Numerical Analysis Qualifying Exam Study Sheet

Trevor Loe

February 28, 2025

This document contains a brief overview of all the topics that I have found may appear on the numerical analysis qualifying exam. I have tried to organize the topics by similarlity. The list of topics was started based off the UCLA qual website `https://ww3.math.ucla.edu/qualifying-exam-dates/` and appended as past qual problems were done.

Past numerical analysis qualifying exams can be found at `https://ww3.math.ucla.edu/past-qualifying-exams/`

Much of the material on undergraduate numerical analysis was taken from [4]. Material on numerical methods for ODEs was mainly taken from [1],[2], and [3]. Material for numerical methods on the one-way wave equation is taken from [6]. Material on conservation laws is taken from Yuxuan Liu's notes and [5]

1 Interpolation

1.1 Lagrange Interpolation

If $x_0, ..., x_n$ are $n+1$ distinct real numbers and f is defined on those numbers, then there is a unique, degree-n polynomial P such that

$$P(x_i) = f(x_i)$$

for all $i = 0, ..., n$. The **n'th Lagrange Interpolating Polynomial** is given by

$$\begin{aligned} L_{n,k}(x) &= \frac{(x-x_0)(x-x_1)...(x-x_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_0)...(x_k-x_{k-1})(x_k-x_{k+1})...(x_k-x_n)} \\ &= \prod_{i=0, i\neq k} \frac{(x-x_i)}{(x_k-x_i)} \end{aligned}$$

The unique polynomial P is then given by

$$P(x) = f(x_0)L_{n,0}(x) + f(x_1)L_{n,1}(x) + ... + f(x_n)L_{n,n}(x)$$

1.2 Divided Differences

The divided differences for a function f and real numbers $x_0, ..., x_n$ is defined by

$$f[x_i] = f(x_i)$$

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

and the kth divided difference is given by

$$f[x_i, x_{i+1}, ..., x_{i+k}] = \frac{f[x_{i+1}, ..., x_{i+k}] - f[x_i, ..., x_{i+k-1}]}{x_{i+k} - x_i}$$

If P is the polynomial made from the Lagrange interpolating polynomials above, then (Newton's Divided Difference Formula)

$$P(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, ..., x_k](x-x_0)...(x-x_{k-1})$$

If $f \in C^n[a,b]$ then there exists a number $\xi \in (a,b)$ such that

$$f[x_0, ..., x_n] = \frac{f^{(n)}(\xi)}{n!}$$

1.3 Cubic Spline Interpolation

For a function f defined on $[a, b]$ and a set of notes $a = x_0 < ... < x_n = b$ the **cubic spline** interpolation is the piecewise-cubic function S such that

1. On each sub-interval $[x_j, x_{j+1}]$, S is a cubic function, denoted S_j
2. $S_j(x_j) = f(x_j)$, $S_j(x_{j+1}) = f(x_{j+1})$
3. $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$
4. $S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$
5. $S''_j(x_{j+1}) = S''_{j+1}(x_{j+1})$

If $S''(x_0) = S''(x_n) = 0$, it is a natural or free boundary. If $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$, it is a clamped boundary.

There is always a unique natural spline interpolating between n data points and a unique clamped spline interpolating between n data points (assuming that $f'(x_0)$ and $f'(x_n)$ are given).

If $f \in C^4[a, b]$ with $\max |f^{(4)}(x)| \leq M$ then there will be some constant C such that

$$|f(x) - S(x)| \leq C \max_j (x_j - x_{j+1})^4$$

for a clamped spline $C = \frac{5M}{384}$

2 Numerical Differentiation

Forward difference formula

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Backward difference formula

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

$(n+1)$-point formula

$$f'(x_j) = \sum_{k=0}^{n} f(x_k) L'_k(x_j) + \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!} \prod_{k=0, k \neq j} (x_j - x_k)$$

which comes from the error bound for the Lagrange interpolating polynomials.

Second derivative midpoint formula

$$f''(x_0) = \frac{1}{h^2}(f(x_0 - h) - 2f(x_0) + f(x_0 + h)) - \frac{h^2}{12} f^{(4)}(\xi)$$

3 Richardson Extrapolation

If your error is of the form

$$M = N(h) + K_1 h + K_2 h^2 + ...$$

then you can define $N_2(h) = N(\frac{h}{2}) + \big(N(\frac{h}{2}) - N(h)\big)$ to get

$$M = N_2(h) - \frac{K_2}{2} h^2 - \frac{3K_3}{4} h^3 + ...$$

increasing the order of the approximation by 1.

4 Numerical Integration

4.1 Trapezoidal Rule

The trapezoidal rule comes from approximating the function by two points and using the trapezoid as an estimate for the integral

$$\int_a^b f(x)dx = \frac{h}{2}(f(a) + f(b)) - \frac{h^3}{12}f''(\xi)$$

where $h = b - a$

4.2 Simpson's Rule

For Simpson's rule we use 3 points

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \frac{h^5}{90}f^{(4)}(\xi)$$

where $h = (b - a)/2$ and the points are evenly spaced.

