Chapter 1

Preconditioning Newton's method

Given an equation $f(x) = 0$, we attempt to precondition Newton's method by finding a fixed point iteration of this equation, $g(x) = x$, then solving $g(x) - x = 0$ using Newton's method:

$$p_{n+1} = p_n - \frac{g(p_n) - p_n}{g'(p_n) - 1}. \tag{1.1}$$

This equates to finding a new function ($f_1(x)$) that shares at least one root with $f(x)$.

Ideal qualities of functions to be solved:

- $f'(x*) \neq 0$ (for accuracy purposes)
- both $f(x)$ and $f'(x)$ are easy to compute (for efficiency purposes)
- $f(x)$ should allow for a wide range of initial guesses (basin of attraction) - this should relate to the region of monotonicity surrounding the root
- $f''(x)$ should be small or zero (for faster convergence)

The iteration above can be rewritten as

$$p_{n+1} = \frac{p_n g'(p_n) - g(p_n)}{g'(p_n) - 1}.$$

The error behaves as

$$\|p^* - p_{n+1}\| = \left\| \sum_{k=2}^{\infty} \frac{(p^* - p_n)^k g^{(k)}(p_n)}{(1 - g'(p_n))k!} \right\|$$

which is the same as Newton's method using $f(x) = x - g(x)$. It can alternatively be written in terms of the unknown variable $\xi \in [x^*, x_n]$:

$$\|p^* - p_{n+1}\| = \frac{1}{2} \|p^* - p_n\|^2 \left\| \frac{g''(\xi)}{1 - g'(p_n)} \right\|.$$

Suppose we consider the preconditioned Newton's method as a fixed point iteration. Then we would like the iterate to satisfy the conditions of the fixed point iteration theorem. In particular, we would like the magnitude of the derivative of the iterate to be less than one near the root:

$$\left| g''(p) \frac{g(p) - p}{(g'(p) - 1)^2} \right| < 1.$$

Thus, we require that either $g''(p)$ or $g(p) - p$ is very small or $g'(p) - 1$ is very large.

Take as an example $g(x) = -\log(x)$. For p near the root $g(p) - p$ is very small. For p away from the root $g''(p)$ is very small. The only issue is when $g'(p) = 1$, in which case there is a singularity. This occurs for $p = -1$, and is a point from which the iterate cannot converge.

1.1 Experiments

1.1.1 Experiment 1

Our first trials are run using the function

$$f(x) = xe^x - 1 = 0. \tag{1.2}$$

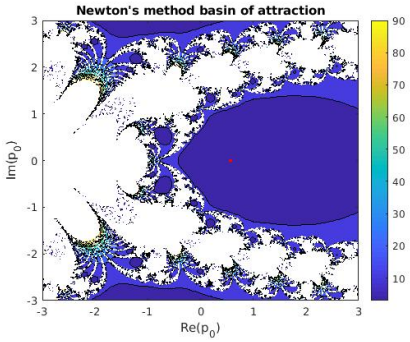
Known as the Lambert W function, the roots of this function are well known and can be called in Matlab using `lambertw(branch,1)`. The function has an infinite number of branches.

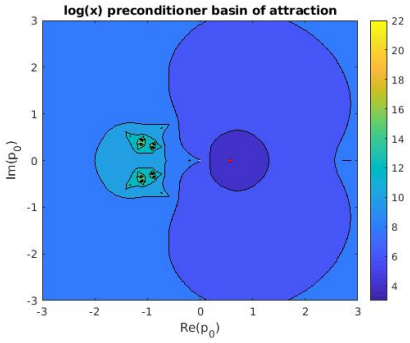
We compare using Newton's method on $f(x)$ with using Newton's method on the rearranged fixed point iteration

$$g(x) = \log(x) + x - ik2\pi, \tag{1.3}$$

where k represents the branch of the function sought. We make the following notes on their differences:

- $g(x)$ (and all its derivatives) has a singularity at $x = 0$
- $g'(x)$ and $f'(x)$ have roots at $x = -1$
- all derivatives of $f(x)$ have roots, which gradually move towards $-\infty$
- all other derivatives of $g(x)$ have no roots
- the limit of all the derivatives of $g(x)$ as x approaches $\pm\infty$ are finite (and zero except for $g'(x)$)
- $\lim_{x\to-\infty} f^{(n)}(x) = 0 \ \forall n \geq 1$ and $\lim_{x\to\infty} f^{(n)}(x) = \infty \ \forall n \geq 0$





Using $g(x)$ shows significant improvement over $f(x)$. Not only does the basin of attraction encompass essentially the entire complex plane (minus the points $x = -1, 0, e$ where the iteration has a singularity, fixed point and root, respectively) but the number of iterations required to reach a given tolerance drops for the majority of the original basin. Moreover, $g(x)$ allows for great control on which root of $f(x)$ we seek. Choosing different k will allow us to converge to a different root of the function. It appears any branch may be selected with a high degree of accuracy resulting.

Testing using $h(x) = x - e^{-x}$ shows an intermediate preconditioner. It widens the basin of attraction, albeit with several remaining fractal areas and 'off-roots' where Newton's method refuses to converge. The function $h(x)$ has the following properties:

- the limit of $h(x)$ as x approaches either infinity is infinite
- the same limits for the derivatives of $h(x)$ are finite for $+\infty$ and infinite for $-\infty$
- $h'(x)$ has a root at $x = 0$, subsequent derivatives are everywhere non-zero

1.1.2 Experiment 2

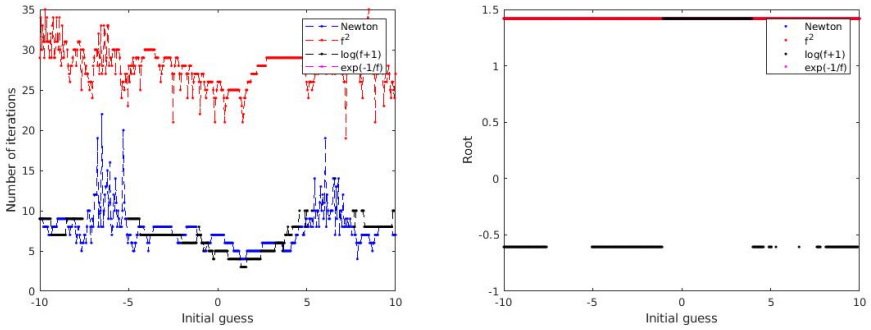
For this experiment we examine a specific instance of Kepler's equation:

$$f(x) = x - 0.8\sin(x) - \frac{2\pi}{10} = 0. \tag{1.4}$$

Newton's method performs adequately with this method. Rather than try to find a simpler equation (a challenge) I will complicate the function in numerous ways to see what will make Newton's method fail.

Taking $f(x)^m$ will obviously induce a root of multiplicity m. This will naturally reduce the convergence rate to linear and induce structure into the error progression.

Taking $\log(f(x)+1)$ significantly reduced the basin of attraction for the real root. It also allowed for imaginary roots to be found. There were several choices



of initial guess that did not converge within 100 iterations. Interestingly, these choices led to the largest number of iterations for the normal case. These areas correspond roughly with very low values of the derivative: $x = \arccos(0.8) = 0.6435 + n2\pi$.

Taking $\exp(-1/|f(x)|)$ results in the iteration

$$p_{n+1} = p_n - \text{sign}(f(p_n))f(p_n)^2/f'(p_n).$$

It is possible to get convergence with this iteration, however it is neither accurate nor efficient. The tolerance must be lowered substantially or significantly more iterations allowed. So far this is the worst form of the equation.

1.1.3 Experiment 3

Finally, we examine the following polynomial:

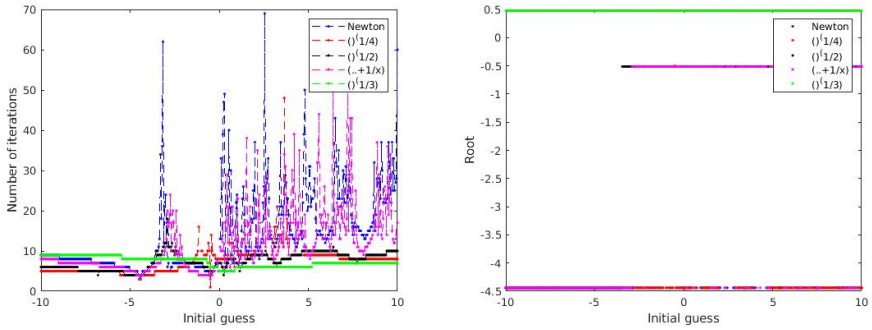
$$x^4 + 4x^3 - 2x^2 + 1 = 0. \tag{1.5}$$

We have tried a number of fixed point iterations as preconditioners:

- $\pm(-4x^3 + 2x^2 - 1)^{1/4}$
- $\pm\sqrt{0.5x^4 + 2x^3 + 0.5}$
- $\frac{1}{2}x^3 + 2x^2 + \frac{1}{2x}$
- $(-0.25x^4 + 0.5x^2 - 0.25)^{1/3}$

of which the second with a minus symbol performs 'best' for finding the real root and the fourth finds both imaginary roots with relative ease.

The leading order of each is 3/4, 2, 3 and 4/3 respectively. I conjecture that it is ideal to have the leading order of the preconditioner be closest to 1 without going under. In this way, the function $x - g(x)$ (where $g(x)$ is the fixed point function used as a preconditioner) will be roughly linear. This conjecture may apply broadly: the more closely linear the function we wish to solve, the quicker we can solve it and with a wider basin of attraction.



1.2 Ideal preconditioning

The best fixed point iteration we can hope for is the constant function:

$$g(x) = x^*$$

where x^* is the root we wish to converge to. Barring this, we would like $g(x)$ to be linear, which would also guarantee single step convergence:

$$g(x) = x^* + a(x - x^*).$$

This last point is of interest: the fixed point iteration will take infinite time to converge to zero, but Newton's iteration converges instantly.

It is curious that Newton's will converge in one step so long as the function is linear while a fixed point iteration will only do so if the function is constant. It suggests a hierarchy of root finding methods, with more complicated functions converging in one step the higher up one goes. The question is how do we go from one level of the hierarchy to another? It seems to be something along the lines of:

$$x_{n+1} = \frac{g(x) + \sum_{k=1}^{n} \frac{(-1)^k x_n^k g^{(k)}(x_n)}{k!}}{1 + \sum_{k=1}^{n} \frac{(-1)^k x_n^{k-1} g^{(k)}(x_n)}{k!}}$$

Failing to have knowledge of the inverse of the function we wish to find the root of, we hope that the function is, or nearly is, linear. Failing even this, we must make compromises:

- if we desire greater accuracy, we require $f''(x^*) = 0$, and possibly further derivatives
- if we desire larger basins of attraction, we require $f''(x) \approx 0$ away from x^*

We look for a second ideal fixed point iteration. That is, we hope to solve

$$x^* = x - \frac{g(x) - x}{g'(x) - 1}.$$

This can be re-arranged into the following singular first order ODE:

$$\begin{cases} (x - x^*)g'(x) - g(x) & = x^* \\ g(x^*) & = x^*. \end{cases}$$

If we make the change of variables $y = x - x^*$ and let $G(y) = g(y + x^*) - x^*$ then the ODE simplifies:

$$yG'(y) - G(y) = 0$$

which has solution $G(y) = ay = a(x - x^*) = g(x) - x^*$, so that $g(x)$ is exactly one of the linear functions we asked for above. Note this ODE is ill-posed: all choices of $a \in \mathbb{R}$ (and possibly $\mathbb{C}$) give solutions.

Since there already exists infinite solutions to the ODE, perhaps there is a solution of a different form. Suppose $f(x)$ also solves the ODE, then the difference $u(x) = g(x) - f(x)$ solves the ODE

$$\begin{cases} (x - x^*)u'(x) - u(x) & = 0 \\ u(x^*) & = 0. \end{cases}$$

This implies $u(x) = (x - x^*)u'(x)$ and $u'(x) = u'(x) + (x - x^*)u''(x)$. It must therefore be that $u''(x) = 0$ and the only type of solutions are linear functions. Alternatively, if we only consider the problem posed on a small domain D then any solution with $u''(x) = 0 \ \forall x \in D$ will also work.

1.3 Newton's method as fixed point iteration

$$z = e^{-z} = g(z)$$

Note for this fixed point iteration there is 2π–periodicity in the imaginary direction. If $Re(z) >> 0$ then $|g(z)| << 1$. For fixed point iterations it is important that $|g'(z)| < 1$. This is true for $Re(z) > 0$ so we would prefer that $Re(g(z)) > 0$. This does not occur for $|Im(z)| > \pi/2$ (again recall periodicity). However, as long as $|Im(g(z))| < \pi/2$ then $Re(g(g(z)) > 0$:

$$Im(g(z)) = \sin\left(Im(z)\right) e^{-Re(z)}.$$

$$z = g(z) = \frac{z + 1}{e^z + 1}$$

This is the Newton's iteration for the previous fixed point iteration. If $Re(z) >> 0$ then $|g(z)| << 1$, like before.

