A tutorial on the pseudo-spectral method

H. Isliker,
University of Thessaloniki,
September 2004

Outline

1. Introducing remarks
2. Basic principles of the pseudo-spectral method
3. Pseudo-spectral methods and Fourier transforms
4. Time-dependent problems
5. Non-linearities
6. Aliasing – de-aliasing
7. Examples
8. Concluding remarks

Introducing remarks

- the pseudo-spectral (PS) methods are methods to solve partial differential equations (PDE)
- they originate roughly in 1970
- the PS methods have successfully been applied to
  - fluid dynamics (turbulence modeling, weather predictions)
  - non-linear waves
  - seismic modeling
  - MHD
  - …
- we have applied them to plasma turbulence simulations, and to the non-linear interaction of grav. waves with plasmas (together with I. Sandberg and L. Vlahos … part I of two talks !)

Basic principles of the pseudo-spectral method

- the ‘pseudo-spectral’ in the method refers to the spatial part of a PDE
- example: a spatial PDE

$$Lu(\mathbf{x}) = s(\mathbf{x}), \qquad \mathbf{x} \in V$$
$$\text{b.c.: } f(u(\mathbf{y})) = 0, \qquad \mathbf{y} \in \partial V$$

L: a spatial differential operator (e.g. $L = \partial_{xx} + \partial_{yy}$, etc.)

- wanted: numerical solution $u^N(\mathbf{x})$ such that the residual R

$$R(\mathbf{x}) := Lu^N(\mathbf{x}) - s(\mathbf{x})$$

is small – but how do we define the smallness ?

- general procedure:

  1. choose a finite set of trial functions (expansion functions) ϕ_j, j = 0,..N-1,
  and expand u^N in these functions

$$u^N(\mathbf{x}) = \sum_{j=0}^{N-1} \hat{u}_j \phi_j(\mathbf{x})$$

  2. choose a set of test functions χ_n, k = 0,1,2, … N-1 and demand that

$$(\chi_n, R) = 0 \qquad \text{for } n=0,1\ldots N-1 \text{ (scalar product)}$$

- 'spectral methods' means that the trial functions ϕ_n form a basis for a certain space of global, smooth functions (e.g. Fourier polynomials)
  (global: extending over the whole spatial domain of interest)

- there are various spectral methods, classified according to the test functions χ_n :
  Galerkin method, tau method, collocation or pseudo-spectral method
- collocation or pseudo-spectral method:

$$\chi_n(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{x}_n),$$

  where the $\mathbf{x}_n$ (n=0,1,… N-1) are special points, the collocation points
- the smallness condition for the residual becomes

$$0 = (\chi_n, R) = (\delta(\mathbf{x}-\mathbf{x}_n), R) = R(\mathbf{x}_n) = Lu^N(\mathbf{x}_n) - s(\mathbf{x}_n)$$

$$\sum_{j=0}^{N-1} \widehat{u}_j L\phi_j(\mathbf{x}_n) - s(\mathbf{x}_n) = 0, \quad n=0,1,2,\ldots,N-1$$

N equations to determine the unknown N coefficients $\widehat{u}_j$

- remark: the solution at the collocation points is exact, in between them we interpolate the solution

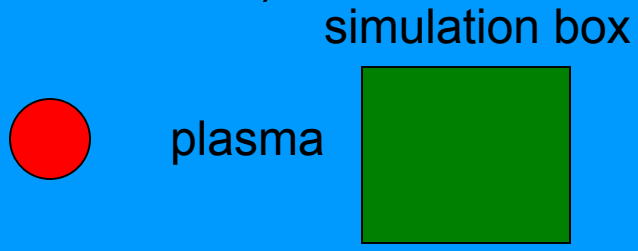
- what trial functions to choose ?

  1. periodic b.c.: trigonometric functions
     (Fourier series)
  2. non-periodic b.c.: orthogonal polynomials
     (main candidate: Chebyshev polynomials)

- in our applications, we assume periodic b.c. and use Fourier series

$$\phi_j(\mathbf{x}) = e^{-i\mathbf{k}_j \mathbf{x}}$$

(periodic b.c. ok if arbitrary, large enough part of an extended plasma is modeled, not bounded by stellar surfaces)



Comparison to analytical Fourier method

- in $Lu(\mathbf{x}) = s(\mathbf{x}), \quad \mathbf{x} \in V$
  b.c.: $f(u(\mathbf{y})) = 0, \quad \mathbf{y} \in \partial V$

  assume 1-D, and e.g. $L = \partial_{zz}$,
  $\partial_{zz} u = s(x)$

- Fourier transform:

$$(-ik_z)^2 \hat{u}(k_z) = \hat{s}(k_z)$$

- in principle, we want to do this numerically, but we have to make sure about a few points …

pseudo-spectral method, the Fourier case

- The aim is to find the expansion coefficients $\hat{u}_j$ such that the residual

$$\sum_{j=0}^{N-1} \hat{u}_j L\phi_j(\mathbf{x}_n) - s(\mathbf{x}_n) = 0, \quad n=0,1,2,\ldots,N-1$$

or

$$\sum_{j=0}^{N-1} \hat{u}_j L e^{-i\mathbf{k}_j\mathbf{x}_n} - s(\mathbf{x}_n) = 0, \quad n=0,1,2,\ldots,N-1$$

vanishes. If L is linear, then $Le^{-i\mathbf{k}_k\mathbf{x}_n} = h(\mathbf{k}_k)\, e^{-i\mathbf{k}_k\mathbf{x}_n}$

$$\sum_{j=0}^{N-1} \hat{u}_j h(\mathbf{k}_j) e^{-i\mathbf{k}_j\mathbf{x}_n} - s(\mathbf{x}_n) = 0, \quad n=0,1,\ldots,N-1$$

the 'trick' is to choose (turning now to 1D for simplicity)

