Fractional PDEs Constrained Optimization: an optimize–then–discretize approach with L–BFGS and Approximate Inverse Preconditioning

Stefano Cipolla[1] Fabio Durastante[2]

[1]Università di Roma ”Tor Vergata”
cipolla@mat.uniroma2.it

[2]Università dell'Insubria
fdurastante@uninsubria.it

Numerical **M**ethods for **O**ptimal **C**ontrol Problems: Algorithms, Analysis and Applications
Rome, June, 21 – 2017

Overview

The Problem

$$\begin{cases} \min J(y,u) = \frac{1}{2}\|y - z_d\|_2^2 + \frac{\lambda}{2}\|u\|_2^2, \\ \text{subject to } e(y,u) = 0. \end{cases} \quad (1)$$

J and e **continuously Fréchet derivable** functionals :

$$J : Y \times U \to \mathbb{R}, \;\; e : Y \times U \to W,$$

Y, U and W *reflexive* Banach spaces, if

- $e_y(\overline{y}, \overline{u}) \in \mathcal{B}(Y, W)$ is a bijection;
- $y(u)$ is a (*locally*) unique solution to the state equation $e(y,u) = 0$.

Then the problem in terms of the *Reduced Cost Functional* $f(u)$,

$$\min_{u \in U} f(u) = \min_{u \in U} J(y(u), u). \quad (2)$$

admits an optimal solution, if $f(u)$ is **continuous**, **convex** and **radially unbounded** (De los Reyes [2015])

Lagrangian conditions

To enforce the optimality conditions of the first order for this problem we consider its **Lagrangian formulation**

$$\mathcal{L}(y,u,p) = J(y,u) - < p, e(y,u) >_{W',W},$$

where $\mathcal{L} : Y \times U \times W' \to \mathbb{R}$, thus the optimality conditions are stated in terms of $\mathcal{L}$ as (see De los Reyes [2015]):

$$\begin{cases} e(y(\overline{u}),\overline{u}) = 0, \\ \mathcal{L}_y(y,\overline{u},p) = 0 \\ \mathcal{L}_u(y,\overline{u},p) = 0 \end{cases} . \quad (3)$$

General framework

In the following we will compute the **Lagrangian conditions** for our problems with **Fractional Partial Differential Equations** (FPDEs) state equations and use them to obtain an expression for both $f(u)$ in (2) and $\nabla f(u)$.

Lagrangian conditions

To enforce the optimality conditions of the first order for this problem we consider its **Lagrangian formulation**

$$\mathcal{L}(y,u,p) = J(y,u) - < p, e(y,u) >_{W',W},$$

where $\mathcal{L} : Y \times U \times W' \to \mathbb{R}$, thus the optimality conditions are stated in terms of $\mathcal{L}$ as (see De los Reyes [2015]):

$$\begin{cases} e(y(\overline{u}),\overline{u}) = 0, \\ \mathcal{L}_y(y,\overline{u},p) = 0 \\ \mathcal{L}_u(y,\overline{u},p) = 0 \end{cases} . \quad (3)$$

General framework

In the following we will compute the **Lagrangian conditions** for our problems with **Fractional Partial Differential Equations** (FPDEs) state equations and use them to obtain an expression for both $f(u)$ in (2) and $\nabla f(u)$.

Overview

- The Control Problem
- Fractional Partial Differential Equations
- Lagrangian Conditions
  - Discretization of the FPDEs
- The Algorithms: LBFGS and Approximate Inverses
  - The LBFGS Optimization Routine
  - Preconditioning
- Numerical Results
- Conclusions

Leibniz (1646–1716)

Leibniz: "[. . .] Thus it follows that $d^{\frac{1}{2}}x$ will be equal to $x\sqrt{dx : x}$. This is an apparent paradox from which, one day, useful consequences will be drawn."

In response to Marquis de l'Hôpital, 1695

We are going to **very briefly** define *some* Fractional Derivative operators and their discretizations.