We've established that sufficient conditions for convergence are met when $g(D) \subset D$ and $|g(x)| < 1$ for all $x \in D$. For preconditioned Newton these conditions change to:

$$(i) \ x - \frac{g(x) - x}{g'(x) - 1} \in D \ \forall x \in D,$$

$$(ii) \ \left| \frac{g''(x)(g(x) - x)}{(g'(x) - 1)^2} \right| < 1 \ \forall x \in D.$$

Also included in the basin of attraction are the pre-images of these. For fixed point iteration, $\{g^{-k}(D)\}_{k=0}^{\infty}$ all lie within the basin. There is again an appropriate replacement of $g(x)$ for the same to be said of Newton's method.

1.3.1 Fixed point iterations

We examine in detail two fixed point iterations:

$$g_1(z) = e^{-z}, \quad g_2(z) = -\log(z). \tag{1.6}$$

First and foremost, these functions have fixed points at the real root of

$$ze^z - 1 = 0 \tag{1.7}$$

and are inverses of each other.

The function $g_1(z)$ is 2π–periodic in the imaginary direction. It is for this reason that it cannot converge to any of the complex roots of function 1.7. Likewise, this means we need only consider $-\pi \leq Im(z) < \pi$. The function $g_2(z)$ also does not converge to complex roots by choice of branch cut. This can be changed with the addition of $in2\pi$, where n is the branch cut of interest.

We are concerned with where each function will converge. We can guarantee convergence in a region D by the fixed point theorem.

Theorem 1 (Fixed point iteration theorem)**.** *If a function $g(z)$ satisfies*

(i) $g(z) \in D$

(ii) $|g'(z)| < 1$

for all $z \in D$ then it has a unique fixed point in D and the iteration $z_{n+1} = g(z_n)$ converges to this fixed point.

Consider $g_1(z)$: condition (ii) of theorem 1 is satisfied when $Re(z) > 0$; for condition (i) note that $g_1(z)$ rotates off the real axis by angle $-Im(z)$. For $g_1(z)$ to satisfy $Re(g_1(z)) > 0$ it is necessary that $|Im(z)| < \pi/2$. Our region D_0 (the region for which $g_1(z)$ satisfies theorem 1) is therefore:

$$D_0 = \{z \in \mathbb{C} | Re(z) > 0, \ -\pi/2 < Im(z) < \pi/2\}. \tag{1.8}$$

Given that $g_2(z)$ is the inverse of $g_1(z)$ and D_0 exists, there is no such region for $g_2(z)$.

Figure 1.4 gives a representation of the region D_0 (purple) and its images and pre-images. For ease of notation we define the sets D_k as:

$$D_{k+1} = g_1(D_k), \quad D_{k-1} = g_2(D_k).$$

Since $D_1 \subset D_0$ by definition of D_0 and $g_2(g_1(z)) = z$ there exists a hierarchy of sets: $D_{k+1} \subset D_k \subset D_{k-1}$ for all $k \in \mathbb{Z}$. Each set D_{k-1} is the pre-image of D_k under the $g_1(z)$ function. As such, $D_{-\infty}$ represents the basin of attraction of $g_1(z)$.

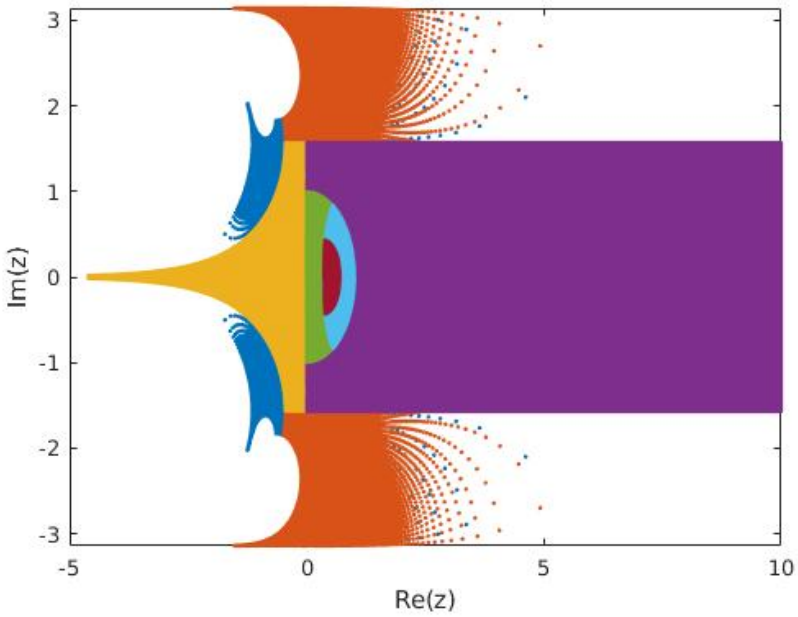


Figure 1.4: The region D_0 and its images under $g_1(z)$ and $g_2(z)$.

1.3.2 Preconditioned Newton

We now look at applying Newton's method to the functions $g_1(z)-z$ and $g_2(z)-z$. This will give the following fixed point iteration functions:

$$f_1(z) = \frac{zg_1'(z) - g_1(z)}{g_1'(z) - 1} = \frac{1+z}{1+e^z}, \quad f_2(z) = \frac{z(1-\log(z))}{1+z}. \tag{1.9}$$

Note that $f_1(z) = f_2(e^{-z})$ and $f_2(z) = f_1(-\log(z))$.

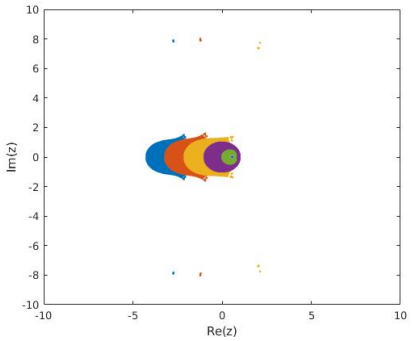
The function $f_1(z)$ has singularities at all branches of $\log(-1)$. Unlike $g_1(z)$, it is not periodic in the imaginary direction. The function $f_2(z)$ has an erroneous fixed point at $z = 0$, a singularity at $z = -1$ and a root at $z = e$. These points will be problematic and must be excluded from the basins of attraction.

We can perform the same analysis as before using theorem 1. Condition (ii) can be written in terms of the fixed point functions:

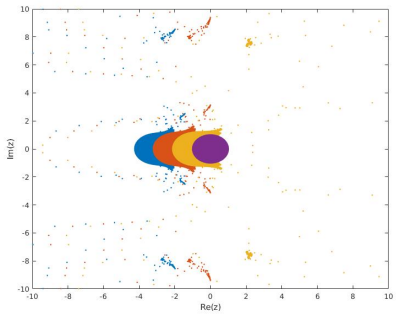
$$|f_1'(z)| = \left|\frac{g_1''(z)(g_1(z)-z)}{(g_1'(z)-1)^2}\right| < 1, \quad \left|\frac{g_2''(z)(g_2(z)-z)}{(g_2'(z)-1)^2}\right| < 1.$$

This ultimately requires

$$|f_1'(z)| = \left|\frac{1-ze^z}{(1+e^z)^2}\right| < 1, \quad |f_2'(z)| = \left|\frac{z+\log(z)}{(1+z)^2}\right| < 1.$$



(a) The region D_0^1, its images and pre-images under $f_1(z)$.



(b) The region D_0^1 and its pre-images under $f_1(z)$.

Condition (ii) holds for $f_2(z)$ except in an elliptical region containing $z = -1$. The region where the fixed point theorem is true for $f_2(z)$, hereafter called D_0^2, is then the complex plane without the pre-images of this ellipse. However, if $z \approx -1$ but not equal then $f_2(z)$ is not within this ellipse. More precisely, the image of the ellipse is outside the ellipse. Thus, D_0^2 is the entire complex plane except for the points -1, 0 and e and their pre-images. This constitutes a countable set. (side note: the pre-image of 0 for $f_2(z)$ is e, so the definition of D_0^2 can be further simplified if desired)

Analysis for $f_1(z)$ is carried out numerically. Through experiments we can establish that the ball of radius 1 in the complex plane represents a region where theorem 1 is satisfied. Call this ball D_0^1. We can also show that the inverse of $f_1(z)$ is

$$f_1^{-1}(z) = z - 1 - W(-ze^{z-1})$$

where $W(z)$ is the Lambert W function (0 branch). Using this, we repeat figure 1.4.

Figure 1.5a shows the hierarchy of sets, with D_0^1 in purple. Its images are inset and converge rapidly to the root. Its pre-images extend onto the negative real line with some scattering. A more resolved D_0^1 (see figure 1.5b) shows greater detail in this scattering, and reveals some fractal structure.

Chapter 2

Newton's method as a time dependent PDE

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \\ \implies \frac{x_{n+1} - x_n}{\Delta t} &= -\frac{f(x_n)}{f'(x_n)},\ \Delta t = 1. \end{aligned}$$

Allowing Δt to represent the change in some path variable t, the left hand side becomes an approximation to the first derivative of x with respect to t:

$$\begin{aligned} \frac{dx}{dt} &= -\frac{f(x)}{f'(x)} \\ \implies \frac{\partial f}{\partial x}\frac{dx}{dt} &= -f(x) \\ \implies \frac{df}{dt} &= -f(x) \\ \implies f(x(t)) &= ae^{-t}. \end{aligned}$$

Newton's method can then be characterized as a specific choice of $g(t)$, Δt and ODE solver in the following method:

$$f(x(t)) = g(t),\ \lim_{t \to T} g(t) = 0,\ \frac{dx}{dt} = \frac{dg/dt}{\partial f/\partial x}.$$

Open questions:

- does the method converge? under what circumstances?
- what makes a good choice of $g(t)$?
- what happens when $f'(x) = 0$? is there a similar way to rewrite modified Newton's? is modified Newton's a different choice of $g(t)$?

- it would be nice if all Newton-like methods could be brought under this umbrella, but this would be a daunting task.

The same techniques can be applied to any fixed point iteration:

$$\begin{aligned} x_{n+1} = g(x_n) &= x_n + g(x_n) - x_n \\ \implies \frac{dx}{dt} &= -x + g(x). \end{aligned}$$

This makes it a perfect candidate for exponential time differencing:

$$\begin{aligned} x(t) &= x(0)e^{-t} + e^{-t}\int_0^t g(x(s))e^s ds, \\ x(t_{n+1}) &= e^{-h}\left(x(t_n) + \int_0^h g(x(t_n+s))e^s ds\right). \end{aligned}$$

2.1 Early experiments

After some initial trials, any choice of $g(t)$ seems suitable. Moreover, it is not necessary for $g(t)$ to be zero in the limit. As long as $g(t)$ is zero somewhere then the method appears to converge.

The convergence seems linked to the time discretization. Using Euler's method gives order one convergence (no surprises there). I should test with RK4 to see if we achieve better convergence. Note that the convergence we are interested in is global, especially for choices of $g(t)$ that are zero in finite time.

Some obvious requirements on $g(t)$:

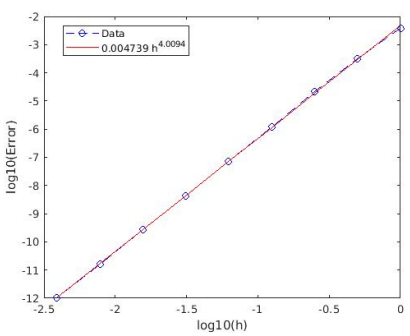
- $\min f \leq \min g < \max g \leq \max f$
- $g(t) = 0$ for some t (although, if we desire a solution to $f(x) = a$ then we would require $g(t) = a$)

2.2 Pathing Methods

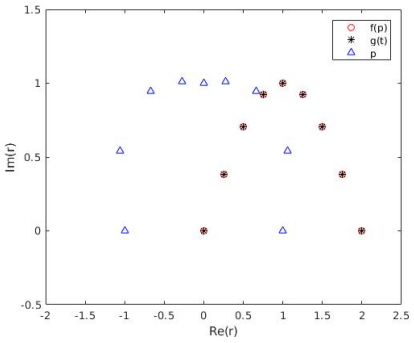
It no longer seems appropriate to name these methods after Newton, as they can arise on their own merit. The main issues facing these methods are the choice of path and the time discretization.

On the issue of time discretization, obviously higher order will give more accurate results. However, lower order discretizations can be used repeatedly to improve accuracy. For example, Newton's method can be considered as repeated application of a pathing method using $g(t) = a(1-t)$, Euler's method and $\Delta t = 1$. Analytically, this is identical to the original derivation of Newton's method.

On the issue of the choice of path, there are a number of problems to consider. Any features of the function must be represented in the path. For example, we



(a) 4th order convergence of pathing method using $g(t) = f(x_0)(1-t)$ and RK4.



(b) Path around singularity using $g(t) = f(x_0) + i\sin(\pi t)$ for $f(x) = 1 - 1/x$.

cannot use the path $g(t) = a(1-t)$ if the function has a local extrema between p_0 and the root, as this behaviour is not represented in the path.

For multiple roots the path will decide on which root is converged to. For singularities, the path will need to route around into complex space, using allowable values of $f(x)$.

