Introducing the Kadath library

Philippe Grandclément

July 9th, 2024

Laboratoire de l'Univers et Théories (LUTh)
CNRS / Observatoire de Paris
F-92195 Meudon, France

philippe.grandclement@obspm.fr

KADATH library

KADATH is a library that implements spectral methods in the context of theoretical physics.

- It is written in C++, making extensive use of object oriented programming.
- Versions are maintained via git.
- Website : *www.kadath.obspm.fr*
- The library is described in the paper : *JCP* **220**, *3334 (2010).*
- Designed to be very modular in terms of geometry and type of equations.
- LateX-like user-interface.
- More general than its predecessor LORENE.
- Well-suited for solving the initial value problem (no time evolution).

Concept in 1D

Given a set of orthogonal functions Φ_i on an interval Λ, spectral theory gives a recipe to approximate f by

$$f \approx I_N f = \sum_{i=0}^{N} a_i \Phi_i$$

Properties

- the Φ_i are called the basis functions.
- the a_i are the coefficients : it is the quantity stored on the computer.
- Multi-dimensional generalization is done by direct product of basis.
- The computation of the a_i comes from the Gauss quadratures.

Coefficient and configuration spaces

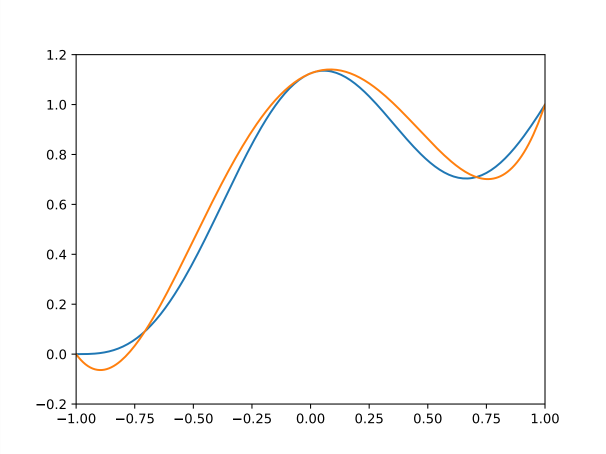
There exist $N+1$ point x_i in Λ such that

$$f\left(x_i\right) = I_N f\left(x_i\right)$$

Two equivalent descriptions

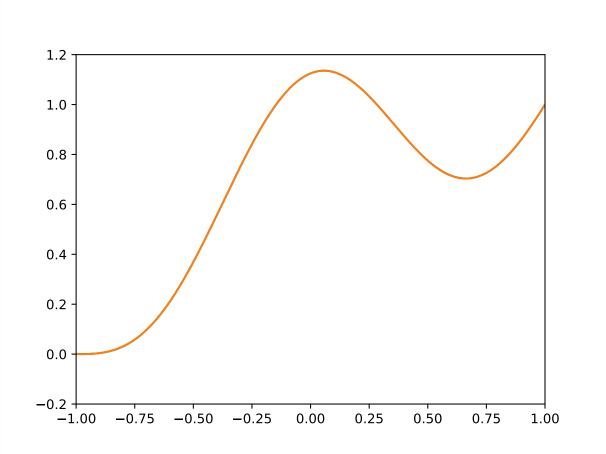
- Formulas relate the coefficients a_i and the values $f\left(x_i\right)$.
- Complete duality between the two descriptions.
- One works in the coefficient space when the a_i are used (for instance for the computation of f').
- One works in the configuration space when the $f\left(x_i\right)$ are employed (for the computation of $\exp\left(f\right)$)

Example of interpolant for $N = 4$



blue curve $f(x) = \cos^3(\pi x/2) + (x+1)^3/8$; *orange* : $I_4 f$.

Example of interpolant for $N = 8$

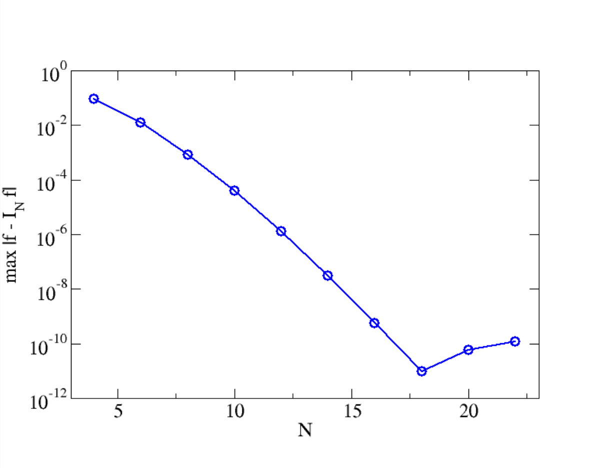


blue curve $f(x) = \cos^3(\pi x/2) + (x+1)^3/8$; *orange* : $I_8 f$.

Spectral convergence

- For smooth functions (i.e. $\mathcal{C}^\infty$), $I_N f$ converges to f (when N increases), faster than any power-law of N (typically exponentially).
- This is called **spectral convergence**.
- this is to be contrasted with finite difference schemes.
- One of the main reason to use spectral methods.

