conjugate gradients

intuition and application

jake roth

February 27, 2020

Outline

Outline

motivation

$$Ax = b, \quad A \succ 0, \quad A = A^\top$$

motivation

motivation

lifecycle of an optimization problem

consider $\min_x f(x)$

- build local model $m_k(d_k) = \langle \nabla_x f(x_k), d_k \rangle + \frac{1}{2}\langle d_k, \hat{H}(x_k) d_k \rangle$
- near a solution
  - expect $\hat{H}(x_k) \succ 0$
  - take Newton step $\hat{H}(x_k) d_k = -\nabla_x f(x_k)$
- boils down to solving $Ax = b$ for $A \succ 0, A = A^\top$

motivation

lifecycle of an optimization problem

consider $\min_x f(x)$

- build local model $m_k(d_k) = \langle \nabla_x f(x_k),\, d_k \rangle + \frac{1}{2}\langle d_k,\, \hat{H}(x_k) d_k \rangle$
- near a solution
  - expect $\hat{H}(x_k) \succ 0$
  - take Newton step $\hat{H}(x_k) d_k = -\nabla_x f(x_k)$
- boils down to solving $Ax = b$ for $A \succ 0, A = A^\top$

other examples

- least squares
  - linear: $\|y - Ax\|_2^2 \implies x^\star = (A^\top A)^{-1} A^\top y$
  - nonlinear: $\|y - f(x)\|_2^2 \implies \Delta x = (J^\top J)^{-1} J^\top y$
- rootfinding: $f(x) = 0$; interpret as $\min_x F(x)$ for $f = \nabla F$

motivation

lifecycle of an optimization problem

consider $\min_x f(x)$

- build local model $m_k(d_k) = \langle \nabla_x f(x_k), \, d_k \rangle + \frac{1}{2} \langle d_k, \, \hat{H}(x_k) d_k \rangle$
- near a solution
  - expect $\hat{H}(x_k) \succ 0$
  - take Newton step $\hat{H}(x_k) d_k = -\nabla_x f(x_k)$
- boils down to solving $Ax = b$ for $A \succ 0, A = A^\top$

other examples

- least squares
  - linear: $\|y - Ax\|_2^2 \implies x^\star = (A^\top A)^{-1} A^\top y$
  - nonlinear: $\|y - f(x)\|_2^2 \implies \Delta x = (J^\top J)^{-1} J^\top y$
- rootfinding: $f(x) = 0$; interpret as $\min_x F(x)$ for $f = \nabla F$

CG...from where?

- presented with algorithm and prove properties about algorithm
- but where does CG come from?

ongoing quadratic example

ongoing quadratic example

optimization

consider the unconstrained convex quadratic function

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) := \frac{1}{2} x^\top A x - b^\top x \tag{1}$$

where $A \succ 0, A = A^\top$

ongoing quadratic example

optimization

consider the unconstrained convex quadratic function

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) := \frac{1}{2} x^\top A x - b^\top x \tag{1}$$

where $A \succ 0, A = A^\top$

linear system

solution to Eq. (1) solves $Ax = b$

$$\nabla_x f(x) = 0 \iff Ax - b = 0 \tag{2}$$

because of convexity in f due to structure of A

history

history

optimization framework

- goal: $f(x_{k+1}) < f(x_k)$
- Newton method: Isaac Newton, 1600s
- gradient descent: Augustin-Louis Cauchy, 1850s
- nonlinear conjugate gradient: R. Fletcher and C.M. Reeves, 1960s

history

optimization framework

- goal: $f(x_{k+1}) < f(x_k)$
- Newton method: Isaac Newton, 1600s
- gradient descent: Augustin-Louis Cauchy, 1850s
- nonlinear conjugate gradient: R. Fletcher and C.M. Reeves, 1960s

linear systems framework

- Jacobi method (diagonally dominant $Ax = b$): Carl Gustav Jacob Jacobi, 1850s
- modified Richardson method (fixed step-size gradient descent $Ax = b$): Lewis Richardson, 1910)
- Krylov methods
  - CG (symmetric, positive-definite $Ax = b$): Magnus Hestenes and Eduard Stiefel, 1950s
  - GMRES (nonsymmetric $Ax = b$): 1950s

