Numerical Methods in Linear Algebra

Dénes Sexty

University of Graz

2020-2021, winter term

UNIVERSITÄT GRAZ
UNIVERSITY OF GRAZ

1. Examples: Bridge, multidim Newton's method, QM, vibrational eigenmodes
2. Linear equations
3. Gauss elimination, Pivoting
4. Numerical errors, condition number
5. LU decomposition, iterative improvement
6. Householder reduction
7. iterative solution: Gauss-Seidel, Successive overrelaxation (SOR), Conjugate gradient (CG) and others
8. Eigenvalues Subspace methods, power iteration, Lánczos method, Krylov Schur
9. Eigenvalues transformation method: Jacobi, QR
10. Diagonalization with Fourier transformation
11. Partial Differential Equations: Boundary value problems, Finite elements
12. **Partial Differential Equations: initial value problems. Courant condition, Crank-Nicholson scheme, Split-Step method**
13. **Outlook: Advanced techniques**

- $B^2 - AC < 0$: *elliptic* PDEs.
  The solutions are smooth if the coefficients are smooth. e.g.: Laplace equation $\triangle f = 0$ has analytic solutions in a region even if the values on the border are not smooth.
  prototypical example: **Poisson equation** for u electrostatic potential, $\rho(x, y)$ charge distribution

$$u_{xx} + u_{yy} = -\rho(x, y) \tag{1}$$

- $B^2 - AC = 0$: *parabolic* PDEs.
  Can be transformed to the **heat equation**. Solutions smooth out as time is increased. heat eq. for u temperature and κ diffusion coefficient:

$$u_t = \kappa u_{xx} \tag{2}$$

- $B^2 - AC > 0$: *hyperbolic* PDEs:
  Wave equations. Discontinuities in the initial conditions are retained for later times. Wave eq. with speed of light (or sound, etc.) v is:

$$u_{tt} = c^2 u_{xx} \tag{3}$$

- **Boundary value problems** (aka. static problem) e.g.: Poisson equation
  a quantity $u(x, y)$ (or its derivative or some other property) is given on the boundary of some region, $u(x, y)$ inside the region is to be calculated.

  Usually some kind of iterative procedure is employed, one looks for efficient solutions.
  After discretisation of linear PDEs one gets a system of linear equations, typically with a sparse matrix.

- **initial value problems** (aka. time evolution problem) e.g: Heat equation, wave equation
  $u(x, t)$ is known at initial times and its time evolution is to be calculated.
  $u(x, t > 0)$ must also be given on the boundary of the region of interest (periodical boundary conditions are also possible)

  Main concern here is to devise algorithms which give a stable solution.

Diffusion equation ($D(x)$ is the diffusion constant)

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(D(x)\frac{\partial u(x,t)}{\partial x}\right) \tag{4}$$

Schrödinger equation

$$i\hbar\frac{\partial \Psi(x,t)}{\partial t} = \hat{H}\Psi(x,t), \quad \hat{H} = -\frac{\hbar^2}{2m}\triangle + V(x) \tag{5}$$

Klein-Gordon equation

$$\frac{\partial^2 \Phi(x,t)}{c^2 \partial t^2} = \triangle\Phi(x,t) - m^2\Phi(x,t) - \lambda\Phi(x,t)^3 \tag{6}$$

When $m = 0$ and $\lambda = 0$, it's also called the wave-equation.
Flux-Conservative problems in 1 Dimension are goverend by the equation

$$\frac{\partial u(x,t)}{\partial t} = -\frac{\partial F(u)}{\partial x} \tag{7}$$

where u and $F(u)$ are vectors

Wave equation can be cast as Flux-Conservative problem:

$$\frac{\partial^2 \Phi(x,t)}{c^2 \partial t^2} = \frac{\partial^2 \Phi(x,t)}{\partial x^2} \tag{8}$$

We define

$$r(x,t) = c\frac{\partial \Phi(x,t)}{\partial x}, \quad s(x,t) = \frac{\partial \Phi(x,t)}{\partial t} \tag{9}$$

This implies (using Young Theorem and the wave equation):

$$\frac{\partial r}{\partial t} = c\frac{\partial s}{\partial x}, \quad \frac{\partial s}{\partial t} = c\frac{\partial r}{\partial x} \tag{10}$$

