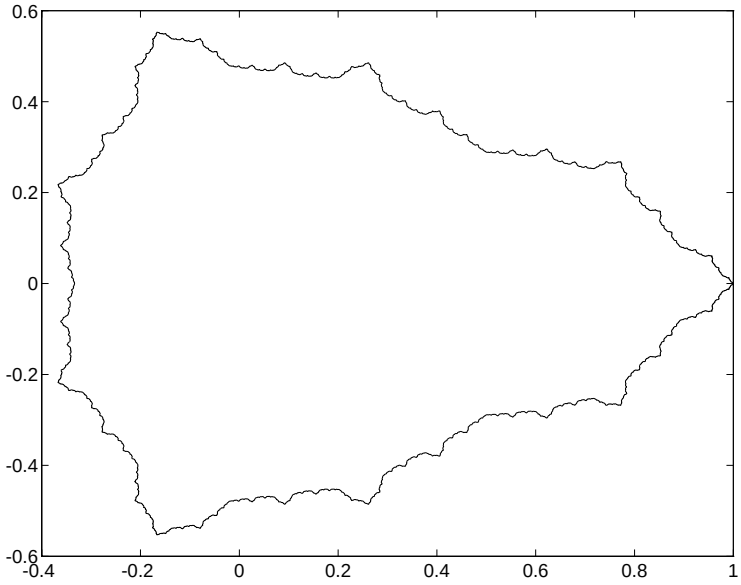

Figure 7: Stability Region for P2-1-2 Method after 10 Iterations

Table 2: Values of Speedup $(M_{k,q} + k + q)/(k + q)$

	q				
k	1	2	3	4	5
1	2.00	2.19	1.39	1.59	1.25
2	2.00	2.48	1.45	1.78	1.31
3	2.66	2.65	1.72	1.91	1.49
4	2.66	2.77	1.76	2.01	1.54
5	3.04	2.85	1.93	2.09	1.66
6	3.04	2.91	1.96	2.15	1.69
7	3.28	2.96	2.08	2.20	1.78
8	3.28	2.99	2.10	2.24	1.81
9	3.45	3.02	2.18	2.28	1.87
10	3.45	3.05	2.20	2.31	1.89

Table 3: Values of speedup and step size ratio for 5 iterations

	Speedup		Step ratio	
k	$q=1$	$q=2$	$q=1$	$q=2$
1	32.0	50.4	1,024	3,766
2	32.0	93.8	3,125	11E3
3	133.2	130.7	13E3	26E3
4	133.2	163.1	26E3	52E3
5	259.6	188.0	60E3	93E3
6	259.6	208.7	10E4	15E4
7	379.6	227.2	18E4	24E4
8	379.6	239.0	28E4	37E4
9	488.8	251.2	44E4	54E4
10	488.8	263.9	63E4	77E4

outer integrator with step size no larger than D times that of the inner integrator for accuracy. Consequently, we have

$$(k+q+M)^m \leq D$$

or the largest m is given by

$$m \approx \log(D)/\log(k+q+M) \tag{5}$$

With this number of iterations of the outer integrator, a total of $N_{k,q,M} = (k+q)^m$ inner integrations will have been used in one level-m outer step, or

$$N_{k,q,M} = (k+q)^{\log(D)/\log(k+q+M)} = D^{\log(k+q)/\log(k+q+M)} \tag{6}$$

Let us define $p_{k,q,M}$ to be the exponent of D in eq. (6). Since D inner integration steps would be used to integrate over the interval if no projective steps were done, the speedup is

$$S = D/N_{k,q,M} = D^{1-p_{k,q,M}}$$

The smaller the value of $p_{k,q,M}$ the greater the speedup. Since $p_{k,q,M}$ decreases as M increases, the smallest value of $p_{k,q,M}$ occurs when M takes its largest possible value consistent with stability, namely the value in Table 1. Let us call this value $p_{k,q}$. Its value is shown in Table 4. We see that $p_{k,q}$ increases with k, indicating that the speedup decreases as k increases.

It is interesting to compare this technique with the explicit Runge-Kutta (RK) methods with extended stability ranges discussed in [2], [3], and [4]. Let m be the maximum iteration level we can use, as given by eq. (5). If we view a single level-m outer step of the TP method as if it were a single RK step, it uses $s = D^{p_{k,q}}$ inner integrations, or *stages* assuming one function evaluation per inner step, to cover an outer step size of $H = Dh_0$.

Table 4: Values of $p_{k,q}$

	q				
k	1	2	3	4	5
1	0.50	0.58	0.81	0.78	0.89
2	0.61	0.60	0.81	0.76	0.88
3	0.59	0.62	0.76	0.75	0.84
4	0.62	0.64	0.77	0.75	0.84
5	0.62	0.65	0.76	0.75	0.82
6	0.64	0.66	0.77	0.75	0.82
7	0.64	0.67	0.76	0.75	0.81
8	0.65	0.68	0.76	0.75	0.81
9	0.65	0.68	0.76	0.76	0.81
10	0.66	0.69	0.77	0.76	0.81

Assume that the inner integration is explicit and its step size is $h_0 \approx 1/|\mathrm{Re}(\lambda)|$ where $\mathrm{Re}(\lambda)$ is the most negative real part of any eigenvalue. Then the method is stable for real eigenvalues λ as negative as given by $H|\lambda| \approx D = s^{1/p_{k,q}}$. For the case $k = 1$ and $q = 1$ from Table 4 we have the stability range $\approx s^2$. This compares with the stability range given in [4] of Cs^2, although the RK methods achieved that with a larger C and also for second-order methods. In that sense, those R-K methods are superior. However, the parameters in the RK methods were chosen to maximize their stability regions at the expense of simplicity and flexible choice of internal step sizes and order.

3.1 Existence of [0,1] stability regions

We will now prove that for any positive integers k and q such that the Pk-q-M TP method is [0,1] stable for all M less than some maximum $M_{k,q}$. We will do this by demonstrating that there is a line segment $[-\beta,1]$ with $0 \leq \beta < \infty$ that maps into itself under (4) and thus remains bounded. This section can be skipped by the reader uninterested in the details of the proof.

3.1.1 Case $q = 1$, first order

Let us first consider $k = 1$ and the plot of σ versus real ρ in eq. (2). It is illustrated in Figure 8. In general, let it be the map

$$\sigma(\rho) = f(M, \rho)$$

In this case, it is a quadratic function that has a minimum at

$$\rho_{min} = 0.5M/(M + 1)$$

and does not exceed +1 for $\rho \in [-\beta, 1]$. If the minimum of σ is $-\gamma = \sigma(\rho_{min})$ and if $\gamma \leq \beta$ then the interval $[-\beta,1]$ maps into itself. In this case, trivial algebra shows this to be true for $0 \leq M \leq 2$.

It is convenient to consider a requirement equivalent to $\gamma < \beta$ that is more useful for larger k and q. First note that the minimum γ is a function of M for any pair k and q. In this case we have

$$\gamma(M) = \frac{1}{M+1}\left[\frac{M}{2}\right]^2$$

The equivalent requirement is that the map of $-\gamma$ does not exceed 1, or

$$f(M,\gamma) = (M+1)\gamma^2 + M\gamma \leq 1 \tag{7}$$

$f(M,\gamma(M))$ is an increasing function of M for positive M. Hence, condition (7) is satisfied for all M in the interval $[0,M_{max}]$, where M_{max} is the smallest positive solution of $f(M,\gamma(M)) = 1$, namely 2.