The most basic pathing method that appears to have great success uses $g(t) = a(1-t)$. This arrives at a root at $t = 1$. For this path to be used, the function in question must be monotonic between the initial guess and the root. This path cannot traverse any valleys or climb hills.

The path $g(t) = a(1-t) + i\sin(\pi t)$ was used for the example function $f(x) = 1 - 1/x$. This path successfully routed the singularity when using an initial guess less than zero. However, this path could not be used if $f(x)$ did not have values that allowed it to traverse this path. For example, if the imaginary part of $f(x)$ was nowhere equal to 1 then the path would not correspond to the function, as at time $t = 0.5$ we require that $f(x(0.5)) = 0.5a + i$.

2.3 Literature review

1953 Davidenko [4] shows the continuous analogy of Newton's method for systems (work cannot be found in English, it seems)

1958 M.K. Gavurin [6] originates the continuous analogy of Newton's method (or does he? what about Davidenko? also in Russian)

1968 Galanov and Malakbovskaya provide a highly general (and uncited?) method of solving nonlinear equations [5]

1972 F.H. Branin [2] suggests small change to equation (attributed to L.E. Kugel through private communication); this paper is critiqued by R.P. Brent [3]

1975 J.P. Abbott and R.P. Brent [1] examine convergence of the analogy (originating the term continuous Newton's method)

1988 Saupe [8] has an interesting paper on several aspects of Newton's method

2005 Hauser [7] gives an interesting discussion on the rigidity of the path $x(t)$ takes, read further

2.3.1 Some other papers of note

65J15 Gibali, On the convergence rate of CNM (2016, Russian)

65H05 Gutierrez, Numerical properties of different root-finding algorithms obtained for approximating CNM (2015, suggests using non-constant step sizes in time discretization)

65J15 Nair, Regularized versions of CNM and CNM under general source conditions (2008, regularization for when the inverse is not well-defined)

49M15 Neuberger, The CNM, inverse functions and Nash-Moser (basic discussions on CNM)

58C15 Neuberger, Integrated form of CNM (what it says on the box)

30D05 Neuberger, CNM for polynomials (examines basins of attraction for CNM on polynomials)

58C15 Castro, An inverse function theorem via CNM (using CNM to prove a theorem)

90C48 Attouch, The second-order in time CNM

65J15 Riaza, Strong singularities and the CNM

65J15 Riaza, Weak singularities and the CNM

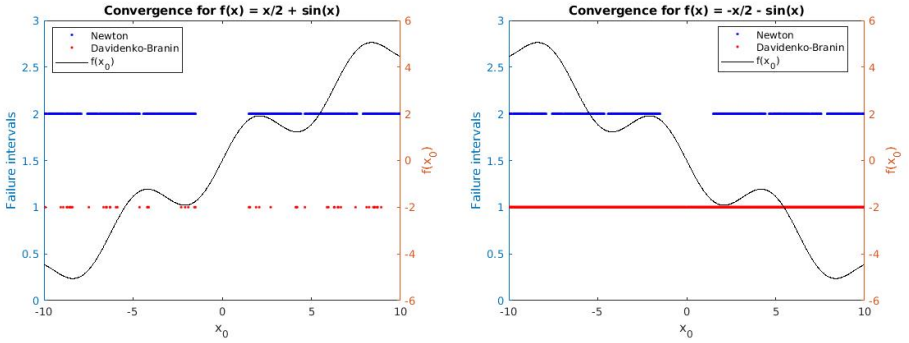
65J10 Airapetyan, CNM and its modification (convergence theorems and derivative calculations)

90C30 Diener, Newton leaves and the CNM (good discussion of Branin's research and consequences, looks at extending ideas to things called Newton leaves)

90C30 Diener, An extended CNM

58C15 Jongen, Some reflections on the CNM for rational functions (I think this refutes some claims by Branin)

65H05 Li, Path following approaches for solving nonlinear equations: homotopy, CN and projection



(a) Initial guesses where the methods failed to converge within 20 iterations to a tolerance of 10^{-8}.

(b) Initial guesses where the methods failed to converge within 20 iterations to a tolerance of 10^{-8}.

2.3.2 1D test of the Davidenko-Branin method

The Davidenko-Branin method (term coined by R.P. Brent) is a slight adjustment to the continuous Newton's method:

$$\frac{dx}{dt} = \frac{\text{adj}J}{|\det J|} f(x).$$

This incorporates a sign change in the right-hand side whenever $f(x)$ passes over a 'hump'. If CNM can be thought of as moving $f(x(t))$ along the path $f(x_0)e^{-t}$ then the DBM can be thought of as moving $f(x(t))$ along a piecewise path composed of $f(x_0)e^{-t}$ and $f(x_n)e^{t}$.

The overall result is that $f(x)$ is capable of surmounting 'humps' (regions of change of $f'(x)$) to continue approaching the root where Newton's method would fail. However, this requires knowledge of where the root is with respect to starting positions. Since $f(x)$ can now climb slopes as well as descend them towards zero, the initial sign on the right-hand side will dictate if $f(x)$ moves towards or away from the root.

We examine the function $f(x) = 0.5x + \sin(x)$ and apply a 1D version of DBM. We achieve the results found in figure 2.2a in 20 iterations of both Newton's method and DBM using Euler's method with $\Delta t = 1$. DBM clearly has a larger basin of attraction than Newton's method. Newton's method in particular fails outside the region of monotonicity surrounding the root.

We flip the function over the real line so that the monotonicity is now reversed (figure 2.2b). DBM fails to converge for almost all points tested. Introducing a sign change in the method will correct this.

Chapter 3

Schwarz methods as fixed point iterations

Consider the alternating Schwarz method

$$\begin{cases} \mathcal{L}_1 u_1^n = f(u_1^n) & x \in (0,\beta) \\ u_1^n(0) = 0 & \\ u_1^n(\beta) = u_2^{n-1}(\beta) & \end{cases}, \quad \begin{cases} \mathcal{L}_2 u_2^n = f(u_2^n) & x \in (\alpha,1) \\ u_2^n(1) = 1 & \\ u_2^n(\alpha) = u_1^n(\alpha). & \end{cases}$$

We can think of the operation to solve for u_1^n as a function $G_1 : \mathbb{R} \to \mathbb{R}$ such that $G_1(u_2^{n-1}(\beta)) = u_1^n(\alpha)$. Likewise, $G_2(u_1^n(\alpha)) = u_2^n(\beta)$ and $G_2(G_1(\gamma)) = \gamma$ is a fixed point iteration.

Let $G(\gamma) = G_2 \circ G_1$. We need to set conditions on $\frac{dG}{d\gamma}$:

$$\begin{aligned} \frac{dG}{d\gamma} &= \frac{\partial G_2}{\partial G_1}\frac{dG_1}{d\gamma} \\ &= g_2(\beta; g_1(\alpha)) \end{aligned}$$

where $g_1(x)$ solves

$$\begin{cases} \left(\mathcal{L}_1 - \frac{\partial f}{\partial u}\right) g_1(x) = 0 \\ g_1(0) = 0 \\ g_1(\beta) = 1 \end{cases}$$

and $g_2(x;\nu)$ solves

$$\begin{cases} \left(\mathcal{L}_2 - \frac{\partial f}{\partial u}\right) g_2(x;\nu) = 0 \\ g_2(1;\nu) = 0 \\ g_2(\alpha;\nu) = \nu. \end{cases}$$

This is arrived at by taking the derivatives of the original iterations with respect to γ.

Note that $G(\gamma)$ is linear in γ as long as $\frac{\partial f}{\partial u}$ does not depend on u. Since g_1 and g_2 do not depend on γ the second derivative of G with respect to γ is zero. Therefore, we should expect Newton to converge in a single step if we apply it to $G(\gamma)$.

For $G(\gamma)$ nonlinear it is possible to establish that it is a convergent fixed point if the operator $\mathcal{L} - \frac{\partial f}{\partial u}$ satisfies a maximum principle. Then g_1 and g_2 attain their maximum values on the boundary and so $|g_2(\beta; g_1(\alpha))| \leq |g_1(\alpha)| \leq 1$. We need to show that G is a contraction mapping. It would suffice that $\mathcal{L}$ also satisfies a maximum principle, then $G : [0,1] \to [0,1]$.

Applying Newton's method to $G(\gamma)$ gives the following algorithm:

$$
\begin{aligned}
&(1) \begin{cases} \mathcal{L}_1 u_1 = f(u_1) \\ u_1(0) = 0 \\ u_1(\beta) = \gamma_n \end{cases} \\
&(2) \begin{cases} \mathcal{L}_2 u_2 = f(u_2) \\ u_2(1) = 1 \\ u_2(\alpha) = u_1(\alpha) \end{cases} \\
&(3) \begin{cases} \left(\mathcal{L}_1 - \frac{\partial f(u_1)}{\partial u}\right) g_1 = 0 \\ g_1(0) = 0 \\ g_1(\beta) = 1 \end{cases} \\
&(4) \begin{cases} \left(\mathcal{L}_2 - \frac{\partial f(u_2)}{\partial u}\right) g_2 = 0 \\ g_2(1) = 0 \\ g_2(\alpha) = g_1(\alpha) \end{cases} \\
&(5) \gamma_{n+1} = \gamma_n - \frac{u_2(\beta) - \gamma_n}{g_2(\beta) - 1} = \frac{g_2(\beta)\gamma_n - u_2(\beta)}{g_2(\beta) - 1}.
\end{aligned}
$$

Steps (2) and (3) can be performed simultaneously.

Early experiments show that by adding steps 3 through 5 (straightforward linear solves) we increase convergence from linear to quadratic (see figure 3.1).

We now look to apply this algorithm to more complicated examples. Suppose the nonlinear function on the right-hand side is also dependent on u', such as in viscous Burgers:

$$
\begin{cases} \epsilon u''(x) & = uu' \\ u(-1) & = 1 \\ u(1) & = -1. \end{cases}
$$

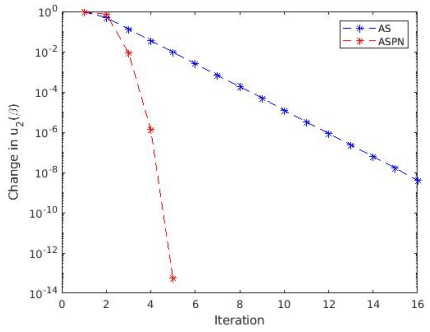


Figure 3.1: Convergence rates of alternating Schwarz and the new algorithm.

Then changes must be made to the algorithm:

$$(1)\begin{cases}\mathcal{L}_1 u_1 = f(u_1, u_1') \\ u_1(0) = 0 \\ u_1(\beta) = \gamma_n\end{cases}$$

$$(2)\begin{cases}\mathcal{L}_2 u_2 = f(u_2, u_2') \\ u_2(1) = 1 \\ u_2(\alpha) = u_1(\alpha)\end{cases}$$

$$(3)\begin{cases}-\frac{\partial f(u_1,u_1')}{\partial u'} g_1' + \left(\mathcal{L}_1 - \frac{\partial f(u_1,u_1')}{\partial u}\right) g_1 = 0 \\ g_1(0) = 0 \\ g_1(\beta) = 1\end{cases}$$

$$(4)\begin{cases}-\frac{\partial f(u_2,u_2')}{\partial u'} g_2' + \left(\mathcal{L}_2 - \frac{\partial f(u_2,u_2')}{\partial u}\right) g_2 = 0 \\ g_2(1) = 0 \\ g_2(\alpha) = g_1(\alpha)\end{cases}$$

$$(5)\gamma_{n+1} = \gamma_n - \frac{u_2(\beta) - \gamma_n}{g_2(\beta) - 1} = \frac{g_2(\beta)\gamma_n - u_2(\beta)}{g_2(\beta) - 1}.$$

Applying this algorithm to steady viscous Burgers (see above) we arrive at a nearly identical picture of the convergence rate as figure 3.1. This improvement holds for ϵ as low as 0.1. For smaller ϵ there is trouble finding solutions for steps (1) and (2). A better nonlinear solver may be required.

Suppose one were to solve the steady viscous Burgers equation iteratively using Newton's method and domain decomposition. This would give the following alternating Schwarz method:

$$\begin{aligned}(1)u_1^{n+1} &= u_1^n - J(u_1^n)^{-1} f(u_1^n), \\ (2)u_2^{n+1} &= u_2^n - J(u_2^n)^{-1} f(u_2^n),\end{aligned}$$

$$f(u) = \epsilon u''(x) - u(x)u'(x).$$

We discretize with a finite difference scheme:

$$x_i = -1 + ih,\ u_i = u(x_i),\ u'(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2h},\ u''(x_i) \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}.$$

We wish to follow the preconditioning algorithm from before. We know there exists a fixed point iteration $G(\gamma)$ for the point $u_1^n(\beta)$. We also know $G'(\gamma) = \frac{\partial u_2^n(\beta)}{\partial u_1^n(\alpha)} \frac{du_1^n(\alpha)}{d\gamma}$. We begin by defining the problem on the first domain.