4.3 Newton-Cotes

The $(n+1)$-point closed Newton-Cotes formula uses the points $x_0 = a$, $x_1 = a+h$, ..., $x_{n-1} = a+(n-1)h, x_n = b$ with $h = (b - a)/n$

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n+1} a_i f(x_i)$$

$$a_i = \int_{x_0}^{x_n} L_i(x)dx = \int_{x_0}^{x_n} \prod_{j=0, j\neq i} \frac{x - x_j}{x_j - x_i}dx$$

The open version is the same thing but without the endpoints so $x_0 = a+h$ and $x_n = b-h$ with $h = (b-a)/(n+2)$

4.4 Composite Numerical Integration

Composite numerical integration is when we break up the integral into a bunch of smaller ones and then apply one of our rules for each one.
The composite Simpson's rule is

$$\int_a^b f(x)dx = \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\right) - \frac{(b-a)h^4}{180}f^{(4)}(\mu)$$

4.5 Romburg Integration

We apply Richardson Extrapolation to the composite trapezoid rule for numerical integration. Denote the integral approximation for $n = 1, 2, 4, 8, ...$ by $R_{1,1}, R_{2,1}, ...$ Then we can increase the order of our approximation via

$$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1} - 1}(R_{k,j-1} + R_{k-1,j-1})$$

4.6 Gaussian Quadrature

We seek to approximate

$$\int_a^b f(x)dx = \sum_{i=1}^{n} c_i f(x_i)$$

where the x_i are not necessarily easily spaced. The Gaussian quadrature is the choice of c_i's and x_i's such that the formula is exact for polynomials up to (and including degree $2n-1$. For $n=2$ the optimal choice is

$$\int_{-1}^{b} f(x)dx \approx f(-\frac{\sqrt{3}}{3}) + f(\frac{\sqrt{3}}{3})$$

which gives correct answers for polynomials of degree less than or equal to $2n-1=3$.
For higher order accurate formulas, define P_n as the n'th Legendre polynomaial. The Legendre polynomials are monic and satisfy

$$\int_{-1}^{1} P(x)P_n(x)dx = 0$$

if $P(x)$ is a polynomial of degree less than n. The first few are

$$P_0 = 1 \quad P_1(x) = x \quad P_2(x) = x^2 - 1/3 \quad P_3(x) = x^3 - 3/5x$$

Then let $x_1, ..., x_n$ be the roots of the n'th Legendre polynomial. And let

$$c_i = \int_{-1}^{1} \prod_{j=1, j\neq i} \frac{x - x_j}{x_j - x_i} dx$$

The above x_i's and c_i's define the n'th order Gaussian quadrature, which is accurate for polynomials of degree $2n-1$.

5 Iterative Methods

5.1 Fixed Point Iteration

If $g \in C[a,b]$ such that $|g'| < k$ for some constant $0 < k < 1$, then for any $p_0 \in [a,b]$, the sequence

$$p_{n+1} = g(p_n)$$

converges to a unique fixed point for g.
In fact this convergence is linear, meaning

$$\lim_{n\to\infty} \frac{p_{n+1} - p^*}{p_n - p^*} = \lambda < 1$$

Note this it may converge faster, but it will be at least linear, always.
If, in addition, $g \in C^2[a,b]$ with $|g''| \leq M$ and $g'(p^*) = 0$, then for p_0 sufficiently close to p^* we get quadratic convergence. In fact for high enough n we have

$$|p_{n+1} - p^*| \leq \frac{M}{2}|p_n - p^*|$$

Quadratic convergence for fixed-point iteration can **only** occur if $g'(p^*) = 0$

5.2 Newton's Method

Newton's method for finding a root of $f \in C^2[a,b]$ is given by

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

If started sufficiently close to p^* where p^* is a **simple** root, then Newton's method will converge quadratically.
If p^* is a root of multiplicity m, then the modification to Newton's method given by

$$p_{n+1} = p_n - m\frac{f(p_n)}{f'(p_n)}$$

will converge quadratically. For a non-simple root of any order, we can apply Newton's method to the function $\mu(x) = \frac{f(x)}{f'(x)}$ giving

$$p_{n+1} = p_n - \frac{f(x)f'(x)}{(f'(x))^2 - f(x)f''(x)}$$

However, this is generally a dumb idea in practice because floating point error in the denominator causes the iterations to be thrown wildly off (boy, sure it great they make us study this modification then...).

6 Numerical Methods for ODEs

Generally, we consider ODEs of the form

$$\frac{dy}{dt} = f(y,t)$$

6.1 Stability and Consistency

A numerical scheme is 0-stable if, for two different solutions, x_n, z_n with potentially different initial conditions, but otherwise all things the same we can bound

$$|x_i - z_i| \leq K|x_0 - z_0|$$

for some constant K and all $i \leq N$ where $N = (b-a)/h$. This basically means that the solutions depend continuously on the initial conditions.
The **local truncation error** for a method is the difference between a the scheme and the actual ODE when applied to a solution. If the difference scheme is given by some update rule

$$y_{n+1} = \Phi_h(y_n)$$

then the local truncation error is defined as

$$y(t_{n+1}) = \Phi_h(y(t_n)) + \tau_n$$

where $y(t)$ is a solution to the ODE.
A numerical scheme is consistant if

$$\tau_n = o(h) \qquad \lim_{h\to 0} \tau_n/h = 0$$

We say that a numerical scheme is convergent of order p if

$$e_n = y_n - y(t_n) = \mathcal{O}(h^p)$$

where $y(t)$ solves the ODE and $e_0 = 0$ (initial conditions match).
Theorem: A numerical method that is both consistent and 0-stable is convergent.
For a 0-stable method, if $\tau_n = \mathcal{O}(h^p)$, then the method will be convergent of oder $p-1$.