We can put these two equations in a vector equation:

$$\mathbf{u} = \begin{pmatrix} r \\ s \end{pmatrix}, \quad \mathbf{F}(\mathbf{u}) = \begin{pmatrix} 0 & -c \\ -c & 0 \end{pmatrix} \mathbf{u} \tag{11}$$

We look at naive discretisation of the simplest flux-conservative PDE:

$$\frac{\partial u(x,t)}{c\partial t} = -\frac{\partial u(x,t)}{\partial x} \tag{12}$$

Discretise coordinates:

$$x_j = j\Delta x, \quad t_n = n\Delta t, \quad j, n = 0, 1, 2, \ldots \tag{13}$$

Using $u_j^n = u(x_j, t_n)$, we discretise the derivatives:

$$\begin{aligned} \left.\frac{\partial u}{\partial t}\right|_{j,n} &= \frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t) \text{ Forward discretisation} \\ \left.\frac{\partial u}{\partial x}\right|_{j,n} &= \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + O(\Delta x^2) \text{ Centered discretisation} \end{aligned} \tag{14}$$

The equation of motion (EoM) becomes:

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{c\Delta t} &= -\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \\ u_j^{n+1} &= u_j^n - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) \end{aligned} \tag{15}$$

This discretisation scheme is called **FTCS** (Forward time centered space)
It's an explicit scheme: u_j^{n+1} can be calculated explicitly from older fields.
Stability can be a problem
von Neumann Stability analysis
General tool to investigate stability of discretisations of initial value problems.
Ansatz for $u(x,t)$

$$u_j^n = \xi(k)^n e^{ikj\Delta x} \tag{16}$$

with the amplification factor $\xi(k)$
Inserted into the eom:

$$\left(\xi(k)^{n+1} - \xi(k)^n\right) e^{ikj\Delta x} = \frac{c\Delta t}{\Delta x}\xi(k)^n \frac{e^{-ik\Delta x} - e^{-ik\Delta x}}{2} e^{ikj\Delta x} \tag{17}$$

Solve for $\xi(k)$

$$\xi(k) = 1 - i\frac{c\Delta t}{\Delta x}\sin(k\Delta x) \implies |\xi(k)| = \sqrt{1 + \frac{c^2\Delta t^2}{\Delta x^2}\sin^2(k\Delta x)} \tag{18}$$

$|\xi(k)| > 1$ divergent modes
FTCS is not stable

Idea: replace u_j^n in the time derivative with $\frac{1}{2}(u_{j-1}^n + u_{j+1}^n)$ (close to the continuum limit this should be OK.)

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) \tag{19}$$

We can repeat the stability analysis:

$$\begin{aligned} \xi(k)^{n+1} e^{ikj\Delta x} &= \xi(k)^n e^{ikj\Delta x}\left[\frac{e^{ik\Delta x} + e^{-ik\Delta x}}{2} - \frac{c\Delta t}{x}\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2}\right] \\ \xi(k) &= \cos(k\Delta x) - i\frac{c\Delta t}{\Delta x}\sin(k\Delta x) \\ |\xi(k)|^2 &= \cos^2(k\Delta x) + \left(\frac{c\Delta t}{\Delta x}\right)^2 \sin^2(k\Delta x) \end{aligned} \tag{20}$$

The discretisation is stable if we have $|\xi(k)| < 1 \implies \frac{c\Delta t}{\Delta x} < 1$. Thus we need to choose Δt accourding to

$$\textbf{Courant condition:} \quad \Delta t < \Delta x/c \tag{21}$$

Alternatively, we can also improve on the time discretisation:

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t) \text{ Forward discretisation} \tag{22}$$

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + O(\Delta t^2) \text{ Leap-frog discretisation}$$

This leads to the EoM:

$$u_j^{n+1} = u_j^{n-1} - \frac{c\Delta t}{\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) \tag{23}$$

Stability analysis leads again to $\Delta t < \Delta x / c$
To start the solution we need u_j^0 and u_j^1 one can calculate u_j^1 from u_j^0 with several small steps using the Lax discretisation.