Fractional Derivatives

- **Riemann-Liouville**: given $\alpha > 0$ and $m \in \mathbb{Z}^+$ such that $m-1 < \alpha \leq m$ the left-side Riemann-Liouville fractional derivative reads as

$$ {}_{\mathrm{RL}}D^{\alpha}_{a,x} y(x) = \frac{1}{\Gamma(m-\alpha)} \left(\frac{d}{dx}\right)^{m} \int_a^x \frac{y(\xi)d\xi}{(x-\xi)^{\alpha-m+1}}, $$

  while the right-side Riemann-Liouville fractional derivative

$$ {}_{\mathrm{RL}}D^{\alpha}_{x,b} y(x) = \frac{1}{\Gamma(m-\alpha)} \left(-\frac{d}{dx}\right)^{m} \int_x^b \frac{y(\xi)d\xi}{(\xi-x)^{\alpha-m+1}}; $$

- **Symmetric Riesz**: given $\alpha > 0$ and $m \in \mathbb{Z}^+$ such that $m-1 < \alpha \leq m$ the symmetric Riesz derivative reads as

$$ \frac{d^{\alpha} y(x)}{d|x|^{\alpha}} = \frac{1}{2}\left({}_{\mathrm{RL}}D^{\alpha}_{a,x} + {}_{\mathrm{RL}}D^{\alpha}_{x,b}\right) y(x); $$

The two FPDEs we consider as State Equations

We consider both the **Fractional Advection Dispersion Equation** (FADE)

$$\begin{cases} -a\left(l\,{}_{\text{RL}}D^{2\alpha}_{0,x} + r\,{}_{\text{RL}}D^{2\alpha}_{x,1}\right)y(x) + b(x)y'(x)\ldots, \quad x \in \Omega = (0,1), \\ \qquad \ldots + c(x)y(x) = u(x) \\ y(0) = y(1) = 0. \end{cases}.$$

for $\alpha \in (1/2, 1)$, $a, l, r > 0$ and $l + r = 1$, $b(x) \in \mathcal{C}^1(\overline{\Omega})$, $c(x) \in \mathcal{C}(\overline{\Omega})$ and such that $c(x) - 1/2 b'(x) \geq 0$,

The two FPDEs we consider as State Equations

We consider both the **Fractional Advection Dispersion Equation** (FADE)

$$\begin{cases} -a\left(l\,{}_{\mathrm{RL}}D^{2\alpha}_{0,x} + r\,{}_{\mathrm{RL}}D^{2\alpha}_{x,1}\right) y(x) + b(x)y'(x)\ldots, & x \in \Omega = (0,1), \\ \qquad \ldots + c(x)y(x) = u(x) & \\ y(0) = y(1) = 0. & \end{cases}.$$

for $\alpha \in (1/2, 1)$, $a, l, r > 0$ and $l + r = 1$, $b(x) \in \mathcal{C}^1(\overline{\Omega})$, $c(x) \in \mathcal{C}(\overline{\Omega})$ and such that $c(x) - 1/2 b'(x) \geq 0$, and the 2D **Riesz Space Fractional Diffusion equation**

$$\begin{cases} -K_{x_1}\frac{\partial^{2\alpha} y}{\partial |x_1|^{2\alpha}} - K_{x_2}\frac{\partial^{2\beta} y}{\partial |x_2|^{2\beta}} + \mathbf{b}\cdot\nabla y + cy = u, & (x_1, x_2) \in \Omega \\ y \equiv 0, & (x_1, x_2) \in \partial\Omega. \end{cases},$$

where $\mathbf{b} \in \mathcal{C}^1(\Omega, \mathbb{R}^2)$, $c \in \mathcal{C}(\Omega)$, $u \in \mathbb{L}^2(\Omega)$, $K_{x_1}, K_{x_2} \geq 0$ and $K_{x_1} + K_{x_2} > 0$, $\alpha, \beta \in (1/2, 1)$, $\Omega = [a, b] \times [c, d]$.

Overview

Lagrangian conditions for the FADE - I

Proposition 1: Lagrange Conditions for the FADE

$U = \mathbb{L}^2(\Omega)$, $Y = H_0^{\alpha}(\Omega)$ for $\alpha > 0$ the closure of $\mathbb{C}_0^{\infty}(\Omega)$ in the norm,

$$\|u\|_{\alpha} = \left(\|u\|_2^2 + \| \,_{\text{RL}}D_{0,x}^{\alpha} u\|_2^2\right)^{1/2}, \tag{4}$$

and $W = Y'$. The Lagrangian conditions for problem

$$\left\{\begin{array}{rl} \min J(y,u) = & \frac{1}{2}\|y - z_d\|_2^2 + \frac{\lambda}{2}\|u\|_2^2, \\ \text{subject to} & -a\left(l \,_{\text{RL}}D_{0,x}^{2\alpha} + r \,_{\text{RL}}D_{x,1}^{2\alpha}\right) y(x) + \\ & +b(x)y'(x) + c(x)y(x) - u(x) = 0, \\ & y(0) = y(1) = 0. \end{array}\right. , \quad (*)$$