The **Region of Absolute Stability** for a method is

$$S = \{\lambda h \in \mathbb{C} : y_n \to 0 \text{ when the scheme is applied to the model problem with } h\}$$

where the model problem is

$$y' = \lambda y$$

This means that, for $\lambda h \in S$, the scheme will have $y_n \to 0$ as $n \to \infty$ when the step size is h, when applied to the model problem.
To get the region of absolute stability, you simply apply the method to $y' = \lambda y$ and find the conditions for $y_n \to 0$.
For example, the region of absolute stability for forward euler is

$$\{z \in \mathbb{C} : |1+z| \leq 1\}$$

6.2 Common Methods

6.2.1 Forward Euler

$$y_{n+1} = y_n + hf(t_n, y_n)$$

We find the local truncation error by plugging in $y(t)$ (an exact solution to the ODE) to get

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \tau_n$$

Then noting that

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}y''(\xi)$$

So we get

$$\tau_n = \frac{h^2}{2}y''(\xi)$$

Suppose f has Lipschitz constant L and y'' is bounded by M. Then we can get the global error as

$$\begin{aligned}
|e_{n+1}| = |y(t_{n+1}) - y_{n+1} &= y(t_n) + hf(t_n, y(t_n)) - y_n - hf(t_n, y_n) + \tau_n| \\
&\leq |y(t_n) - y_n| + h|f(t_n, y(t_n)) - f(t_n, y_n)| + |\tau_n| \\
&\leq e_n + hL(y(t_n) - y_n) + |\tau_n| \\
&= (1 + hL)e_n + |\tau_n|
\end{aligned}$$

So, by repeatedly applying this inequality we get

$$|e_n| \leq (1 + hL)e_{n-1} + |\tau_n| \leq (1 + hL)^2 e_{n-2} + (1 + hL)|\tau_{n-1}| + |\tau_n|$$

etc. Finally arriving at

$$\begin{aligned}
|e_n| &\leq (1 + hL)^n e_0 + \sum_{j=0}^{n-1}(1 + hL)^j |\tau_j| \\
&\leq (1 + hL)^n e_0 + \max_{i=1,\ldots,n} |\tau_i| \frac{(1 + hL)^n - 1}{1 - (1 + hL)} \\
&\leq (1 + hL)^n e_0 + \frac{h^2 M}{2} \frac{(1 + hL)^n - 1}{-hL} \\
&\leq \frac{h^2 M}{2} \frac{(1 + hL)^n - 1}{hL} \\
&= \frac{hM}{2L}((1 + hL)^n - 1)
\end{aligned}$$

Where we have assumed that $e_0 = 0$. Now we use the fact that $(1 + hL)^n \leq e^{nLh}$ and $nh = T$ (the length of the time interval we are considering to get

$$|e_n| \leq \frac{hM}{2L}(e^{TL} - 1) \leq \frac{hM}{2L}e^{TL}$$

So forward Euler is an order-1 method.

6.2.2 Backward Euler

The backward Euler method is an implicit scheme given by

$$y_{n+1} = y_n + hf(t_n, y_{n+1})$$

The local truncation error is found by noting that

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}y''(\xi)$$

and

$$f(t_n, y(t_{n+1})) = f(t_n, y(t_n)) + (y(t_{n+1}) - y(t_n))f_y(t_n, y(t_n)) + \mathcal{O}(h^2) = f(t_n, y(t_n)) + hff_y + \mathcal{O}(h^2)$$

So

$$y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}y''(\xi) = y(t_n) + h(f(t_n, y(t_n)) + hff_y + \mathcal{O}(h^2)) + \tau_n$$

Note that $ff_y = y''$ so

$$\tau_n = h^2 y''(\xi)(\frac{1}{2} - 1) = -\frac{h^2}{2}y''(\xi)$$

If we let M be a bound on y'' and f have Lipschitz constant L. Then we have

$$\begin{aligned} e_{n+1} &= |y(t_{n+1}) - y_{n+1}| = |y(t_n) + hf(t_n y(t_{n+1})) + \tau_n - y_n - hf(t_{n+1}, y_{n+1})| \\ &\leq |y(t_n) - y_n| + hL|y(t_{n+1}) - y_{n+1}| + |\tau_n| \\ &= e_n + hLe_{n+1} + |\tau_n| \end{aligned}$$

So $e_{n+1} \leq \frac{e_n}{1-hL} + \frac{|\tau_n|}{1-hL}$ Repeatedly applying this formula gives

$$e_n \leq \frac{e_0}{(1-hL)^n} + \sum_{j=0}^{n-1} \frac{|\tau_j|}{(1-hL)^{n-j}}$$

Our bound gives us that $|\tau_n| \leq \frac{Mh^2}{2}$ for all n so, assuming $e_0 = 0$ we get

$$e_n \leq \sum_{j=1}^{n} \frac{Mh^2}{2} \frac{1}{(1-hL)^j}$$

We then apply the geometric sum formula to get

$$e_n \leq \frac{h^2 M}{2} \frac{1 - \left(\frac{1}{1-hL}\right)^n}{1 - \frac{1}{1-hL}} = \frac{Mh^2}{2} \frac{(1 - \left(\frac{1}{1-hL}\right)^n)(1-hL)}{hL}$$

which we simply as

$$e_n \leq \frac{Mh}{2L}(1 - \left(\frac{1}{1-hL}\right)^n)$$

Note that $\frac{1}{1-hL} = 1 + \frac{hL}{1-hL}$ and that

$$(1 + \frac{hL}{1-hL})^n \leq e^{\frac{NhL}{1-hL}}$$

So

$$e_n \leq \frac{Mh}{2L}(1 + e^{\frac{NhL}{1-hL}}) = \frac{Mh}{2L}(1 + e^{\frac{TL}{1-hL}})$$

So implicit Euler is an order-1 method.

6.2.3 Heun's Method (Explicit Trapezoid, Modified Euler's Method)

Heun's method is given by the update rule

$$y_{n+1} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n)))$$

Heun's method will be an order 2 method.

6.2.4 Implicit Trapezoid

The implicit trapezoid method is given by the update rule

$$y_{n+1} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))$$

7 Numerical Methods for PDEs

7.1 The One-Way Wave Equation

The simplest one-way wave equation (and simpliest PDE for that matter) is the

$$u_t + au_x = f(t,x)$$

for $x \in [x_0, x_1]$, $t \geq 0$, with initial data given by $u(0,x) = u_0(x)$.