The i–th element of $f(u)$ on the domain $[-1, \beta]$ can be defined as:

$$\begin{aligned}
f_i(u) &= \frac{\epsilon}{h^2}\left(u_{i-1} - 2u_i + u_{i+1}\right) - \frac{u_i}{2h}\left(u_{i+1} - u_{i-1}\right), \\
f_1(u) &= \frac{\epsilon}{h^2}\left(1 - 2u_1 + u_2\right) - \frac{u_1}{2h}\left(u_2 - 1\right), \\
f_{N-1}(u) &= \frac{\epsilon}{h^2}\left(u_{N-2} - 2u_{N-1} + \gamma\right) - \frac{u_{N-1}}{2h}\left(\gamma - u_{N-2}\right).
\end{aligned}$$

Therefore, we can define the Jacobian $J(u)$ as:

$$J(u) = \begin{bmatrix}
\frac{-2\epsilon}{h^2} - \frac{u_1 - 1}{2h} & \frac{\epsilon}{h^2} - \frac{u_1}{2h} & 0 & \cdots & 0 \\
\frac{\epsilon}{h^2} + \frac{u_2}{2h} & \frac{-2\epsilon}{h^2} - \frac{u_3 - u_1}{2h} & \frac{\epsilon}{h^2} - \frac{u_2}{2h} & \cdots & 0 \\
0 & \frac{\epsilon}{h^2} + \frac{u_3}{2h} & \frac{-2\epsilon}{h^2} - \frac{u_4 - u_2}{2h} & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \frac{\epsilon}{h^2} + \frac{u_{N-1}}{2h} & \frac{-2\epsilon}{h^2} - \frac{\gamma - u_{N-2}}{2h}
\end{bmatrix}.$$

It is then a matter of solving $J(u^n)(u^{n+1} - u^n) = -f(u^n)$ for u^{n+1}.

We can then calculate the derivative of u^{n+1} with respect to γ by taking such a derivative of the whole equation:

$$\begin{aligned}
\left(\frac{\partial J(u^n)}{\partial \gamma}\right)(u^{n+1} - u^n) + J(u^n)\frac{du^{n+1}}{d\gamma} &= -\frac{df(u^n)}{d\gamma} \\
\implies \frac{du^{n+1}}{d\gamma} &= -J(u^n)^{-1}\left(\left(\frac{\partial J(u^n)}{\partial \gamma}\right)(u^{n+1} - u^n) + \frac{df(u^n)}{d\gamma}\right) \\
&= -J(u^n)^{-1}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{\epsilon}{h^2} - \frac{u^{n+1}(\beta - h)}{2h} \end{pmatrix}.
\end{aligned}$$

Note we only need the value of the derivative at the point α. Therefore, we pull out a single element of the matrix $J(u^n)^{-1}$:

$$\frac{du_1^n(\alpha)}{d\gamma} = \left(\frac{u_1^{n+1}(\beta - h)}{2h} - \frac{\epsilon}{h^2}\right) J(u_1^n)_{\frac{\alpha+1}{h}, N-1}$$

where $J(u_1^n)_{i,j}$ is the element in the i–th row and j–th column of $J(u_1^n)$.

By similar argument, we have the following formula for $\frac{\partial u_2^{n_1}(\beta)}{\partial u_1^{n+1}(\alpha)}$:

$$\frac{\partial u_2^{n+1}(\beta)}{\partial u_1^{n+1}(\alpha)} = -\left(\frac{\epsilon}{h^2} + \frac{u_2^{n+1}(\alpha+h)}{2h}\right) J(u_2^n)_{\frac{\beta-\alpha}{h},1}.$$

$G'(\gamma)$ is then a product of these two formulae:

$$G'(\gamma) = -\left(\frac{\epsilon}{h^2} + \frac{u_2^{n+1}(\alpha+h)}{2h}\right)\left(\frac{u_1^{n+1}(\beta-h)}{2h} - \frac{\epsilon}{h^2}\right) J(u_2^n)_{\frac{\beta-\alpha}{h},1} J(u_1^n)_{\frac{\alpha+1}{h},N-1}.$$

The issue is the function $G(\gamma)$ is dependent on the initial guess u^n. While we converge to the fixed point γ_n using this iteration, the function $G(\gamma)$ is changing at each iteration. Therefore, we have no guarantee that the Newton-Raphson iteration at step n will provide an appropriate step along the function $G(\gamma)$ at step $n+1$.

We can still use this iteration by finding the fixed point of a given function $G(\gamma)$, then using this fixed point to update the solution. That is, we run Newton's method on $G(\gamma)$, find a fixed point, then perform one iteration of Newton's method to get u^{n+1}.

Summary of current findings:

- the first algorithm described in this chapter fully solves a nonlinear system for a given transmission condition, then performs a Newton iteration to improve the transmission condition;
- the second algorithm runs Newton's method on the transmission condition for a given u^n, then performs a Newton iteration to find u^{n+1} using the 'best possible' transmission condition.
- Note that the second algorithm can be made more efficient as only a rank 1 addition to the Jacobian needs to be made at each transmission condition step.

3.1 Breaking the algorithms

We seek an example for which the algorithm(s) presented in this chapter fail. That is, we look for example problems where the Newton preconditioning on the transmission condition prevents an otherwise good alternating Schwarz iteration. This should occur when $G'(\gamma) = 1$, as this is where the function $G(\gamma) - \gamma$ has a local extrema which prevents Newton-Raphson from converging. The algorithm would also fail if we found a cycle, but this is excluded from this analysis. Note that $G'(\gamma) < 1$ if and only if $G(\gamma) - \gamma$ is monotonic, which would mean Newton-Raphson could not be expected to fail for any reasonable choice of initial guess.

We will use the first algorithm presented in this chapter, with $u(0) = 0$ and $u(1) = 1$ as boundary conditions. In this context, $G'(\gamma) = g_2(\beta)$. Consider

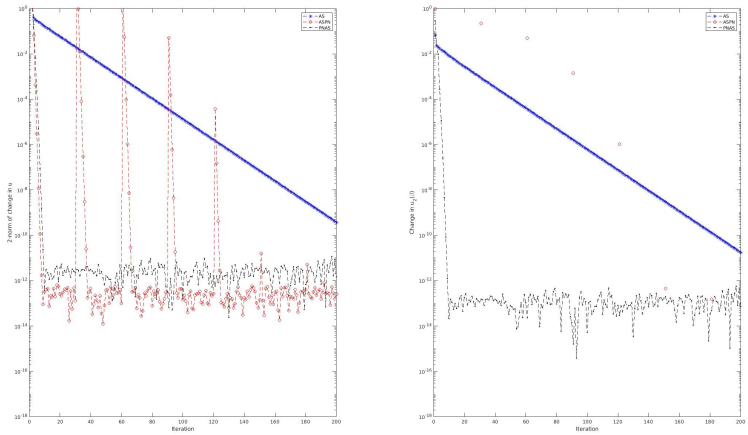


Figure 3.2: Comparison of standard additive Schwarz acting on the Newton iteration with preconditioning of the transmission condition by Newton's method both before and after an iteration of the Newton iteration on the solution. The 'iterations' along the x–axis represent roughly equal numbers of computations for all three methods. The grid used contains 1001 points, $\alpha = -\beta = -0.2$, $\epsilon = 0.1$.

what it means for $g_2(\beta) = 1$. We have already that $g_2(\alpha) = g_1(\alpha)$ and that $g_1(\beta) = 1$. Including the new condition that $g_2(\beta) = 1$ means that both $g_2(x)$ and $g_1(x)$ satisfy:

$$\begin{cases} \left(\mathcal{L} - \frac{\partial f}{\partial u}\right) g = 0 \\ g(\alpha) = g_1(\alpha) \\ g(\beta) = 1 \end{cases} \tag{3.1}$$

in the limit that $u_1 = u_2$ in the region of overlap. We must consider ourselves to be in this limit or nearby it for Newton-Raphson to fail on any meaningful interval of initial guesses.

This means that either (3.1) is singular or $g_1(x) = g_2(x)$. The first possibility can be true for a limited number of α and β (prove this?). Therefore, changing the overlap regions slightly would fix this issue. The second possibility means the linear ODE

$$\begin{cases} \left(\mathcal{L} - \frac{\partial f}{\partial u}\right) g = 0 \\ g(0) = 0 \\ g(1) = 0 \end{cases}$$

has at least one (and therefore infinite) nontrivial solutions. The problem is then singular.

Compare this problem with that for u:

$$\begin{cases} \mathcal{L}u = f(u) \\ u(0) = 0 \\ u(1) = 1. \end{cases}$$

If we rewrite these equations in a new form we will see they share many properties:

$$F_u(x, u, u', u'') = \mathcal{L}u - f(u) = 0, \quad F_g(x, g, g', g'') = \left(\mathcal{L} - \frac{\partial f(\tilde{u})}{\partial u}\right) g = 0$$
$$\implies \frac{\partial F_u(x, \tilde{u}, u', u'')}{\partial u} = \frac{\partial F_g}{\partial g}, \quad \frac{\partial F_u}{\partial u'} = \frac{\partial F_g}{\partial g'}, \quad \frac{\partial F_u}{\partial u''} = \frac{\partial F_g}{\partial g''}.$$

Note the difference between $\frac{\partial F_u}{\partial u}$ and $\frac{\partial F_g}{\partial g}$. The latter represents a particular choice of u in the former. Therefore, if $\frac{\partial F_u}{\partial u}$ satisfies a given property for all $u \in \mathbb{R}$ then $\frac{\partial F_g}{\partial g}$ also satisfies that property. Likewise, if $\frac{\partial F_g}{\partial g}$ satisfies a given property then $\frac{\partial F_u}{\partial u}$ also satisfies that property for at least a subset of $\mathbb{R}$.

The vast majority of uniqueness theorems on solutions to boundary value problems rely on properties satisfied by the various derivatives of the functions F_u and F_g (nb: cite some theorems). As such, if the problem in u can be proven to have a unique solution by one of these theorems then the problem in g will likewise have a unique solution, since they share the values of these derivatives. This means, by extension, that $G'(\gamma) \neq 1$ for all γ and $G(\gamma) - \gamma$ is monotonic. Newton-Raphson will then only have a problem if it encounters a cycle.

If we consider instead $u_1 \neq u_2$ in the overlap region then the problem (3.1) becomes:

$$\begin{cases} \left(\mathcal{L} - \frac{\partial f}{\partial u_1}\right)(g_1 - g_2) = \left(\frac{\partial f}{\partial u_1} - \frac{\partial f}{\partial u_2}\right) g_2 \\ g_1(\alpha) = g_2(\alpha) \\ g_1(\beta) = g_2(\beta). \end{cases}$$

Because of the nonzero right hand side in the first line of the problem it is no longer necessary for there to be a singularity. However, if u_1 and u_2 continue to converge to the same limit then the right hand side converges to zero. Even if u_1 and u_2 are close to one another the problem in the overlap region is 'nearly' singular.

By the same principles as above a near singularity in the overlap region of the g problem can only occur if a near singularity exists in the u problem. As such, we would expect similar instabilities in solving for u as we would for the transmission condition. This suggests that Newton-Raphson performed on the transmission condition will always improve or maintain the convergence of the alternating Schwarz method, and any failures of the Newton-Raphson should be concomitant with failures in the alternating Schwarz.

Theorem 2. *Let $u(x)$ solve the second order ODE*

$$\begin{cases} F(u, u', u'') = 0 \quad x \in [0, 1] \\ u(0) = a \\ u(1) = b. \end{cases}$$

Let $u_1(x)$ solve the same problem on $[0, \beta]$ with $u_1(\beta) = \gamma$ and $u_2(x)$ solve the problem on $[\alpha, 1]$ with $u_2(\alpha) = u_1(\alpha)$. Let $G(\gamma) = u_2(\beta)$. Let $J(x, y, z)$ be the Jacobian of $F(x, y, z)$.

If $G(\gamma) \in C^2(\mathbb{R})$ and $J(u, u', u'')$ is nonsingular on both $[0, 1]$ and $[\alpha, \beta]$, in the sense that there does not exist a nontrivial function $v(x)$ that satisfies

$$\begin{cases} J(u, u', u'') \cdot (v, v', v'') = 0 & x \in \Omega \\ v(x) = 0 & x \in \partial\Omega, \end{cases}$$

then $G'(\gamma) \neq 1$ in an interval around $u(\beta)$. As a corollary, the function $G(\gamma) - \gamma$ is monotonic in this interval.