If k is odd and larger than 1, the plot of $\sigma(\rho)$ is very similar to Figure 8 except that there is a multiple zero at the origin. In this case we find that the minimum occurs at

$$\rho_{min} = \frac{kM}{(k+1)(M+1)}$$

and

$$\gamma = -\sigma(\rho_{min}) = \left[\frac{M}{k+1}\right]^{k+1}\left[\frac{k}{M+1}\right]^k$$

The condition equivalent to eq (7) is

$$f_k(M,\gamma) = \gamma^k[(M+1)\gamma + M] \leq 1 \tag{8}$$

In eq. (8) f is an increasing function of M so we find that the condition is satisfied for all M in the interval $[0,M_{max}]$ where M_{max} is the smallest positive solution of $f_k(M,\gamma(M)) = 1$.

If k is even, the plot changes to that shown in Figure 9.

Now the condition on γ is that it must be less than β, where $-\beta$ is the value at which the graph of $\sigma(\rho)$ intersects the line $\sigma = \rho$ in the lower left quadrant. This is equivalent to

$$\gamma^k[(M+1)\gamma + M] \leq \gamma$$

or, if we define $f_k(M,\gamma)$ for even k as

$$f_k(M,\gamma) = \gamma^{k-1}[(M+1)\gamma + M$$

and by eq. (8) for odd k, the condition for each $k \geq 1$ is

$$f_k(M,\gamma) \leq 1$$

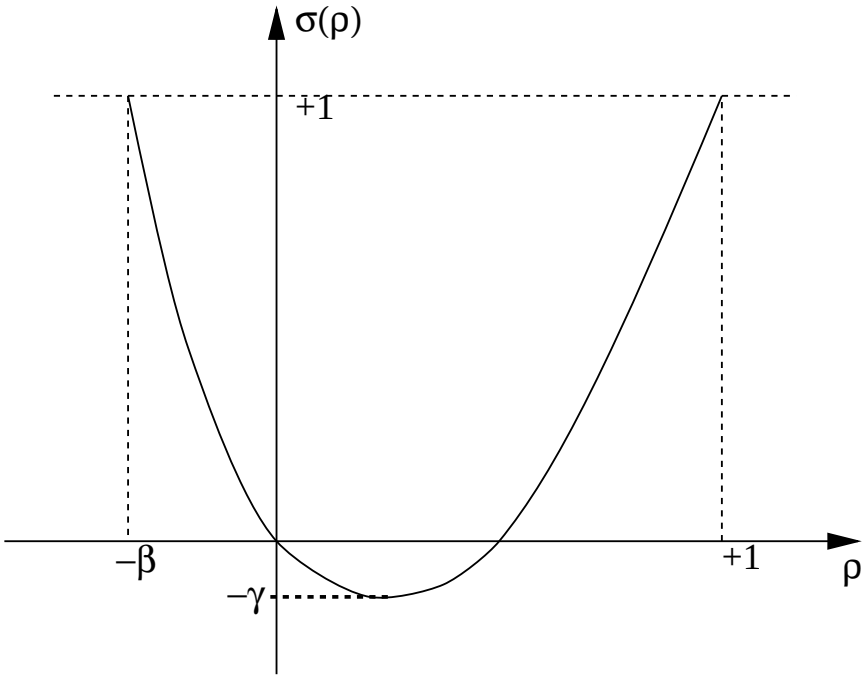

Figure 8: Real σ-ρ map, $k = 1, q = 1$.

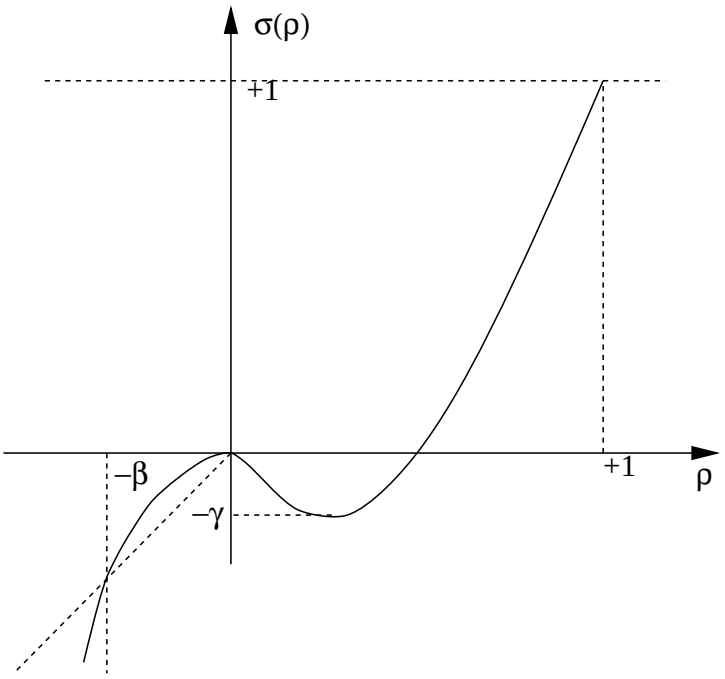

Figure 9: Real σ-ρ map, k even, $q = 1$

3.1.2 Case $q = 2$, second order

When $q = 2$ the extrapolation formula leads to

$$\sigma(\rho) = \rho^k \left[\frac{(M+1)(M+2)}{2} \rho^2 - M(M+2)\rho + \frac{M(M+1)}{2} \right]$$

If $k = 1$ the σ-ρ graph is as shown in Figure 10.

Because its slope at $\rho = 0$ exceeds $+1$, no part of the negative real axis can be stable. However, as long as σ does not lie outside of [0,1] for $\rho \in [0, 1]$ then the method has a [0,1] stability region. By simple algebra we note that

$$\sigma(\rho) = \left[\frac{(M+1)M+2}{2} \right] (\rho - \rho_1)(\rho - \rho_2)\rho$$

where

$$\rho_i = \frac{M(M+2) \pm \sqrt{-M(M+2)}}{(M+1)(M+2)}$$

Thus the only real zero of σ is at $\rho = 0$ so σ is everywhere positive for $\rho > 0$. Hence the method has a [0,1] stability region as long as the local maximum does not exceed one. The maximum occurs at

$$\rho_0 = \frac{2M(M+2) - \sqrt{(M-1)M(M+2)(M+3)}}{3M(M+2)}$$

for $M > 1$ and the largest M for which we have a [0,1] stability region is the smallest value of M for which $\sigma(\rho_0) = 1$.

If k is even, the graph takes the form in Figure 11.

In this case the method has a [0,1] stability region if the local maximum does not exceed 1 and then the stability region includes the segment $[-\beta, 1]$ of the real axis.

If $k > 1$ and is odd, the graph takes the form shown in Figure 12. While this is similar to Figure 10, the high order contact at the origin means that there is the section of the negative real axis $[-\beta,0]$ that maps into itself where $-\beta$ is the intersect of the line $\sigma = \rho$ and the graph. Again, for M less than the critical value at which the local maximum is $+1$ the method has a [0,1] stability region.

3.1.3 Case $q > 2$

The earlier figures illustrate the general features for all q. If $k + q$ is even, σ is positive when ρ is negative, as in Figures 8 and 11. However, there may be more maxima and minima than shown, so the conditions are that no maxima exceed $+1$ and no minima be less than $-\beta$. If $k + q$ is odd, σ is negative when ρ is negative, as in Figures 9, 10, and 12. In this case, the conditions are that no maxima exceed $+1$ and that no minima be less than $-\beta$. (When k is one and q is even, we have the case in Figure 10 and β is effectively zero.)