$z_n = n\,\Delta$, $\quad n = 0,1,2,\ldots N-1$

and

$k_j = 2\pi j / (N\Delta)$, $\quad j = -N/2, \ldots, N/2$

(Δ: spatial resolution)

- z_n and k_j are equi-spaced, and the condition on the residual becomes

$$\sum_{j=0}^{N} \hat{u}_j h(k_j) e^{-2\pi i j n/N} - s(\mathbf{x}_n) = 0, \quad n=0,\ldots,N-1$$

- we define the **discrete Fourier Transform DFT** as

$$\hat{u}_j = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u_n e^{2\pi i n j / N}$$

- with $u_n = u(x_n)$, and the **inverse DFT^{-1}** as

$$u_n = \frac{1}{\sqrt{N}} \sum_{j=-N/2}^{N/2} \hat{u}_j e^{-2\pi i n j / N}$$

- it can be shown that with the specific choice of k_j and z_n

$$\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{2\pi i n j'/N} e^{-2\pi i n j/N} = \delta(j - j')$$

[algebraic proof, using $\sum_{n=0}^{N-1} q^n = (1-q^N)/(1-q)$]
so that

$$u = \mathrm{DFT}^{-1}(\mathrm{DFT}(u))$$

(but just and only at the collocation points, actually

$$\{u_n\} = \mathrm{DFT}^{-1}(\mathrm{DFT}(\{u_n\}))$$!!!)

- the condition on the residual

$$\sum_{j=0}^{N-1} \hat{u}_j h(k_j) e^{-2\pi i j n/N} - s(\mathbf{x}_n) = 0$$

can thus, using the DFT, be written as

$$DFT^{-1}[\hat{u}_j h(k_j)](x_n) - s(x_n) = 0$$

and, on applying DFT,

$$\hat{u}_j h(k_j) - DFT[s(x_n)](k_j) = 0$$

$\Rightarrow$ we can manipulate our equations numerically with the DFT analogously as we do treat equations analytically with the Fourier transform

Remarks:

- z_n and k_j are equi-spaced only for trigonometric polynomials, every set of expansion functions has its own characteristic distribution of collocation points – equi-distribution is an exception (Chebychev, Legendre polynomials etc)
- the sets $\{u_n\}$ and $\{u_j^*\}$ are completely equivalent, they contain the same information

Summary so-far

- we have defined a DFT, which has analogous properties to the analytic FT, it is though finite and can be implemented numerically
- the PS method gives (in principle) exact results at a number of special points, the collocation points
- From the definition of DFT^{-1},

$$u_n = \frac{1}{\sqrt{N}} \sum_{k=-N/2}^{N/2} \hat{u}_k e^{-2\pi i n j/N}$$

it follows immediately that (z corresponds to n, to differentiate we assume n continuous)

$$DFT[\{\partial_z u_n\}](k_j) = \{-2\pi i (j/N)\hat{u}_j\}$$

as “usual”, and the like for other and higher derivatives, and where we concentrate just on the collocation points

The pseudo-spectral method and time-dependent problems

- example: diffusion equation in 1D:

$$\partial_t u = \nu \partial_{zz} u$$
$$u(z,0) = u_0(z)$$

- we consider the equation only at the collocation points $\{z_n = n\Delta,\ n=0,1,\ \ldots\ N-1\}$, writing symbolically

$$\partial_t u_n = \nu \partial_{zz} u_n$$
$$u(z_n,0) = u_0(z_n)$$

- apply a spatial DFT

$$\partial_t \hat{u}_j = -\nu 4\pi^2 (j/N)^2 \hat{u}_j$$
$$\hat{u}(k_j,0) = \hat{u}_0(k_j)$$

where j=-N/2, …,N/2

$\Rightarrow$ we have a set of N ODEs !

$\Rightarrow$ the temporal integration is done in Fourier space

Temporal integration

- The idea is to move the initial condition to Fourier space, and to do the temporal integration in Fourier space, since there we have ODEs
- since we have a set of ODEs, in principal every numerical scheme for integrating ODEs can be applied
- often good is Runge-Kutta 4th order, adaptive step-size
- 4th order RK: d**u**/dt = **F**(**u**,t) (**u** has N components)

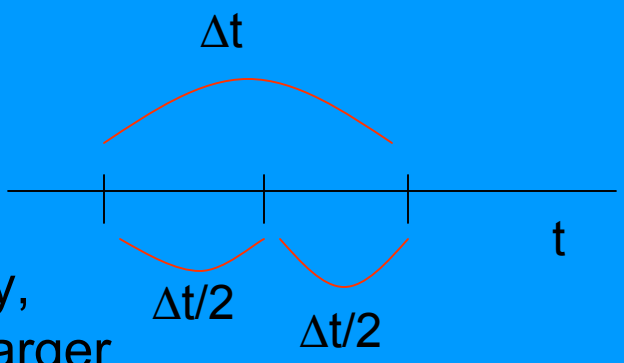
  $\mathbf{u}^{n+1} = \mathbf{u}^n + 1/6(\mathbf{r}_1 + 2\mathbf{r}_2 + 2\mathbf{r}_3 + \mathbf{r}_4)$

  $\mathbf{r}_1 = \Delta t\ \mathbf{F}(\mathbf{u}^n, t_n)$

  $\mathbf{r}_2 = \Delta t\ \mathbf{F}(\mathbf{u}^n + 1/2\mathbf{r}_1,\ t_n + 1/2\Delta\ t)$