LAGRANGIAN CONDITIONS FOR THE FADE - I

Proposition 1: Lagrange Conditions for the FADE

$U = \mathbb{L}^2(\Omega)$, $Y = H_0^\alpha(\Omega)$ for $\alpha > 0$ the closure of $\mathbb{C}_0^\infty(\Omega)$ in the norm,

$$\|u\|_\alpha = \left(\|u\|_2^2 + \| \,_{\text{RL}}D_{0,x}^\alpha u\|_2^2\right)^{1/2}, \tag{4}$$

and $W = Y'$. The Lagrangian conditions for problem (*) are expressed in *weak form* as

$$\begin{cases} B(y,v) = F(v), & \forall\, v \in Y, \\ \tilde{B}(p,w) = < y - z_d, w >, & \forall w \in Y, \\ < p,h > = -\lambda < u,h >, & \forall\, h \in Y. \end{cases},$$

LAGRANGIAN CONDITIONS FOR THE FADE - I

Proposition 1: Lagrange Conditions for the FADE

$U = \mathbb{L}^2(\Omega)$, $Y = H_0^{\alpha}(\Omega)$ for $\alpha > 0$ the closure of $\mathbb{C}_0^{\infty}(\Omega)$ in the norm,

$$\|u\|_{\alpha} = \left(\|u\|_2^2 + \|\,{}_{\mathrm{RL}}D_{0,x}^{\alpha} u\|_2^2\right)^{1/2}, \tag{4}$$

and $W = Y'$. The Lagrangian conditions for problem (*) where $u \in U$, $y \in Y$ and

$$\begin{aligned}
B(y, v) = &- al < {}_{\mathrm{RL}}D_{0,x}^{\alpha} y(x), {}_{\mathrm{RL}}D_{x,1}^{\alpha} v(x) > + \\
&- ar < {}_{\mathrm{RL}}D_{x,1}^{\alpha} y(x), {}_{\mathrm{RL}}D_{0,x}^{\alpha} v(x) > \\
&+ < b(x) y'(x) + c(x) y(x), v(x) >, \\
\tilde{B}(p, w) = &- al < {}_{\mathrm{RL}}D_{0,x}^{\alpha} w(x), {}_{\mathrm{RL}}D_{x,1}^{\alpha} p(x) > + \\
&- ar < {}_{\mathrm{RL}}D_{x,1}^{\alpha} w(x), {}_{\mathrm{RL}}D_{0,x}^{\alpha} p(x) > + \\
&- < (b(x) p(x))', w > + < c\, p, w >, \\
F(v) = &< u, v > .
\end{aligned}$$

Lagrangian conditions for the FADE - II

Gradient and objective function

The gradient for the objective function $f(u)$ for the FADE problem reads as

$$\nabla f(u) = \mathcal{L}_u(y(u), u, p(u)) = p + \lambda u,$$

where p is the solution of the adjoint equation:

$$\begin{cases} -a\left(r\,{}_{\text{RL}}D^{2\alpha}_{0,x} + l\,{}_{\text{RL}}D^{2\alpha}_{x,1}\right)p(x) - \frac{d}{dx}(b(x)p(x)) + \ldots \\ \qquad \ldots + c(x)p(x) = y(x) - z_d(x), \\ p(0) = p(1) = 0. \end{cases}$$

Lagrangian conditions for the Riesz FPDE - I

Lagrange Conditions for the Riesz FPDE

$U = \mathbb{L}^2(\Omega)$, $Y = H_0^{\alpha}(\Omega) \cap H_0^{\beta}(\Omega)$ and $W = Y'$. The Lagrangian conditions (3) for problem

$$\begin{cases} \min J(y,u) = \frac{1}{2}\|y - z_d\|_2^2 + \frac{\lambda}{2}\|u\|_2^2, \\ \text{subject to } -K_{x_1}\frac{\partial^{2\alpha} y}{\partial |x_1|^{2\alpha}} - K_{x_2}\frac{\partial^{2\beta} y}{\partial |x_1|^{2\beta}} + \mathbf{b}\cdot\nabla y + cy = u \\ y \equiv 0, (x_1, x_2) \in \partial\Omega. \end{cases} \quad (**)$$

Lagrangian conditions for the Riesz FPDE - I

Lagrange Conditions for the Riesz FPDE

$U = \mathbb{L}^2(\Omega)$, $Y = H_0^{\alpha}(\Omega) \cap H_0^{\beta}(\Omega)$ and $W = Y'$. The Lagrangian conditions (3) for problem (**) are expressed in *weak form* as

$$\left\{ \begin{array}{ll} B(y, v) = F(v), & \forall v \in Y, \\ \tilde{B}(p, w) = < y - z_d, w >, & \forall w \in Y, \\ < p, h > = -\lambda < u, h >, & \forall h \in Y. \end{array} \right. , \qquad (***)$$