In the general case we have

$$\sigma(\rho) = \rho^k \frac{M(M+1)\cdots(M+q)}{q!} \int_0^1 t^{M-1}(\rho - t)^q dt \tag{9}$$

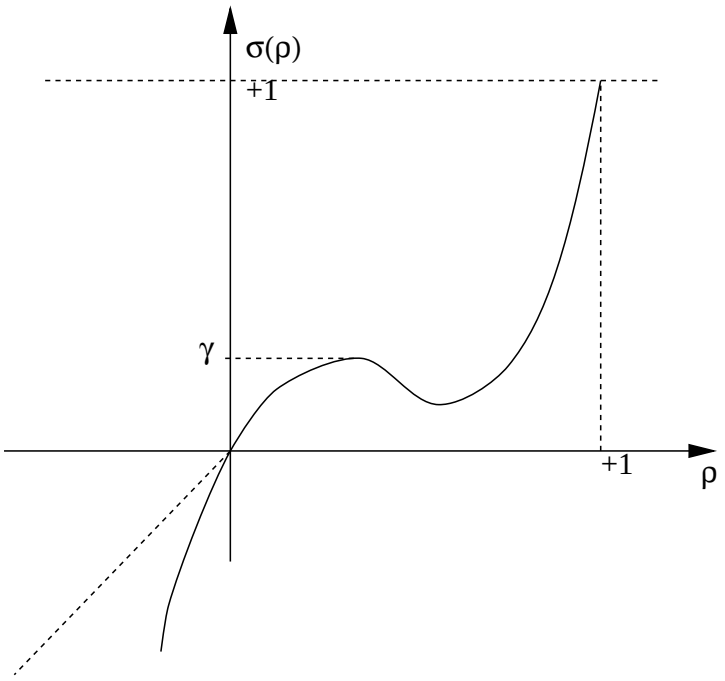

Figure 10: Real σ-ρ map, $k = 1, q = 2$

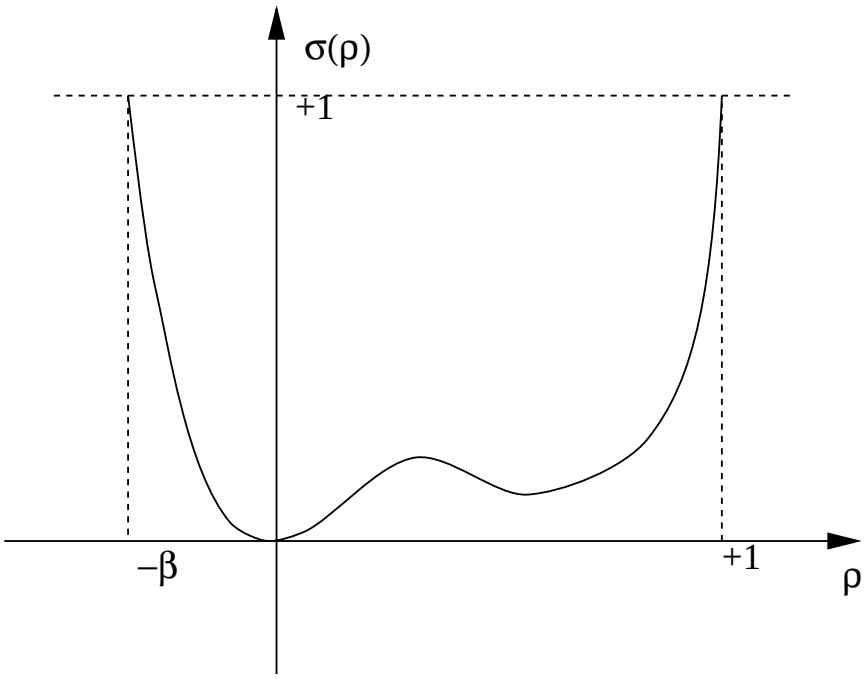

Figure 11: Real σ-ρ map, k even, $q = 2$

for the q-th order extrapolation. (This can be derived by tedious algebra or verified by expansion as a power series in ρ.) It uses the fact that the q-th order extrapolant is simply the q-th order polynomial through the last $q+1$ points k, $k+1, \cdots, k+q$, and then $\sigma(\rho)$ is obtained by substituting ρ^{k+i} for the $k+i$ data value.

Since $\sigma(\rho)$ (given by eq. (9)) is a polynomial in ρ whose coefficients are continuous (and differentiable) functions of M, we only need to verify that the method is [0,1] stable for one value of M. Then we know that it is [0,1] stable for all values of M up to the first that violates the criteria discussed above. When $M = 0$ we have $\sigma(\rho) = \rho^{k+q}$ and it is trivially true, hence we have [0,1] stability for some non-negative M.

From the expression in eq. (9) we see that σ is positive for all positive ρ when q is even. Hence for these cases we need only look at local maxima in the interval [0,1] and ensure that they do not exceed 1. When q is odd, we must compute both the positive local maxima and the negative local minima and ensure that (i) the largest maxima does not exceed 1, and (ii) the smallest local minima, $-\gamma$, is such that the map $\sigma(-\gamma)$ is less than one if $k+q$ is even, or is less than γ is $k+q$ is odd. These criteria were used in an iteration to compute the $M_{k,q}$ shown in Table 1. (In fact, for the k and q considered, there was only one local minimum and no local maximum when q was odd, and only one local maximum when q was even.)

4 Spectra with multiple gaps

We are interested in problems whose spectrum is clustered into a number of reasonably well separated regions as shown in Figure 3. We assume that all eigenvalues can be *covered* by a set of $N+1$ disjoint[4] disks, each disk corresponding to a cluster. We describe the region by disks because it provides the simplest description (fewest parameters) and also because many parts of the stability regions of the methods we discuss are approximately circular. The eigenvalues fall into a set of *fast* clusters which lie in the disks $\mathrm{S}_i, i = 0, \cdots, N-1$ and the active cluster, which lies in S_A.

In this section we will refer to the *stability set* as the set of points in the ρ-plane in which the method is stable, while *stability region* will refer to a connected region in the stability set. In this section we will see how the TP method can be used for problems with multiple gaps, providing a stability region for each cluster.

In the previous section we showed that by making M small enough it was possible to get a stability set that included the real axis between zero and one in the ρ-plane. If we make M larger and use the TP method, the stability set, as defined in the previous section consists of infinitely many disjoint regions. After any finite number of iterations, it will consist of a finite number of regions. In this section we will plot the stability regions in the $(\rho - 1)$-plane which is the $h\lambda$-plane if the inner integrator is Forward Euler. We will do this because we will be scaling the plots about the origin to look at different regions in more detail.

If, for example, we use two levels of the P2-1-9 method, we get the stability set shown in Figure 13. Note that there are five disjoint regions of stability: one attached to the origin for the active components, two straddling the negative real axis, and two more off the real axis. We call the first three regions *principal* stability regions. We will see that we can choose method parameters to place these where we like on the negative real axis. The other two stability regions are artifacts of the value of k - if k were larger and M

[4] while they need not be disjoint, the overlapping disks case is partially covered in the previous section.