  $\mathbf{r}_3 = \Delta t\ \mathbf{F}(\mathbf{u}^n + 1/2\mathbf{r}_2,\ t_n + 1/2\Delta\ t)$

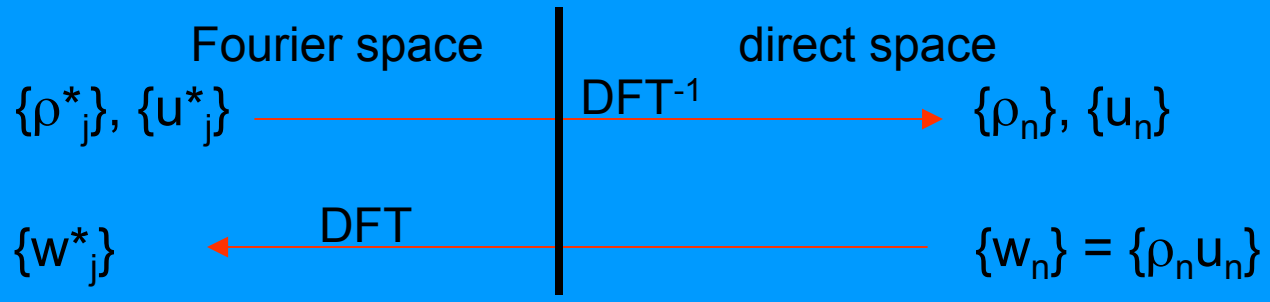
  $\mathbf{r}_4 = \Delta t\ \mathbf{F}(\mathbf{u}^n + \mathbf{r}_3,\ t_n + \Delta\ t)$
- adaptive step-size:
  (for efficiency of the code)
  advance Δt, and also $\Delta t/2 + \Delta t/2$,
  compare the results with prescribed accuracy,
  depending on the result make Δt smaller or larger



How to treat non-linearities

- assume there is a term $\rho(z)u(z)$ in the original PDE
- we are working in F-space, using DFT, so at a given time we have available ρ^*_j and u^*_j
- ρu corresponds to a convolution in F-space, but convolutions are expensive (CPU time !) and must be avoided ($\sim N^2$)
- the procedure to calculate $(\rho u)^*_j$ is as follows ($\sim N \log_2 N$):

1. given at time t are ρ^*_j and u^*_j
2. calculate $\rho_n = DFT^{-1}(\rho^*_j)$ and $u_n = DFT^{-1}(u^*_j)$
3. multiply and store $w_n = \rho_n u_n$
4. move w_n to F-space, $w^*_k = DFT(w_n)$
5. use w^*_j for $(\rho u)^*_j$



Aliasing

- the Fourier modes used are

$$e^{ik_j z_n}$$

with wave-vectors $k_j = 2\pi j/N\Delta,\ j = -N/2, ...N/2$
and grid-points $z_n = n\Delta,\ n = 0, 1, ..., N-1,$
i.e. the modes are $e^{2\pi ijn/N}$

- at the grid points z_n, $e^{2\pi inj/N}$ equals

$$e^{2\pi i(j+lN)n/N},\ l = ..., -2, -1, 0, 1, 2, ...$$

this implies that modes with $k = 2\pi(j + lN)/(N\Delta)$

contribute to the DFT as if they had $k = 2\pi j/N\Delta$

i.e. high k modes alias/bias the amplitude a lower k modes !

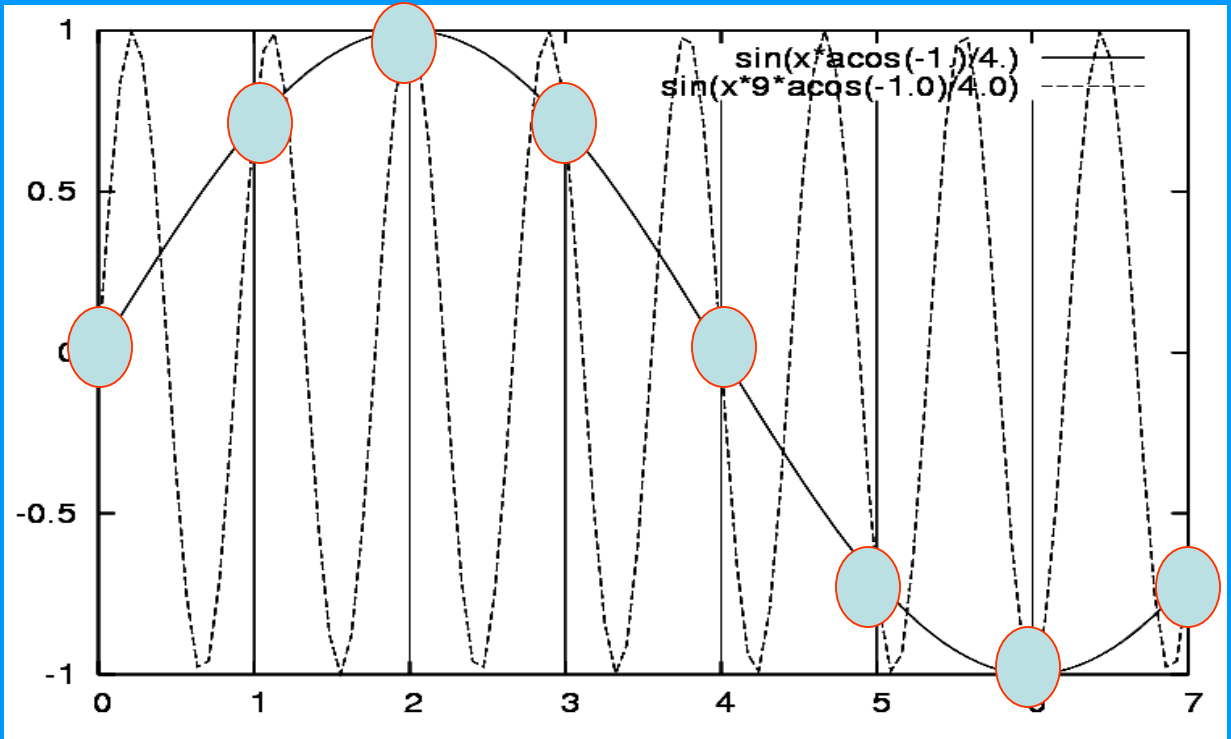
- example: for Δ=1, N=8, our wave vectors are

