Porting Turchin's algorithm to Julia

Tatyana Abramova

Moscow Institute of Physics and Technology

August 12, 2019

Experimental data	Observed value	Apparatus function	A bit of noise

$$f(y) = \int dx \quad \varphi(x) \quad * \quad K(x,y) \quad + \quad \varepsilon_y$$

$\varphi(x)$ - **?**

Fredholm equation is **ill-posed**.
Regularization: introduce additional information about $\varphi(x)$

Algorithm: Solution

$$f(y) = \int K(x,y)\varphi(x)dx$$

Decomposition of $\varphi(x)$ on basis $\{T_n(x)\}$:

$$\varphi(x) = \sum_n \varphi_n T_n(x)$$

$$K_{mn} = \int K(x, y_m) T_n(x) dx$$

$$f_m = f(y_m)$$

Matrix form:

$$f_m = K_{mn}\varphi_n$$

Choose solution based on the strategy $\hat{S}$, which is using prior information about $\varphi(x)$:

$$\text{Optimal } \varphi_n = \hat{S}_n[f] = E[\varphi_n|f] = \int \varphi_n P(\varphi|f)d\varphi$$

$$P(\varphi|f) = \frac{P(\varphi)P(f|\varphi)}{\int d\varphi P(\varphi)P(f|\varphi)}$$

Algorithm: Prior information

Conditions on prior information

$$I[P(\varphi)] = \int \ln P(\varphi) P(\varphi) d\varphi \to min$$

$$\int P(\varphi) d\varphi = 1$$

$$\int \langle \varphi, \hat{\Omega}\varphi \rangle P(\varphi) d\varphi = \omega$$

where $\hat{\Omega} = |\frac{d^2}{dx^2}\rangle\langle\frac{d^2}{dx^2}|$ – operator of smoothness

$$P_\alpha(\vec{\varphi}) = \frac{\alpha^{Rg(\Omega)/2} \det \Omega^{1/2}}{(2\pi)^{N/2}} \exp(-\frac{1}{2}(\vec{\varphi}, \alpha\Omega\vec{\varphi})),$$

where $\alpha = \frac{1}{\omega}$ – parameter of smoothness that should be selected:

- Manually using known smoothness
- Using most probable parameter: $\alpha^* = \mathrm{argmax} P(\alpha|f)$
- Using prior information about smoothness $P(\alpha)$:

$$P(\varphi) = \int P_\alpha(\varphi) P(\alpha)\ d\alpha$$

Algorithm: Integration

$$P(\varphi|f) = \frac{P(\varphi)P(f|\varphi)}{\int d\varphi P(\varphi)P(f|\varphi)}$$

Gaussian errors

In case of Gaussian errors

$$P(f|\varphi) = \frac{1}{(2\pi)^{M/2}\Sigma|^{1/2}} exp(-\frac{1}{2}(f - K\varphi)^T\Sigma^{-1}(f - K\varphi)),$$

the integral can be calculated analytically
Desired vector and covariance matrix:

$$\varphi = (K^T\Sigma^{-1}K + \alpha^*\Omega)^{-1}K^T\Sigma^{-1}f$$

$$\Sigma_\varphi = (K^T\Sigma^{-1}K + \alpha^*\Omega)^{-1}$$

MCMC integration

For any other kinds of errors the integral should be calculated numerically.

Kernels and bases

Kernels

- rectangular
- diffraction
- gaussian
- triangular
- dispersive
- exponential
- heaviside
- can be set manually

Bases

- Fourier
- Legendre polynomials
- Bernstein polynomials + boundary conditions
- Cubic Spline + boundary conditions (custom realization)

NOTE: for MCMC sampling omega matrix of basis should be nonsingular

$$P_\alpha(\vec{\varphi}) = \frac{\alpha^{Rg(\Omega)/2} \det \Omega^{1/2}}{(2\pi)^{N/2}} \exp(-\frac{1}{2}(\vec{\varphi}, \alpha\Omega\vec{\varphi}))$$

Sampling with singular covariance matrix?