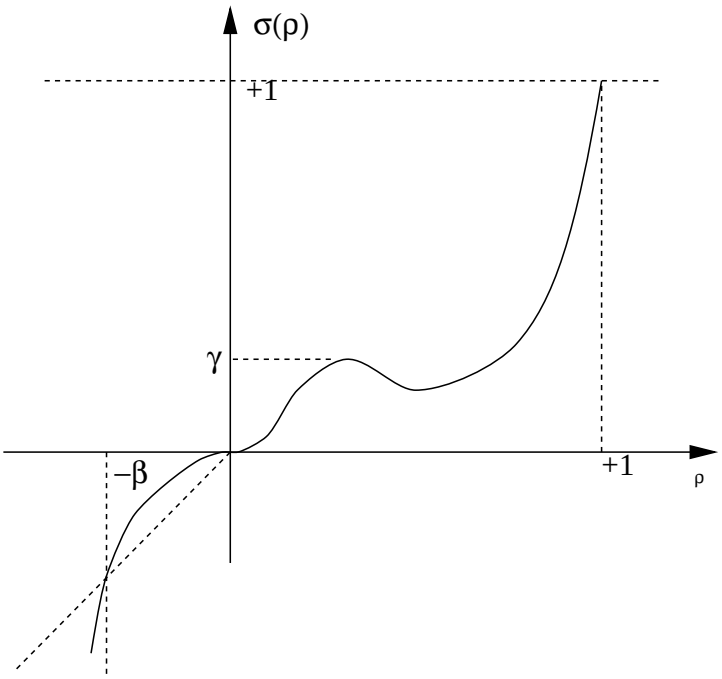

Figure 12: Real σ-ρ map, $k \geq 3$ and odd, $q = 2$

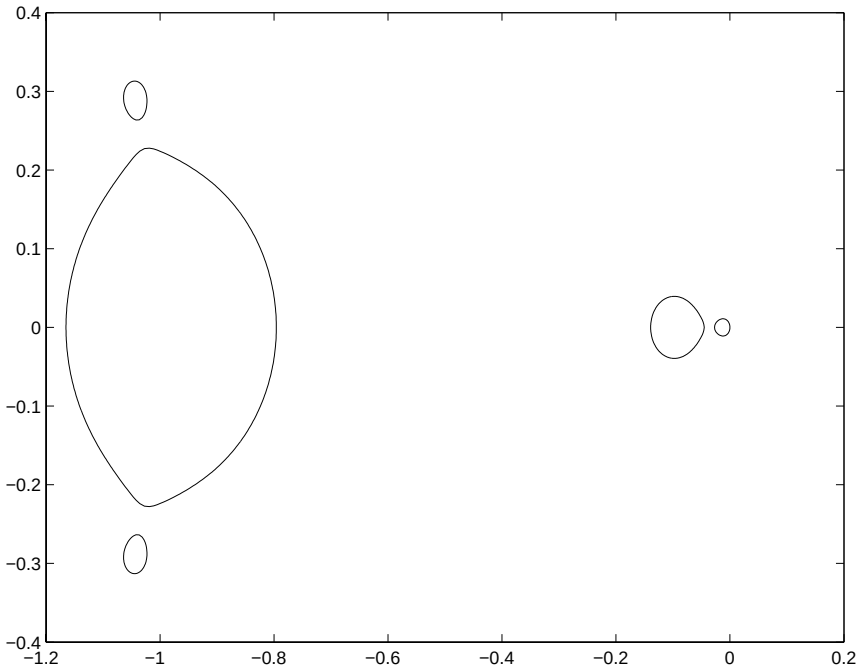

Figure 13: Stability set of 2nd-level outer P2-1-9 method in $h_0\lambda = \rho - 1$ plane

were large enough for the stability set to consist of more than one region there would still be three principal stability regions but there would be k artifacts. (In the case of odd k one of the artifacts will straddle the real axis, but, as with the other artifacts, its position cannot be controlled independently, so we will not use it.) At the next level of iteration of the TP method we would get four principal regions straddling the real axis and $3k+k^2$ artifacts. The m-level TP method has $m+1$ principal stability regions which can be placed to cover the clusters of eigenvalues by choice of (possibly) different k_i, q_i, and M_i at each projective level. We will name the principal stability regions of the TP method $\mathbf{R}_0$, $\mathbf{R}_1$, $\cdots$, $\mathbf{R}_{m-1}$, and $\mathbf{R}_A$. Each is intended to deal with the corresponding cluster $\mathbf{S}_i$ in Figure 3. Note that these regions are somewhat close to being circular. As M increases, they will asymptotically become circular (but spread much further apart so be difficult to see on a single plot). Note also that the principal stability regions in Figure 13 are somewhat close to being centered at -1, -1/10, and -1/120 respectively. The leftmost region always surrounds -1 (which is 0 in the ρ-plane). The other two values are $-h_0/[h_{i-1}(M_i+1)]$ for $i=1,2$ respectively. (The reasons for these values can be seen in the later analysis.) Figures 14, 15, and 16 show the regions $\mathbf{R}_0$, $\mathbf{R}_1$, and $\mathbf{R}_A$ from Figure 13 scaled by 1, 10, and 120 respectively.

In the clusters shown in Figure 3 the disk S_i is centered at $-\lambda_i$ with $\lambda_0 > \lambda_1 > \cdots > \lambda_{N-1} > 0$. For $i = 0, \cdots, N-1$ the disks have radius $\lambda_i r_i$ with $r_i \leq r < 1$, that is, r_i is their *relative radius*. The disk S_A is centered at the origin and contains the eigenvalues corresponding to the active components in the system (i.e., the components still of a size to be of interest in the computation). It may contain some eigenvalues with real parts slightly into the positive half plane. Such eigenvalues do not cause computational problems, but to avoid making exceptions for these cases in the discussion below, we will assume that all eigenvalues are strictly in the negative half plane or at the origin.

In the early stages of an integration, fast components may still be active. Only after some initial transients have decayed will Figure 3 be relevant. While all components are active, there will only be the one cluster, S_A, and N will effectively be zero as far as the possibility of any projective integration. At this stage we should be using an integration method of appropriately high order. (It can be explicit because there are no inactive, fast components.) After the fastest components have become inactive, cluster S_0 will be identifiable and N will effectively be 1. We can now begin projective integration. At this time, there is probably no reason to use other than a first-order integrator as the inner integrator since its step size is now small compared to time constants of the active components. The order of the outer integrator should be chosen commensurate with the accuracy requirements of the problem. As time passes, other components will decay and other clusters will become identifiable. Thus the usable N will slowly increase, permitting iteration of the projective step and a larger outermost step. It is unlikely that there is any reason to use other than first order methods in all but the outermost integrator.

At any point in the integration we will have an effective N and a collection of N different step parameters, $k_i, q_i, M_i, i = 0, \cdots, N$ typically with $q_i = 1$ for $i < N$. The i-th level step size, h_i must be commensurate with the time constant on the i-th cluster for $i = 0, \cdots, N-1$, while h_N must be commensurate with the accuracy needs of the fastest active component. h_0 is the actual step size of the inner integrator. For the outer levels of integration, "step size" refers to the combination of $k+q$ steps of the lower-level integrator plus the M-fold projective step of the lower level, that is,

$$h_{i+1} = (k_i + q_i + M_i)h_i$$

We will assume in the analysis below that PFE is used for the all outer integrators except