$$k_j = -\pi, -3\pi/4, ..., -\pi/4, 0, \pi/4, ..., 3\pi/4, \pi$$

now e.g. to k=π/4 also the modes k=9π/4, 17π/4, …. etc. contribute !
i.e. modes outside the k-range we model bias the modeled k-range

example: grid of N=8 points, $\Delta = 1$:

sin(z π/4) and sin(z 9π/4) appear as being the same function when sampled



First consequence of the aliasing effect:

prescribed functions such as initial conditions u(z,t=0) or source functions s(z,t) are best provided as superpositions of the explicitly available modes,

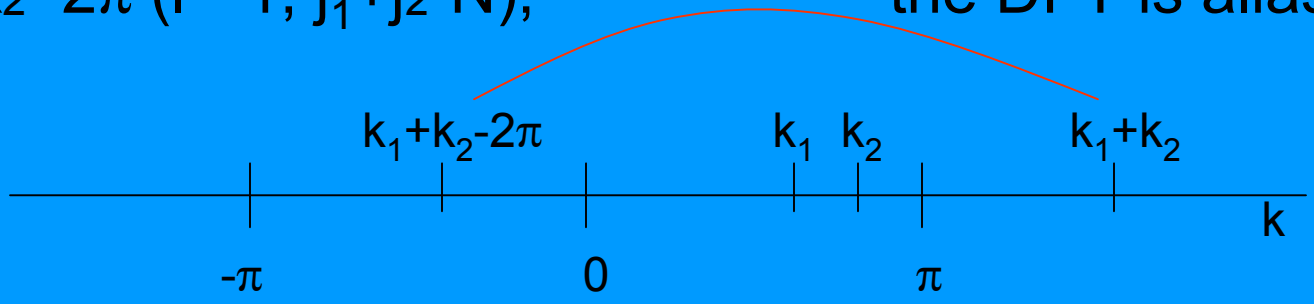
$u(z_n,t=0) = \sum_j u^*_{0,j} e^{2\pi\, i\, jn/N}$

Aliasing and nonlinearities

- assume we have a non-linear term ρu in our PDE, and
  $\rho(z) = \sin(k_1 z)$, $u(z) = \sin(k_2 z)$,
  with k_1, k_2 from our set of available wave-vectors k_j

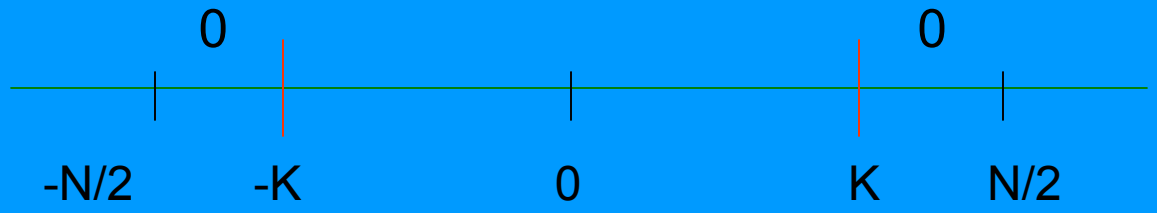
- now
  $\rho u \sim -\cos[(k_1+k_2) z] + \cos[(k_2-k_1)z]$,
  and k_1+k_2 may lie outside our range of k's,
  and the available Fourier amplitudes might get aliased !

- k_1+k_2 outside range if $k_1+k_2 > \pi$,
  and the amplitude appears wrongly in the range of k's at
  $k_1+k_2-2\pi$ ($l=-1$, j_1+j_2-N), the DFT is aliased

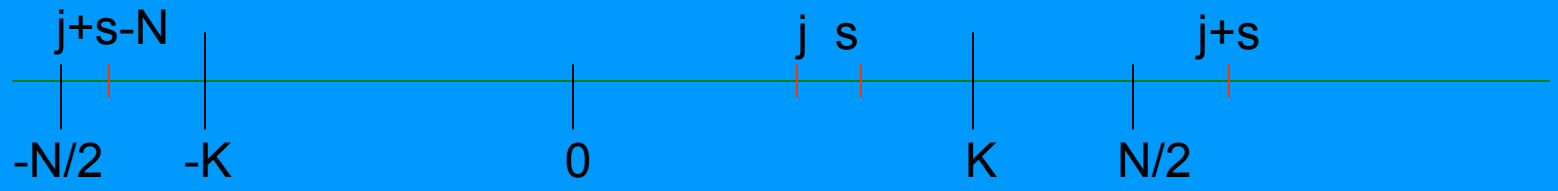


De-aliasing

- Several methods exist to prevent aliasing:
  zero-padding (3/2-rule), truncating (2/3-rule), phase shift
- we apply 2/3-rule:
  - simple to apply,
  - low cost in computing time
- Basic idea:
  set part of the amplitudes to zero always prior to (non-linear) multiplications:



  full index range of k-vectors: [-N/2,N/2]
  $\rightarrow$ keep the sub-range [-K,K] free of aliasing
  method: set Fourier amplitudes $u^*_j = 0$ in [-N/2,-K] and [N/2,K]
- why does this work ? and how to choose K ?



