Notes: von Neumann Stability Analysis

R. R. Rosales (MIT, Math. Dept., room 2-337, Cambridge, MA 02139)

April 26, 2022

Contents

List of Figures

1 von Neumann Stability Analysis for Numerical Schemes

A von Neumann stability analysis can be carried out for constant coefficients linear finite differences schemes only. It is based on the fact that (for this class of schemes) the general solution to the scheme equations can be found as a linear combination of the special solutions found by separation of variables. This works because:

A. For linear schemes, linear combination of solutions are solutions.

B. If the scheme coefficients are constant (independent of the indexes), exponential and power dependencies on the indexes factor out, leaving algebraic equations for the parameters.

These are the same reasons that allow the solution of linear constant coefficients differential equations using exponential functions.

The limitation to constant coefficients linear equations is not as restrictive as it might seem. One can often get information on the behavior of schemes where these conditions do not apply (e.g: either non-linear or non-constant coefficients situations) by doing a von Neumann stability analysis on a "frozen" coefficients version of the schemes (this can be tricky, so we will not go into this now).

Note 1.1 *For a more mathematically rigorous presentation of the meaning of consistency and stability for numerical schemes, see the section on* Convergence of numerical schemes *in* Various lecture notes for 18311. *The presentation here is at an informal level.* ♣

1.1 Example: the "good" scheme from the lectures

Here we show an example of a von Neumann stability analysis, to illustrate the idea. We pick the "good" scheme used in the GBNS_lecture script in the 18.311 MatLab Toolkit. The aim of this scheme is to solve the wave equation, written as the system of equations:

$$u_t = v \quad \text{and} \quad v_t = u_{x\,x}, \tag{1.1}$$

where the subscripts indicate partial derivatives and the equations are written using nondimensional variables (thus the wave speed is $c = 1$).

Description of the scheme.

Consider a uniform grid in space and time $\{(x_n,\, t_m)\}$, with grid spacings Δx and Δt (assumed "small"). That is $x_{n+1} = x_n + \Delta x$ and $t_{m+1} = t_m + \Delta t$. On this grid we assume that the solution is approximated by the grid functions u_n^m and v_n^m. That is:

$$u(x_n,\, t_m) \approx u_n^m \quad \text{and} \quad v(x_n,\, t_m) \approx v_n^m, \tag{1.2}$$

where the grid functions satisfy the following discretized version of equations (1.1) (the "good" numerical scheme):

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} = v_n^m + \frac{1}{2}\,\delta\,\Delta t\,\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\Delta x)^2}, \tag{1.3}$$

$$\frac{v_n^{m+1} - v_n^m}{\Delta t} = \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\Delta x)^2} + \frac{1}{2}\,\delta\,\Delta t\,\frac{v_{n+1}^m - 2v_n^m + v_{n-1}^m}{(\Delta x)^2}, \tag{1.4}$$

where $\delta > 1$ is a constant (for $\delta = 2$ this is the "good" scheme in the GBNS_lecture script script of the 18.311 MatLab Toolkit). The reason for the condition $\delta > 1$ is stability, as shown below.

Equations (1.3 –1.4) are obtained by adding a small amount of *numerical viscosity* [1] to the intuitive (but naive) discretization used in the "bad" scheme of the GBNS_lecture script in the 18.311 MatLab Toolkit. Namely:

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} = v_n^m \qquad \text{and} \qquad \frac{v_n^{m+1} - v_n^m}{\Delta t} = \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\Delta x)^2}. \tag{1.5}$$

Note that, **to get a consistent scheme** the extra terms added in (1.3–1.4) must vanish as Δt and Δx do. Hence we **require $\boldsymbol{\delta\,\Delta t \ll 1}$**. In other words: **Do not take $\boldsymbol{\delta}$ too large, or problems can arise.**

Normal modes for the scheme.