Lagrangian conditions for the Riesz FPDE - I

Lagrange Conditions for the Riesz FPDE

$U = \mathbb{L}^2(\Omega)$, $Y = H_0^\alpha(\Omega) \cap H_0^\beta(\Omega)$ and $W = Y'$. The Lagrangian conditions (3) for problem where in (***) $u \in U$, $y \in Y$ and, by setting $C_x = K_{x_1} c_{2\alpha}$ and $C_{x_2} = K_{x_2} c_{2\beta}$,

$$\begin{aligned}
B(y,v) =& C_{x_1} \left(< {}_{\mathrm{RL}}D^\alpha_{a,x_1} y, {}_{\mathrm{RL}}D^\alpha_{x_1,b} v > + < {}_{\mathrm{RL}}D^\alpha_{x_1,b} y, {}_{\mathrm{RL}}D^\alpha_{a,x_1} v > \right) \\
& + C_{x_2} \left(< {}_{\mathrm{RL}}D^\beta_{c,x_2} y, {}_{\mathrm{RL}}D^\beta_{x_2,d} v > + < {}_{\mathrm{RL}}D^\beta_{x_2,d} y, {}_{\mathrm{RL}}D^\beta_{c,x_2} v > \right) \\
& + < \mathbf{b} \cdot \nabla y, v > + < cy, v >, \\
\tilde{B}(p,w) =& C_{x_1} \left(< {}_{\mathrm{RL}}D^\alpha_{a,x_1} p, {}_{\mathrm{RL}}D^\alpha_{x_1,b} w > + < {}_{\mathrm{RL}}D^\alpha_{x_1,b} p, {}_{\mathrm{RL}}D^\alpha_{a,x_1} w > \right) \\
& + C_{x_2} \left(< {}_{\mathrm{RL}}D^\beta_{c,x_2} p, {}_{\mathrm{RL}}D^\beta_{x_2,d} w > + < {}_{\mathrm{RL}}D^\beta_{x_2,d} p, {}_{\mathrm{RL}}D^\beta_{c,x_2} w > \right) \\
& - < \nabla \cdot (\mathbf{b} p), w > + < cp, w >, \\
F(v) =& < u, v > .
\end{aligned}$$

LAGRANGIAN CONDITIONS FOR THE RIESZ FPDE - II

Gradient and objective function

The gradient for the objective function $f(u)$ for the Riesz problem reads as

$$\nabla f(u) = \mathcal{L}_u(y(u), u, p(u)) = p + \lambda u,$$

where p is the solution of the adjoint equation for $\mathbf{x} \in \Omega$:

$$\begin{cases} -K_{x_1} \frac{\partial^{2\alpha} p}{\partial |x_1|^{2\alpha}} - K_{x_2} \frac{\partial^{2\beta} p}{\partial |x_2|^{2\beta}} - \nabla \cdot (p\mathbf{b}) + cp = y - z_d, & (x_1, x_2) \in \Omega \\ p \equiv 0, & (x_1, x_2) \in \partial\Omega. \end{cases}.$$

DISCRETIZATIONS

We used, for the solution of both the *state and adjoint equation* the *finite difference discretizations* over a grid $\{x_k\}_k = \{a + kh\}_k$ on the domain $\Omega = [a, b]$ with stepsize $h = {}^{b-a}/_{n}$ with the right p–shifted Grünwald–Letnikov formula as

$$ {}_{\mathrm{RL}}D^{\alpha}_{a,x} y(x)\big|_{x=x_k} \approx \frac{1}{h^{\alpha}} \sum_{j=0}^{k+p} \omega_j^{(\alpha)} \left[y(x_{k-j+p}) - y(a) \right] + \frac{y(a) x_k^{-\alpha}}{\Gamma(1-\alpha)} + O(h^2), $$

where the coefficients $\omega_j^{(\alpha)}$ are defined as $\omega_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}$, and similarly for the left–sided operator as

$$ {}_{\mathrm{RL}}D^{\alpha}_{x,b} y(x)\big|_{x=x_k} \approx \frac{1}{h^{\alpha}} \sum_{j=0}^{n-k+p} \omega_j^{(\alpha)} \left[y(x_{k+j-p}) - y(b) \right] + \frac{y(b) x_k^{-\alpha}}{\Gamma(1-\alpha)} + O(h^2). $$

Discretization of *Riesz Space Fractional derivative* is done in an analogous way.