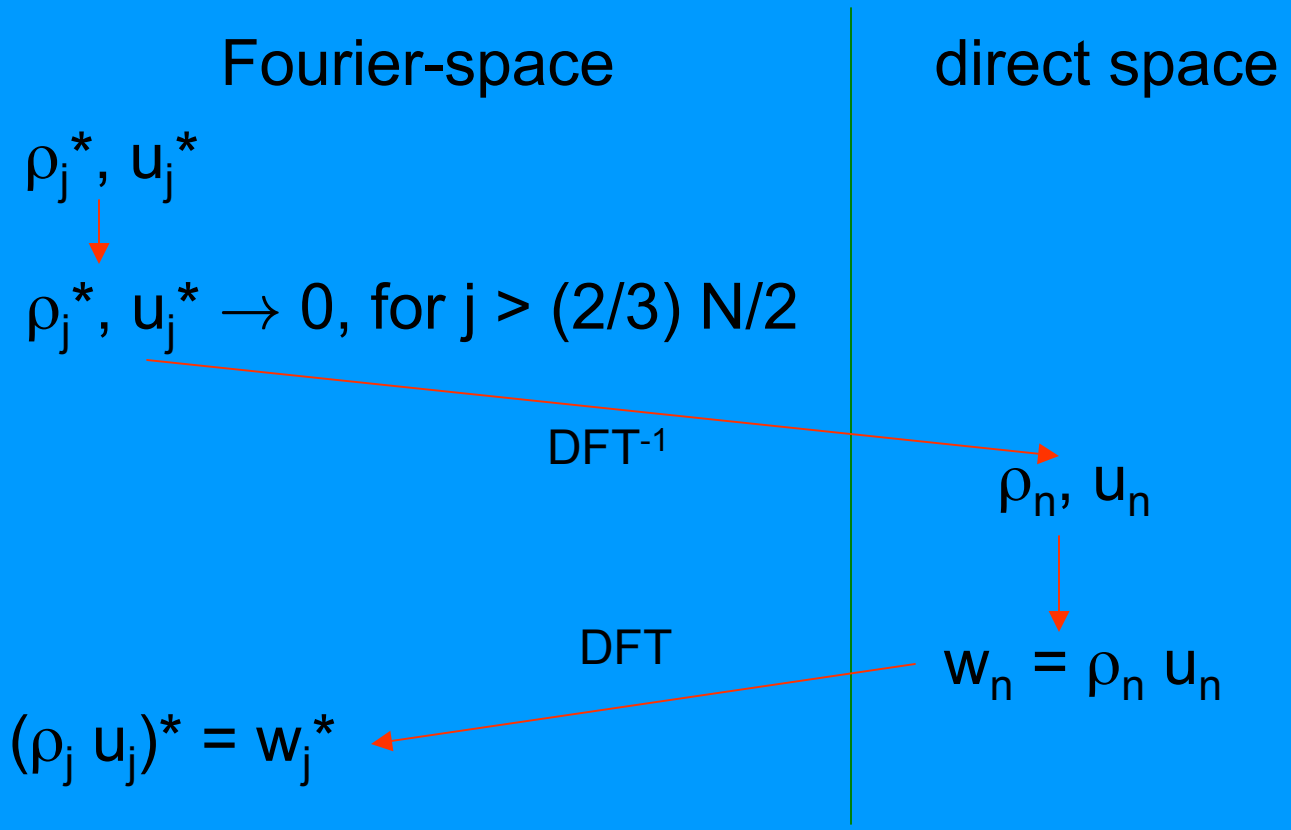
- let j and s be in [0,K]
- if j+s > N/2 (outside range), then the amplitude corresponding to j+s will be aliased to j+s-N
- we demand that j+s-N < -K (in the not used part of the spectrum), the largest j, s in the range are j=s=K: j+s-N <= 2K-N
  i.e. we demand 2K-N <-K or K<N/3
- we set K = N/3 = (2/3) N/2: **'2/3-rule'**

- for j, s in [K,N/2] and j+s > N/2 the amplitude is aliased to j+s-N, which may lie in [-K,0], but we do not have to care,
  the amplitudes at j and s are set to zero

$\Rightarrow$ the range [-K,K] is free of aliasing

non-linearities, de-aliased

- assume you need to evaluate DFT($\rho_i u_i$), having given the Fourier transforms ρ_j^* and u_j^*:



Stability and convergence

- … theory on stability on convergence …
- reproduce analytically known cases
- reproduce results of others, or results derived in different ways
- test the individual sub-tasks the code performs
- monitor conserved quantities (if there are any)
- apply fantasy and physical intuition to the concrete problem you study, try to be as critical as you can against your results