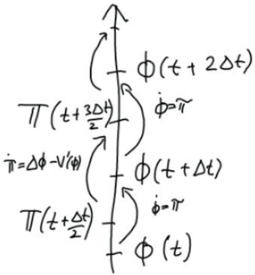
Here the R.H.S. is not a divergence because of the m^2 and Φ^3 terms

$$\frac{\partial^2\Phi(x,t)}{c^2\partial t^2} = \triangle\Phi(x,t) - m^2\Phi(x,t) - \lambda\Phi(x,t)^3 \tag{24}$$

Using momenta we can break up this into two equations:

$$\begin{aligned}\partial_t\Phi(x,t) &= \pi(x,t) \\ \partial_t\pi(x,t) &= \triangle\Phi(x,t) - m^2\Phi(x,t) - \lambda\Phi^3(x,t)\end{aligned} \tag{25}$$

This suggests the Leap-Frog discretisation (using centered derivatives, the order is improved)



Eliminating π we get the EoM:

$$\begin{aligned}\Phi(x,t+\Delta t) &= 2\Phi(x,t) - \Phi(x,t-\Delta t) \\ &\quad + \triangle\,\Phi(x,t) + m^2\Phi(x,t) + \lambda\Phi(x,t)\end{aligned}$$

For the Laplace operator we can use the usual discretisation or an improved one:

$$\triangle_{imp}\Phi(x,t) = \sum_{i,\pm} A\Phi(x\pm 2a_i,t) + B\Phi(x\pm a_i,t) + C\Phi(x,t)$$

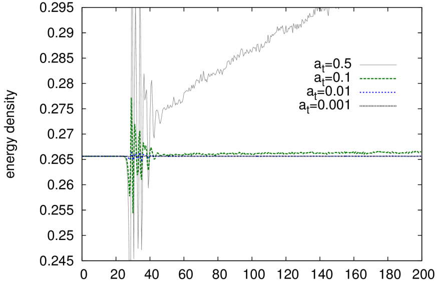
with $A = -\frac{1}{12},\ B = \frac{4}{3},\ C = -\frac{5}{2}$

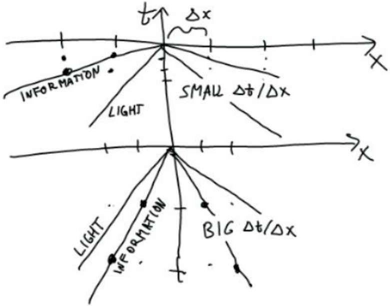
asymmetric lattice $a_i = a_s \quad a_0 = a_t$

Courant condition $\frac{a_t}{a_s} < 0.1$ Otherwise the solution is linearly instable

Speed of light should "fit into lattice"

Energy conservation fulfilled in limit $a_t \rightarrow 0$





or heat equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \tag{26}$$

We use the following discretisation for the second derivative

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{j,n} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} + O(\Delta x^2) \text{ Centered discretisation} \tag{27}$$

So the FTCS discretisation reads

$$u_j^{n+1} = u_j^n + \frac{D\Delta t}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{28}$$

Stability analysis using $2\sin^2(\phi/2) = 1 - \cos\phi$:

$$\xi(k) = 1 + \frac{D\Delta t}{\Delta x^2}(e^{ik\Delta x} + e^{-ik\Delta x} - 2) = 1 - \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2) \tag{29}$$

$|\xi| < 1$ if we have $\frac{D\Delta t}{\Delta x^2} \leq \frac{1}{2}$

$$u_j^{n+1} = u_j^n + \frac{D\Delta t}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{30}$$

Now inserting the Taylor expansion of the field centered on u_j^n we get

$$u_t\Delta t + \frac{1}{2}u_{tt}\Delta t^2 + \ldots = \frac{D\Delta t}{\Delta x^2}(u_{xx}\Delta x^2 + \frac{1}{12}u_{xxxx}\Delta x^4 + \ldots) \tag{31}$$

For $\Delta t \to 0$ and $\Delta x \to 0$ we get back the original PDE, so the discretisation is **consistent**

Nonzero corrections come with Δt so the scheme is **first order** in time, and Δx^2 corrections show its second order in space.

A scheme is **stable** if we have $|\xi(k)| < 1$ for all k

A scheme is **convergent** if the solution approaches the exact solution of the PDE for $\Delta t, \Delta x \to 0$.