We look now for solutions of the form

$$\boldsymbol{u_n^m = U\,G^m\,e^{i\kappa n}} \quad \text{and} \quad \boldsymbol{v_n^m = V\,G^m\,e^{i\kappa n}}, \tag{1.6}$$

for the scheme equations in (1.3 – 1.4). Here $\boldsymbol{U}$ and $\boldsymbol{V}$ are constants, $\boldsymbol{G}$ is the **growth factor** and $\boldsymbol{-\pi < \kappa \le \pi}$ is the **grid wave number.**

Remark 1.1 *The restriction $-\pi < \kappa \le \pi$ follows because the terms $e^{i\kappa n}$ — above, in (1.6) — are periodic in κ, of period 2π. Thus there is no point in using values of κ outside of a 2π range — though, some times it is convenient to restrict the range by $0 \le \kappa < 2\,\pi$.*

Note that, for $\kappa = p/q$ a rational number, the solution above is periodic, of period q in n. ♣

Substituting (1.6) into (1.3 – 1.4), we obtain the equations:

$$\left.\begin{aligned} GU &= \left(1 - 2\,\delta\,\lambda^2\,\sin^2(\kappa/2)\right) U + \Delta t\, V, \\ GV &= -\frac{4\,\lambda^2}{\Delta t}\,\sin^2(\kappa/2)\, U + \left(1 - 2\,\delta\,\lambda^2\,\sin^2(\kappa/2)\right) V, \end{aligned}\right\} \tag{1.7}$$

[1] For a motivation, see the introduction to the AENS (Associated Equation to a Numerical Scheme) set of problems. Equations (1.3 –1.4) mimic the system $\boldsymbol{u_t = v + \frac{1}{2}\,\delta\,\Delta t\, u_{x\,x}}$ and $\boldsymbol{v_t = u_{x\,x} + \frac{1}{2}\,\delta\,\Delta t\, v_{x\,x}}$.

where $\boldsymbol{\lambda} = \dfrac{\boldsymbol{\Delta t}}{\boldsymbol{\Delta x}}$, and we have used that $e^{i\kappa} - 2 + e^{-i\kappa} = -4\,\sin(\kappa/2)$.

This is an **eigenvalue problem, with eigenvalue G** given by

$$\begin{aligned} G &= 1 - 2\,\delta\,\lambda^2\,\sin^2(\kappa/2) \pm 2\,i\,\lambda\,\sin(\kappa/2). && (1.8)\\ \text{In particular} \quad |G|^2 &= \left(1 - 2\,\delta\,\lambda^2\,\sin^2(\kappa/2)\right)^2 + 4\,\lambda^2\,\sin^2(\kappa/2) \\ &= 1 - 4\,(\delta - 1)\,\sin^2(\kappa/2)\,\lambda^2 + 4\,\delta^2\,\sin^4(\kappa/2)\,\lambda^4. && (1.9) \end{aligned}$$

The general solution to the scheme equations (1.3–1.4) can be written as a linear combination of the normal modes (1.6 – 1.8). In order for the scheme to behave appropriately, none of these modes should grow (otherwise numerical errors are amplified). Thus we need $|G| \le 1$, which leads to

Stability conditions are conditions that guarantee $|G| \le 1$ for all κ. (1.10)

As we show below — see (1.14) — this is possible provided that λ and δ are restricted appropriately. In fact, we want more (consistency):

The restrictions for stability should allow $\Delta t \to 0$, $\Delta x \to 0$, and $\delta\,\Delta t \to 0$ — so that the scheme equations in (1.3 – 1.4) approach the pde in (1.1). (1.11)

This is also allowed by (1.14).

Implementation of the stability condition (1.10).

For any fixed κ, (1.9) expresses $|G|^2$ as a a quadratic polynomial in λ^2. It follows that $|G|^2 \le 1$ if and only if[2]

$$\delta > 1 \quad \text{and} \quad \lambda^2 \le \frac{\delta - 1}{\delta^2\,\sin^2(\kappa/2)}. \tag{1.12}$$

Thus the **stability conditions** for the scheme in (1.3 – 1.4), obtained by requiring that (1.12) hold for all the possible values of κ, are: $\delta > 1$ and $\lambda^2 \le (\delta - 1)/\delta^2$. However (see note **1.3**)

> It is always a good idea to have any "high frequency" (grid scale) oscillations, killed by the algorithm. This means that **we want $|G| < 1$ for κ away from zero, and particularly near $\kappa = \pi$.** (1.13)

Thus, finally, we arrive at the stability conditions:

$$\delta > 1 \quad \text{and} \quad \lambda < \frac{\sqrt{\delta - 1}}{\delta}, \qquad \text{which yield } |G| < 1 \text{ for } \kappa \ne 0. \tag{1.14}$$

Of course, $G(0) = 1$, so that $|G(0)| = 1$.

