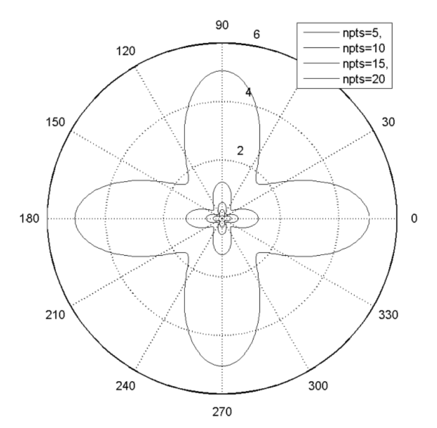
Questions

- Why Chebyshev?
- How about anisotropy?
- Are there also cases where Chebyshev is mixed with Fourier?
- What are the advantages and drawbacks with Chebyshev?

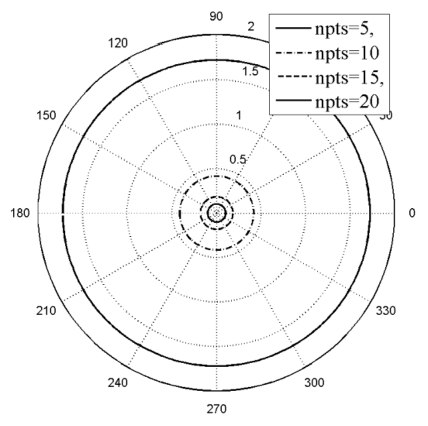
Repetition: Why pseudospectral method?

- reduced number of grid points and therefore less memory is required
- space-dependent fields are known exactly (up to machine precision) at the grid points
- no grid staggering required
- well suited for anisotropic problems
- mixed case (e.g. Tessmer (1995):
  - need for free surface condition on top boundary $\rightarrow$ Chebyshev
  - while horizontally no special boundary conditions are needed $\rightarrow$ Fourier (computationally easier than Chebyshev)
- avoids numerical anisotropy by the use of spectral derivatives

Numerical anisotropy



Numerical anisotropy from FD (picture from Igel, unpublished)



Numerical anisotropy from PS (picture from Igel, unpublished)

Why pseudospectral Chebyshev?

- implementation of boundary conditions like free-surface or non-reflecting boundaries
- ... with (almost) the same accuracy as within the medium

The arcoustic 1D wave equation

$$\frac{\partial^2 p(x,t)}{\partial t^2} = \frac{\partial^2 p(x,t)}{\partial x^2} c(x)^2 + s(x,t)$$

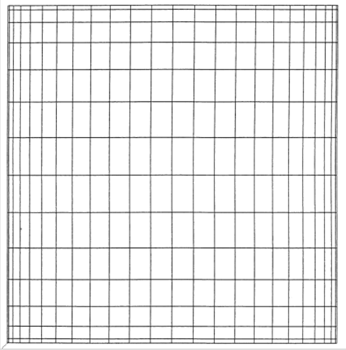
- L.h.s.: treated via FD ($\frac{p(x,t+\mathrm{d}t)-2p(x,t)+p(x,t-\mathrm{d}t)}{\mathrm{d}t^2}$)
- R.h.s: the spatial derivative is calculated using the Chebyshev polynomials T_n

Computational grid

- non uniform grid defined in the interval $[-1, 1]$
- field variables are defined on the Chebyshev (or Gauss-Lobatto) collocation points:

$$x_i = cos(\frac{i\pi}{N}) \qquad i = 0, ..., N$$

- with N being the number of gridpoints per dimension
- resulting in a denser grid at the boundaries



Chebyshev collocation points in a 2D grid (picture from Carcione and Wang (1993))