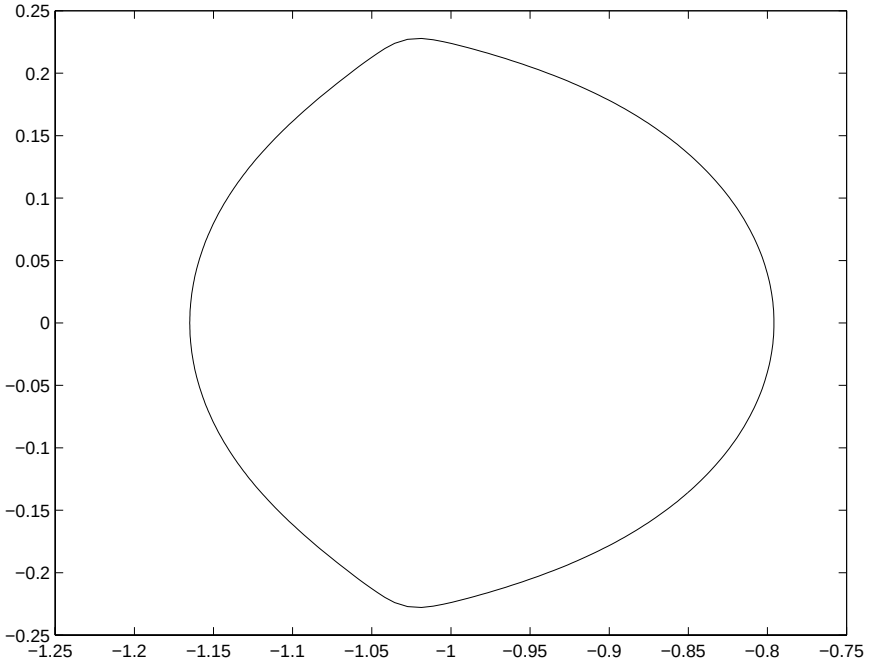

Figure 14: 2nd-level outer P2-1-9 method, R_0 region in $h_0\lambda$-plane

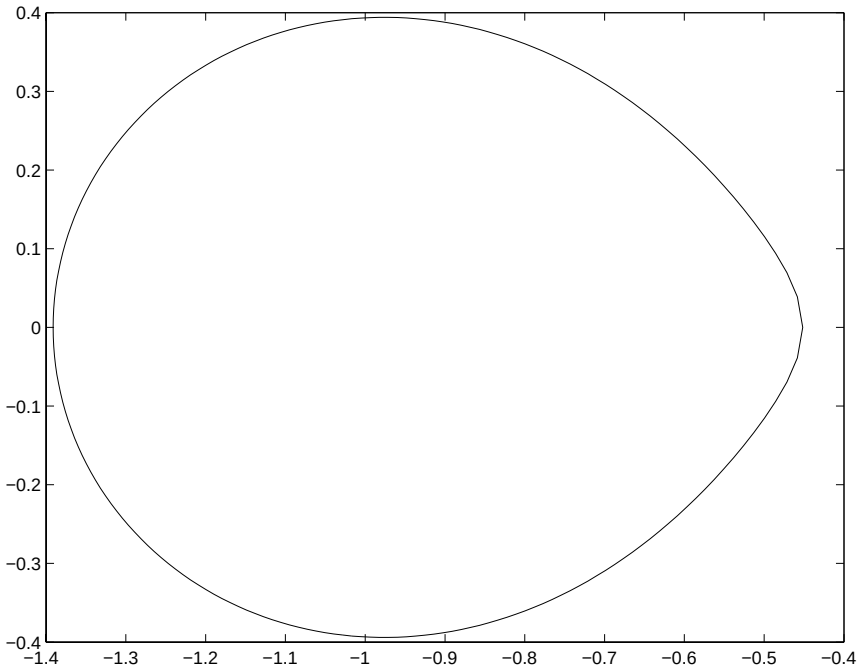

Figure 15: 2nd-level outer P2-1-9 method, R_1 region in $10h_0\lambda$-plane

possibly the outermost. (If we are operating in the context of legacy software, we may not have a choice of integration methods for the innermost integrator.)

In [1] we showed that, for asymptotically large outer step sizes and for a polynomial extrapolation outer integrator, the stability regions of a single projective level consisted of a disk which should contain all of the fast eigenvalues and the stability region of the outer integrator. Here we will show that a multiple level method can achieve the extension of those features: If q_i and M_i have been chosen appropriately to the problem (accuracy needed in each cluster and the speed of each cluster) then there exists a set of k_i such that the method is stable for those clusters.

We will assume that Forward Euler method is used for the innermost integrator. The analysis can be adjusted for other integrators since the analysis depends on (i) the inner integrator damping the fastest components by a multiple of at least r at each step, and (ii) the inner integrator being sufficiently accurate for the other components.

We will show that there is a k such that if $k_i \geq k$ then the method is stable. The important feature is that k is independent of the number of levels, N, of outer integration, so can be chosen once and for all. k does depend on r and M. In a practical code these values may be determined dynamically as the code proceeds so k may have to be adjusted. This result guarantees that k will not become arbitrarily large as N increases.

The proof of the result is outlined below. Because of the number of parameters that would be involved if each cluster were treated with different parameters, we will analyze the case $q_i = 1, r_i \leq r < 1$ and $M_i \leq M$ and derive results for the maximum values r and M. These lead to the largest value for k. It will be obvious how smaller relative radii, r_i, and projective step lengths, M_i, could result in a smaller k_i in particular cases.

4.1 Proof Outline

The proof proceeds by considering eigenvalues in each cluster in turn and showing that a component with eigenvalue ξ_i in cluster $\mathbf{S}_i$ is integrated accurately by the lower-level integrators ($m < i$) so that the amplification of the component over an interval of size L is an accurate representation of $\exp(\xi_i L)$ and less than a value near one in magnitude for eigenvalues in the negative half plane, while it is damped by a factor of at least r by integrators of the same and higher levels. (An exception is the active cluster for which the top level integrator is also expected to be accurate rather than damping.)

When Forward Euler is used for the inner integrator and first order projective steps are used in the all outer integration steps, the step sizes are chosen as follows:

$$h_0 = 1/\lambda_0$$

$$h_i = 1/\lambda_i + k_{i-1}h_{i-1}, i < N$$

The important criterion is that each of these makes the level i integrator maximally damping at the center of disk in $\mathbf{S}_i$ in Figure 3. For Forward Euler this leads to the smallest k to ensure sufficient damping. A different value might be more efficient for higher-order integrators. Note that the definition of h_i could also be written as

$$(M_{i-1} + 1)h_{i-1} = 1/\lambda_i \tag{10}$$

that is, the projective step itself corresponds to the time constant of the center of the i-th cluster.

In drawing a figure like Figure 3 it is tempting to think of clusters that are distributed roughly uniformly along the negative axis. However, the multiplier, M_i, for the projective step is approximately the ratio of successive λ_i. In order for there to be a reasonable speedup, this ratio should be significantly different from 1. Thus we expect that

$$\lambda_{i-1}/\lambda_i \geq d > 1 \tag{11}$$

for which M is about d or larger. This means that the distribution is approximately *logarithmic* and we expect d to be significantly larger than 1.

We also need a reasonable gap between the clusters in order for there to be much speedup. This means that

$$\frac{\lambda_i(1+r_i)}{\lambda_{i-1}(1-r_{i-1})} \leq 1-c$$

where $c > 0$ is a measure of the relative gap size. In the analysis below it will be convenient to assume instead that

$$r \leq b < 1$$

and

$$(1+r)/d \leq \epsilon << 1 \tag{12}$$

(where ϵ is not necessarily very small, but smaller is better since the analysis is asymptotic in ϵ).

We indicated above that we would consider the effect of the various integrator levels on an eigenvalue in each cluster. Let $\mathbf{e}$ be any complex number of magnitude no greater than one. Then any point in the disk S_i can be written as

$$\xi_i = -\lambda_i(1+r\mathbf{e})$$

and we will use the notation ξ_i to stand for an eigenvalue in cluster $\mathbf{S}_i$.