7.1.1 The Method of Characteristics

Often the qual is concerned with the well-posedness of an IBVP. For a one-way wave equation, the standard way to do this is with the method of characteristics. Note that while the method is often used for *solving* equations, you do not necessarily need to solve the equation to determine if the problem is well-posed.
We can consider the one-way wave with variable coefficient

$$u_t + a(t,x)u_x = f(t,x)$$

Then we seek a transform of the form $t \to \tau$ and $x \to \xi$ such that the differential operator on the left hand side will simplify. This will be a specific curve in the xt-plane. Let us consider $x = x(\tau)$ and $t = t(\tau)$. Then

$$\frac{d}{d\tau}u(t(\tau), x(\tau)) = t'u_t + x'u_x$$

If we pick $t(\tau)$ such that $t' = 1$ and $x(\tau)$ such that $x' = a(t,x)$ then this will be accomplished. Immediately, we see that $t = \tau$ will work. The second condition will be satisfied if we can solve the ODE

$$\frac{dx}{dt} = a(x,t)$$

Once we do this we have

$$\frac{d}{dt}u(t,x(t)) = f(t,x)$$

We could equivalently write $w(t) = u(t,x(t))$ and solving the PDE becomes a problem of solving the ODE

$$w' = f(t,x)$$

The well-posedness of the problem will then come down to the solvability of this ODE and its dependence on initial conditions. Typically we will be concerned with solving the PDE on a bounded interval $[x_0, x_1]$. The characteristic curves we get from solving the ODE for $x(t)$ will inform the dependence on initial conditions. If $x(0)$ is outside the interval, but $x(t) \in [x_0, x_1]$ for some t later, then a value inside the specified interval will depend on an initial value outside the interval. Because the initial conditions $u_0(x)$ are only specified in the interval $[x_0, x_1]$ we will need more information, in the form of boundary values.
We can then say that for the problem on $x \in [a,b]$, if $x'(t) > 0$ when $x = x_0$ for any t then we will need boundary conditions on the boundary $x = x_0$. If $x'(t) < 0$ when $x = x_1$ for any t, then we will need boundary conditions on the boundary for $x = x_1$. Here $x(t)$ is the characteristic curve described above. Otherwise, we cannot specify boundary conditions and have a solution necessarily exist.
Often a will just be a function of x, $a(x)$, which case, we get $x'(t) = a(x)$, so the boundary conditions required will depend on the sign of $a(x_0)$ and $a(x_1)$.

7.1.2 Stability and the CLF Conditions

A numerical finite difference scheme approximating the one-way wave (or any PDE) is **stable** in stability region Γ (a region in the kh-plane) if there exists some integer J such that for any positive time T we have

$$h\sum_{m=-\infty}^{\infty} |v_m^n|^2 \leq C_T h \sum_{j=1}^{J}\sum_{m=-\infty}^{\infty} |v_m^j|^2$$

for all $n \leq T/k$. where C_T can depend on T but not h or k. Note that this is saying

$$h||v^n||_{l_2} \leq C_T h \sum_{j=1}^{J} ||v^j||_{l_2}$$

where $||\cdot||_{l_2}$ denotes the spacial l_2 discrete norm. So stability states that every l_2 norm for each timestep is bounded by a constant times the sum of the first few l_2 norms.

The **Courant-Friedrichs-Lewy** condition states that, for a finite difference scheme, trying to approximate the one-way wave equation, if the scheme is of the form

$$v_m^{n+1} = \alpha v_{m-1}^n + \beta v_m^n + \gamma v_{m+1}^n$$

A necessary condition for stability of the scheme is

$$|a\lambda| \leq 1$$

where $\lambda = k/h$. This can be thought of as the condition that the speed propagation of information of the wave (given by a) is less than the speed at which the finite difference scheme calculates (which is given by the ratio between the temporal step size and spacial step size).
The CFL conditions will generally not be sufficient.

7.1.3 Consistancy and The Lax-Richtmyer Equivalence Theorem

Consistency of a method say whether the difference operator approximates the differential operator. A finite difference scheme $P_{k,h}v = f$ is consistant with the differential equation $Pu = f$ if, for any smooth ϕ

$$P_{k,h}\phi - P\phi \to 0 \quad \text{as } h, k \to 0$$

The Lax Richtmyer Equivalence theorem tells us necessary and sufficient conditions for a finite difference scheme to be convergent. The theorem states that for a well-posed PDE problem, a consistant finite difference approximation will converge if and only if it is stable. So it should be thought of as

$$\text{consistency} + \text{stability} \iff \text{convergence}$$

The theorem will be useful for finding difference schemes for the one-way wave, heat equation, and wave equation. However it will **not** be useful for conservation laws, as the solutions to such problems may not be unique.

7.1.4 Von Neumann Analysis

This method for determining the stability of a method relies on the fact that the L_2 (discrete and continuous) norm is preserved under the Fourier transform, and that in the Fourier domain, derivatives become multiplication by $i\omega$. For a linear, constant-coefficient PDE, the finite difference scheme (when written in an update-rule form) will turn out to be

$$\hat{v}^{n+1}(\omega) = g(h\omega, k, h)\hat{v^n}(\omega)$$

we typically write $h\omega = \theta$. So the stability of a method will depend nicely on the initial conditions if and only if $|g| \leq 1$. Von Neumann analysis is the process of finding g.
The quickest way to do this to to plug in $v_m^n \to g^n e^{im\theta}$ into the scheme and then solve for g. Note that an increment of the index in the space domain shows up as multiplication by $e^{i\theta}$ in the Fourier domain.
The full theorem for stability with Von Neumann analysis states that if $g = g(\theta)$ the method is stable if and only if $|g| \leq 1$. And if $g = g(\theta, h, k)$, the method is stable if and onnly if

$$|g(\theta, h, k)| \leq 1 + \alpha k$$

Another important thing to remember is that the stability of a scheme will be independent of f. So when determining stability, we typically set $f = 0$ for simplicity.