Cubic Spline basis

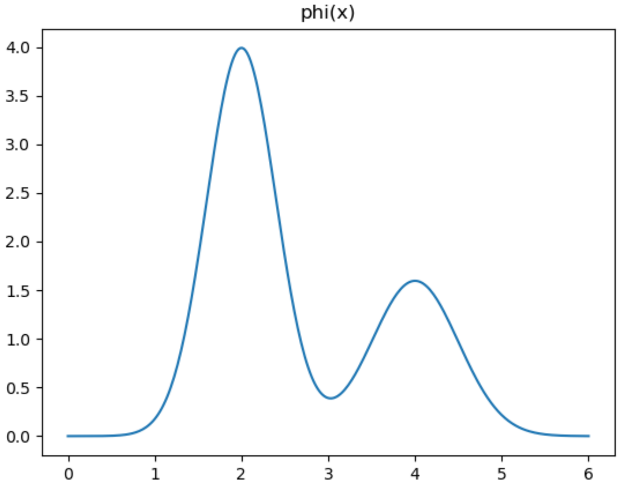
- arbitrary set of nodes (including repeating ones)
- arbitrary spline degree
- derivatives

TODO: refactor to polynomials for analytical integration

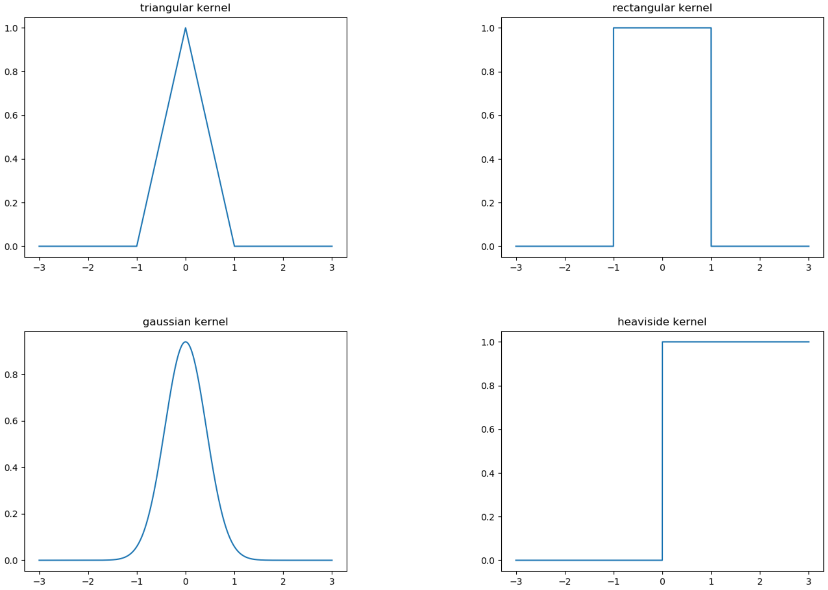
```
struct BSpline
    i::Int64
    k::Int64
    knots::Array{Float64, 1}
    func::Function
end

BSpline(i::Int64, k::Int64, knots::Array{Float64, 1})

derivative(b_spline::BSpline, x::Float64, deg::Int64)
```

Example function

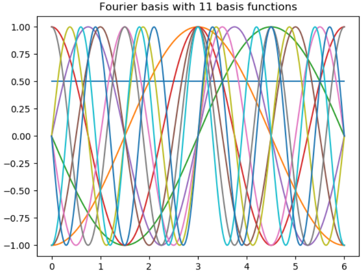
$$\phi(x) = \frac{4}{\sqrt{2\pi \cdot 0.4^2}} exp\left(-\frac{(x-2)^2}{2 \cdot 0.4^2}\right) + \frac{2}{\sqrt{2\pi \cdot 0.5^2}} exp\left(-\frac{(x-4)^2}{2 \cdot 0.5^2}\right)$$