Discretizations: Matrix Representation - I

In the **1D case**, i.e., the FADE we have that:

$$A\mathbf{y} = \mathbf{u}, \text{ with}$$
$$A = a(lT_{(\alpha)} + rT_{(\alpha)}^T) + O_n,$$

with,

$$T_{(\alpha)} = -\frac{1}{h^\alpha} \begin{bmatrix} \omega_1 & \omega_0 & 0 & \dots & 0 \\ \omega_2 & \omega_1 & \omega_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \omega_1 & \omega_0 \\ \omega_{n-1} & \dots & \dots & \omega_2 & \omega_1 \end{bmatrix}$$

Discretizations: Matrix Representation - I

In the **1D case**, i.e., the FADE we have that:

$$A\mathbf{y} = \mathbf{u}, \text{ with}$$
$$A = a(lT_{(\alpha)} + rT_{(\alpha)}^T) + O_n,$$

with,

$$O_n = \begin{bmatrix} c_1 & \frac{b_1}{2h} & & \\ -\frac{b_2}{2h} & \ddots & \ddots & \\ & \ddots & c_3 & \frac{b_{n-2}}{2h} \\ & & -\frac{b_{n-1}}{2h} & c_{n-1} \end{bmatrix}$$

Discretizations: Matrix Representation - I

In the **1D case**, i.e., the FADE we have that:

$$A\mathbf{y} = \mathbf{u}, \text{ with}$$
$$A = a(lT_{(\alpha)} + rT_{(\alpha)}^T) + O_n,$$

with,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1}. \end{bmatrix}$$

Discretizations: Matrix Representation - II

For the 2D *Riesz Space Fractional Diffusion equation*, we get, through the use of Kronecker sums, given an (i, j)–finite difference grid with N_{x_1}, N_{x_2} nodes and amplitude h, k respectively

$$A = K_{x_1}(R_{x_1}^{(\alpha)} \otimes I_{x_2}) + K_{x_2}(I_{x_1} \otimes R_{x_2}^{(\beta)}) \\ + B_{x_1}(T_{x_1} \otimes I_{x_2}) + B_{x_2}(I_{x_1} \otimes T_{x_2}) + C,$$

where

- I_{x_i}, for $i = 1, 2$ for the relative grid size,
- $R_{x_1}^{(\alpha)}$ and $R_{x_2}^{(\beta)}$ are the **dense Toeplitz matrix** for the 1D fractional order derivatives in the two directions,
- T_{x_i} are the finite difference matrix for the **convective terms** obtained with centered differences,
- the $\{B_{x_i}\}_{i=1,2}$ and C are respectively the evaluation of the functions $\mathbf{b} = (b^{(1)}, b^{(2)})$ and c on the relative nodes of the grid.

Overview

OPTIMIZATION ROUTINE: L–BFGS

Input: $\mathbf{x}_0$ starting point, $M \in \mathbb{N}$;
Compute $f_0 = f(\mathbf{x}_0)$ and $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
Set $\mathbf{d}_0 = -\mathbf{g}_0$, $\gamma_0 = 1$, $i = 0$;
while $\|\mathbf{g}_i\| > 0$ **do**
  Choose an initial Hessian approx.: $H_i^0 = \gamma_i I$;
  Compute the step direction $\mathbf{d}_i = -\mathbf{r}$ using *TwoLoopRecursion*;
  Compute the step length α_i and set $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$;
  Set $\mathbf{g}_i = \nabla f(\mathbf{x}_{i+1})$;
  if $i > M$ **then**
    Discard vectors $\{\mathbf{s}_{i-M}, \mathbf{y}_{i-M}\}$ from storage ;
  end
  Set and store $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and $\mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$;
  Set $i = i + 1$;
end
Output: Output $\mathbf{x}_i \approx \mathbf{x}_*$;

The L–BFGS method, originally introduced in Liu and Nocedal [1989], continuously updates the quasi–Newton matrix, approximation of the Hessian of problem (2), using a fixed limited amount of storage.