If we have a problem where $a = a(x)$, then technically, the analysis in the Fourier domain will not work the same as the Fourier transform of a will not be the same as a and we will get a convolution in general. However, we can still analyze this problem using Von Neumann analysis if we consider the **frozen coefficient problems** for every possible value of a. If a scheme is stable for every frozen coefficient problem, that is the problem where we consider a fixed, then the scheme will be stable for the actual problem. So we can treat a as a constant until the Von Neumann analysis is complete and then derive a condition on $a\lambda$ that will allow the scheme to be stable for all a.

7.1.5 L_∞ Analysis

The definition of stability is actually independent of the choice of norm, so a scheme will still satisfy the Lax-Richtmyer Equivalence Theorem if it is stable for a norm other than the L_2 norm. In particular, it may be helpful to consider the L_∞ norm given by

$$||v^n||_\infty = \max_m |v_m^n|$$

The L_∞ can be conducted by writing v_m^{n+1} as a function of $v_{m-1}^n, v_m^n, v_{m+1}^n$ and then taking the L_∞ norm of both sides. If we can bound $||v^{n+1}||_\infty$ by $c||v^n||_\infty$, where $c \leq 1$, the scheme will be stable.
This is a useful method for analyzing nonlinear problems, including conservation laws. It can be used for the one-way wave equation when $a = a(x)$.

7.1.6 Symbol Analysis

The **symbol** of a differential operator, $p(s, \xi)$ is given by

$$P(e^{st}e^{i\xi x}) = p(s, \xi)e^{st}e^{i\xi x}$$

The symbol of a differential operator can be used to determine if the problem posed by the operator is well-posed. In particular, it can be used to determine if the problem

$$Pu = 0$$

is well-posed. It can be determined by finding the roots of $p(s, \xi)$ in s. That is the values $q(\xi)$ such that $p(q(\xi), \xi) = 0$. Then if $\mathrm{Re}(q(\xi)) \leq \bar{q}$ for some $\bar{q}$, the PDE is well-posed, in the sense that the solution will not blow up. For linear PDEs, this is equivalent to examining the equation in the Fourier domain, and making sure the solution

$$\hat{u}(t, \xi) = e^{q(\xi)t}\hat{u}_0(\xi)$$

does not run off to ∞ as we let $\xi \in \mathbb{R}$.

The symbol for a difference operator is analogously defined to be $p_{k,h}(s, \xi)$ where

$$P_{k,h}(e^{skn}e^{imh\xi}) = p_{k,h}(s, \xi)e^{skn}e^{imh\xi}$$

If we let $r_{k,h}(s, \xi)$ be the symbol for $R_{k,h}$, then a difference scheme of the form

$$P_{k,h}v = R_{k,h}f$$

to solve the equation

$$Pu = f$$

is consistent if and only if

$$p_{k,h}(s, \xi) - r_{k,h}(s, \xi)p(s, \xi) = \mathcal{O}(h^p) + \mathcal{O}(k^q)$$

where $p, q > 0$.

7.1.7 Common Schemes and their Stability

The **Lax-Friedrich Scheme** is given by

$$\frac{v_m^{n+1} - \frac{1}{2}(v_{m+1}^n + v_{m-1}^n)}{k} + a\frac{v_{m+1}^n - v_{m-1}^n}{2h} = f_m^n$$

To analyze the stability of this scheme, we plug in $g^n e^{im\theta}$ to get

$$\frac{g^{n+1}e^{im\theta} - \frac{1}{2}(g^n e^{i(m+1)\theta} + e^{i(m-1)\theta})}{k} + a\frac{g^n e^{i(m+1)\theta} - g^n e^{i(m-1)\theta}}{2h} = f_m^n$$

Now we set f_m^n for simplicity. Dividing the above expression by $g^n e^{im\theta}$ gives

$$g - \frac{1}{2}(e^{i\theta} + e^{-i\theta}) + \frac{a\lambda}{2}(e^{i\theta} - e^{-i\theta}) = 0$$

Solving for g gives

$$g = \frac{1}{2}(e^{i\theta} + e^{-i\theta}) - \frac{a\lambda}{2}(e^{i\theta} - e^{-i\theta}) = \cos(\theta) - ia\lambda\sin(\theta)$$

Then

$$|g|^2 = \cos^2(\theta) + a^2\lambda^2\sin^2(\theta)$$

which will be less than 1 if and only if $|a\lambda| \leq 1$. So the stability conditions for Lax-Friedrich is the same as the CFL condition.

The **Crank-Nicolson Scheme** is given by

$$\frac{v_m^{n+1} - v_m^n}{k} + a\frac{v_{m+1}^{n+1} - v_{m-1}^{n+1} + v_{m+1}^n - v_{m-1}^n}{4h} = \frac{f_m^{n+1} + f_m^n}{2}$$

This is an implicit scheme, but the stability can be found in the same way using Von Neumann analysis. We plug in $v_m^n \to g^n e^{im\theta}$ and simplify (using $f = 0$) to get

$$\frac{g-1}{k} + a\frac{ge^{i\theta} - ge^{-i\theta} + e^{i\theta} - e^{-i\theta}}{4h} = 0$$

and we can find g to be

$$g = \frac{1 - \frac{a\lambda}{2}i\sin(\theta)}{1 + \frac{a\lambda}{2}i\sin(\theta)}$$

which will have $|g| = 1$ for any $a\lambda$. So this scheme is **unconditionally** stable. The Crank-Nicolson scheme is also accurate to order $(2, 2)$, that is order 2 in time and 2 in space.