Proof. It suffices to show $G'(u(\beta)) \neq 1$. To do so we must calculate $G'(\gamma)$:

$$G'(\gamma) = \frac{du_2(\beta)}{d\gamma} = \frac{\partial u_2(\beta)}{\partial u_1(\alpha)} \frac{du_1(\alpha)}{d\gamma}.$$

To establish how the solution of an ODE at a given point depends on the value of its endpoint, one can differentiate the ODE with respect to the endpoint:

$$\begin{cases} \frac{\partial F}{\partial u}\frac{du_1}{d\gamma} + \frac{\partial F}{\partial u'}\frac{du_1'}{d\gamma} + \frac{\partial F}{\partial u''}\frac{du_1''}{d\gamma} = 0 \quad x \in [0, \beta] \\ \frac{du_1(0)}{d\gamma} = 0 \\ \frac{du_1(\beta)}{d\gamma} = 1. \end{cases}$$

Let $g_1(x) = \frac{du_1}{d\gamma}$, then the ODE above may be written as:

$$\begin{cases} J(u_1, u_1', u_1'') \cdot (g_1, g_1', g_1'') = 0 \quad x \in [0, \beta] \\ g_1(0) = 0 \\ g_1(\beta) = 1. \end{cases}$$

Then $\frac{du_1(\alpha)}{d\gamma} = g_1(\alpha)$. Likewise, if $g_2(x) = \frac{du_2}{d\gamma}$ then

$$\begin{cases} J(u_2, u_2', u_2'') \cdot (g_2, g_2', g_2'') = 0 \quad x \in [\alpha, 1] \\ g_2(\alpha) = \frac{du_1(\alpha)}{d\gamma} = g_1(\alpha) \\ g_2(1) = 0. \end{cases}$$

Therefore, $G'(\gamma) = g_2(\beta)$.

Consider the difference between $g_1(x)$ and $g_2(x)$ in the region of overlap, $g(x) = g_2(x) - g_1(x)$. This function satisfies:

$$\begin{cases} J(u_2, u_2', u_2'') \cdot (g, g', g'') = (J(u_2, u_2', u_2'') - J(u_1, u_1', u_1'')) \cdot (g_1, g_1', g_1'') & x \in [\alpha, \beta] \\ g(\alpha) = 0 \\ g(\beta) = G'(\gamma) - 1. \end{cases}$$

For $\gamma = u(\beta)$ $u_1 = u_2$ in the region of overlap. If $G'(u(\beta)) = 1$ then $g(x) = 0$ by the assumption that $J(u, u', u'')$ is nonsingular on $[\alpha, \beta]$. Define the function $\hat{g}(x)$:

$$\hat{g}(x) = \begin{cases} g_1(x) & x \in [0, \alpha] \\ g_2(x) & x \in (\alpha, 1]. \end{cases}$$

This function satisfies:

$$\begin{cases} J(u, u', u'') \cdot (\hat{g}, \hat{g}', \hat{g}'') = 0 \\ \hat{g}(0) = \hat{g}(1) = 0. \end{cases}$$

By the assumption that $J(u, u', u'')$ is nonsingular on $[0, 1]$ it must be that $\hat{g}(x) = 0$. This contradicts $G'(u(\beta)) = 1$. Therefore, $G'(u(\beta)) \neq 1$. Since $G(\gamma) \in C^2(\mathbb{R})$ there is a neighbourhood of $u(\beta)$ for which $G'(\gamma) \neq 1$. □

Theorem 3. *If the problem*

$$\begin{cases} F(u, u', u'') = 0 & x \in \Omega \\ u(x) = h(x) & x \in \partial\Omega \end{cases}$$

is nonsingular on $\Omega \setminus \Omega_1$ and $\Omega \setminus \Omega_2$, in the sense that there exists a unique solution to the problem on those domains and that the continuations of these solutions are also unique, then the function $G(\gamma)$ is strictly monotonic.

Proof. It suffices to show that $G(\gamma_1) \neq G(\gamma_2)$ for all $\gamma_1 \neq \gamma_2$.

Let u_1^j solve the problem on $\Omega_1 = [a, \beta]$ with $u_1^j(\beta) = \gamma_j$. Likewise, u_2^j solves the problem on $\Omega_2 = [\alpha, b]$ with $u_2^j(\alpha) = u_1^j(\alpha)$. We proceed by contradiction: suppose $u_2^1(\beta) = u_2^2(\beta)$. Then both u_2^1 and u_2^2 solve the same problem on $\Omega \setminus \Omega_1$. By assumption, this must mean $u_2^1 = u_2^2$ and $u_1^1(\alpha) = u_1^2(\alpha)$.

By a similar argument, this implies u_1^1 and u_1^2 solve the same problem on $\Omega \setminus \Omega_2$. Again by assumption $u_1^1 = u_1^2$ and $\gamma_1 = \gamma_2$. □

Lemma 1. *If $f \in C^2([a, b])$, $f''(x) \neq 0$ for all $x \in (a, b)$ and $f'(c) = 0$ for some $c \in (a, b)$ then there exists $x_0 \in (a, b)$ such that $f(x_0)$ is equal to either $f(a)$ or $f(b)$. Moreover, there is only one local extremum on this interval.*

Proof. We begin by showing that if $f'(c_1) = f'(c_2) = 0$ then $c_1 = c_2$. Consider the region $[c_1, c_2]$. By the mean value theorem there must exist a point $c_3 \in (c_1, c_2)$ such that $f''(c_3) = 0$. This contradicts the assumption that $f''(x) \neq 0$ for all $x \in [a, b]$, unless (c_1, c_2) is the empty set. Therefore, $c_1 = c_2$.

Suppose the statement is not true, that is for all $x_0 \in (a,b)$ $f(x_0) \neq f(a)$ and $f(x_0) \neq f(b)$. Then it must be that either $f(a) < f(x_0) < f(b)$ for all $x_0 \in (a,b)$ or $f(a) > f(x_0) > f(b)$. Then $f(x)$ is monotonic on $[a,c_1]$ and $[c_2,b]$ where $f'(c_1) = f'(c_2) = 0$. We have already shown that $c_1 = c_2 = c$. Thus $f(x)$ is monotonic on $[a,b]$, and $f'(x) \geq 0$ or $f'(x) \leq 0$. WLOG we consider the former.

Let $S_1 = [a,c)$ and $S_2 = (c,b]$, and let $s_1 = \max_{x \in S_1} f'(x) > 0$ and $s_2 = \max_{x \in S_2} f'(x)) > 0$. WLOG, $s_1 > s_2$. Since $f'(x)$ is continuous, there exists $x_1 \in S_1$ such that $f'(x_1) = s_2 = f'(x_2)$ for some $x_2 \in S_2$. Therefore, again by the mean value theorem, $f''(x_3) = 0$ for some $x_3 \in (x_1, x_2)$. This contradicts the assumption that $f''(x) \neq 0$ on the interval. Thus, it must be that the statement is true. □

Corollary 1. *If $G(\gamma) \in C^2$ and $G''(\gamma^*) = 0 \implies G(\gamma^*) = \gamma^*$ then either $G'(\gamma) \neq 1$ for all γ or the problem is singular.*

Proof. Consider $f(\gamma) = G(\gamma) - \gamma$ and the domains $(-\infty, \gamma^*]$ and $[\gamma^*, \infty)$. Apply the previous lemma to $f(\gamma)$, noting that $f(\gamma^*) = 0$ and $\lim_{\pm\gamma \to \infty} f(\gamma) = \pm\infty$. Therefore, if $f'(\gamma_1) = 0$ for some $\gamma_1 \neq \gamma^*$ then $f(\gamma_2) = 0$ for some $\gamma_2 \neq \gamma^*$. This implies $G(\gamma)$ has two fixed points, and therefore the problem has two solutions. □

These results show that, failing to have a singular problem, Newton will fail only when the second derivative of the fixed point iteration changes sign away from the root. To this end, we present the following problem:

$$\begin{cases} u''(x) - \sin(15u(x)) = 0 \quad x \in [-1,1] \\ u(-1) = u(1) = 0. \end{cases} \tag{3.2}$$

The problem is uniquely solved by $u(x) = 0$.

Figure 3.3 shows the function $G(\gamma)$ for equation (3.2) for $\alpha = -\beta = -0.2$ with the space between points equal to 0.01. Also shown is the derivative $G'(\gamma)$, the function $G(\gamma) - \gamma$ and the line $y = \gamma$. The intersection of $G(\gamma)$ and γ represents a fixed point and a root of $G(\gamma) - \gamma$.

As can be seen in the figure, $G'(\gamma) = 1$ for some values of γ, and the derivative of $G(\gamma) - \gamma$ is zero. Thus, Newton-Raphson applied to $G(\gamma) - \gamma$ is not guaranteed to converge. After testing, it appears that Newton-Raphson does converge for all choices of γ tested, albeit slowly when the initial guess is chosen near the points where $G'(\gamma) = 1$. The importance here is the existence of a nonsingular problem for which $G'(\gamma) = 1$.

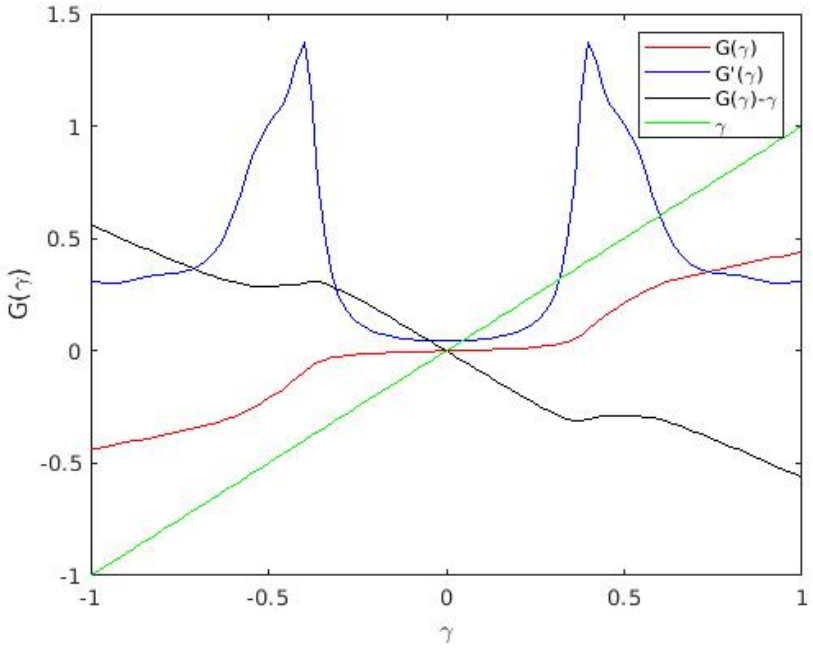


Figure 3.3: The fixed point iterate $G(\gamma)$ and associated functions for equation (3.2). The point where $G(\gamma) = \gamma$ represents the fixed point.

Chapter 4

Continuing thesis work

4.1 Inverse operators

Having found the inverses of differentiation matrices with boundary conditions, we now seek the inverses of more general differential operators. Consider the linear differential operator $\mathcal{L}$:

$$\mathcal{L}u(x) = u^{(m)}(x) + \sum_{n=1}^{m} q_n(x)u^{(m-n)}(x) \tag{4.1}$$

with a fundamental set of solutions to the homogeneous equation $\mathcal{L}u(x) = 0$ represented by $\{P_k(x)\}_{k=1}^{m}$. Let the matrix A be constructed as described in Chapter **??**.

The round-off error in the matrix $D^{(m)}$ increases with N and m. This causes the system to be poorly conditioned, and troublesome to solve. Rather than solve the system directly, we look for a matrix R that acts as a right inverse to A:

$$AR \approx I. \tag{4.2}$$

4.1.1 Construction of the inverse operator

Let the j–th column of R be an N–th degree polynomial $R_j(x)$ evaluated at the Chebyshev points, such that R can be defined element-wise as:

$$R_{ij} = R_j(x_i). \tag{4.3}$$

Lemma 2. *Let A be constructed as in Chapter **??** for the linear differential operator $\mathcal{L}$ and m boundary conditions $\{\mathcal{B}_k\}$. Then $AR = I$ if and only if $R_j(x)$*

satisfy:

$$\mathcal{L}R_j(x_i) = \begin{cases} \delta_{ij} & x_j \notin V \\ 0 & x_j \in V \end{cases}, \qquad x_i \notin V \tag{4.4}$$

$$\mathcal{B}_k R_j(\pm 1) = \begin{cases} 0 & x_j \neq v_k \in V \\ 1 & x_j = v_k \in V \end{cases}. \tag{4.5}$$

Proof. The proof is identical to that for lemma **??**, with m–th order differentiation exchanged for the more general operator $\mathcal{L}$. □

Recall that the functions $\{P_k(x)\}$ are homogeneous solutions for the operator $\mathcal{L}$. We use the following ansatz for the form of $R_j(x)$:

$$R_j(x) = \sum_{k=1}^{m} G_{k,j}(x) P_k(x). \tag{4.6}$$

To find the function $G_{k,j}(x)$, we use variation of parameters. This enforces the following conditions:

$$\sum_{k=1}^{m} G'_{k,j}(x) P_k^{(l)}(x) = 0, \quad l = 0, ..., m-2. \tag{4.7}$$

The first $m-1$ derivatives of $R_j(x)$ are then:

$$\begin{aligned} R'_j(x) &= \sum_{k=1}^{m} G'_{k,j}(x) P_k(x) + G_{k,j}(x) P'_k(x) &&= \sum_{k=1}^{m} G_{k,j}(x) P'_k(x) \\ R_j^{(l)}(x) &= \sum_{k=1}^{m} G'_{k,j}(x) P_k^{(l-1)}(x) + G_{k,j}(x) P_k^{(l)}(x) &&= \sum_{k=1}^{m} G_{k,j}(x) P_k^{(l)}(x), \quad l \leq m-1. \end{aligned} \tag{4.8}$$