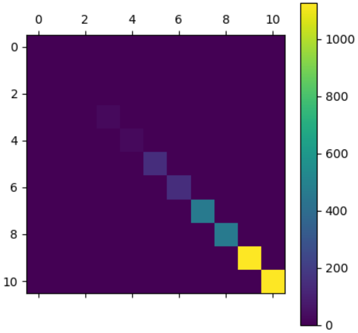


Example kernels

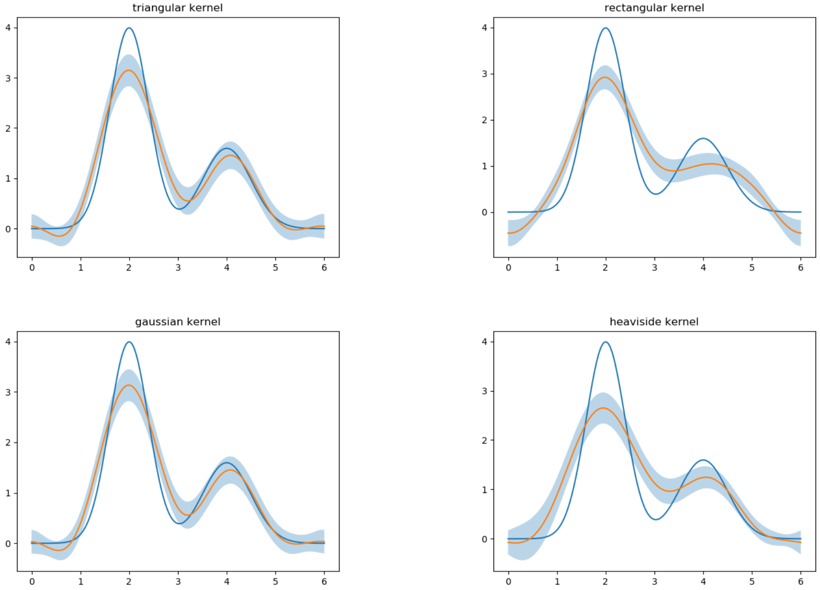


Fourier basis

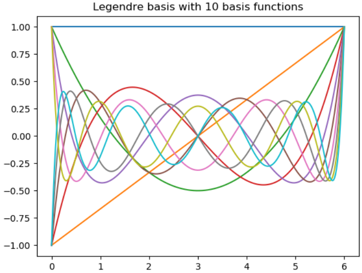


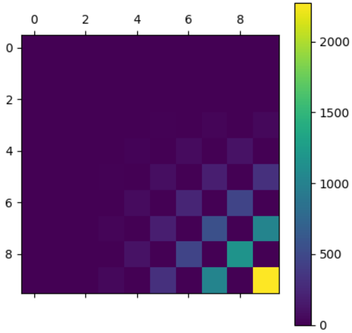


Fourier basis: 31 basis functions

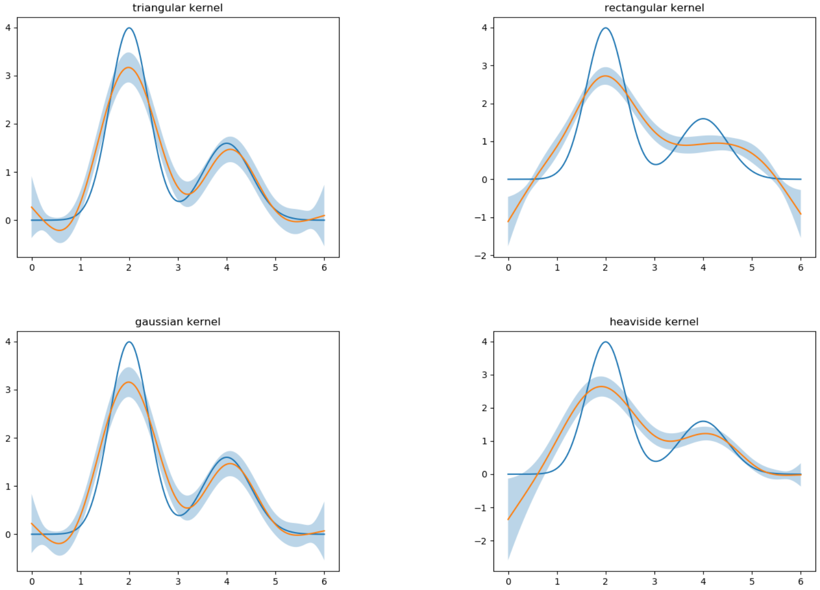


Legendre polynomials basis

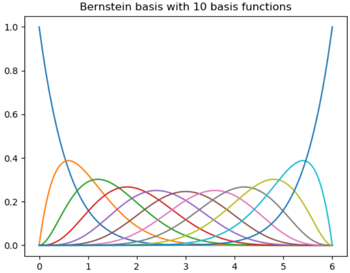


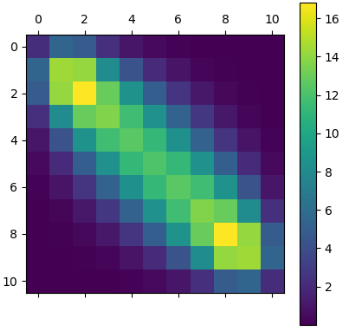


Legendre polynomials basis: 20 basis functions

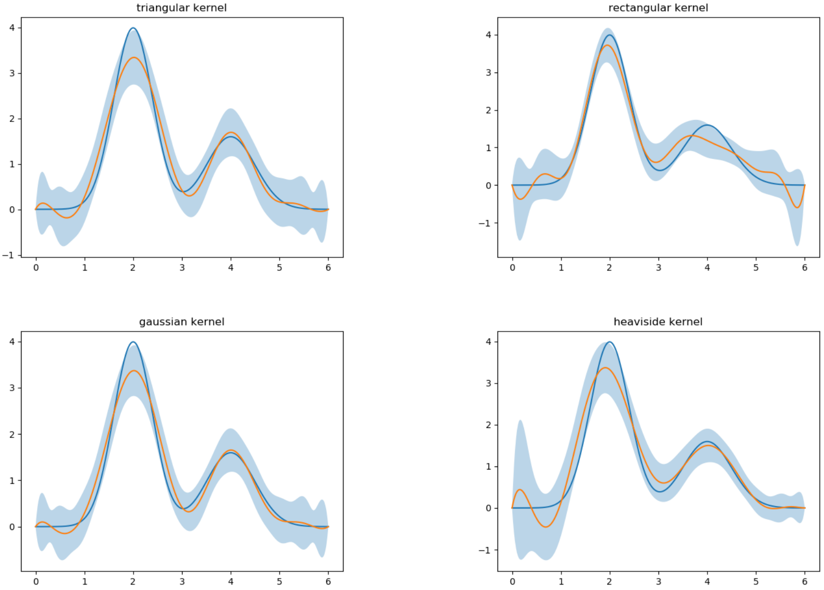


Bernstein polynomials basis

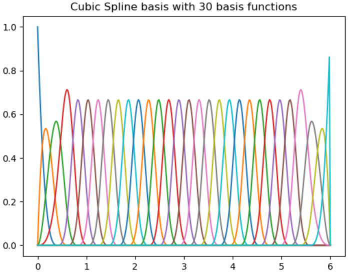


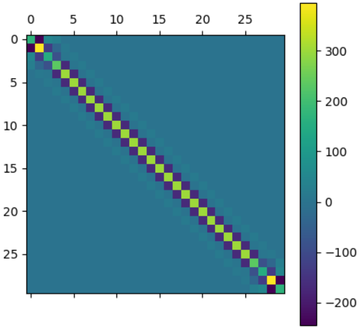


Bernstein polynomials basis: 20 basis functions + zero boundary conditions