Optimization Routine: L–BFGS

Input: $\mathbf{x}_0$ starting point, $M \in \mathbb{N}$;
Compute $f_0 = f(\mathbf{x}_0)$ and $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
Set $\mathbf{d}_0 = -\mathbf{g}_0$, $\gamma_0 = 1$, $i = 0$;
while $\|\mathbf{g}_i\| > 0$ **do**
  Choose an initial Hessian approx.: $H_i^0 = \gamma_i I$;
  Compute the step direction $\mathbf{d}_i = -\mathbf{r}$ using *TwoLoopRecursion*;
  Compute the step length α_i and set $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$;
  Set $\mathbf{g}_i = \nabla f(\mathbf{x}_{i+1})$;
  if $i > M$ **then**
    Discard vectors $\{\mathbf{s}_{i-M}, \mathbf{y}_{i-M}\}$ from storage ;
  end
  Set and store $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and $\mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$;
  Set $i = i + 1$;
end
Output: Output $\mathbf{x}_i \approx \mathbf{x}_*$;

$$H_i = (I - \rho \mathbf{s}\mathbf{y}^T) H (I - \rho \mathbf{y}\mathbf{s}T) + \rho \mathbf{s}\mathbf{s}^T$$

Input: $\mathbf{g}_i = \nabla f(\mathbf{x}_i)$;
Set $\mathbf{q} = \mathbf{g}_i$;
for $k = i-1, \ldots, i-M$ **do**
  $a_k = \frac{\mathbf{s}_k^T \mathbf{q}}{\mathbf{y}_k^T \mathbf{s}_k}$;
  $\mathbf{q} = \mathbf{q} - a_k \mathbf{y}_k$;
end
$\mathbf{r} = H_i^0 \mathbf{q}$;
for $k = i-M, i-M+1, \ldots i-1$ **do**
  $b = \frac{\mathbf{y}_k^T \mathbf{r}}{\mathbf{y}_k^T \mathbf{s}_k}$;
  $\mathbf{r} = \mathbf{r} + (a_k - b)\mathbf{s}_k$;
end
Output: Output $\mathbf{r} = H_i \mathbf{g}_i$;

Optimization Routine: L–BFGS

Input: $\mathbf{x}_0$ starting point, $M \in \mathbb{N}$;
Compute $f_0 = f(\mathbf{x}_0)$ and $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
Set $\mathbf{d}_0 = -\mathbf{g}_0$, $\gamma_0 = 1$, $i = 0$;
while $\|\mathbf{g}_i\| > 0$ **do**
  Choose an initial Hessian approx.: $H_i^0 = \gamma_i I$;
  Compute the step direction $\mathbf{d}_i = -\mathbf{r}$ using *TwoLoopRecursion*;
  Compute the step length α_i and set $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$;
  Set $\mathbf{g}_i = \nabla f(\mathbf{x}_{i+1})$;
  if $i > M$ **then**
    Discard vectors $\{\mathbf{s}_{i-M}, \mathbf{y}_{i-M}\}$ from storage ;
  end
  Set and store $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and $\mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$;
  Set $i = i + 1$;
end
Output: Output $\mathbf{x}_i \approx \mathbf{x}_*$;

The computational cost of the L–BFGS algorithm:

1. $O(Mn)$ flops of the optimization procedure,
2. The cost required for the evaluation of the function and the gradient $\Rightarrow$ this costs *at least* two solutions of FPDEs!

Building the preconditioner

We consider sparse approximations for nonsymmetric matrices, the **AINV preconditioner** of the form:

$$A^{-1} \approx WD^{-1}Z^T.$$

with

- W lower triangular,
- Z upper triangular,
- D diagonal.

Computed within a **GPU architecture**, see Bertaccini and Filippone [2016].
We stress that for *nonsymmetric* or *non Hermitian* matrices we speak about *biorthogonalization* and *conjugation* for symmetric or Hermitian ones.

Short–Memory Principle

Lets look into some properties of the matrices $T_{(\alpha)}$ and $T_{(\alpha)}^T$:

1. They are *Toeplitz* matrices,
2. The entries are such that:

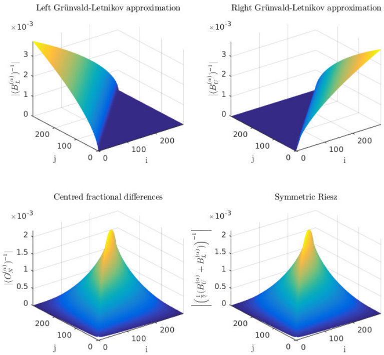
$$|\omega_j^{(\alpha)}| = O(j^{-\alpha-1}), \quad \text{for } j \to +\infty.$$

- Property (1) have been used to develop **preconditioners for iterative Toeplitz solvers**, e.g., [Donatelli et al., 2016].
- Property (2) have been used to **discard entries** from $T_{(\alpha)}$ and $T_{(\alpha)}^T$ and then use **direct solvers**, e.g., [Popolizio, 2013].
- In Bertaccini and Durastante [2017] **Property (2)** have been exploited for obtaining sparse approximate inverses of the discretization matrix A.