This implies:

$$\begin{aligned}\mathcal{L}R_j(x) &= R_j^{(m)}(x) + \sum_{n=1}^{m} q_n(x) R_j^{(m-n)}(x) \\ &= \sum_{k=1}^{m} \left[G'_{k,j}(x) P_k^{(m-1)}(x) + G_{k,j}(x) P_k^{(m)}(x) + \sum_{n=1}^{m} q_n(x) G_{k,j}(x) P_k^{(m-n)}(x) \right] \\ &= \sum_{k=1}^{m} G'_{k,j}(x) P_k^{(m-1)}(x) + G_{k,j}(x) \mathcal{L}P_k(x) \\ &= \sum_{k=1}^{m} G'_{k,j}(x) P_k^{(m-1)}(x), \end{aligned} \tag{4.9}$$

$$\begin{aligned}\mathcal{B}_l R_j(\pm 1) &= \sum_{n=1}^{m} a_n^l R_j^{(m-n)}(\pm 1) \\ &= \sum_{k=1}^{m} \sum_{n=1}^{m} a_n^l G_{k,j}(\pm 1) P_k^{(m-n)}(\pm 1) \\ &= \sum_{k=1}^{m} G_{k,j}(\pm 1) \mathcal{B}_l P_k(\pm 1). \end{aligned} \tag{4.10}$$

Equations (4.4), (4.7) and (4.9) can be combined for a set of conditions on $G_{k,j}(x)$ and $P_k(x)$:

$$\sum_{k=1}^{m} G'_{k,j}(x_i) P_k^{(l)}(x_i) = \begin{cases} 0 & l < m-1 \\ 0 & x_j \in V \\ 0 & x_i \neq x_j,\ x_i \notin V \\ 1 & x_i = x_j,\ x_i \notin V,\ l = m-1. \end{cases} \tag{4.11}$$

By the last two conditions, $G_{k,j}(x)$ is a multiple of a Birkhoff interpolant seen in Chapter **??**. As discussed there, the value of $G'_{k,j}(x)$ can be specified at all but one of the CGL points (**??**). The point at which $G'_{k,j}(x)$ is unknown prescribes the row removal. Thus, to each $G_{k,j}(x)$ we assign the point $v_k \in V$ as the point where $G'_{k,j}(x)$ is unknown.

The conditions on $G_{k,j}(x)$, as defined by equation (4.11) and allowable by the algorithm in Chapter **??**, are:

$$G'_{k,j}(x_i) = \begin{cases} \beta_{k,j} & x_i = x_j \\ 0 & x_i \neq x_j, v_k \end{cases} \tag{4.12}$$

where v_k is that element in V associated with $G_{k,j}(x)$ and $\beta_{k,j}$ is the scalar multiplier of the Birkhoff interpolant. Following the algorithm from Chapter

??, $G_{k,j}(x)$ is found to be:

$$\begin{aligned} G_{k,j}(x) &= \beta_{k,j} \sum_{n=0}^{N-1} b_{nj}^k \partial_x^{-1} T_n(x), \\ b_{nj}^k &= \frac{2}{c_n c_j N} \left(T_n(x_j) - \frac{T_N(x_j)}{T_N(v_k)} T_n(v_k) \right). \end{aligned} \tag{4.13}$$

There is a remaining degree of freedom in $G_{k,j}(x)$: adding any constant will not change any of the conditions $G'_{k,j}(x)$ needs to satisfy (4.12, 4.7). That is, replacing $G_{k,j}(x)$ with $G_{k,j}(x) + C_{k,j}$ for any constant $C_{k,j}$ in the ansatz (4.6) will not change any of the above results.

With $G_{k,j}(x)$ now defined, equation (4.7) enforces:

$$G'_{k,j}(v_k) P_k^{(l)}(v_k) = 0, \ l = 0, ..., m-2, \ k = 1, ..., m. \tag{4.14}$$

As the value of $G'_{k,j}(v_k)$ cannot be specified, this requires $P_k(x)$ be the homogeneous solution satisfying:

$$\mathcal{L} P_k(x) = 0, \ P_k^{(l)}(v_k) = \begin{cases} 0 & l = 0, ..., m-2 \\ 1 & l = m-1 \end{cases}. \tag{4.15}$$

Note the value of $P_k^{(m-1)}(v_k)$ does not need to be 1, but all scalar multipliers can be placed on $G_{k,j}(x)$ and its scalar multiplier $\beta_{k,j}$ (4.12).

This scalar multiplier $\beta_{k,j}$ is currently unknown. Equation (4.11) enforces the following conditions on the values $\beta_{k,j}$:

$$\sum_{k=1}^{m} \beta_{k,j} P_k^{(l)}(x_j) = \begin{cases} 1 & l = m-1 \\ 0 & l = 0, ..., m-2 \end{cases} \tag{4.16}$$

for $x_j \notin V$. The system for equation (4.16) can be written as:

$$\begin{bmatrix} P_1(x_j) & \dots & P_m(x_j) \\ \vdots & \ddots & \vdots \\ P_1^{(m-1)}(x_j) & \dots & P_m^{(m-1)}(x_j) \end{bmatrix} \begin{bmatrix} \beta_{1,j} \\ \vdots \\ \beta_{m,j} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{4.17}$$

The system for this matrix is different for each j, meaning $(N-m)$ such $m \times m$ systems need to be solved in order to construct R.

The function $R_j(x)$ can be written in its entirety as:

$$R_j(x) = \sum_{k=1}^{m} (C_{k,j} + G_{k,j}(x)) P_k(x), \tag{4.18}$$

where $C_{k,j}$ is the arbitrary constant added to $G_{k,j}(x)$. Regardless of the values of $C_{k,j}$ this formula enforces $\mathcal{L} R_j(x_i) = \delta_{ij}$ if $x_j, x_i \notin V$. All that remains are the boundary conditions: $\mathcal{B}_k R_j(\pm 1) = 0$ for all $k = 1, ..., m$.

Equation (4.10) requires that $\mathcal{B}_s R_j(\pm 1) = \sum_{k=1}^{m} C_{k,j}\mathcal{B}_s P_k(\pm 1) + G_{k,j}(\pm 1)\mathcal{B}_s P_k(\pm 1)$. As in Chapter **??**, this leads to two systems of equations:

$$\begin{bmatrix} \mathcal{B}_1 P_1(1) & \dots & \mathcal{B}_1 P_m(1) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_{k_0} P_1(1) & \dots & \mathcal{B}_{k_0} P_m(1) \end{bmatrix} \begin{bmatrix} C_{1,j} \\ \vdots \\ C_{m,j} \end{bmatrix} = - \begin{bmatrix} \sum_{k=1}^{m} \mathcal{B}_1 P_k(1) G_{k,j}(1) \\ \vdots \\ \sum_{k=1}^{m} \mathcal{B}_{k_0} P_k(1) G_{k,j}(1) \end{bmatrix}$$
$$\begin{bmatrix} \mathcal{B}_{k_0+1} P_1(-1) & \dots & \mathcal{B}_{k_0+1} P_m(-1) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_m P_1(-1) & \dots & \mathcal{B}_m P_m(-1) \end{bmatrix} \begin{bmatrix} C_{1,j} \\ \vdots \\ C_{m,j} \end{bmatrix} = - \begin{bmatrix} \sum_{k=1}^{m} \mathcal{B}_{k_0+1} P_k(-1) G_{k,j}(-1) \\ \vdots \\ \sum_{k=1}^{m} \mathcal{B}_m P_k(-1) G_{k,j}(-1) \end{bmatrix}. \tag{4.19}$$

For the function $R_j(x)$ such that $x_j = v_k \in V$, $G_{k,j}(x) = 0$ and the systems in equation (4.19) have their right hand sides replaced by portions of the identity matrix.

4.1.2 The Wronskian

We use the particular set of homogeneous solutions, $\{P_k(x) \mid P_k(x) \text{ satisfies equation (4.15)}\}$, to construct R. Note that given any fundamental set of solutions, $\{\hat{P}_n(x)\}$, we can always calculate $\{P_k(x)\}$ by using the following system for each k:

$$P_k(x) = \sum_{n=1}^{m} \gamma_{kn} \hat{P}_n(x), \quad \begin{bmatrix} \hat{P}_1(v_k) & \dots & \hat{P}_m(v_k) \\ \vdots & \ddots & \vdots \\ \hat{P}_1^{(m-1)}(v_k) & \dots & \hat{P}_m^{(m-1)}(v_k) \end{bmatrix} \begin{bmatrix} \gamma_{k1} \\ \vdots \\ \gamma_{km} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{4.20}$$

Notice the similarities between this system and that for $\beta_{k,j}$ (4.17).

The matrices presented in equations (4.17) and (4.20) have well-known inverses that rely on the Wronskians of the homogeneous solutions. The Wronskian of a set of functions $\{f_k(x)\}_{k=1}^{n}$, denoted here as $W(\{f_k\}, x)$, is itself a function defined as the determinant of the matrix:

$$\begin{bmatrix} f_1(x) & \dots & f_n(x) \\ \vdots & \ddots & \vdots \\ f_1^{(n-1)}(x) & \dots & f_n^{(n-1)}(x) \end{bmatrix}. \tag{4.21}$$

Using Cramer's rule, the solution to the system:

$$\begin{bmatrix} f_1(x) & \dots & f_n(x) \\ \vdots & \ddots & \vdots \\ f_1^{(n-1)}(x) & \dots & f_n^{(n-1)}(x) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{4.22}$$

is equal to:

$$a_j = \frac{(-1)^{j+n} W(\{f_k\}_{k \neq j}; x)}{W(\{f_k\}; x)}. \tag{4.23}$$

Therefore, the systems in equations (4.17) and (4.20) do not need to be solved if the Wronskians are calculated instead.

To simplify calculating the Wronskians, one can apply Abel's identity [**?**, **?**]: if the functions $\{f_k\}_{k=1}^n$ form the fundamental solution set to a linear operator $\mathcal{L}$ such that $\mathcal{L}u(x) = u^{(n)}(x) + \sum_{k=1}^n q_k(x)u^{(n-k)}(x)$, then the Wronskian can be expressed as:

$$W(\{f_k\};x) = W(\{f_k\};0)\exp\left(-\int_0^x q_1(s)ds\right). \tag{4.24}$$

It is then a matter of finding the linear operator $\mathcal{L}$ for which the functions form a fundamental solution set.

For the problem at hand $\mathcal{L}$ is known for $\{\hat{P}_k\}$. It is not known for $\{\hat{P}_k\}_{k\neq j}$. Note that each $P_j(x)$ is the fundamental solution of a first order linear operator $\mathcal{L}_j$: $\mathcal{L}_jP_j(x) = P_j'(x) + r_j(x)P_j(x) = 0$, so that $P_j(x) = \alpha\exp\left(-\int r_j\right)$. Moreover, $\mathcal{L}$ can be written as a product of several of these first order linear operators: $\mathcal{L} = \tilde{\mathcal{L}}_m\ldots\tilde{\mathcal{L}}_1$. The coefficient function $q_1(x)$ is then $\sum\tilde{r}_j(x)$.

No matter what each of the $\tilde{\mathcal{L}}_j$ are, $\tilde{\mathcal{L}}_1$ is always one of the $\mathcal{L}_j$, defined for each $P_j(x)$. Suppose $\tilde{\mathcal{L}}_1 = \mathcal{L}_j$ in a particular formulation and we search for the Wronskian of the set $\{P_k\}_{k\neq j}$. Define $\bar{\mathcal{L}}_j$ as $\tilde{\mathcal{L}}_m\ldots\tilde{\mathcal{L}}_2$, in essence $\mathcal{L}$ without $\mathcal{L}_j$. Since the fundamental solution set is linearly independent, $P_j(x)$ is not a homogeneous solution of $\bar{\mathcal{L}}_j$. The second order coefficient function for this is simply $q_1(x) - r_1(x)$. However, the fundamental solution set of $\bar{\mathcal{L}}_j$ is not $\{P_k\}_{k\neq j}$ but $\{\mathcal{L}_jP_k\}_{k\neq j}$.