Note 1.2 *It is, generally, desirable for a numerical scheme to preserve the equilibrium solutions for the equation. In this case this means that we want $u_n^m =$ constant and $v_n^m = 0$ to be solutions of the scheme. This requirement forces $G(0) = 1$, and it is not guaranteed by consistency.* ♣

Note 1.3 *Generally, a situation where $|G| \equiv 1$ is not desirable, unless $G = G_e =$ exact growth factor*[3] *— for more on G_e, see § 1.1.2.*
The reason is that then modes with κ away from zero ("high frequencies") will be evolved incorrectly, without being killed. For solutions with a substantial amount of high frequencies (e.g.: the initial conditions have discontinuities or singularities of some type) this leads to oscillations that appear in the numerical solution, as the modes loose coherence. An example of this is provided in problems vNSA01h *and* vNSA04h *(Crank-Nicolson scheme for $u_t \pm u_x = 0$).*

[2] Assume $\lambda > 0$, since $\lambda = 0$ is useless. Then note that: (i) $\delta \le 1$ yields $|G^2| > 1$ in (1.9). (ii) For $\delta > 1$ the polynomial in (1.9) is below 1 for small λ^2, and crosses above 1 for $\lambda^2 > (\delta - 1)/(\delta^2\,\sin^2(\kappa/2))$.

[3] Of course, $G = G_e$ is something that can only be achieved for very simple equations, such as $u_t + u_x = 0$.

1.1.1 Further considerations regarding stability

The restriction $|G| \leq 1$ for all κ in (1.10) is a bit of an over-kill in terms of stability, though for the equation in (1.1) this is fine — even desirable, as demonstrated by (1.13 – 1.14). However, there are situations where one cannot use (1.10), as explained in item (c) below. Let us now examine the issue of stability in more detail.

(a) Consistency (see § 1.1.2) guarantees that the normal modes with small grid number $\kappa = O(\Delta x)$ are treated correctly by the scheme.[4] It is the normal modes with κ away from zero that must be kept under control (no growth), since for these modes the scheme cannot be accurate: Finite differences are not accurate approximations of the partial derivatives for functions that vary too rapidly relative to the grid scales Δx and Δt.

(b) A **less restrictive**, but technically more complicated, **definition of stability**[5] is that the solutions provided by the numerical scheme (for fixed initial conditions) should remain bounded on any fixed time interval $0 \leq t_m \leq T$, as $\Delta t \to 0$. In terms of the growth factor G, this translates to

$$|G|^m \leq f(t_m) \quad \text{for all } \; m, \, \Delta t \text{ and } \kappa, \tag{1.15}$$

where $f = f(t)$ is a continuous function which does not depend on neither Δt or Δx. Clearly (1.10) is the special example of this corresponding to the simple choice $f \equiv 1$. See item (c) for another example.

Typically G is a function of κ, Δx, Δt, and any parameters in the scheme (such as δ in the current case). In order to prevent errors in the calculation from growing, it is necessary that $|G| \leq 1$ apply — or, more generally, (1.15).

If (with suitable restrictions on the scheme parameters, Δx, and Δt) this can be achieved, then we say that **the scheme is stable.** (1.16)

A scheme will be useful if we can do this, while simultaneously letting the scheme equations become better and better approximations to the equations that we want to solve (consistency can be implemented). This *requires that the stability restrictions allow Δx and Δt to vanish, but may also impose restrictions on the scheme parameters* — e.g.: $\delta \, \Delta t \to 0$ in (1.11).

For some schemes, such as the "bad" scheme of the GBNS_lecture script in the 18.311 MatLab Toolkit, (1.16) is impossible. Such schemes are called **unstable.** An unstable scheme is useless, because it cannot keep the errors under control, which grow till they overwhelm the solution.

(c) When the solutions of the pde the scheme aims to solve are bounded — e.g.: (1.1), then it is reasonable to require $|G| \leq 1$ for stability, as in (1.10). However, when this is not true, the definition in (1.15 – 1.16) must be used.