One can prove that for a linear initial value problem, the stability and consistency of a finite difference scheme is necessary and sufficient for its convergence (Lax equivalence theorem)

No theorem for non linear PDEs, but they generally converge if they are stable and consistent.

$|\xi| < 1$ if we have $\frac{D\Delta t}{\Delta x^2} \leq \frac{1}{2}$

$$\Delta t \leq \frac{1}{2}\frac{\Delta x^2}{D} \tag{32}$$

Diffusion time over length λ given by:

$$\tau \sim \frac{\lambda^2}{D} \tag{33}$$

If we want to describe distances of scale λ we need to calculate N_t time steps

$$N_t = \frac{\tau}{\Delta t} \sim \frac{\lambda^2}{\Delta x^2} \tag{34}$$

Since we have $\lambda \gg \Delta x$, N_t can be quite large.
Can we come up with a better discretisation scheme?

We can change the discretisation of the time derivtive:

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t) \text{ Forward discretisation} \tag{35}$$

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n} - u_j^{n-1}}{\Delta t} + O(\Delta t) \text{ Backward discretisation}$$

This gives the EoM:

$$u_j^n = u_j^{n-1} + \frac{D\Delta t}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{36}$$

To get u_j^n from u_j^{n-1}, we need to solve a system of linear equations. **fully implicit** scheme
Stability:

$$1 = \frac{1}{\xi(k)} - \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2) \tag{37}$$

$$\xi(k) = \frac{1}{1 + \frac{4D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2)}$$

Stable for any Δt

We can improve accuracy if we combine backward an forward in time schemes:

$$u_j^{n+1} = u_j^n + \frac{D\Delta t}{2\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n + u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) \tag{38}$$

Both sides are now centered on $(N+1/2)\Delta t$, so the equation is valid with $O(\Delta t^2)$ corrections, better than both the fully explicit and the fully implicit schemes. Space accuracy is also second order.
Defining $H_{jk} = \frac{D\Delta t}{\Delta x^2}(\delta_{j+1,k} - 2\delta_{jk} + \delta_{j-1,k})$, we can write the EoM as

$$\sum_k (1-H)_{jk} u_k^{n+1} = \sum_l (1+H)_{jl} u_l^n \tag{39}$$

$$u^{n+1} = (1-H)^{-1}(1+H)u^n$$

Stability analysis:

$$\xi(k) = \frac{1 - 2\frac{D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2)}{1 + 2\frac{D\Delta t}{\Delta x^2}\sin^2(k\Delta x/2)} \tag{40}$$

Stable for any Δt

using units such that $\hbar = 1, m = 1/2$

$$i\frac{\partial \Psi(x,t)}{\partial t} = -\frac{\partial^2 \Psi(x,t)}{\partial x^2} + V(x)\Psi(x,t) \tag{41}$$

fully implicit discretisation:

$$i\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -\frac{\Psi_{j+1}^{n+1} - 2\Psi_j^{n+1} + \Psi_{j-1}^{n+1}}{\Delta x^2} + V_j\Psi_j^{n+1} \tag{42}$$

using $i\partial_t \Psi = \hat{H}\Psi$ with $H = -\partial_x^2 + V(x)$

$$(1 + iH\Delta t)\Psi_j^{n+1} = \Psi_j^n \tag{43}$$

Problem: The norm of the wave function is not conserved.

$$\int_{-\infty}^{\infty} |\Psi(x,t)|^2 dx = 1 \tag{44}$$

This is becouse the evolution operator is not unitary

$$(1 + iH\Delta t)^\dagger = 1 - iH\Delta t \neq (1 + iH\Delta t)^{-1} \tag{45}$$

The explicit scheme has the same problem:

$$\Psi_j^{n+1} = (1 - iH\Delta t)\Psi_j^n \tag{46}$$

Formal solution using $i\partial_t \Psi = \hat{H}\Psi$ with $H = -\partial_x^2 + V(x)$

$$\Psi(x, t + \Delta t) = e^{-i\hat{H}\Delta t}\Psi(x, t) \tag{47}$$

Both explicit and implicit schemes are approximations of the evolution operator

$$e^{-i\hat{H}\Delta t} \approx 1 - iH\Delta t \approx (1 + iH\Delta t)^{-1} \tag{48}$$