Example 1

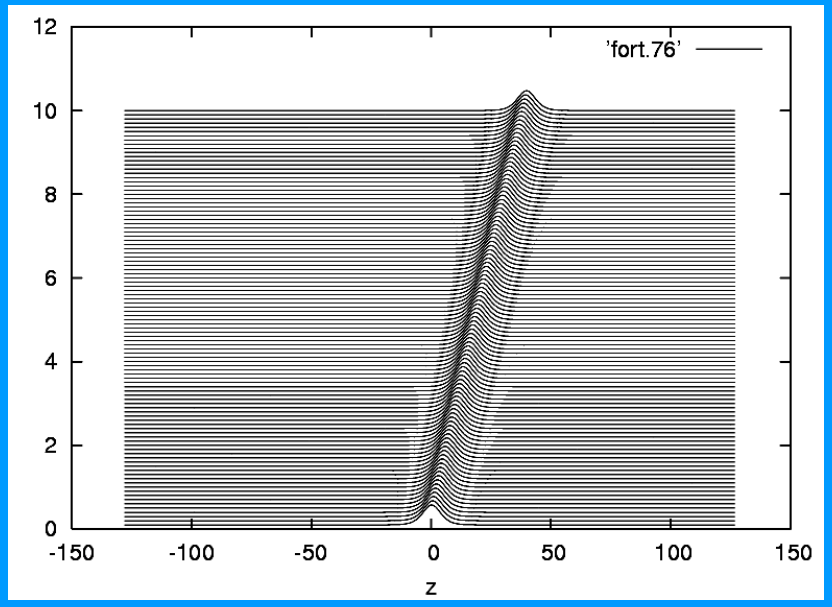
- Korteweg de Vries equation (KdV)

$$\partial_t u = -u\partial_z u - \partial_{zzz} u$$

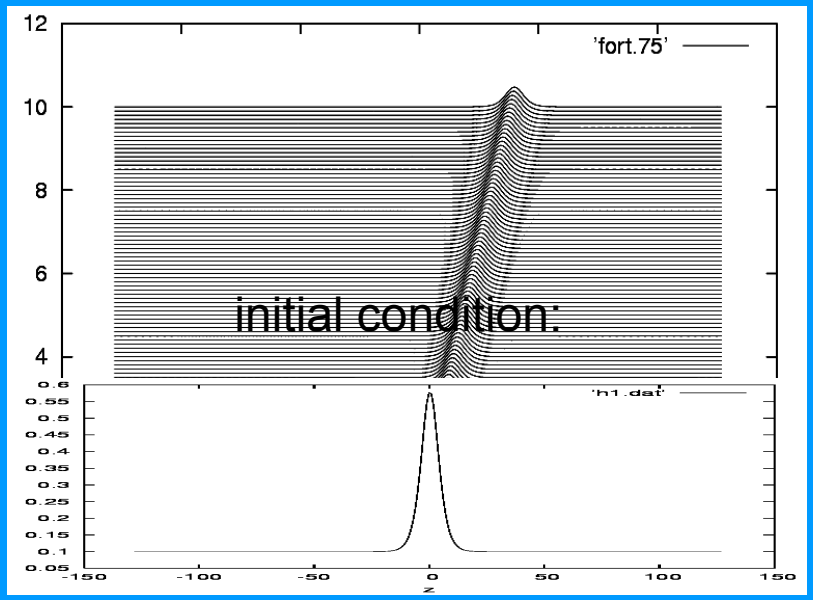
admits soliton wave solutions:

$$u(x,t) = 3\alpha^2 / \cosh^2 \tfrac{1}{2}\left(\alpha z - \alpha^3 t\right)$$

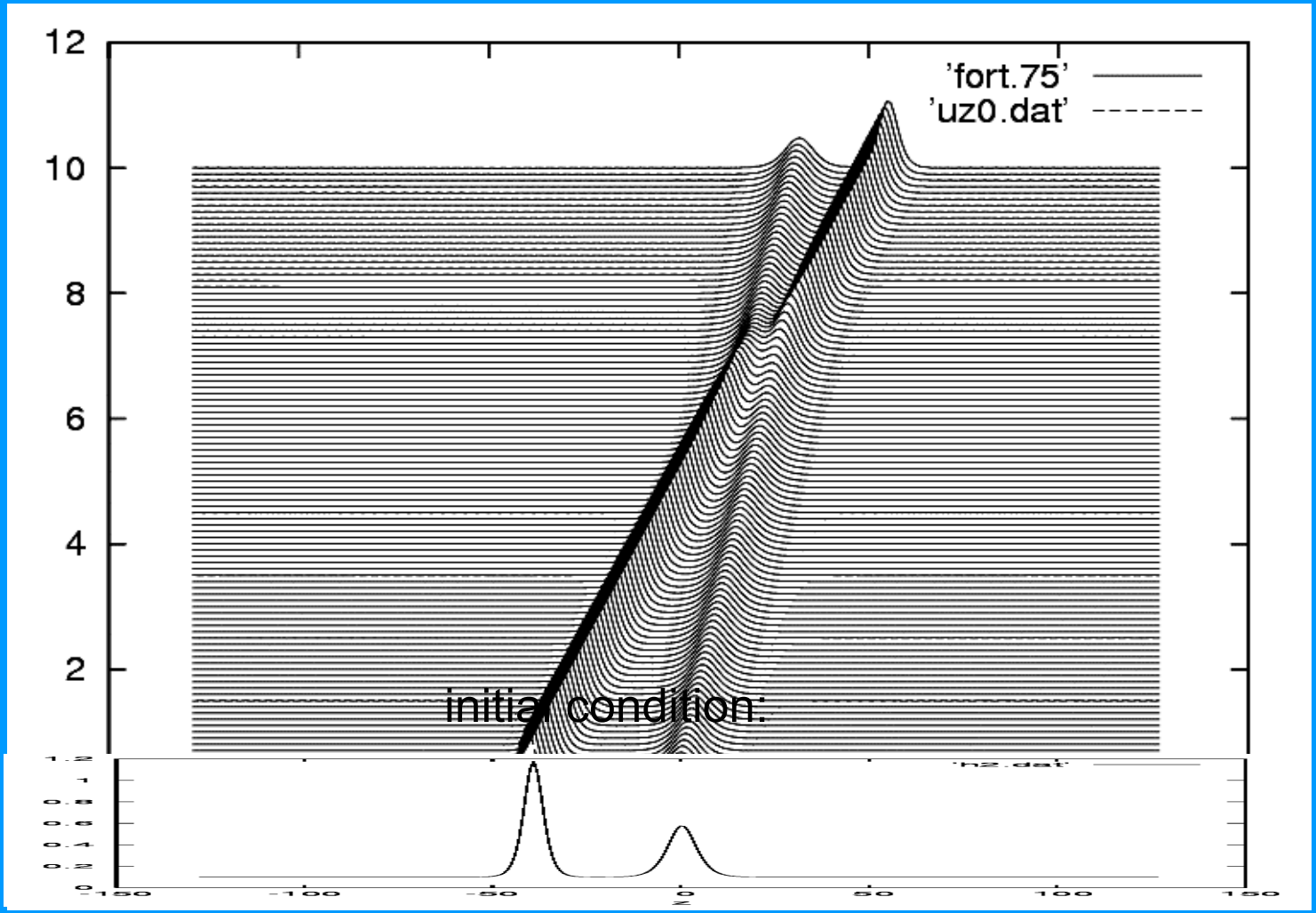
analytically:



numerically:

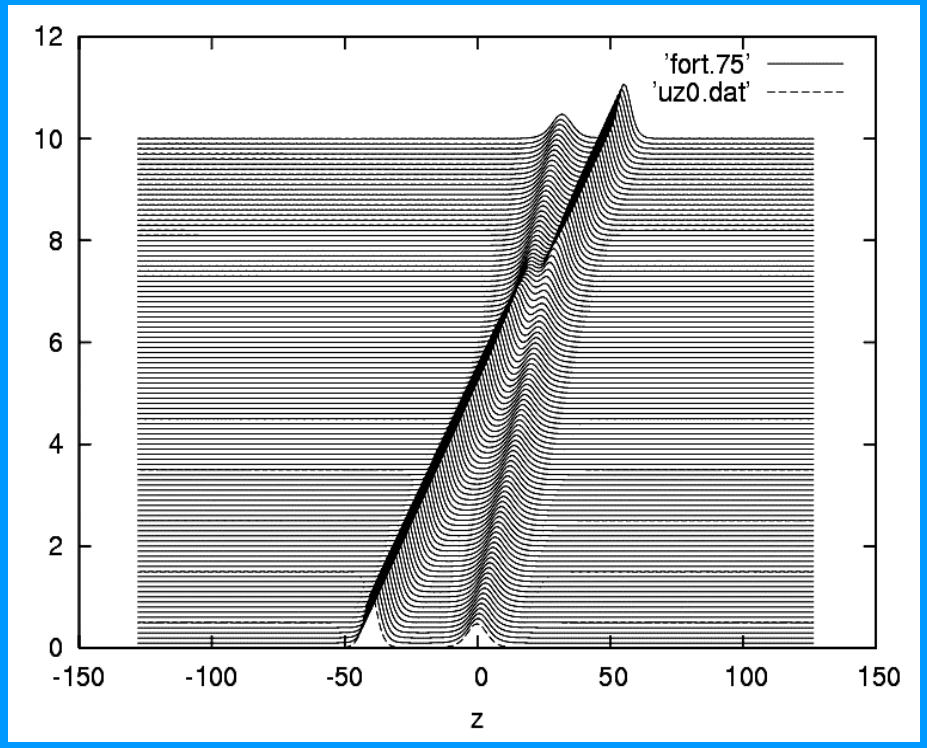


Numerically, two colliding solitons

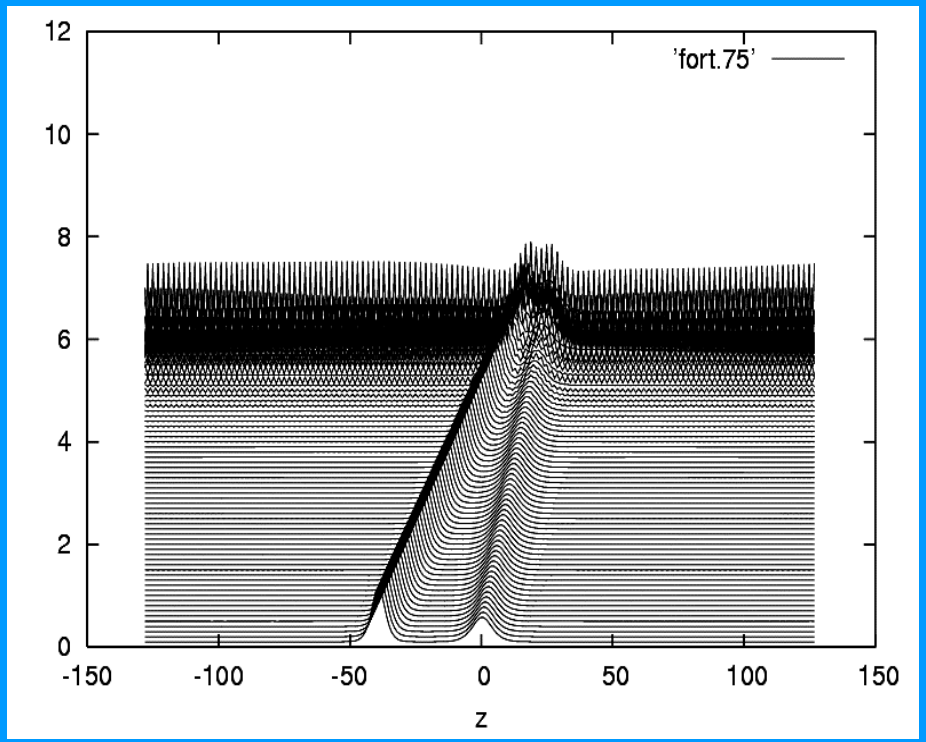


Aliasing

de-aliased



not de-aliased



Example 2: Two-fluid model for the formation of large scales in plasma turbulence

$$(\partial_t - \tau v_n \partial_y)\nabla_\perp^2 \phi + v_g(1+\tau)\partial_y n = \tau \mathrm{div}\{\nabla_\perp \phi, n\} + \{\nabla_\perp^2 \phi, \phi\} + \mu\nabla^4(\phi + \tau n)$$