For example, consider (see § 1.3) the equation $u_t + u_x = u$. In this case the solutions, $u = e^t \, f(x - t)$, grow exponentially. Hence a restriction of the form $|G| \leq 1$ would be unreasonable for a numerical scheme. The appropriate thing here is to require (1.15) with f an exponential, which occurs if we require

$$|G| \leq 1 + a \, \Delta t, \quad \text{for some constant } a > 0. \tag{1.17}$$

This guarantees (1.15), because $\ln(1 + a \, \Delta t) \leq a \, \Delta t$, so that $\boldsymbol{|G|^m \leq e^{m \, a \, \Delta t} = e^{a \, t_m}}$.

A final point regarding the example in item (c) above: it may seem that allowing the errors to grow like $e^{a \, t}$, with (possibly) $a > 1$ would create a problem when the solution itself grows at a rate e^t. This is true, however, only if we want to calculate the solution for a very long time.[6] For a "finite" time, since the errors "start" small, this is not a show-stopper for convergence. Nevertheless, *it is always desirable — see (1.13) — to have the values of κ away from zero (particularly the high frequencies) killed,* since these cannot be computed accurately by the scheme (any scheme).

[4] For $\kappa = \alpha \, \Delta x$, $e^{i \, \kappa \, n} = e^{i \, \alpha \, (x_n - x_0)}$. Thus u_n^m and v_n^m in (1.6) follow from smooth functions of x evaluated on the grid, for which approximating derivatives via finite differences makes sense.

[5] See the section on *Convergence of numerical schemes* in *Various lecture notes for 18311.*

[6] Which is, generally, not easy to do accurately — particularly if the solutions grow in amplitude.

1.1.2 von Neumann normal modes and consistency

For a constant coefficients linear pde, a Fourier mode in space (i.e.: space dependence via an exponential $e^{i\,k\,x}$) has an associated time dependence of the form $e^{\mu\,t}$, where $\mu = \mu(k)$ is a function of the **wave number** k — when $\mu = -i\,\omega$ is purely imaginary, ω is called the **wave frequency**. When these Fourier modes are evaluated on the numerical grid, as in equation (1.2), they give rise to a dependence on the grid indexes of the form

$$u_n^m \propto \exp(i\,k\,x_n + \mu\,t_m) \propto \exp(i\,k\,\Delta x\,n + \mu\,\Delta t\,m) = (G_e)^m\,e^{i\,\kappa\,n}, \tag{1.18}$$

where $\kappa = k\,\Delta x$ and $G_e = e^{\mu\,\Delta t}$. *This has the same form as the dependence in equation (1.6), with the numerical G replaced by the exact G_e.* For the the numerical solution to approximate the actual solution of the pde, it must be that (this is what **consistency** means)

$$\boldsymbol{G_e} \textbf{ and } \boldsymbol{G} \textbf{ are close to each other when } \boldsymbol{\Delta t} \textbf{ and } \boldsymbol{\Delta x} \textbf{ are small.} \tag{1.19}$$

To be more precise, in general G is a function of κ, Δx, Δt, and whatever parameters the scheme equations include. That is $G = G(\kappa,\,\Delta t,\,\Delta x,\,\text{parameters})$. Then what is needed is

$$\boldsymbol{G(\kappa,\,\Delta t,\,\Delta x,}\ \text{parameters}\boldsymbol{) = 1 + \mu(k)\,\Delta t + \epsilon\,\Delta t,} \tag{1.20}$$

where $\epsilon \to 0$ as Δx and Δt vanish, with k fixed and $\kappa = k\,\Delta x$.

Remark 1.2 *(1.20) is precisely what is needed to get $G^m \to e^{\mu\,t}$ as $m \to \infty$, with $\Delta t = \frac{t}{m}$. Thus the scheme solutions converge to the true solution, provided that the scheme is stable — where stable means that the high frequency$^{\#}$ contributions can be neglected.* ♣
$^{\#}$That is, contributions from values of κ which are not $O(\Delta x)$, as assumed in (1.20)).