We need to find unitary approximation to $e^{-iH\Delta t}$, which is given by:

$$e^{-i\hat{H}t} \approx \frac{1 - \frac{1}{2}i\hat{H}\Delta t}{1 + \frac{1}{2}i\hat{H}\Delta t} \tag{49}$$

The EoM is than given by:

$$\left(1 + \frac{1}{2}i\hat{H}\Delta t\right)\Psi_j^{n+1} = \left(1 - \frac{1}{2}i\hat{H}\Delta t\right)\Psi_j^n \tag{50}$$

this is the same as the Crank-Nicholson scheme that we had for diffusion eq.

$$\left(1 + \frac{1}{2} i\hat{H}\Delta t\right) \Psi_j^{n+1} = \left(1 - \frac{1}{2} i\hat{H}\Delta t\right) \Psi_j^{n} \tag{51}$$

The inverse of $1 + \frac{1}{2} i\hat{H}\Delta t$ can be calculated e.g. by LU decomposition. Decomposing needs to be calculted once, then reused in every time step. Alternatively, in 1D, using Dirichelet boundary conditions, $1 + \frac{1}{2} i\hat{H}\Delta t$ is tridiagonal:

$$\begin{pmatrix} b_1 & c_1 & & & \\ \ddots & \ddots & \ddots & & \\ & a_j & b_j & c_j & \\ & & \ddots & \ddots & \ddots \\ & & & a_N & b_N \end{pmatrix} \tag{52}$$

The LU decomposition (without pivoting) for such tridiagonal matrices, solving $Mx = r$, starting with $u_1 = b_1,\ y_1 = r_1$

$$\underbrace{l_j = a_j / u_{j-1}, \quad u_j = b_j - l_j c_{j-1},}_{\text{decomposition}} \quad \underbrace{y_j = r_j - l_j y_{j-1}}_{\text{forward substitution}} \tag{53}$$

back-substitution:

$$x_n = y_n / u_n, \quad x_j = (y_j - c_j x_{j+1}) / u_n \text{ for } j = n-1, \ldots, 1 \tag{54}$$

Consider diffusion equation in two dimensions

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{55}$$

The Crank-Nicholson scheme

$$u_{j,l}^{n+1} = u_{j,l}^{n} + \frac{1}{2}\frac{D\Delta t}{\Delta x^2}\left(\delta_x^2 u_{j,l}^{n+1} + \delta_y^2 u_{j,l}^{n+1} + \delta_x^2 u_{j,l}^{n} + \delta_y^2 u_{j,l}^{n}\right) \tag{56}$$

$$\delta_x^2 u_{j,l}^{n} = u_{j+1,l}^{n} - 2u_{j,l}^{n} + u_{j-1,l}^{n}$$

For the solution of $u_{j,l}^{n+1}$ we have to decompose the a large matrix. The matrix is sparse but no longer tridiagonal

Alternating-direction implicit (ADI) method. Still second order in time and space and unconditionally stable.

$$u_{j,l}^{n+1/2} = u_{j,l}^{n} + \frac{1}{2}\frac{D\Delta t}{\Delta x^2}\left(\delta_x^2 u_{j,l}^{n+1/2} + \delta_y^2 u_{j,l}^{n}\right) \tag{57}$$

$$u_{j,l}^{n+1} = u_{j,l}^{n+1/2} + \frac{1}{2}\frac{D\Delta t}{\Delta x^2}\left(\delta_x^2 u_{j,l}^{n+1/2} + \delta_y^2 u_{j,l}^{n+1}\right)$$

A tridiagonal solution is needed in both steps.

Generally, considder an initial value problem

$$\frac{\partial u}{\partial t} = Lu \tag{58}$$

where L is some operator:

$$Lu = L_1 u + L_2 u + \ldots + L_m u \tag{59}$$

Suppose you know a differencing scheme for each of the L_i valid if that was the only term on the RHS:

$$u^{n+1} = U_i(u^n, \Delta t) \tag{60}$$

Now we use the following scheme:

$$\begin{aligned} u^{n+1/m} &= U_1(u^n, \Delta t) \\ u^{n+2/m} &= U_2(u^{n+1/m}, \Delta t) \\ &\vdots \\ u^{n+1} &= U_m(u^{n+(m-1)/m}, \Delta t) \end{aligned}$$