If the $\tilde{\mathcal{L}}_j$ commute, such as in the case of constant coefficient problems, then we do have that the fundamental solution set of $\bar{\mathcal{L}}_j$ is indeed $\{P_k\}_{k\neq j}$, as we can commute $\mathcal{L}_j$ to the front of the formulation and then remove it. In this case it is straightforward to show:

$$\frac{W(\{P_k\}_{k\neq j};x)}{W(\{P_k\};x)} = \frac{W(\{P_k\}_{k\neq j};0)}{W(\{P_k\};0)}\exp\left(\int_0^x r_1(s)ds\right) = \frac{W(\{P_k\}_{k\neq j};0)}{W(\{P_k\};0)}\frac{P_j(0)}{P_j(x)}.$$

We apply this to equation (4.20):

$$P_k(x) = \sum_{n=1}^m(-1)^{n+m}\frac{W(\{\hat{P}_j\}_{j\neq n};0)}{W(\{\hat{P}_j\};0)}\frac{\hat{P}_n(x)\hat{P}_n(0)}{\hat{P}_n(v_k)}.$$

Consider $W(\{P_n\}_{n\neq k};x)$ as the determinant of the product of two rectangular matrices:

$$\left[P_n^{(l)}(x)\right]_{n\neq k,l=0,\ldots,m-2} = \left[\hat{P}_j^{(l)}(x)\right]_{l=0,\ldots,m-2}\left[\gamma_{n,j}\right]_{n\neq k},$$

$$\gamma_{n,j} = (-1)^{j+m}\frac{W(\{\hat{P}_i\}_{i\neq j};0)}{W(\{\hat{P}_i\};0)}\frac{\hat{P}_j(0)}{\hat{P}_j(v_n)} = \frac{\omega_j}{\hat{P}_j(v_n)}.$$

The Cauchy-Binet formula then provides this determinant in terms of determinants already calculated:

$$\begin{aligned}(-1)^{k+m}\frac{\left|P_n^{(l)}(x)\right|_{n\neq k,l<m-1}}{\left|P_n^{(l)}(x)\right|} &= \sum_{i=1}^{m}(-1)^{k+m}\frac{\left|\hat{P}_j^{(l)}(x)\right|_{j\neq i,l<m-1}}{\left|\hat{P}_n^{(l)}(x)\right|}\left|\tfrac{\omega_j}{\hat{P}_j(v_n)}\right|_{n\neq k,j\neq i}\\ &= \sum_{i=1}^{m}(-1)^{k-i}\frac{\omega_i}{\hat{P}_i(x)}\Pi_{j\neq i}\omega_j\left|\tfrac{1}{\hat{P}_j(v_n)}\right|_{n\neq k,j\neq i}\\ &= \left(\Pi_{j=1}^{m}\omega_j\right)\sum_{i=1}^{m}\frac{(-1)^{k-i}}{\hat{P}_i(x)}\left|\tfrac{1}{\hat{P}_j(v_n)}\right|_{n\neq k,j\neq i}\end{aligned}$$

4.2 Wronskians

Lemma 3. *Wronskians of exponentials*

$$W(\{e^{\lambda_k x}\}_{k=1}^m;x) = \begin{vmatrix}1 & \dots & 1\\ \lambda_1 & \dots & \lambda_m\\ \vdots & \ddots & \vdots\\ \lambda_1^{m-1} & \dots & \lambda_m^{m-1}\end{vmatrix} e^{-x\sum_{k=1}^m \lambda_k}.$$

Proof. The case of any two λ_k being equal is trivially true as both sides are necessarily zero. Therefore, without loss of generality, $\lambda_k \neq \lambda_i$ for all $k \neq i$.

Consider a linear operator $\mathcal{L}$ such that $\mathcal{L} = \sum_{j=0}^m a_j u^{(j)}(x)$. It is straightforward to see that $\mathcal{L}e^{\lambda_k x} = e^{\lambda_k x}\sum_{j=0}^m a_j\lambda_k^j$. Therefore, $e^{\lambda_k x}$ is a homogeneous solution of $\mathcal{L}$ if λ_k is a root of the polynomial $p(x) = \sum_{j=0}^m a_j x^j$.

Let $p(x)$ be the polynomial with roots λ_k:

$$p(x) = \Pi_{k=1}^m (x-\lambda_k) = \sum_{j=0}^m a_j x^j.$$

The set $\{e^{\lambda_k x}\}_{k=1}^m$ is then the fundamental solution set of the linear operator with constant coefficients a_j. By Abel's identity this provides that $W(\{e^{\lambda_k x}\}_{k=1}^m;x) = W(\{e^{\lambda_k x}\}_{k=1}^m;0)e^{-\int_0^x a_{m-1}/a_m ds}$. Based on the above formula for $p(x)$, $a_m = 1$ and $a_{m-1} = \sum_{k=1}^m \lambda_k$. The remainder follows from the definition of $W(\{e^{\lambda_k x}\}_{k=1}^m;0)$.

(nb: this lemma and proof likely already exists in literature) □

Lemma 4. *Wronskians of polynomials*

$$\begin{aligned}&(i) && W(\{x^k/k!\}_{k=0}^m;x) = 1\\ &(iii) && W(\{x^k/k!\}_{k=0,k\neq j}^m;x) = W(\{x^k/k!\}_{k=1}^{m-j};x)\\ &(iii) && W(\{x^k/k!\}_{k=1}^m;x) = x^m/m!\end{aligned}$$

Proof. (i)

$$W(\{x^k/k!\}_{k=0}^m; x) = \begin{vmatrix} 1 & x & x^2/2 & \dots & x^m/m! \\ 0 & 1 & x & \dots & x^{m-1}/(m-1)! \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{vmatrix} = 1$$

(ii)

$$W(\{x^k/k!\}_{k=0,k\neq j}^m; x) =$$
$$\begin{vmatrix} 1 & x & \dots & x^{j-1}/(j-1)! & x^{j+1}/(j+1)! & \dots & x^m/m! \\ 0 & 1 & \dots & x^{j-2}/(j-2)! & x^j/j! & \dots & x^{m-1}/(m-1)! \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \dots & 1 & x^2/2! & \dots & x^{m-j+1}/(m-j+1)! \\ & & & 0 & x & & x^{m-j}/(m-j)! \\ & & & 0 & 1 & & x^{m-j-1}/(m-j-1)! \\ & & & \vdots & \vdots & \ddots & \vdots \\ & & & 0 & 0 & \dots & 1 \end{vmatrix}$$
$$= W(\{x^k/k!\}_{k=0}^{j-1}; x) W(\{x^k/k!\}_{k=1}^{m-j}; x)$$
$$= W(\{x^k/k!\}_{k=1}^{m-j}; x).$$

(iii) The statement is trivially true for $m = 1$. Apply strong induction by supposing it is true for $n < m$. Calculate the Wronskian by expanding through the top row of the matrix:

$$W(\{x^k/k!\}_{k=1}^m; x) = \begin{vmatrix} x & \dots & x^m/m! \\ 1 & \ddots & \vdots \\ 0 & \ddots & \\ \vdots & \ddots & \\ 0 & \dots & 1 \end{vmatrix}$$
$$= \sum_{n=1}^m (-1)^{n+1} x^n/n! W(\{x^k/k!\}_{k=0,k\neq n}^{m-1}; x)$$
$$= \sum_{n=1}^m (-1)^{n+1} x^n/n! W(\{x^k/k!\}_{k=1}^{m-n}; x) \quad \text{by } (ii)$$
$$= \sum_{n=1}^m (-1)^{n+1} x^n/n! x^{m-n}/(m-n)! \quad \text{by our induction hypothesis}$$
$$= x^m/m! \sum_{n=1}^m (-1)^{n+1} \binom{m}{n}$$
$$= x^m/m! \quad \text{by an identity of the binomial coefficients. (nb:cite)}$$

□

Lemma 5.

$$W(\{f_k g\}_{k=1}^m; x) = g^m W(\{f_k\}_{k=1}^m; x)$$

Proof. It is trivially true for $m = 1$. Apply strong induction by supposing it is true for $n < m$, for any set of functions. Calculate the Wronskian by expanding through the bottom row of the matrix:

$$\begin{aligned}
&W(\{f_k g\}_{k=1}^m; x) = \\
&\quad \begin{vmatrix} f_1 g & \cdots & f_m g \\ \vdots & & \vdots \\ \sum_{j=0}^{m-1} \binom{m-1}{j} f_1^{(m-1-j)} g^{(j)} & \cdots & \sum_{j=0}^{m-1} \binom{m-1}{j} f_m^{(m-1-j)} g^{(j)} \end{vmatrix} \\
&\quad = \sum_{i=1}^{m} (-1)^{i+m} W(\{f_k g\}_{k\neq i}; x) \sum_{j=0}^{m-1} \binom{m-1}{j} f_i^{(m-1-j)} g^{(j)} \\
&\quad = \sum_{i=1}^{m} (-1)^{i+m} g^{m-1} W(\{f_k\}_{k\neq i}; x) \sum_{j=0}^{m-1} \binom{m-1}{j} f_i^{(m-1-j)} g^{(j)} \\
&\quad\quad \text{by our induction hypothesis} \\
&\quad = g^{m-1} \sum_{j=0}^{m-1} \binom{m-1}{j} g^{(j)} \sum_{i=1}^{m} (-1)^{i+m} W(\{f_k\}_{k\neq i}; x) f_i^{(m-1-j)} \\
&\quad = g^{m-1} \sum_{j=0}^{m-1} \binom{m-1}{j} g^{(j)} \begin{vmatrix} f_1 & \cdots & f_m \\ \vdots & & \vdots \\ f_1^{(m-2)} & \cdots & f_m^{(m-2)} \\ f_1^{(m-1-j)} & \cdots & f_m^{(m-1-j)} \end{vmatrix} \\
&\quad = g^{m-1} \sum_{j=0}^{m-1} \binom{m-1}{j} g^{(j)} \times \begin{cases} 0 & j \neq 0 \\ W(\{f_k\}_{k=1}^m; x) & j = 0 \end{cases} \\
&\quad = g^m W(\{f_k\}_{k=1}^m; x).
\end{aligned}$$

□

Lemma 6. *Wronskian of linear sums of functions*

$$W(\{\sum_{k=1}^{m} a_{k,j} f_k(x)\}_{j=1}^{m-1}; x) = \sum_{n=1}^{m} W(\{f_k\}_{k\neq n}; x) \begin{vmatrix} a_{1,1} & \cdots & a_{1,m-1} \\ \vdots & & \vdots \\ a_{n-1,1} & \cdots & a_{n-1,m-1} \\ a_{n+1,1} & \cdots & a_{n+1,m-1} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,m-1} \end{vmatrix}$$

Proof. Let $g_j(x) = \sum_{k=1}^{m} a_{k,j} f_k(x)$.

$$
\begin{aligned}
W(\{g_j\}_{j=1}^{m-1}; x) &= \begin{vmatrix} g_1 & \dots & g_{m-1} \\ \vdots & & \vdots \\ g_1^{(m-2)} & \dots & g_{m-1}^{(m-2)} \end{vmatrix} \\
&= \det\left(\begin{bmatrix} f_1 & \dots & f_m \\ \vdots & & \vdots \\ f_1^{(m-2)} & \dots & f_m^{(m-2)} \end{bmatrix} \begin{bmatrix} a_{1,1} & \dots & a_{1,m-1} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,m-1} \end{bmatrix} \right) \\
&= \sum_{n=1}^{m-1} W(\{f_k\}_{k\neq n}; x) \begin{vmatrix} a_{1,1} & \dots & a_{1,m-1} \\ \vdots & & \vdots \\ a_{n-1,1} & \dots & a_{n-1,m-1} \\ a_{n+1,1} & \dots & a_{n+1,m-1} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,m-1} \end{vmatrix}
\end{aligned}
$$

by the Cauchy-Binet formula.

□

Lemma 7. *Let f_k represent the vector such that the j–th element of f_k is equal to the j–th derivative of the function $x^k/k!e^{\lambda x}$ ($\lambda \neq 0$). Then $f_k = \sum_{n=0}^{k} x^{k-n}/(k-n)!\tilde{f}_n$, where $\tilde{f}_n$ is defined elementwise as*

$$
\tilde{f}_{n,j} = e^{\lambda x}\lambda^{j-n} \begin{cases} 0 & j < n, \\ \binom{j}{n} & j \geq n. \end{cases}
$$

Proof. The j–th derivative of the function $x^k/k!e^{\lambda x}$ is straightforward to calculate:

$$
f_{k,j} = e^{\lambda x} \sum_{n=0}^{\min\{k,j\}} \binom{j}{n} x^{k-n}/(k-n)!\lambda^{j-n}.
$$

The remainder is a matter of padding with zeros and indexing in the desired fashion. □

Lemma 8. *Let f_k and $\tilde{f}_k$ be defined as in the previous lemma. Suppose we are concerned with the determinant of a matrix that contains, as columns, the set $\{f_k\}_{k=0,k\neq j}^{m}$:*

$$
\begin{vmatrix} \dots & f_0 & \dots & f_{j-1} & f_{j+1} & \dots & f_m & \dots \end{vmatrix}.
$$

Then the determinant can be written as:

$$\sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad \ldots\right|.$$

Proof. The statement is true for $m = j$ by inspection. We suppose the statement is true for m and prove it is true for $m+1$. Note that we could also proceed by supposing it is true for j and prove it is true for $j-1$.