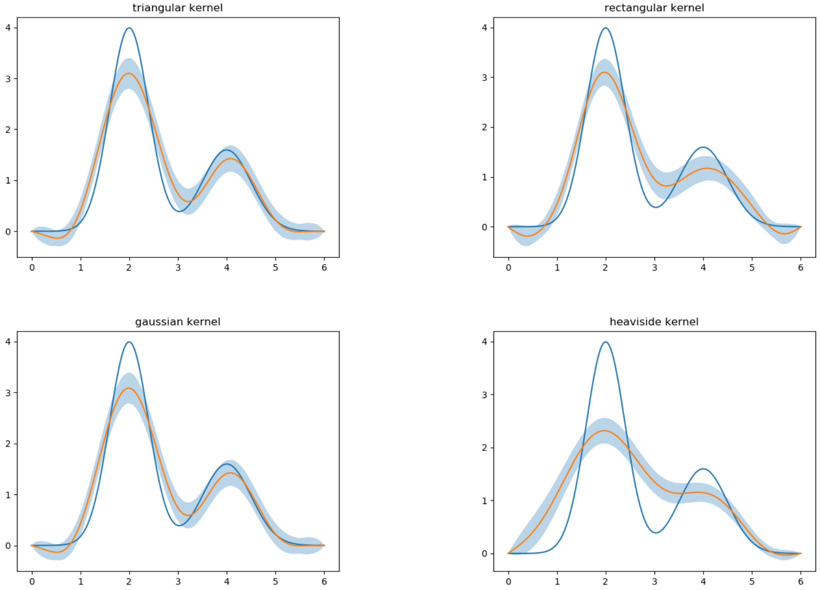


Cubic spline basis

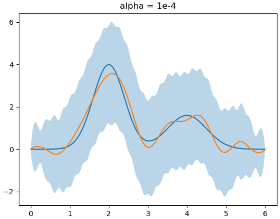


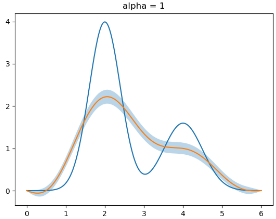


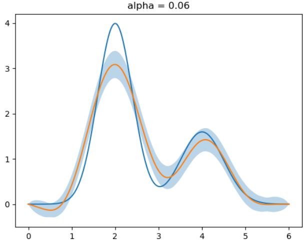
Cubic spline basis: 30 basis functions + zero boundary conditions



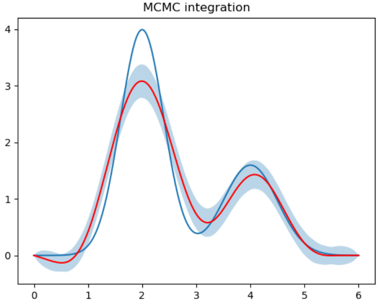
Cubic spline basis with 30 basis functions + boundary conditions, gaussian kernel.

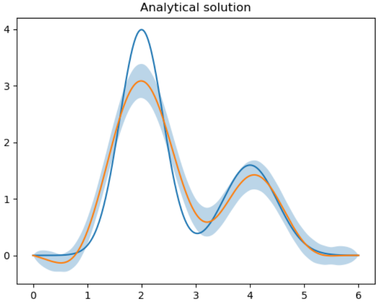






Test sampling with gaussian model:
Mamba.jl package, 1 chain 1000 samples, Cubic spline basis with 30 basis functions and zero boundary conditions.





TODO: BAT.jl integration

Documentation:

- Theoretical introduction
- User's guide
- Getting started
- Example of reconstruction

Config file

```
using Logging

RTOL_QUADGK = 1e-8
MAXEVALS_QUADGK = 1e5
X_TOL_OPTIM = 1e-8
ORDER_QUADGK = 500

global_logger()
```

- integration constants
- optimization constants
- logger

Done

- 4 bases (2 with zero boundary conditions) and set of kernels
- User-defined or optimal α
- Gaussian errors and MC integration
- Documentation

Problems

- Choosing integration parameters (atol, xtol, maxevals)
- Choosing alpha boundaries(lower, higher limits and initial value for optimisation)
- Config file
- Logger
- Heaviside kernel: optimal alpha actually is not optimal

To do

- BAT MCMC integration
- BSpline refactoring
- Documentation structure
- Testing
- Release