It is instructive to first consider $\lambda = \xi_0$ in S_0. The amplification of the inner integrator for ξ_0 is

$$\sigma_1(\xi_0) = (1+h_0\xi_0)^{k_0}(1+(M_0+1)h_0\xi_0)) \tag{13}$$

Since $h_0 = 1/\lambda_0$, eq. (13) then implies that

$$|\sigma_1(\xi_0)| \leq r^{k_0}|(1-(M_0+1)(1+\mathbf{e}r))| \leq r^k|(1-(M+1)(1+\mathbf{e}r))| \tag{14}$$

The second term in eq. (14) is bounded by $(M+1)(1+r)$. In [1] we chose k to make $|\sigma_1(\xi_0)|$ in eq. (14) no greater than one. This can be done by requiring that

$$r^k < \frac{1}{(M+1)(1+r)} \tag{15}$$

However, we are going to project the first-level result using the second-level projective step. It will result in an amplification of

$$\sigma_2(\xi_0) = {\sigma_1}^{k_1}(\xi_0)[1+(M_1+1)(\sigma_1(\xi_0)-1)]$$

For $|\sigma_2(\xi_0)|$ to be less than one we need $|\sigma_1(\xi_0)|$ to be less than one and k_1 to be sufficiently large. If we chose k in eq. (14) such that $|\sigma_1(\xi_0)| \leq r$ (which requires that k be only one

larger required by eq. (15)) then, if we set $k_1 = k$ we find that $|\sigma_2(\xi_0)|$ is also less than r.

Since the stability of the i-th level is determined by

$$\sigma_{i+1}(\xi_0) = {\sigma_i}^{k_i}(\xi_0)[1 + (M_i + 1)(\sigma_i(\xi_0) - 1)] \tag{16}$$

the same reasoning shows that all $|\sigma_i(\xi_0)|$ are less than r if $k_i \geq k$.

We now need to consider λ in other disks. Suppose λ is in S_i (that is, $\lambda = \xi_i$) with $i > 0$.

Let us first consider the j-th-level integrators for $j < i$. We have

$$|\xi_i|/\lambda_j \leq (1 + r)/d \leq \epsilon$$

from assumption (12). Since we are taking $h_i = \mathrm{O}(1/\lambda_i)$ we also have

$$h_j|\xi_i| = \mathrm{O}(\epsilon)$$

The inner integrator has an amplification factor of

$$\sigma_0(\xi_i) = (1 + h_0\xi_i) = \exp(h_0\xi_i) + \mathrm{O}(\epsilon^2)$$

The first level outer integrator will have amplification factor

$$\sigma_1(\xi_i) = (1 + h_0\xi_i)^{k_0}[1 + (M_0 + 1)(\sigma_0 - 1)]$$

$$= \exp(k_0 h_0 \xi_i)[1 + (M_0 + 1)h_0\xi_i] + \mathrm{O}(\epsilon^2)$$

$$= \exp((k_0 + M_0 + 1)h_0\xi_i) + \mathrm{O}(\epsilon^2)$$

$$= \exp(h_1\xi_i) + \mathrm{O}(\epsilon^2)$$

since $h_i = (k_{i-1} + 1 + M_{i-1})h_{i-1}$. Proceeding in this way, we find that

$$\sigma_j(\xi_i) = \exp(h_j\xi_i) + \mathrm{O}(\epsilon^2)$$

When we come to the i-th level outer integrator we get the error amplification

$$\sigma_i(\xi_i) = \sigma_{i-1}^{k_{i-1}}[1 + (M_{i-1} + 1)(\sigma_{i-1} - 1)]$$

$$= \exp(k_{i-1}h_{i-1}\xi_i)[1 + (M_{i-1} + 1)h_{i-1}\xi_i] + \mathrm{O}(\epsilon^2) \tag{17}$$

Since ξ_i is in the left half plane the magnitude of the first term is bounded by one. (Since we only have an order ϵ^2 approximation to the exponential, the bound could be slightly larger than one, but this can be handled with greater care and detail in the proof.) From the definition of h in eq. (10) the second term can is bounded by r. Hence we have

$$|\sigma_i(\xi_i)| \leq r$$

Now we can guarantee that $|\sigma_j(\xi_i)| \leq r$ for $j > i$ using exactly the same argument as we used for $i = 0$. Incidentally, it is eq. (17) that tells us the amplification is zero when

$$\xi_i \approx \frac{-1}{h_{i-1}(M_i + 1)}$$

so that the i-th stability region is centered near that location.

For eigenvalues in cluster $\mathbf{S}_A$ the analysis proceeds in the same manner until the outermost step. In that step the step size h_N has been chosen to make the outermost step accurate for eigenvalues in $\mathbf{S}_A$, so the last step yields

$$\sigma_N(\xi_A) = \exp(h_N \xi_A) + \mathrm{O}(\epsilon^2)$$

Thus we conclude that it is possible to select h_i and k_i such that the method is stable for each cluster, at least in a linear analysis.

The above is an asymptotic argument. For any given situation we can plot the stability regions.

4.2 A Related Method

It is instructive to consider a related method: that of using a selection of step sizes chosen to achieve the desired stability regions. This is the process that would result if, instead of using polynomial extrapolation through past points to project forward, we used a one-step method that evaluated the derivatives as needed. For example, we could take any 1-step method such as Forward Euler or Runge-Kutta and apply it as follows:

1. Integrate for k_0 steps of size h_0
2. Integrate for one step of size h_1
3. Repeat steps 1 to 2 k_1 times
4. Integrate for one step of size h_2
5. Repeat steps 1 to 4 k_2 times
6. Integrate for one step of size h_3
7. $\cdots$

This is related to methods based on using a Chebyshev polynomial to create a stability region. In these methods, a set of step sizes $h_i = -1/\lambda_i$ are used, and the stability region is related to the region in which the polynomial

$$\prod_{i=0}^{i+N} (\lambda - \lambda_i)$$

is less than one in magnitude. The difference between this method and the TP method is that a regular integration step is used for the outer steps rather than a projective step. This difference leads to poorer stability properties so a larger number of smaller steps are needed than in the TP method.

We will consider the Forward Euler (FE) method as the integrator. We start with the inner integrator performing FE on step size h_0. Let us suppose that at the i-th level ($i = 0, \cdots, N-1$) we do k_{i-1} steps of the $i-1$-st level integrator and then "junp forward" using one Forward Euler step of length $h_i = (M_{i-1} + 1)h_{i-1}$, where M_i is the value used in the method discussed above. (This definition of h_i differs from the earlier, but if the

size of the k_i's were the same, the amount of computation would be similar.) The error amplification of one step of the first outer level for the linear test equation $y' = \lambda y$ is

$$\rho_1(\lambda) = (1 + h_0\lambda)^{k_0}(1 + h_1\lambda) \tag{18}$$

(We have used ρ here to designate the amplification to distinguish it from σ in the regular projective method.) In fact, this is exactly the same as for the first-level projective method so we do not need to analyze it further since $h_1 = (M_0 + 1)h_0$. The stability formula for the i-th level of projective integration is

$$\rho_i(\lambda) = ((\cdots((1 + h_0\lambda)^{k_0}(1 + h_1\lambda))^{k_1} \cdots (1 + h_{i-1}\lambda))^{k_{i-1}})(1 + h_i\lambda) \tag{19}$$

or

$$\rho_i(\lambda) = \rho_{i-1}(\lambda)^{k_{i-1}}(1 + h_i\lambda) \tag{20}$$

This should be compared with eq. (16). The difference is that the $(1 + (M_{i-1} + 1)(\sigma_{i-1} - 1))$ has been replaced by $(1 + h_i\lambda)$ as the second term in the product. Because the size of $|\sigma_{i-1}|$ was controlled to be no more than r and M_{i-1} is bounded by M, the form in eq. (16) can be bounded using a k_{i-1} that is independent of i. On the other hand, the form in eq. (20) contains a term $h_i\lambda$ which is given by

$$h_i\lambda = h_0\lambda \prod_{j=0}^{i-1}(M_j + 1)$$

When λ is in $\mathbf{S}_0$ this is of order M^i so that k_i will have to grow approximately linearly with i. This compares with a k that is independent of i in the TP method proposed here.