Short–Memory Principle

To build the preconditioner we need only another *ingredient*:

- transport the **decay** of the element of the **discretization matrix** to a **decay** of the element of the **inverse**.



This **decaying property** of these entries is a **structural property** of the fractional differential operators:

$$E(x) = |\,{}_{\mathrm{RL}}D^{\alpha}_{a,x}y(x) - {}_{\mathrm{RL}}D^{\alpha}_{x-L,x}y(x)| \leq \frac{\sup_{x\in[a,b]} y(x)}{L^{\alpha}|\Gamma(1-\alpha)|},$$

where $a \leq L < x$, is a *memory length*: the Short–Memory Principle.

Overview

A 2D Example

As a last example, we consider the following choice of coefficients for the problem,

$$\begin{cases} \min J(y,u) = \frac{1}{2}\|y - z_d\|_2^2 + \frac{\lambda}{2}\|u\|_2^2, \\ \text{subject to } -K_{x_1}\frac{\partial^{2\alpha} y}{\partial |x_1|^{2\alpha}} - K_{x_2}\frac{\partial^{2\beta} y}{\partial |x_2|^{2\beta}} + \mathbf{b}\cdot\nabla y + cy = u \\ y \equiv 0, (x_1, x_2) \in \partial\Omega. \end{cases}$$

with coefficients

$$K_{x_1} = 2,\ K_{x_2} = 1.5,\ c(x,y) = 1 + 0.5\cos(x_1 x_2),$$
$$\mathbf{b} = (\beta + 0.5\sin(4\pi x_1)\cos(5\pi x_2), \ldots$$
$$\ldots \alpha + 0.7\sin(7\pi x_2)\cos(4\pi x_1)),$$

On $\Omega = [0,1]^2$ for the desired state

$$z_d(x_1, x_2) = \sin(5\pi x_1)\sin(5\pi x_2).$$

A 2D Example

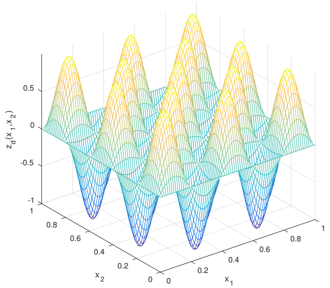
	Preconditioner	BiCGstab, $2\alpha = 1.3, 2\beta = 1.2, \lambda = 1e-6$					Direct
	I	$\sharp A\mathbf{v}$	$\sharp A'\mathbf{v}$	Func. Eval.	$\lVert\nabla f\rVert_2$	T(s)	T(s)
N	20	46476	73524	976	2.71e-04	4.9912	6.23
	40	80834	130061	899	1.15e-03	25.4829	146.40
	60	92609	158198	745	3.19e-03	116.5496	1669.18
	80	108270	197247	710	5.49e-03	488.6336	6029.44
	AINV(1e-1)	$\sharp A\mathbf{v}$	$\sharp A'\mathbf{v}$	Func. Eval.	$\lVert\nabla f\rVert_2$	T(s)	T(s)
N	20	20832	41173	1155	2.99e-04	3.2712	6.2328
	40	36156	73400	976	1.21-03	16.1750	146.40
	60	51793	109713	958	3.35e-03	89.8328	1669.18
	80	66275	140296	933	5.70e-03	385.6470	6029.44
	AINV(1e-2)	$\sharp A\mathbf{v}$	$\sharp A'\mathbf{v}$	Func. Eval.	$\lVert\nabla f\rVert_2$	T(s)	T(s)
N	20	10216	12361	947	3.22e-04	**1.7162**	6.23
	40	24771	38206	1497	1.40e-03	**14.2141**	146.40
	60	20346	34563	836	2.99e-03	**46.0631**	1669.18
	80	33924	59922	1079	4.37e-03	**207.9970**	6029.44