Splitting could be advantegous for and eq like

$$\frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x} + D\frac{\partial^2 u}{\partial x^2} \tag{61}$$

The ADI method is an other variation of this idea:
suppose U_i is the differencing scheme that involves all terms on the RHS, but is only stable with respect to the term L_i
Than the updateing is written as:

$$\begin{aligned}
u^{n+1/m} &= U_1(u^n, \Delta t/m) \\
u^{n+2/m} &= U_2(u^{n+1/m}, \Delta t/m) \\
&\vdots \\
u^{n+1} &= U_m(u^{n+(m-1)/m}, \Delta t/m)
\end{aligned}$$

In practice it is often enough to have a stable scheme for some of the terms on the RHS, most notably the one with the highest number of spatial derivatives.

aka. Gross-Pitaevskii equation. Governs the time evolution of Bose-Einstein condensates (Dilute bose gas, where interaction of the atoms are dominted with s-channel interaction)

$$i\hbar\frac{\partial\Psi(x,t)}{\partial t} = -\frac{1}{2m}\triangle\Psi(x,t) + V(x)\Psi(x,t) + U|\Psi(x,t)|^2|\Psi(x,t) \tag{62}$$

We often consider the case where $V(x) = 0$
Particle number conserved:

$$n_{tot} = \int d^d x|\Psi(x,t)|^2 \tag{63}$$

Here we also need a unitary time evolution

The Crank-Nicholson scheme is given by:

$$i\frac{\Psi_j^{n+1}-\Psi_j^n}{\Delta t} = \frac{1}{2}\left(-\frac{\Psi_{j+1}^{n+1}-2\Psi_j^{n+1}+\Psi_{j-1}^{n+1}}{\Delta x^2} + U|\Psi_j^{n+1}|^2\Psi_j^{n+1} - \frac{\Psi_{j+1}^{n}-2\Psi_j^{n}+\Psi_{j-1}^{n}}{\Delta x^2} + U|\Psi_j^n|^2\Psi_j^n\right)$$

Stability analysis shows that it is an unconditionally stable scheme.

It's also non-linear, how do we solve it? Iteratively
Using $H = -\triangle$, we write

$$\left(1+\frac{1}{2}i\hat{H}\Delta t\right)\Psi_j^{n+1} = \left(1-\frac{1}{2}i\hat{H}\Delta t\right)\Psi_j^n + \frac{1}{2}\left(U|\Psi_j^{n+1}|^2\Psi_j^{n+1} + U|\Psi_j^n|^2\Psi_j^n\right) \quad (64)$$

The right hand side we take as given, using $\Psi_j^{n+1} = \Psi_j^n$, and solve for Ψ_j^{n+1} by inverting $1+\frac{1}{2}iH\Delta t$ (with e.g. LU decomposition).
Now use the new Ψ_j^{n+1} in the RHS, and solve again for Ψ_j^{n+1}
Repeat until convergence

We want to calculate $e^{-iH\Delta t}\Psi(x,t)$
With a hermitian operator H, however there are two terms in H, which do not commute

$$H = \underbrace{-\frac{1}{2m}\triangle}_{=H_1} + \underbrace{U|\Psi(x,t)|^2}_{=H_2} \tag{65}$$

Baker-Campbell-Hausdorf:

$$e^{(A+B)\Delta t} = e^{A\Delta t}e^{B\Delta t}e^{-\frac{1}{2}[A,B]\Delta t^2} + O(\Delta t^3) \quad \Longrightarrow \quad e^{(A+B)\Delta t} = e^{A\Delta t}e^{B\Delta t} + O(\Delta t^2)$$

Higher order discretisation: $e^{i(A+B)\Delta t} = e^{iA\Delta t/2}e^{iB\Delta t}e^{iA\Delta t/2} + O(\Delta t^3)$
One can show that

$$e^{-iH_1\Delta t/2}\dots e^{-iH_L\Delta t/2}e^{-iH_L\Delta t/2}\dots e^{-iH_1\Delta t/2} = e^{-i\sum H_i\Delta t} + O(\Delta t^3)$$

Doing many steps with the higher order formula:

$$\prod e^{iA\Delta t/2}e^{iB\Delta t}e^{iA\Delta t/2} = e^{iA\Delta t/2}\left(\prod e^{iB\Delta t}e^{iA\Delta t}\right)e^{iB\Delta t}e^{iA\Delta t/2} \tag{66}$$