5 Test Examples

The following examples are in the nature of a "Proof of Principle." They are not comparisons of the proposed methods with other methods, but demonstrations that the methods do have the claimed properties. Since these methods have not yet been developed into good codes (which involves all of the issues of step and order selection, plus a number of other issues for these methods) comparisons could only be made with simple versions of other methods (such as a fixed step size Runge-Kutta method). But nobody uses such simple codes anymore, so it is a meaningless comparison. Furthermore, there are already too many papers that compare the author's new method with a hastily coded standard method for us to add to the list!

Below we give a very simple example of a double gap problem. Since it is linear, and the theory above applies to linear problems, it should be no surprise that it works. It does, however, illustrate very well the nature of the method. Then we give two examples of problems with a spread of eigenvalues along the real axis, one a direct version of the heat equation, and one a non-linear modification. All calculations are done in MATLAB on a PC.

5.1 Example 1

Consider the equation for the 2-dimensional vector y:

$$y' = -\begin{bmatrix} 5050 & 4950 \\ 4950 & 5050 \end{bmatrix}\left(y - \begin{bmatrix} \sin(t/10) \\ \cos(t/10) \end{bmatrix}\right) + 0.1\begin{bmatrix} \cos(t/10) \\ -\sin(t/10) \end{bmatrix}$$

with initial values such that the solution is

$$y = \begin{bmatrix} \sin(t/10) \\ \cos(t/10) \end{bmatrix}$$

Its eigenvalues are -10,000 and -100, so it is stiff. This was integrated using two levels of the projective integrator. The inner step h_0 was 0.0001. First order methods were used at all levels. The first-level projective step was done with $k = 1$ and $M = 99$ - which corresponds to $h_1 = 0.0101$ and $(M + k)h_0 = 0.01$ to handle the second eigenvalue. The second level projective step was done with $k = 1$ and $M = 75.7622$. This was chosen so that $h_2 = \pi/4$ and 20 steps of the second level integrator covered a quarter revolution of $\sin(t/10)$. Table 5 shows the errors after 10, 20, and 30 steps of second-level integrations, followed in each case by the results of the next 7 calculated points. These consist of two points from the inner integrator ($k + q = 2$), one point from the first-level outer integrator, two more points from the inner integrator, one from the first-level integrator, and one from the second-level outer integrator - in other words, one complete second-level integration step. Two errors are shown: the tangential error and the radial error (with reference to the circular orbit of the true solution). The first column is the time value converted to degrees of rotation of the true solution. The second and third columns are the tangential and radial errors.

Since the correct solution is a unit vector rotating with period 20π, and since a first order method is used, the local error is largely in the radial direction at each step. At 45 degrees, the radial error is about 20 times the tangential error. We note an error of about 0.003 in the outermost projective step. This is almost exactly the truncation error of a first-order method, which is approximately $h^2 d^2y/dt^2/2 = (\pi/4)^2/200 = 0.031$. Now it is interesting to observe what happens in subsequent steps. At 45 degrees, the next inner step annihilates the radial error. This is because the eigenvector corresponding to the eigenvalue 10,000 is [1,1]. This is the radial direction at 45 degrees, so errors in that direction are annihilated in one inner step. That step doesn't do anything, however, for the tangential error which is not reduced until the next first-level projective step. In contrast, at 135 degrees, the radial error is in the direction of the eigenvector correspond to the eigenvalue of -100. The error is not changed very much in the next two inner steps, but it is annihilated by the first level outer step. At 90 degrees, the error is equally composed of both eigenvectors, so half of it is eliminated in the subsequent inner steps, and the remainder is eliminated in the first first-level outer step.

This example points out where the method could have difficulties: if the eigenspaces were changing with time (due to non-linearities or time-dependent Jacobians) errors introduced in one eigendirection at one step may be in another eigendirection by the time we take the projective step whose size is intended to damp that direction. Consequently, the rate of change of eigenvectors must be limited over the length of one outer projective step.

It is natural to ask about the behavior of the error as a function of step size. It is not going to exhibit the simple behavior of conventional methods for two reasons. First,

Table 5: Results for Example 1

degrees	tangential error	radial error
45.000000	1.52003e-004	2.96239e-003
45.000573	1.50706e-004	9.89910e-009
45.001146	1.49199e-004	8.19619e-009
45.057869	3.28302e-010	4.95050e-007
45.058442	1.69956e-010	5.01732e-011
45.059015	1.68306e-010	5.01734e-011
45.115737	3.23333e-010	4.95050e-007
49.500000	1.52003e-004	2.96239e-003
...	...	...
90.000000	1.52001e-004	2.96240e-003
90.000573	1.39005e-003	1.39201e-003
90.001146	1.37614e-003	1.37809e-003
90.057869	2.14416e-009	4.97527e-007
90.058442	2.47332e-007	2.47887e-007
90.059015	2.44881e-007	2.45436e-007
90.115737	2.14415e-009	4.97532e-007
94.500000	1.52001e-004	2.96240e-003
...	...	...
135.000000	1.52003e-004	2.96240e-003
135.000573	2.92749e-008	2.93276e-003
135.001146	5.79873e-008	2.90344e-003
135.057869	3.33214e-010	5.00000e-007
135.058442	5.04972e-010	4.95050e-007
135.059015	5.04833e-010	4.90149e-007
135.115737	3.38279e-010	5.00000e-007
139.500000	1.52004e-004	2.96240e-003

Table 6: Results for modified Example 1 at second order

# outer steps	tangential error	radial error
8	6.48423e-002	4.64337e-002
16	9.60296e-003	2.96321e-003
32	1.24296e-003	1.84880e-004
64	1.56327e-004	1.15698e-005
128	1.94525e-005	7.17206e-007
256	2.40220e-006	4.75877e-008

we cannot change the inner step sizes in an example like the above - they are set by the eigenvalue clusters. Second, the errors after an outer projective step are much larger that errors after inner steps, so that errors after outer steps tend to behave like h^{p+1} for order p methods. This is particularly true if the problem is moderately stiff because errors introduced in the stiff components by the outermost projective step are subsequently damped by the inner steps. Table 6 shows the results for a similar example using a second-order outer method. The driving period was $2\pi/100$ rather than $2\pi/10$ to allow for much longer outermost steps. The problem was run using $k = 1$ for both projective levels, $h_0 = 10^{-4}, M_1 = 99$, and M_2 varied so that 8, 16, 32, 64, 128, or 256 outer steps were used to integrate one full cycle (to $t = 200\pi$). (Note that M_2 is approximately 8000 in the first case.) We see that the errors are behaving more like h^3 in the tangential direction (which has the larger errors in the case of a second order outer integrator).