Spectral convergence

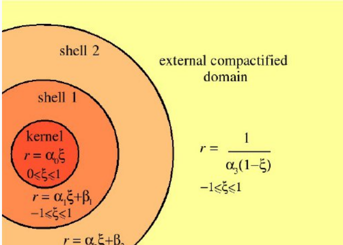


Multi-domain setting

Numerical coordinates

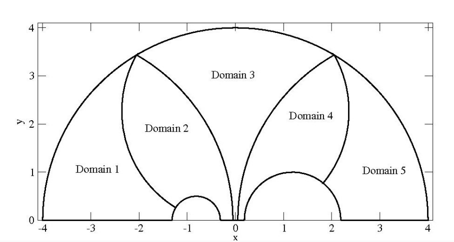
- Space is divided into several numerical domains.
- In each domain there is a link between the physical coordinates X and the numerical ones $X^\star$.
- Spectral expansion is performed with respect to $X^\star$.
- Non-periodic coordinates are expanded wrt to polynomials.
- Periodic coordinates (i.e. angles) are described by trigonometrical functions.

Example spherical space



Other spaces available

- Cylindrical space.
- Bispherical space.
- Spaces with periodic time coordinates.
- Spaces with adaptable domains.
- Spaces with various symmetries.
- Additional ones relatively easy to include.



Code 1

Scalar field management

For a scalar field, in each domain:

- one array for the values at the collocation points.
- one array for the values of the coefficients.
- one object describing the spectral basis.
- should be transparent to the user.

Setting a scalar field

- `KADATH` is not intended to implement complex analytic expressions.
- The space gives access to the coordinates that can be assigned to scalar fields.
- Usually the basis must be set by hand.
- Once the basis is known, additional methods are available (derivative, value at any point etc).

Code 2

The standard spectral base

- A scalar field is regular if it is expressed as a sum of polynomials of Cartesian coordinates $x^m y^n z^p$.
- From that assumption one can deduce some appropriate choice of spectral basis by expressing x, y, z in terms of r, θ, φ, for instance.
- Details depend on the space considered.
- For a spherical space it leads to :
  - for φ : $\cos(m\varphi)$ and $\sin(m\varphi)$.
  - for θ : $\cos(2j\theta)$ for m even and $\sin((2j+1)\theta)$ for m odd.
  - Chebyshev polynomials with respect to $r^\star$
  - In the nucleus : $T_{2i}(r^\star)$ for m even and $T_{2i+1}(r^\star)$ for m odd.

KADATH management of the spectral basis

- For every computation, KADATH tries to assert the basis of the result.
- Straightforward for things like the product, inverse, sum etc...
- For other computations (like $\exp$, $\cos$, $\sqrt{}$) the base cannot be directly obtained and is lost.
- **Important rule** set the base by hand if and only if it is required.
- Be careful when enforcing the standard base. For instance $\rho = \sqrt{x^2 + y^2}$ is not expanded onto the standard base.
- Most of the errors in using KADATH come from inappropriate setting of the basis.

Code 3

Weighted residual method

Consider a field equation $R = 0$ (ex. $\Delta f - S = 0$). The discretization demands that

$$(R, \xi_i) = 0 \quad \forall i \leq N$$

Properties

- $(,)$ is the same scalar product as the one used for the spectral approximation.
- the ξ_i are called the test functions.
- For the τ-method, the ξ_i are the basis functions.
- Amounts to cancel the coefficients of R.
- Some equations are relaxed and must be replaced by appropriate boundary and matching conditions.

The discrete system

Original system

- Unknowns : tensorial fields.
- Equations : partial derivative equations.

Discretized system

- Unknowns : coefficients $\vec{u}$.
- Equations : algebraic system $\vec{F}(\vec{u}) = 0$.

Properties

- For a linear system $\vec{F}(\vec{u}) = 0 \Longleftrightarrow A^i_j u^j = S^i$
- In general $\vec{F}(\vec{u})$ is even not known analytically.
- $\vec{u}$ is sought numerically.

Newton-Raphson iteration

Given a set of field equations with boundary and matching equations, `KADATH` translates it into a set of algebraic equations $\vec{F}(\vec{u}) = 0$, where $\vec{u}$ are the unknown coefficients of the fields.

The non-linear system is solved by Newton-Raphson iteration

- Initial guess $\vec{u}_0$.
- Iteration :
  - Compute $\vec{s}_i = \vec{F}(\vec{u}_i)$
  - If $\vec{s}_i$ if small enough $\Longrightarrow$ solution.
  - Otherwise, one computes the Jacobian : $\mathbf{J}_i = \dfrac{\partial \vec{F}}{\partial \vec{u}}(\vec{u}_i)$
  - One solves : $\mathbf{J}_i \vec{x}_i = \vec{s}_i$.
  - $\vec{u}_{i+1} = \vec{u}_i - \vec{x}_i$.

Convergence is very fast for good initial guesses.