A 2D Example

	Preconditioner	BiCGstab, $2\alpha = 1.1, 2\beta = 1.8, \lambda = 1e-9$					Direct
	I	$\sharp A\mathbf{v}$	$\sharp A'\mathbf{v}$	Func. Eval.	$\|\nabla f\|_2$	T(s)	T(s)
N	20	34008	40346	380	3.02e-04	3.1414	2.67
	40	49603	58500	245	1.22e-03	13.0047	32.82
	60	81056	94216	237	3.02e-03	81.4658	335.74
	80	121244	140609	233	5.21e-03	421.9529	1432.13
	AINV(1e-1)	$\sharp A\mathbf{v}$	$\sharp A'\mathbf{v}$	Func. Eval.	$\|\nabla f\|_2$	T(s)	T(s)
N	20	13098	12349	388	3.11e-04	1.3566	2.67
	40	15854	14694	218	1.20e-03	4.8374	32.82
	60	23496	24406	237	3.15e-03	30.9931	335.74
	80	29293	29717	215	5.18e-03	112.3551	1432.13
	AINV(1e-2)	$\sharp A\mathbf{v}$	$\sharp A'\mathbf{v}$	Func. Eval.	$\|\nabla f\|_2$	T(s)	T(s)
N	20	5715	6036	382	3.03e-04	**0.8399**	2.67
	40	7682	7841	228	1.15e-03	**3.3400**	32.82
	60	13626	13914	238	3.18e-03	**25.2802**	335.74
	80	18825	20112	219	5.21e-03	**90.6548**	1432.13

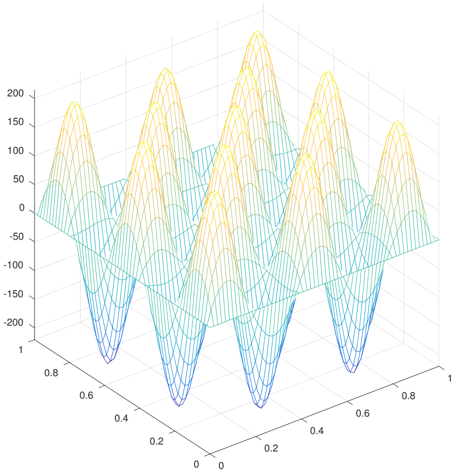
A 2D Example

The **desired state** and the **computed solution**.



A 2D Example

Obtained with the **control**



Overview

Conclusions and Future Work

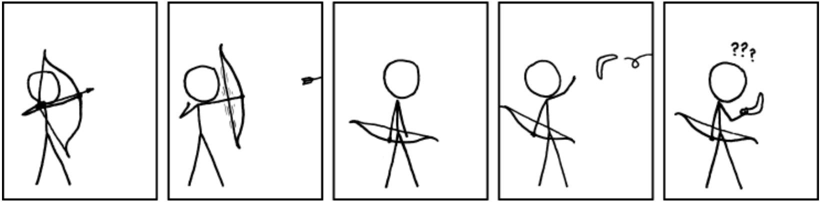
We have,

- Fully described two FPDEs–constrained optimization problem with tracking type cost functional,
- Devised a suitable coupling of the L-BFGS algorithm and an approximate inverse preconditioner for speeding up the computations,

We plan to,

- Add box–constraints on both *state* and *control* variable,
- Extend the model to account for fractional operators in the definition of $J(y, u)$,
- Extend the technique for considering also time–fractional problems,
- Adapt LQN optimization algorithms for these cases; see Cipolla et al. [2015].

Questions?



(From XKCD: `http://xkcd.com`)

Thanks for your attention!

References I

Daniele Bertaccini and Fabio Durastante. Solving mixed classical and fractional partial differential equations using short–memory principle and approximate inverses. *Numer. Algorithms*, 74(4):1061–1082, 2017.

Daniele Bertaccini and Salvatore Filippone. Sparse approximate inverse preconditioners on high performance GPU platforms. *Comp. Math. Appl.*, 71(3): 693–711, 2016.

S Cipolla, C Di Fiore, F Tudisco, and P Zellini. Adaptive matrix algebras in unconstrained minimization. *Linear Algebra Appl.*, 471:544–568, 2015.

Juan Carlos De los Reyes. *Numerical PDE–constrained optimization*. Springer, 2015.

Marco Donatelli, Mariarosa Mazza, and Stefano Serra-Capizzano. Spectral analysis and structure preserving preconditioners for fractional diffusion equations. *J. Comp. Phys.*, 307:262–279, 2016.

CT Kelley and EW Sachs. Quasi-newton methods and unconstrained optimal control problems. *SIAM J. Control Optim.*, 25(6):1503–1516, 1987.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1-3):503–528, 1989.

Marina Popolizio. A matrix approach for partial differential equations with Riesz space fractional derivatives. *Eur. Phys. J-Spec. Top.*, 222(8):1975–1985, 2013.

Infinite vs Finite

We interpret **discretized optimal control problems** as **optimization problems** and then solve them by a quasi–Newton method.

Discretized problems do not solve the original infinite-dimensional control problem but rather approximate it up to a certain accuracy!

This is justified by the mesh independence property; see De los Reyes [2015], Kelley and Sachs [1987]