5.2 Example 2

The heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + g(x,t)$$

was solved on $x \in [0,1]$ and $t \geq 0$ with Dirichlet boundary conditions. Initial and boundary values and g are chosen to be consistent with the solution $u = sin((x+t/100)\pi)$. The space interval Δx was 1/100 so there were 99 internal points and unknowns. The problem was integrated using an inner step size of $h_0 = 0.000025$ which is consistent with the stability limit of an explicit Euler integrator. (The most negative eigenvalue is about -39,998.) Up to eight levels of projective integration were used. All levels used $k = 1, q = 1, M = 2$ so that the outermost step size was $h_8 = 1.6384$ at level 8.

Table 7 shows the results from integrating over the interval [0,6.5536] using from three to eight levels of projective steps. The error shown is the L_2 error. (In all cases in these tests, the L_1 error was roughly nine times larger, while the L_∞ error was seven times smaller.) The ratio of step sizes at adjacent projective levels is four, so one should expect to see a factor of four reduction if we were in the asymptotic region. This appears to be the case until we get to six projective levels and above in this example.

However, it is not the number of projective levels themselves that cause the problem, as can be seen in Table 8 which shows the result of running the same problem with the

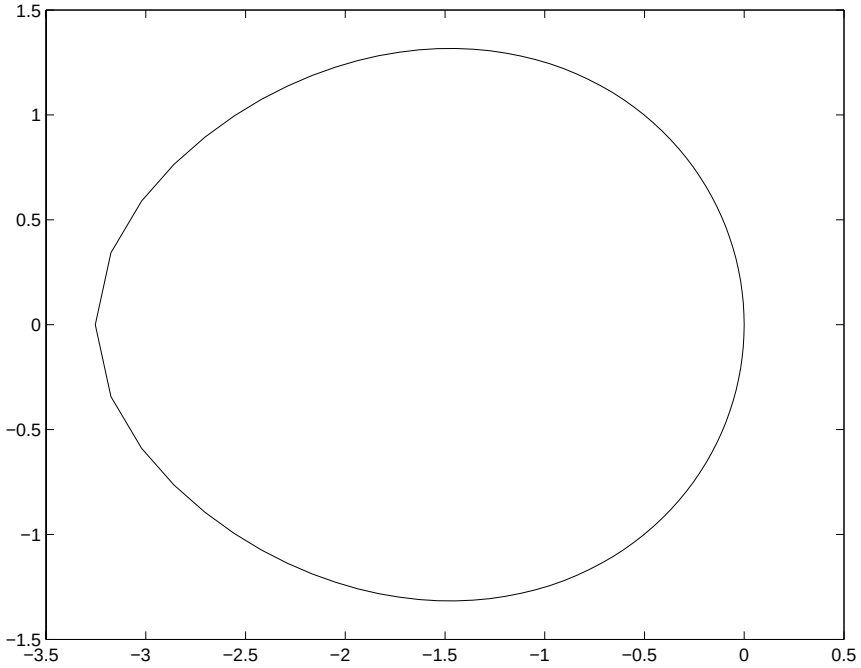

Figure 16: 2nd-level outer P2-1-9 method, R_A region in $120h_0\lambda = 10h_1\lambda$-plane

Table 7: Results for Example 2 with various Projective levels

Projective Level	L2 Error
8	1.1252e-002
7	2.5722e-004
6	2.3622e-005
5	4.7326e-006
4	1.1311e-006
3	2.8257e-007

Table 8: Results for Example 2 with various inner step sizes and 8 projective levels

Inner Step	L2 Error
2.5000e-005	1.1252e-002
1.2500e-005	1.7137e-003
6.2500e-006	2.5913e-004
3.1250e-006	6.1977e-005
1.5625e-006	2.4666e-005
7.8125e-007	1.0555e-005
3.9063e-007	4.9571e-006
1.9531e-007	2.3792e-006

inner step being reduced by two a number of times. Here we see similar behavior - only as we get to the smaller step sizes do we see the first order behavior. Close inspection of the intermediate results indicates that the error immediately after a projective step are larger than other steps (as would be expected and was seen in the the previous example). Since the problem is mildly damping, these large errors are damped in subsequent small steps. Indeed, the whole structure of the method is based on using the smaller inner integrations to damp out perturbations caused by the large projective steps. Note that each projective step immediately follows a projective step at each lower level. Thus, in the example above, the steps immediately prior to the output at level eight consisted of projective steps of length 4, 16, 64, 256, 1024, 4096, 16384, and $65536h_0$ taken with no intervening inner steps. (One might think that it would be better to introduce additional inner steps, but, for very technical reasons, it appears to introduce potential instabilities.) Therefore, it would be much better to take output before a projective step. Table 9 shows the output from the same problem, but taken immediately prior to the start of the sequence of outermost projective steps. At this point the method has applied the maximum damping using a sequence of k projective steps from the next to largest to the smallest. Note that the table also shows the t output point, and it changes slightly. When the maximum projective level is reduced, the same output points are not directly available (in a code we would use interpolation to get the desired output, but doing that will muddy our view of the errors). However, we would not expect the errors to change significantly over a small range of t. The table indicates that the error is very close to first order noting that each projective level is a factor of four bigger that the previous. (The eighth projective level is not shown because its output point is very different.)

5.3 Example 3

The third example is a mildly non-linear version of the second. The equation is

$$\frac{\partial u}{\partial t} = (1 + u^2)\frac{\partial^2}{\partial x^2} + g(x, t)$$

where the boundary conditions and g are chosen to make the solution the same as before. Because the largest eigenvalue can be about twice the value in the previous case, the

Table 9: Results for Example 2, output before projective steps

Final t	Projective level	L2 Error
5.05175	7	4.2685e-005
4.94935	6	1.6587e-005
4.89815	5	4.4032e-006
4.88535	4	1.1187e-006
4.88375	3	2.8384e-007

inner step size is $h_0 = 0.0000125$ - half the value previously. All other parameters are the same. The results are similar to those above so the details are not listed. However, the final error was nearly three times larger than Example 2 when it was run at half step size. Some of this increase is accounted for by the larger derivatives.

6 Conclusion

We have shown how the projective method can be iterated to achieve either larger regions of absolute stability or to handle problems with multiple gaps. It is clear that a combination of these two objectives could be used so that there are fewer gaps and larger stability regions where needed. While the methods do not appear to be as efficient as Runge-Kutta methods designed to have extended stability regions, they are conceptually much simpler and it appears that it will be much easier to design an automatic code to adapt to the clusters of eigenvalues in a specific problem. We are working on a code to do that, but it is too early to report on specifics.

References

[1] Gear, C. W. and Kevrekidis, I. G, Projective Methods for Stiff Differential Equations: *problems with gaps in their eigenvalue spectrum*, NEC Research Institute Report 2001-029, submitted to SISC.

[2] Lebedev, V. I., Explicit Difference Schemes with Variable Time Steps for Solving Stiff Systems of Equations, in *Numerical Analysis and Its Applications*, Proc. WNAA '96, Bulgaria, ed. Vulkov, Waśniewski, and Yalamov, Springer, Berlin, 1997, pp274 - 283.

[3] Medovikov, A. L., High-Order Explicit Methods for Parabolic Equations, *BIT* **38**, #2, pp372 - 390, 1998.

[4] Verwer, J. G., Explicit Runge-Kutta Methods for Parabolic Partial Differential Equations, *Applied Numerical Mathematics* **22**, pp359-379, 1996.