The **Lax-Windroff Scheme** is given by

$$\begin{aligned}\frac{v_m^{n+1} - v_m^n}{k} + a\frac{v_{m+1}^n - v_{m-1}^n}{2h} - \frac{a^2k}{2}\frac{v_{m+1}^n - 2v_m^n + v_{m-1}^n}{h^2} \\ = \frac{1}{2}(f_m^{n+1} + f_m^n) - \frac{ak}{4h}(f_{m+1}^n - f_{m-1}^n)\end{aligned}$$

The amplification factor, g can be found for this scheme to be

$$|g|^2 = 1 - 4a^2\lambda^2(1 - a^2\lambda^2)\sin^4(\frac{\theta}{2})$$

which will be less than 1 provided the CFL conditions are met, i.e. that $|a\lambda| \leq 1$.

7.1.8 Higher Dimensional Analogues

There may be higher-dimensional versions of the one-way wave equation such as those of the form

$$u_t + a(t, x, y)u_x + b(t, x, y)u_y = f(t, x, y)$$

The method of characteristics will work the same for the one-dimensional case, except that we will have $x'(t) = a(t, x, y)$ and $y'(t) = b(t, x, y)$ and the well-posedness will depend on x' and y' on the boundaries for x and y respectively.

The higher-dimensional Lax-Friedrich Scheme (for 2 spacial dimensions) can be formulated as

$$\frac{v_{m,l}^{n+1} - \frac{1}{4}(v_{m+1,l}^n + v_{m-1,l}^n + v_{m,l+1}^n + v_{m,l-1}^n)}{k} + a\frac{v_{m+1,l}^n - v_{m-1,l}^n}{2h_x} + b\frac{v_{m,l+1}^n - v_{m,l-1}^n}{2h_y} = f_m^n$$

So rather than taking the average of 2 values in the forward time difference, we take the average of 4.

The higher-dimensional Crank-Nicholson scheme (for 2 spacial dimensions) can be formulated as

$$\begin{aligned}\frac{v_{m,l}^{n+1} - v_{m,l}^n}{k} + \frac{1}{2}\left(a\frac{v_{m+1,l}^n - v_{m-1,l}^n}{2h_x} + b\frac{v_{m,l+1}^n - v_{m,l-1}}{2h_y}\right) \\ + \frac{1}{2}\left(a\frac{v_{m+1,l}^{n+1} - v_{m-1,l}^{n+1}}{2h_x} + b\frac{v_{m,l+1}^{n+1} - v_{m,l-1}^{n+1}}{2h_y}\right) = 0\end{aligned}$$

However, most of the time on the qual, if presented with a multi-dimensional equation, the best solution is often transforming the equation into a system of PDEs, each of which have a simpler form.

7.2 Conservation Laws

A conservation law is a PDE of the form

$$u_t + (f(u))_x = 0$$

For some smooth f. There is also the "vanishing viscosity version" of the conservation law which will take the form

$$u_t + (f(u))_x = \epsilon u_{xx}$$

and we will want to convergent method for when $\epsilon \to 0$. These problems are also given with smooth initial data $u(x,0) = u_0(x)$. The first thing to note about these problems is technically, they are **not** well-posed. This is because solutions to the problem may not be unique. Because of this, the Lax equivalence theorem no longer applies. In the qual we will typically be asked to construct a convergent finite difference scheme which converges for $t \geq 0$, even as $\epsilon \to 0$.

7.2.1 Convergence of Conservation Law FD Schemes

First I should note that the CFL conditions for conservation laws state that a necessary condition for convergence of a scheme is that

$$|\lambda f'(u)| \leq 1$$

for any u in the range that numerical approximation can take.

Our main (and pretty much only) theorem for convergence of a finite difference scheme for a conservation law is theorem 15.2 in [5]. This theorem says that for a finite difference scheme to converge on a conservation law we need 3 things:

1. The numerical method can be written in conservative form with a Lipschitz continuous numerical flux
2. The numerical flux is consistent with the conservation law we wish to solve
3. The method is TV-stable (total-variation stable)

In this case, converge means that $d(U_k, W) \to 0$ as $k \to 0$, where U_k is the numerical solution and W is the set of solutions to the actual PDE. So it is important to note that the numerical approximations won't necessarily head towards 1 solution and without further requirements, we do not know what solution it will head towards. But it will converge. The TV-stability of the method ensures that the functions output by our numerical scheme live in a compact space, meaning that we can be sure the Cauchy sequence given by our scheme will converge.
First, the conservation form of a finite difference scheme is the following

$$v_m^{n+1} = v_m^n - \lambda \left(F(v_m^n, v_{m+1}^n) - F(v_{m-1}^n, v_m^n) \right)$$

with $\lambda = \frac{k}{h}$. The function F is called the **numerical flux**. Note that this is only the definition for a one-step function, but that should be sufficient for the qual.
The second condition is that the numerical flux is consistent with the conservation law we wish to solve and it is Lipschitz continuous. We actually check both these at the same time, by verifying the below inequality

$$|F(v,w) - f(\overline{u})| \leq K \max\{|v - \overline{u}|, |w - \overline{u}|\}$$

for some constant K that may depend on $\overline{u}$. Once the numerical flux is determined, this is typically not too hard to verify, so long as the original f is Lipschitz.
Finally, to check the method is TV-stable, we have a few possibilities. Checking directly is difficult and will generally not be done. It is much easier to check slightly stronger conditions that imply TV stability, meaning that the total variation of all approximations v^n is bounded by a fixed constant (usually denoted R). The first condition we could check is that the method is **total variation diminishing**, meaning that $TV(v^n) \leq TV(v^{n-1})$ for all n. In that case, the total variation of any time step would be bounded by the total variation of the first.
A slightly easier condition to check is that the method is l_1**-contracting**. This means that for any v^n, w^n numerical solutions to the conservation law obtained by our method, we have

$$||v^{n+1} - w^{n+1}||_1 \leq ||v^n - w^n||_1$$

where $||\cdot||_1$ is the l_1 norm defined by

$$||v^n||_1 = h \sum_{j=-\infty}^{\infty} |v_j^n|$$

or a similarly truncated sum if the grid is finite. From [5] we know that an l_1-contracting method is total variation diminishing, meaning it is TV-stable.
However the easier way to verify that a method is TV-stable is to show that the method is **monotone**. This means that

$$v_m^n \geq w_m^n \quad \forall m \implies v_m^{n+1} \geq w_m^{n+1} \quad \forall m$$