Let us check (1.20) for the case of (1.8). Here $\mu = \pm i\,k$ so that:

$$\begin{aligned} G_e &= \exp(\pm i\,k\,\Delta t) \\ &= 1 \pm i\,k\,\Delta t - \frac{1}{2}\,k^2\,(\Delta t)^2 \mp \frac{1}{6}\,i\,k^3(\Delta t)^3 + O\left((\Delta t)^4\right), \qquad (1.21) \\ G &= 1 - 2\,\delta\,\lambda^2\,\sin^2\left(\frac{k\Delta x}{2}\right) \pm 2\,i\,\lambda\,\sin\left(\frac{k\Delta x}{2}\right) \\ &= 1 \pm i\,k\,\Delta t - \frac{1}{2}\,\delta\,k^2\,(\Delta t)^2 \mp i\,\frac{1}{24}\,k^3\,(\Delta t)(\Delta x)^2 + O\left((\Delta t)^2(\Delta x)^2,(\Delta t)(\Delta x)^4\right). \qquad (1.22) \end{aligned}$$

It is then clear that ϵ above in (1.20) satisfies[7]

$$\epsilon = O\left(\delta\,\Delta t,(\Delta x)^2\right), \quad \text{i.e.: the scheme is consistent.} \tag{1.23}$$

Notice that, by taking $\delta = 1$, we can have agreement of the first three terms in the expansions for G_e and G above in (1.21 – 1.22), not just the first two. This makes the scheme higher order (thus, in principle, more accurate). *Unfortunately, it also makes it unstable* — as can be seen from equation (1.9), which shows that $|G| > 1$ for all $\lambda \neq 0$ and $\kappa \neq 0$.

1.1.3 The CFL (Courant-Friedrichs-Lewy) condition

A necessary condition, that all convergent schemes must satisfy, is the CFL condition: **The numerical domain of dependence <u>must</u> include the p.d.e. domain of dependence.**
The reason why this is needed should be obvious: a numerical scheme cannot possibly compute the correct solution

[7] Recall that we assume $\delta\,\Delta t \ll 1$ — see below (1.5).

(converge) if it does not have access to the data that determines the solution. This conditions says that *a numerical scheme must respect the speeds for information transport in the physical problem.*

For explicit schemes and hyperbolic problems (with finite speeds of propagation) the CFL condition translates into *limits on the time step* of the form $\Delta t \leq c\,\Delta x$, where c is related to the maximum propagation speed in the problem, and the shape of the scheme stencil.

For explicit schemes in problems with infinite speeds of propagation (e.g.: the heat equation) *restrictions on the time step, of the form* $\Delta t \leq c\,(\Delta x)^p$ *— where* $p > 1$ *— are needed.* This tends to make explicit schemes not very efficient, as it forces very small time steps (even if not needed for accuracy).

1.2 Example: implicit forward differences for $u_t + u_x = 0$

The aim is to solve the equation

$$u_t + u_x = 0. \tag{1.24}$$

To do so we replace the equation by the following finite difference approximation

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} + \frac{u_{n+1}^{m+1} - u_n^{m+1}}{\Delta x} = 0, \tag{1.25}$$

where we use the same notation introduced in (1.2). Expand the solution to the equation as follows

$$\begin{array}{lllll} u_n^m & = & u(x_n,\, t_{m+1} - \Delta t) & = & u_n^{m+1} - \Delta t\, u_t + O((\Delta t)^2), \\ u_{n+1}^{m+1} & = & u(x_n + \Delta x,\, t_{m+1}) & = & u_n^{m+1} + \Delta x\, u_x + O((\Delta x)^2), \end{array}$$

where u_t and u_x are evaluated at $(x_n,\, t_{m+1})$. Substituting this into (1.25) shows that it approximates the equation with an error $O(\Delta x,\, \Delta t)$. Hence, (1.25) gives rise to the *consistent* scheme

$$\boldsymbol{(1 - \lambda)\, u_n^{m+1} + \lambda\, u_{n+1}^{m+1} = u_n^m,} \quad \text{where} \quad \boldsymbol{\lambda = \frac{\Delta t}{\Delta x} > 0.} \tag{1.26}$$

But, **can λ be selected so that the scheme is stable ?** We investigate this question below.