Half step only at the very beginning and at the very end $\Longrightarrow$ more accuracy almost free

$$H = \underbrace{-\frac{1}{2m}\triangle}_{=H_1} + \underbrace{U|\Psi(x,t)|^2}_{=H_2} \tag{67}$$

So we approximate:

$$\Psi(x, t+\Delta t) \approx e^{-iH_2\Delta t} e^{-iH_1\Delta t}\Psi(x,t) \tag{68}$$

(for higher order approximation, do half steps at the beginning and end, see above)
The second operator to use:

$$e^{-iH_2\Delta t} = e^{-iU|\Psi(x,t)|^2} \tag{69}$$

is diagonal in space coordinates $\implies$ trivially implemented.
The first operator is diagonal in k-space: $\triangle \to -k^2$ in Fourier space

$$\Psi(x, t+\Delta t) \approx e^{-iH_2\Delta t} F^{-1}\left[e^{-ik_{LAT}^2\Delta t/2m} F\left[\Psi(x,t)\right]\right] \tag{70}$$

with F the Fourier transformation

Using FFT, a Fourier transformation takes $O(N \log N)$ operations

when multiplying with k^2, the lattice version for the used discretisation has to be used

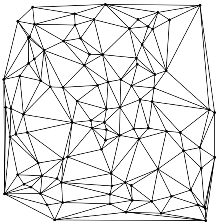
$$\Psi(x+a) - 2\Psi(x) + \Psi(x-a) \quad \Longrightarrow \quad k_{LAT}^2 = (2\sin(\pi k/N))^2, \quad k = 0, \dots, N-1$$

In d dimensions, similarly:

$$k_{LAT}^2 = \sum_i (2\sin(\pi k_i/N))^2, \quad k_i = 0, \dots, N-1 \tag{71}$$

This can also be used for the linear Schrödinger equation. The potential term is than in $H_2 = V(x)$

The space coordinates must remain periodic in order to use the Fourier Transformation. If the potential is large at the boundaryes, this could be OK.



Alternative discretisation scheme for PDEs.
Based on writeing the PDE
as a conservation of some charge:

$$\partial_t A(x,t) + \nabla F(x,t) = S(x,t)$$

with A charge density, F flux and S source terms
Defines an irregular mesh of control volumes,
each cell is called a **control volume**
The conservation law is integrated over the control volumes:

$$\partial_t Q + \text{ Flux on boundary} = \int S \qquad (72)$$

The A, F and S depends on some fields u, and by construction the Flux is built such that there is no loss in the boundaries (i.e. $F_a = -F_b$ where F_a and F_b are the flux through a face connecting two control volumes a and b) The conservation law is then turned into an equation for the u fields

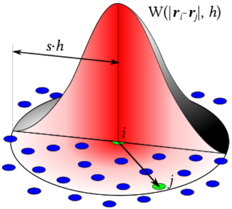
Used often in CFD (Computational Fluid Dynamics)

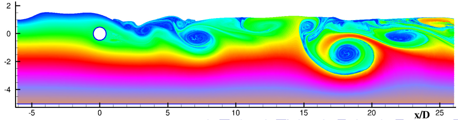
A regular mesh can become broken if the material being simulated moves around (e.g. hydrodinamic flow around a complex object)

Smoothed-particle Hydrodynamics (one of the oldest meshfree methods) the path of particles in a hydrodinamical flow.
The physical properties of the flow are calculated using a kernel function:

$$A(r) = \sum_j V_j A_j W((r - r_j), h) \tag{73}$$

where the sum is over particles, V_j is the volume of the particle, A_j is the quantity A carried by particle, W is the kernel function which has a characteristic length h.
This allows converting e.g. the Euler equation into an EoM for the particles.





Imagine solving the Poisson equation on a fine lattice $\Longrightarrow$ slow momentum modes take long to equilibrate $\rightarrow$ approximate them on a rough grid.
Typical elements of the algorithm:
Residual computation on the fine lattice
Restriction the residual is downsampled to the coarse grid
Solution on the coarse grid
Interpolation of the correction to the fine grid and adding it to the solution