This can be checked rather easily by writing the method as

$$v_m^{n+1} = \mathcal{H}(v_{m-1}^n, v_m^n, v_{m+1}^n)$$

and verifying

$$\frac{\partial \mathcal{H}}{\partial v_j^n} \geq 0$$

for all j. This is much easier to do than verifying l_1 stability, but it can only be done for first order accurate methods. Namely, the Lax-Friedrich method will be monotone (as it is first-order accurate). But if we need a second order accurate method it will not be monotone.

7.2.2 Example Method: Lax-Friedrich

For solving the PDE

$$u_t + (f(u))_x = 0$$

the Lax-Friedrich method can be defined analogously to the advection case as

$$v_m^{n+1} = \frac{1}{2}(v_{m-1}^n + v_{m+1}^n) - \frac{\lambda}{2}(f(v_{m+1}^n) - f(v_{m-1}^n))$$

where $\lambda = k/h$. This will then have numerical flux given by

$$F(v_m^n, v_{m+1}^n) = \frac{1}{2\lambda}(v_m^n - v_{m+1}^n) + \frac{1}{2}(f(v_m^n) + f(v_{m+1}^n))$$

Then we can see

$$\begin{aligned}
v_m^{n+1} &= v_m^n - \lambda \left(F(v_m^n, v_{m+1}^n) - F(v_{m-1}^n, v_m^n)\right) \\
&= v_m^n - \frac{1}{2}(v_m^n - v_{m+1}^n) - \frac{\lambda}{2}(f(v_m^n) + f(v_{m+1}^n)) + \frac{1}{2}(v_{m-1}^n - v_m^n) + \frac{\lambda}{2}(f(v_{m-1}^n) + f(v_m^n)) \\
&= v_m^n - v_m^n + \frac{1}{2}(v_{m+1}^n + v_{m-1}^n) + \frac{\lambda}{2}(f(v_{m-1}) - f(v_{m+1}^n)) \\
&= \frac{1}{2}(v_{m+1}^n + v_{m-1}^n) - \frac{\lambda}{2}(f(v_{m+1}^n) - f(v_{m-1}^n))
\end{aligned}$$

recovering the original method. If instead we have the PDE

$$u_t + (f(u))_x = \epsilon u_{xx}$$

Then we use the modified Lax-Friedrich scheme given by

$$\frac{v_m^{n+1} - kv_m^n - \frac{1-k}{2}(v_{m+1}^n + v_{m-1}^n)}{k} + \frac{f(v_{m+1}^n) - f(v_{m-1}^n)}{2h} = \epsilon \frac{v_{m+1}^n - 2v_m^n + v_{m-1}^n}{h^2}$$

For this method to converge as $\epsilon \to 0$, we must have that $\epsilon = \frac{h^2}{2}$. To write this out in conservative form we note that

$$\begin{aligned}
v_m^{n+1} &= kv_m^n + \frac{1-k}{2}(v_{m+1}^n + v_{m-1}^n) + \frac{k}{2h}(f(v_{m+1}^n) - f(v_{m-1}^n)) + \frac{\epsilon k}{h^2}(v_{m+1} - 2v_m^n + v_{m-1}^n) \\
&= v_m^n + \frac{1-k}{2}(v_{m+1} - v_m^n - v_m^n + v_{m-1}^n) + \frac{k}{2h}(f(v_m^n) + f(v_{m+1}) - (f(v_m^n) + f(v_{m-1}^n))) \\
&+ \frac{\epsilon k}{h^2}(v_{m+1} - v_m^n - (v_m^n + v_{m-1}^n)) \\
&= v_m^n + \lambda\frac{1-k}{2\lambda}(v_{m+1} - v_m) + \frac{\lambda}{2}(f(v_m^n) + f(v_{m+1}^n)) + \frac{\epsilon\lambda}{h}(v_{m+1}^n - v_m^n) \\
&- \left(\lambda\frac{1-k}{2\lambda}(v_m^n - v_{m-1}^n) + \frac{\lambda}{2}(f(v_{m-1}^n + f(v_m^n))) + \frac{\epsilon\lambda}{h}(v_m^n + v_{m-1}^n)\right) \\
&= v_m^n - \lambda\left(F(v_m^n, v_{m+1}^n) - F(v_{m-1}^n, v_m^n)\right)
\end{aligned}$$

if $F(v_m^n, v_{m+1}^n) = \frac{1-k}{2\lambda}(v_{m+1}^n - v_m^n) + \frac{1}{2}(f(v_m^n) + f(v_{m+1}^n)) + \frac{\epsilon}{h}(v_{m+1}^n - v_m^n)$.
Now we will show for the vanishing viscosity case, that the scheme converges via theorem 15.2 in [5]. We have already written the method in conservation form. We merely need to check that the numerical flux is consistent with f. To do this, note that

$$\begin{aligned}
|F(v,w) - f(\overline{u})| &= \left|\frac{1-k}{2\lambda}(w-v) + \frac{1}{2}(f(v) + f(w)) + \frac{\epsilon}{h}(w-v) - f(\overline{u})\right| \\
&\leq \left|\left(\frac{1-k}{2\lambda} + \frac{\epsilon}{h}\right)(w - \overline{u} + \overline{u} - v)\right| + \frac{1}{2}\left|f(v) - f(\overline{u}) + f(w) - f(\overline{u})\right| \\
&\leq \left(\frac{1-k}{2\lambda} + \frac{\epsilon}{h}\right)(|w - \overline{u}| + |v - \overline{u}|) + \frac{1}{2}|f(v) - f(\overline{u})| + \frac{1}{2}|f(w) - f(\overline{u})|
\end{aligned}$$