Remark 1.3 *Note that (1.26) is consistent* precisely *because of (1.25). That is: the scheme is equivalent to a finite differences approximation to the equation. The* **order of the scheme** *is given by the order of the errors in this approximation — in this case first order in both space and time.*
The order of a scheme characterizes the size of the errors that computing with the scheme produces (how good the solution is) — as long as the scheme is stable. ♣

Remark 1.4 *Note that this scheme is* **implicit.** *That is, to find the solution at time* t_{m+1} *a system of (in this case linear) equations has to be solved. There is no expression that gives, directly,* u_n^{m+1} *in terms of* u_n^m*. Implicit schemes are more expensive (per time step) to implement than explicit schemes. On the other hand, they tend to have better stability properties — they allow larger time steps than explicit schemes (in some cases much, much, larger). See* § *1.1.3.* ♣

To perform a von Neumann stability analysis, we now **look for solutions to the scheme of the form $u_n^m = G^m e^{i\kappa n}$,** where G is a constant to be found and $-\pi \leq \kappa \leq \pi$ is the grid wave number. Substituting this into the scheme equation, we find $\left(1 - \lambda + \lambda\, e^{i\kappa}\right) G = 1$. Thus

$$G = \frac{1}{1 - \lambda + \lambda\, e^{i\,\kappa}}. \tag{1.27}$$

Note that $\boldsymbol{1/G}$ lies on a circle of radius $\boldsymbol{\lambda}$ in the complex plane, centered at $\boldsymbol{1 - \lambda}$. This circle has to be outside the unit circle for G to be inside it. Hence all values of G fall inside the unit disk provided that $\lambda > 1$. We conclude that

$$\textbf{The scheme is stable provided } \boldsymbol{\lambda > \lambda_c = 1.} \tag{1.28}$$

Now notice:

1. For $\boldsymbol{\kappa = 0}$, $\boldsymbol{G = 1}$. See note **1.2**.
2. This **scheme is a little unusual,** in the sense that the *stability requirement is that* Δt *should be "large" enough* — instead of the more common Δt "small" enough. Nevertheless, this is not a problem, as it is still possible to let both Δt and Δx vanish, while satisfying (1.28). This is all that is needed in order to get a convergent scheme.
3. Note that, for $\lambda = 1$ (i.e.: $\Delta t = \Delta x$) the scheme is exact: in this case (1.26) is explicit, and it reproduces the fact that the general solution to (1.24) has the form $u = f(x - t)$, for some function f. In this case $G = e^{-i\,\kappa}$.

   However, we have excluded this case from (1.28) based on the *general principle that we always want to have "high frequency", grid scale $\kappa \approx \pi$, oscillations killed by a numerical scheme.* In the particular case of this very special scheme, for this very special equation, one can get away with not doing this (i.e.: use $\lambda = 1$). But, in general, this is not a good idea.
4. Note that the scheme in (1.26) is *singular* for $\lambda = 1/2$, that is: the equations determining u_n^{m+1} may not have a solution. This is easy to see from (1.27), since $G = \infty$ for $\lambda = 1/2$ and $\kappa = \pi$. This is a very *extreme case of instability.*
5. This **scheme is implemented by the script scheme01i in the 18311 MatLab Toolkit.** You should check the behavior of the scheme for various values of λ. For example:
   - For $\lambda < 1$ the scheme is unstable. However, when λ is very close to one (say: $\lambda = 0.99$), the instabilities grow very slowly, unless Δx is taken fairly small.
   - For $\lambda > 1$ the scheme is stable. However, when λ is large (say: $\lambda = 8$), the scheme is very dissipative. Everything decays rather fast, unless Δx is taken fairly small.
   - For $\lambda = 1$ the scheme is "exact". Check what happens with the discontinuous initial data with the script scheme01i — only for $\lambda = 1$ the discontinuity is not smeared by the scheme (this is particularly bad for λ large).
6. Finally, we check (1.20)

$$\begin{aligned} G_e &= \exp(-i\,k\,\Delta t) \;=\; 1 - i\,k\,\Delta t - \frac{1}{2}\,k^2\,(\Delta t)^2 + \ldots \\ G &= \frac{1}{1-\lambda+\lambda\,e^{i\,k\,\Delta x}} \;=\; \frac{1}{1 + i\,k\,\Delta t - \frac{1}{2}\,k^2\,\Delta t\,\Delta x + \ldots} \\ &= 1 - i\,k\,\Delta t + \frac{1}{2}\,k^2\,\Delta t\,\Delta x - k^2\,(\Delta t)^2 + \ldots \end{aligned} \tag{1.29}$$

Thus $\epsilon = O(\Delta t,\, \Delta x)$ in (1.20).

1.3 Example: backward differences for $u_t + u_x = u$

This subsection is yet to be written.

THE END.