Moving to $m+1$ means the inclusion of the column f_{m+1}. To begin, we can suppose this column is part of the section of the matrix not represented by these columns:

$$\left|\ldots \quad f_0 \quad \ldots \quad f_{j-1} \quad f_{j+1} \quad \ldots \quad f_m \quad f_{m+1} \quad \ldots\right| =$$

$$\sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad f_{m+1} \quad \ldots\right| =$$

$$\sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad \frac{x^{m+1-n}}{(m+1-n)!}\tilde{f}_n + \tilde{f}_{m+1} \quad \ldots\right|$$

by the previous lemma

$$= \sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad \tilde{f}_{m+1} \ldots\right| +$$

$$\sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \frac{x^{m+1-n}}{(m+1-n)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad \tilde{f}_n \quad \ldots\right|$$

$$= \sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad \tilde{f}_{m+1} \ldots\right| +$$

$$\frac{x^{m+1-j}}{(m+1-j)!} \sum_{n=j}^{m} (-1)^{m+1-n} \binom{m+1-j}{n-j} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_m \quad \ldots\right|$$

$$= \sum_{n=j}^{m} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_m \quad \tilde{f}_{m+1} \ldots\right| +$$

$$\frac{x^{m+1-j}}{(m+1-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_m \quad \ldots\right| \sum_{n=0}^{m-j} (-1)^{m+1-j-n} \binom{m+1-j}{n} =$$

$$\sum_{n=j}^{m+1} \frac{x^{n-j}}{(n-j)!} \left|\ldots \quad \tilde{f}_0 \quad \ldots \quad \tilde{f}_{n-1} \quad \tilde{f}_{n+1} \quad \ldots \quad \tilde{f}_{m+1} \quad \ldots\right|$$

by the same property of the binomial coefficient used earlier in lemma 4. □

Theorem 4. *Wronskians for constant coefficient operators Let* $\mathcal{L}u(x) = \sum_{k=0}^{m} a_k u^{(k)}(x)$. *Let* $\{\lambda_j\}_{j=1}^{M}$ *be the roots of the polynomial* $p(x) = \sum_{k=0}^{m} a_k x^k$, *each with multiplicity* m_j ($M = \sum m_j$). *Let* $E_j = \{x^k/k! e^{\lambda_j x}\}_{k=0}^{m_j-1}$ *and* $E = \cup_{j=1}^{M} E_j$. *Note*

that E is the fundamental solution set of $\mathcal{L}$. Let $\{\tilde{f}_k(\lambda_j)\}_{k=0,j=1}^{m_j-1,M}$ be the vector valued functions defined in lemma 7 for the function $\frac{x^k}{k!}e^{\lambda_j x}$. Then the Wronskian for $E \setminus \{x^n/n!e^{\lambda_j x}\}$ is equal to:

$$W(E \setminus \{x^n/n!e^{\lambda_j x}\}; x) =$$
$$\sum_{k=n}^{m_j} \frac{x^{k-n}}{(k-n)!} \left|\tilde{f}_0(\lambda_1) \quad \dots \quad \tilde{f}_{m_1-1}(\lambda_1) \quad \tilde{f}_0(\lambda_2) \quad \dots \quad \tilde{f}_{k-1}(\lambda_j) \quad \tilde{f}_{k+1}(\lambda_j) \quad \dots \quad \tilde{f}_{m_M-1}(\lambda_M)\right|.$$

Proof. The theorem has been proven for the case where $m_j = 1$ for all j and $M = m$ in lemma 3. The case where $M = 1$ can be covered by combining lemmas 4 and 5:

$$W(\{x^k/k!e^{\lambda_1 x}\}_{k\neq j}; x) = e^{(m-1)\lambda_j x} W(\{x^k/k!\}_{k\neq j}; x).$$

The special case where $\lambda_j = 0$ for all j is treated in lemma 4. For $1 < M < m$ we can make use of lemmas 7 and 8.

For $1 < M < m$ the desired Wronskians can be written as:

$$W(E \setminus \{x^n/n!e^{\lambda_j x}\}; x) =$$
$$\left|f_0(\lambda_1) \quad \dots \quad f_{m_1-1}(\lambda_1) \quad f_0(\lambda_2) \quad \dots \quad f_{n-1}(\lambda_j) \quad f_{n+1}(\lambda_j) \quad \dots \quad f_{m_M-1}(\lambda_M)\right|.$$

By lemma 7 and properties of the determinant, we can replace $f_k(\lambda_i)$ by $\tilde{f}_k(\lambda_i)$ except for $k > n$, $i = j$. Lemma 8 then defines the remaining simplifications:

$$W(E \setminus \{x^n/n!e^{\lambda_j x}\}; x) =$$
$$\left|\tilde{f}_0(\lambda_1) \quad \dots \quad \tilde{f}_{m_1-1}(\lambda_1) \quad \tilde{f}_0(\lambda_2) \quad \dots \quad \tilde{f}_{n-1}(\lambda_j) \quad f_{n+1}(\lambda_j) \quad \dots \quad \tilde{f}_{m_M-1}(\lambda_M)\right| =$$
$$\sum_{k=n}^{m_j} \frac{x^{k-n}}{(k-n)!} \left|\tilde{f}_0(\lambda_1) \quad \dots \quad \tilde{f}_{m_1-1}(\lambda_1) \quad \tilde{f}_0(\lambda_2) \quad \dots \quad \tilde{f}_{k-1}(\lambda_j) \quad \tilde{f}_{k+1}(\lambda_j) \quad \dots \quad \tilde{f}_{m_M-1}(\lambda_M)\right|.$$

□

It is informative to write out this sum and the determinants in full:

$$W(E\setminus\{x^n/n!e^{\lambda_j x}\};x)\exp\left(-\sum_{k=1}^{M}m_k\lambda_k x+\lambda_j x\right)=$$

$$\begin{vmatrix} 1 & 0 & & 1 & \dots & 0 & 0 & \dots & 1 & \dots & 0\\ \lambda_1 & 1 & \dots & \lambda_j & \ddots & & & & \lambda_M & \ddots & 0\\ \lambda_1^2 & 2\lambda_2 & \ddots & & & \vdots & & & \vdots & & \vdots\\ & \vdots & & & & & & & & & \\ & & & & & 1 & 0 & & & & \\ & & & & & n\lambda_j & 0 & & & & \\ & & & & & \binom{n+1}{n-1}\lambda_j^2 & 1 & & & & \\ & & & & & \vdots & \vdots & & & & \end{vmatrix} +$$

$$x\begin{vmatrix} 1 & 0 & & 1 & \dots & 0 & 0 & \dots & 1 & \dots & 0\\ \lambda_1 & 1 & \dots & \lambda_j & \ddots & & & & \lambda_M & \ddots & 0\\ \lambda_1^2 & 2\lambda_2 & \ddots & & & \vdots & & & \vdots & & \vdots\\ & \vdots & & & & & & & & & \\ & & & & & 1 & 0 & & & & \\ & & & & & (n+1)\lambda_j & 0 & & & & \\ & & & & & \binom{n+2}{n}\lambda_j^2 & 1 & & & & \\ & & & & & \vdots & \vdots & & & & \end{vmatrix} + \dots$$

$$+\frac{x^{m_j-n}}{(m_j-n)!}\begin{vmatrix} 1 & 0 & & 1 & \dots & 0 & 0 & \dots & 1 & \dots & 0\\ \lambda_1 & 1 & \dots & \lambda_j & \ddots & & & & \lambda_M & \ddots & 0\\ \lambda_1^2 & 2\lambda_2 & \ddots & & & \vdots & & & \vdots & & \vdots\\ & \vdots & & & & & & & & & \\ & & & & & 1 & \lambda_{j+1}^{m_j-1} & & & & \\ & & & & & m_j\lambda_j & \lambda_{j+1}^{m_j} & & & & \\ & & & & & \binom{m_j+1}{m_j-1}\lambda_j^2 & \lambda_{j+1}^{m_j+1} & & & & \\ & & & & & \vdots & \vdots & & & & \end{vmatrix}.$$

The exponential function present in each determinant is moved to the left hand side for ease of notation. While this may look large in practice the determinants are for matrices of size $(m-1)\times(m-1)$ and the order of the polynomial is at most $m-1$, where m is the order of the linear operator $\mathcal{L}$. Certain unusual problems may have order 5 or higher but for most problems routinely encountered we can expect $m\leq 4$.

4.2.1 Example

To illustrate the application of this work we consider a problem with $m = 4$ and two roots, each with multiplicity two. This gives $E = \{e^{\lambda_1 x}, xe^{\lambda_1 x}, e^{\lambda_2 x}, xe^{\lambda_2 x}\}$. From Abel's identity and lemma 7 we know

$$W(E;x) = \begin{vmatrix} 1 & 0 & 1 & 0 \\ \lambda_1 & 1 & \lambda_2 & 1 \\ \lambda_1^2 & 2\lambda_1 & \lambda_2^2 & 2\lambda_2 \\ \lambda_1^3 & 3\lambda_1^2 & \lambda_2^3 & 3\lambda_2 \end{vmatrix} \exp\left((2\lambda_1 + 2\lambda_2)x\right).$$

Define the sets $\hat{E}_{n,j} = E \setminus \{x^n/n!e^{\lambda_j x}\}$. To calculate the IOM we need the functions $W(\hat{E}_{n,j};x)$, of which there are four:

$$W(\hat{E}_{1,1};x)\exp\left(-(\lambda_1+2\lambda_2)x\right) = \begin{vmatrix} 0 & 1 & 0 \\ 1 & \lambda_2 & 1 \\ 2\lambda_1 & \lambda_2^2 & 2\lambda_2 \end{vmatrix} + x \begin{vmatrix} 1 & 1 & 0 \\ \lambda_1 & \lambda_2 & 1 \\ \lambda_1^2 & \lambda_2^2 & 2\lambda_2 \end{vmatrix},$$

$$W(\hat{E}_{2,1};x)\exp\left(-(\lambda_1+2\lambda_2)x\right) = \begin{vmatrix} 1 & 1 & 0 \\ \lambda_1 & \lambda_2 & 1 \\ \lambda_1^2 & \lambda_2^2 & 2\lambda_2 \end{vmatrix},$$

$$W(\hat{E}_{1,2};x)\exp\left(-(2\lambda_1+\lambda_2)x\right) = \begin{vmatrix} 1 & 0 & 0 \\ \lambda_1 & 1 & 1 \\ \lambda_1^2 & \lambda_1 & \lambda_2 \end{vmatrix} + x \begin{vmatrix} 1 & 0 & 1 \\ \lambda_1 & 1 & \lambda_2 \\ \lambda_1^2 & 2\lambda_1 & \lambda_2^2 \end{vmatrix},$$

$$W(\hat{E}_{2,2};x)\exp\left(-(2\lambda_1+\lambda_2)x\right) = \begin{vmatrix} 1 & 0 & 1 \\ \lambda_1 & 1 & \lambda_2 \\ \lambda_1^2 & 2\lambda_1 & \lambda_2^2 \end{vmatrix}.$$

Note that all of the 3×3 determinants could be calculated as part of calculating the 4×4 determinant of $W(E;x)$ if one expands along the bottom row. As such, calculating the other four Wronskians should be no more than $\mathcal{O}(m)$ additional operations.

4.3 Algorithm for IOM for constant coefficients

Step 1: Identify the roots and their multiplicities $\{\lambda_k; m_k\}_{k=1}^{M}$ of the polynomial associated with the linear operator.

Step 2: Calculate the 'sub'–Wronskians, $W(E \setminus \{x^n/n!e^{\lambda_k x}\};x)$, using the formulas of lemma 4 and theorem 4 . Store the $(m-1)\times(m-1)$ sized determinants for later use.

Step 3: Divide the 'sub'–Wronskians by $W(E;x)$, using the previously stored determinants to aid the calculation. Multiply by -1 where necessary to form the functions $\tilde{\omega}_i(x)$.

Step 4: Evaluate these new $\tilde{\omega}_i(x)$ at the points $v_k \in V$ to find the coefficients γ_{kn} of equation (4.20).

Step 5: Derive the functions of the set E and use the coefficients γ_{kn} to create the desired fundamental solution set $\{P_k(x)\}$.

Step 6: Apply lemma 6 to get the Wronskians of this fundamental solution set.

Step 7: Use equations (4.17,4.23) to get the coefficients $\beta_{k,j}$.

Step 8: Calculate the Birkhoff interpolants $G_{k,j}(x)$ with equation (4.13).

Step 9: Calculate the constants $C_{k,j}$. Currently only a brute-force approach is provided, but it is expected that there are simplifications to be made given constant coefficients.

Step 10: Form the functions $R_j(x)$ and by extension the IOM.

Bibliography

[1] James P Abbott and Richard P Brent. Fast local convergence with single and multistep methods for nonlinear equations. *The ANZIAM Journal*, 19(2):173–199, 1975.

[2] Franklin H Branin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, 16(5):504–522, 1972.

[3] RP Brent. On the davidenko-branin method for solving simultaneous nonlinear equations. *IBM Journal of Research and Development*, 16(4):434–436, 1972.

[4] DF Davidenko. On a new method of numerical solution of systems of nonlinear equations. In *Dokl. Akad. Nauk SSSR*, volume 88, pages 601–602, 1953.

[5] BA Galanov and SN Malakhovskaya. Realization of a variation of the relaxation method. *Ukrainian Mathematical Journal*, 20(5):503–508, 1968.

[6] Mark Konstantinovich Gavurin. Nonlinear functional equations and continuous analogues of iteration methods. *Izvestiya Vysshikh Uchebnykh Zavedenii. Matematika*, (5):18–31, 1958.

[7] Raphael Hauser and Jelena Nedic. The continuous newton–raphson method can look ahead. *SIAM Journal on Optimization*, 15(3):915–925, 2005.

[8] Dietmar Saupe. Discrete versus continuous newtons method: a case study. In *Newtons Method and Dynamical Systems*, pages 59–80. Springer, 1988.