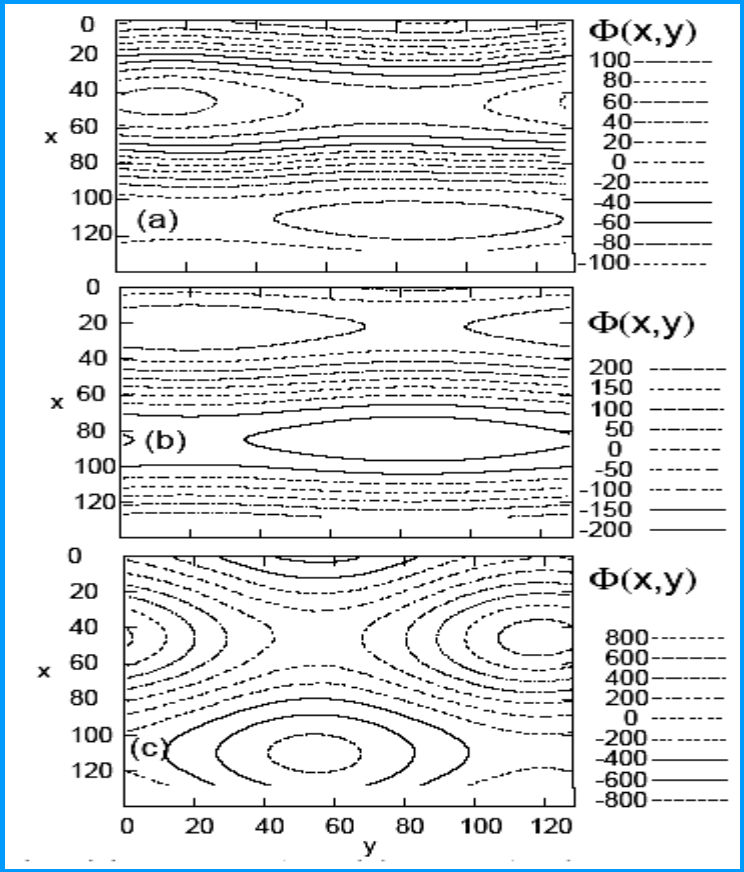
$$(\partial_t + v_g \partial_y) n + (v_n - v_g)\partial_y \phi = \{n, \phi\} + D\nabla^2 n$$

ϕ: electric potential
n: density
τ, v_n, v_g, μ, D: constants

initial condition:
random, small amplitude perturbation (noise)

$\rightarrow$ large scale structures are formed

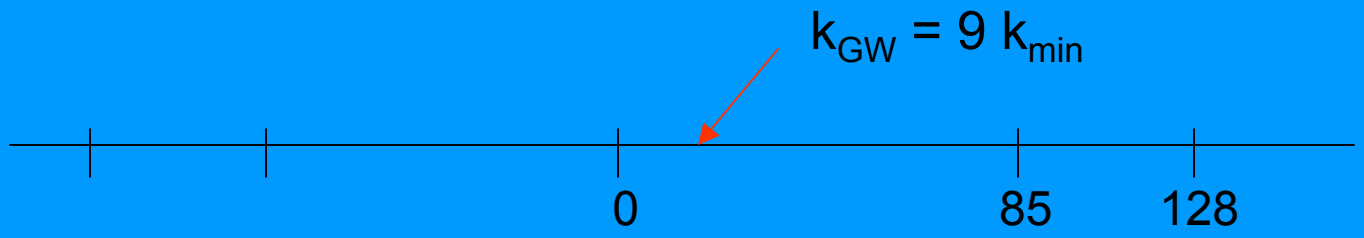
(three different ion temperatures τ)



(Sandberg, Isliker, Vlahos 2004)

Example 3

- relativistic MHD equations, driven by a gravitational wave
- emphasis on the full set of equations, including the non-linearities
  $\rightarrow$ numerical integration
  $\rightarrow$ pseudo-spectral method, de-aliased,
  N=256, effective number of k-vectors: (2/3) 128 = 85
- we use ω_{GW} = 5 kHz, so that $k_{GW} \approx 10^{-6}$ cm^{-1},
  and the range of modeled k's is chosen such that



  i.e. the 1-D simulation box has length 9 $\times$ the wave-length of the grav. wave
- … to be continued at 15:30, by I. Sandberg

(Isliker / Sandberg / Vlahos)

Concluding remarks

Positive properties of the pseudo-spectral (PS)method:

- for analytic functions (solutions), the errors decay exponentially with N, i.e. very fast
- non-smoothness or even discontinuities in coefficients or the solutions seem not to cause problems
- often, less grid points are needed with the PS method than with finite difference methods to achieve the same accuracy
  (computing time and memory !)

Negative properties of the pseudo-spectral method:

- certain boundary conditions may cause difficulties
- irregular domains (simulation boxes) can be difficult or impossible to implement
- strong shocks can cause problems
- local grid refinement (for cases where it is needed) seems not possible, so-far