Chebyshev polynomials

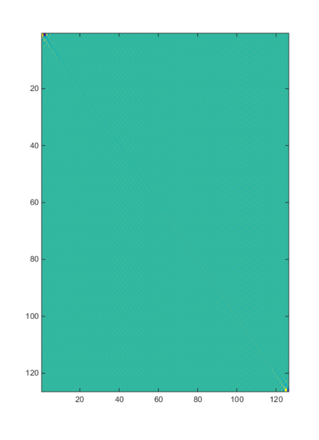
- On this grid each field variable $u(x_i)$ ($x_i = cos(\frac{i\pi}{N})$) is expanded as

$$u(x_i) = \frac{1}{2}a_0 T_0(x_i) + \sum_{n=1}^{N} a_n T_n(x_i)$$

  with Chebyshev polynomials $T_n(cos(\frac{i\pi}{N})) = cos(n\frac{i\pi}{N})$
- N.B.: Chebyshev polynomials form a set of orthogonal basis functions (analogous to the Fourier method)

Drawbacks with Chebyshev

- stretching of the grid near the boundaries required in order to fulfill a given stability criterion
- global communication of the spatial derivative difficult to implement on modern computer architectures
- application of characteristic variables (see e.g. Tessmer (1995)) require additional computational effort



Spatial derivative matrix

Fourier vs. Chebyshev

	Fourier	**Chebyshev**
collocation points	$x_i = \frac{2\pi}{N} i$	$x_i = cos(\frac{\pi}{N} i)$
domain	periodic functions	limited area $[-1, 1]$
basis functions	$cos(nx)$, $sin(nx)$	$T_n(x) = cos(n\psi)$, $x = cos(\psi)$
interpolating function	$f(x_i) = \frac{1}{2} a_0 + \frac{1}{2} a_m cos(kx_i) + \sum_{k=1}^{m-1} (a_k cos(kx_i) + b_k sin(kx_i))$	$f(x_i) = \frac{1}{2} c_0 T_0 + \sum_{k=1}^{m} (c_k T_k(x_i))$
coefficients	$a_k = \frac{2}{N} \sum_{i=1}^{N} f(x_i) cos(kx_i)$ $b_k = \frac{2}{N} \sum_{i=1}^{N} f(x_i) sin(kx_i)$	$c_k = \frac{2}{N} \sum_{i=1}^{N} f(cos(\psi_i)) cos(k\psi_i)$

from www.math.nus.edu.sg/ matgkv/Lecture8.pdf

Chebyshev polynomials (cont.)

- The partial derivative of order q is given by:

$$\frac{\partial^q u(x)}{\partial x^q} = \sum_{n=0}^{N} a_n^{(q)} T_n(x)$$

with

$$c_0 = 2,\ \ c_n = 1\ (n > 0)$$

and

$$c_{n-1} a_{n-1}^{(q)} - a_{n+1}^{(q)} = 2n a_n^{(q-1)}\ (n \geq 1)$$

Chebyshev polynomials (cont.)

Hence, defining $a_n = a_n^{(0)}$ and $b_n = a_n^{(1)}$, the first-order derivative is

$$\frac{\partial u}{\partial \zeta} = \left(\sum_{n=0}^{N}\right)' b_n T_n(\zeta), \tag{15}$$

where

$$b_{n-1} = b_{n+1} + 2na_n, \qquad n = N, \cdots, 1, \qquad b_{N+1} = b_N = 0. \tag{16}$$

The expansion of $u(\zeta)$ and its coefficients can be written explicitly as

$$\boxed{u(\zeta_j) = \left(\sum_{n=0}^{N}\right)' a_n \cos\frac{\pi n j}{N}, \tag{17}$$

$$a_n = \frac{2}{N}\left(\sum_{j=0}^{N}\right)' u(\zeta_j)\cos\frac{\pi n j}{N}.} \tag{18}$$

Let us define $N' = 2N$, and $u(\zeta_j) = 0$ for $j = N'/2 + 1, ..., N' - 1$. Then

$$a_n = \frac{4}{N'}\sum_{j=0}^{N'-1} u(\zeta_j)\cos\frac{2\pi n j}{N'} \tag{19}$$

is a real Fourier transform which can be calculated by using the Fast Fourier Transform (FFT). Afterwards, we get the b_n's from the a_n's by using the recursion equation (16), and again, the calculation of (15) is carried out with a real Fourier transform. In particular, we compute the FFT's with the prime factor algorithm of Temperton [5] in its vectorized form.