Now, here we must assume that f is a Lipschitz function with Lipschitz constant K. Giving us

$$\leq \left(\frac{1-k}{2\lambda} + \frac{\epsilon}{h} + K\right)\max\{|w - \overline{u}|, |v - \overline{u}|\}$$

Note that $\epsilon = h^2/2$ and λ is constant, so the multiple of the maximum above is bounded by a constant. Thus the numerical flux is consistent with the PDE.
Finally, we can check that the method is TV stable by checking it is monotone preserving. To do this, we write

$$v_m^n = \mathcal{H}(v_{m-1}^n, v_m^n, v_{m+1}^n)$$

and check that $\partial\mathcal{H}/\partial v_m^n \geq 0$, $\partial\mathcal{H}/\partial v_{m-1}^n \geq 0$, $\partial\mathcal{H}/\partial v_{m+1}^n \geq 0$. Based on our equation above we get that

$$\mathcal{H}(v_{m-1}^n, v_m^n, v_{m+1}^n) = kv_m^n + \frac{1-k}{2}(v_{m+1}^n + v_{m-1}^n) + \frac{k}{2h}(f(v_{m+1}^n) - f(v_{m-1}^n)) + \frac{\epsilon k}{h^2}(v_{m+1} - 2v_m^n + v_{m-1}^n)$$

Then

$$\frac{\partial\mathcal{H}}{\partial v_m^n} = k - \frac{2\epsilon k}{h^2} \geq k - k = 0$$

because $\epsilon = h^2/2$. We also have

$$\frac{\partial\mathcal{H}}{\partial v_{m-1}^n} = \frac{1-k}{2} - \frac{\lambda}{2}f'(v_{m-1}^n) + \frac{\epsilon k}{h^2} = \frac{1}{2} - \frac{k}{2} - \frac{\lambda}{2}f'(v_{m-1}^n) + \frac{k}{2} = \frac{1 - \lambda f'(v_{m-1}^n)}{2}$$

Using the CFL condition $|\lambda f'(u)| \leq 1$ for all u we get

$$\frac{\partial\mathcal{H}}{\partial v_{m-1}^n} \geq 0$$

And finally

$$\frac{\partial\mathcal{H}}{\partial v_{m+1}} = \frac{1-k}{2} + \frac{\lambda}{2}f'(v_{m+1}^n) + \frac{\epsilon k}{h^2} = \frac{1}{2} - \frac{k}{2} + \frac{\lambda f'(v_{m+1}^n)}{2} + \frac{k}{2} = \frac{1 + \lambda f'(v_{m+1}^n)}{2} \geq 0$$

So the scheme is monotone preserving and thus TV-stable. Then by our theorem, we know that the scheme will converge to a solution of the PDE.

7.2.3 Second Order Schemes

Sometimes the qual problem will ask you to provide a second order accurate scheme. If this is the case, they will also ask for a second order accurate scheme (see S21 Problem 6). An important note with this case is that a second order accurate scheme **will not converge** for large T. We cannot show that a second order accurate scheme will be TV-stable, so even though the scheme is stable in the traditional sense, the convergence will be violated as soon as the Lax equivalence theorem fails to apply.
The question will typically ask you to give an estimate on $\frac{k}{h}$ in order for the scheme to converge for small time T. This estimate will just be the CFL condition, i.e.

$$|f'(u)\frac{k}{h}| \leq 1$$

for all u. The problem will also usually ask why you cannot get convergence for large T. The answer to this is because shocks and rarefactions mean that solutions to the PDE will no longer be continuous and then the CFL condition will no longer apply. This is because the linearization of the conservation law will no longer be valid, but it should be sufficient to say that shocks and rarefactions make the PDE no longer well-posed.

References

[1] Uri M. Ascher and Linda R. Petzold. "Chapter 3: Basic Methods, Basic Concepts". In: *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, pp. 37–72. DOI: 10.1137/1.9781611971392.ch3. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611971392.ch3. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611971392.ch3.

[2] Uri M. Ascher and Linda R. Petzold. "Chapter 4: One-Step Methods". In: *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, pp. 73–122. DOI: 10.1137/1.9781611971392.ch4. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611971392.ch4. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611971392.ch4.

[3] Uri M. Ascher and Linda R. Petzold. "Chapter 5: Linear Multistep Methods". In: *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, pp. 123–159. DOI: 10.1137/1.9781611971392.ch5. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611971392.ch5. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611971392.ch5.

[4] R.L. Burden and J.D. Faires. *Numerical Analysis*. Cengage Learning, 2010. ISBN: 9780538733519. URL: https://books.google.com/books?id=zXnSxY9G2JgC.

[5] Randall J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, 1992. ISBN: 3764327235. URL: http://tevza.org/home/course/modelling-II_2016/books/Leveque%20-%20Numerical%20Methods%20for%20Conservation%20Laws.pdf.

[6] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, Second Edition*. Society for Industrial and Applied Mathematics, 2004. DOI: 10.1137/1.9780898717938. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898717938. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898717938.