Computation of the Jacobian

Explicit derivation of the Jacobian can be difficult for complicated sets of equations.

Automatic differentiation

- Each quantity x is supplemented by its infinitesimal variation δx.
- The dual number is defined as $\langle x, \delta x\rangle$.
- All the arithmetic is redefined on dual numbers. For instance $\langle x, \delta x\rangle \times \langle y, \delta y\rangle = \langle x \times y, x \times \delta y + \delta x \times y\rangle$.
- Consider a set of unknown $\vec{u}$, and a its variations $\delta\vec{u}$. When $\vec{F}$ is applied to $\langle \vec{u}, \delta\vec{u}\rangle$, one then gets : $\left\langle \vec{F}\left(\vec{u}\right), \delta\vec{F}\left(\vec{u}\right)\right\rangle$.
- One can show that
$$\delta\vec{F}\left(\vec{u}\right) = \mathbf{J}\left(\vec{u}\right) \times \delta\vec{u}$$

The full Jacobian is generated *column by column*, by taking all the possible values for $\delta\vec{u}$, at the price of a computation roughly twice as long.

Defining the system

All the information needed to solve the problem are contained in the class `System_of_eqs`

- The region of space of interest must be passed.
- The various constants and unknowns are defined, along with the name by which they will be recognized.
- At this point the total number of unknown (i.e. the total number of coefficients) is computed.

Passing the equations

Main types of equations

- `Eq_inside` : bulk equations, second order PDE, to be solved inside a given domain.
- `Eq_matching` : ensures the matching of a quantity at the boundary between two domains.
- `Eq_bc` : enforces a boundary condition at the boundary of one domain.

The equations are passed with a formalism inspired by LateX.

- Various reserved word can be used (like `Lap`, `Sqrt`...)
- If a metric is defined, it is used to manipulate indices.
- Einstein's summation on repeated indices is used.
- for consistency reasons, the equations must have the same symmetries as the unknowns.

Different types of metric

If a metric is associated to the system, one has access to additional methods : covariant derivative D, Christoffel Γ, Riemann and Ricci tensors R etc...

No possibility to have two metrics associated to a system.

Main types of metrics:

- **Flat metrics.** In Cartesian of spherical tensorial basis.
- **Constant metrics.** It gives access to the methods but they are not considered as unknowns.
- **General metrics.** It is an unknown and the code will try to find its value.

Numerical resources

Consider N_u unknown fields, in N_d domains, with d dimensions. If the resolution is N in each dimension, the Jacobian is an $m \times m$ matrix with:

$$m \approx N_d \times N_u \times N^d$$

For $N_d = 5$, $N_u = 5$, $N = 20$ and $d = 3$, one reaches $m = 200\,000$

Solution

- The matrix is distributed on several processors.
- Easy because the Jacobian is computed column by column.
- The library `SCALAPACK` is used to invert the distributed matrix.

- $d = 1$ problems : sequential.
- $d = 2$ problems : 100 processors (mesocenters).
- $d = 3$ problems : 1000 processors (national supercomputers).

A test problem

Find the conformal factor Ψ of the Schwarzschild black hole in QI coordinates.

System of equations

- Bulk : $\Delta\Psi = 0$.
- Inner BC : $\Psi_{,r} + \frac{1}{2a}\Psi = 0$
- Outer BC : $\Psi = 1$

a is the radius of the black hole and the solution is

$$\Psi\left(r\right) = 1 + \frac{a}{r}.$$

Code 4

Advanced topics : tensors

- Any valence and types of indices are possible.
- Tensorial basis : Cartesian, orthonormal spherical basis.
  For instance : $\vec{V} = \left(V^x\left(r,\theta,\varphi\right), V^y\left(r,\theta,\varphi\right), V^z\left(r,\theta,\varphi\right)\right).$
- Be careful in setting the spectral base of each components.
- Tensors can be used as constants or unknowns.

Advanced topics : definitions

- Some expressions of the unknowns appear often in the equations (e.g. the extrinsic curvature tensor).
- Instead of explicitly replacing them by their expression, they can be made definitions.
- Definitions are computed only when the unknowns are modified.
- Definitions can also be used to do computations.

Advanced topics : global unknowns

- Some unknowns are not field but global quantities (orbital velocity of a binary system for instance).
- The system must then be supplemented with some global equations (i.e. not field equations).
- Example of global equations :
    - surface integrals (at infinity, on horizons).
    - value of a field at a given point.
    - value of one coefficient of an expression.

A complicated case

Binary black holes in the extended thin-sandwich approach.

Geometry
Bispherical coordinates, solve outside of the two spheres.

Unknowns
Lapse N, conformal factor Ψ, shift vector B^i and orbital velocity Ω.

Bulk equations
Hamiltonian and momentum constraints, trace of the evolution equations.

Boundary conditions
Apparent horizons in equilibrium and asymptotic flatness.

Equation for Ω
Relativistic virial theorem : $M_{\rm Komar} = M_{\rm ADM}$.

Code 5

Try it...

The tutorials can be found on the Kadath website (https://kadath.obspm.fr). Have fun...