history

optimization framework

- goal: $f(x_{k+1}) < f(x_k)$
- Newton method: Isaac Newton, 1600s
- gradient descent: Augustin-Louis Cauchy, 1850s
- nonlinear conjugate gradient: R. Fletcher and C.M. Reeves, 1960s

linear systems framework

- Jacobi method (diagonally dominant $Ax = b$): Carl Gustav Jacob Jacobi, 1850s
- modified Richardson method (fixed step-size gradient descent $Ax = b$): Lewis Richardson, 1910)
- Krylov methods
  - CG (symmetric, positive-definite $Ax = b$): Magnus Hestenes and Eduard Stiefel, 1950s
  - GMRES (nonsymmetric $Ax = b$): 1950s

interpret CG from optimization and linear systems perspectives

history

optimization framework

- goal: $f(x_{k+1}) < f(x_k)$
- Newton method: Isaac Newton, 1600s
- gradient descent: Augustin-Louis Cauchy, 1850s
- nonlinear conjugate gradient: R. Fletcher and C.M. Reeves, 1960s

linear systems framework

- Jacobi method (diagonally dominant $Ax = b$): Carl Gustav Jacob Jacobi, 1850s
- modified Richardson method (fixed step-size gradient descent $Ax = b$): Lewis Richardson, 1910)
- Krylov methods
  - CG (symmetric, positive-definite $Ax = b$): Magnus Hestenes and Eduard Stiefel, 1950s
  - GMRES (nonsymmetric $Ax = b$): 1950s

interpret CG from optimization and linear systems perspectives

Outline

representing the error

representing the error

total error

from an initial guess x_0 dente the error

$$e_0 := x^\star - x_0 \tag{3}$$

where $x^\star = \arg\min f(x)$ from Eq. (1)

representing the error

total error

from an initial guess x_0 dente the error

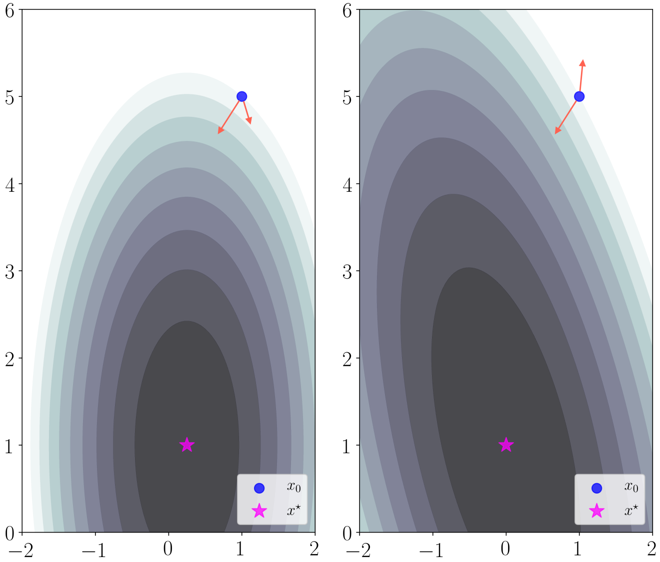
$$e_0 := x^\star - x_0 \tag{3}$$

where $x^\star = \arg\min f(x)$ from Eq. (1)

reconstruct the error

- suppose we have n linearly independent vectors $\{d_0, d_1, \ldots, d_{n-1}\}$
- can we build the error in one go?
  - $e_0 = \sum_i \alpha_i d_i$ for $\alpha_i \in \mathbb{R}$
  - easy if we know α_i
  - how can we find α_i?
- can we build the error iteratively? $e_k = x^\star - x_k$

linearly independent vectors



first-order iterative methods

search direction

- *first-order methods* use current (and possibly historical) gradient information to determine the next iterate
- update x_k with a step in direction d_k with

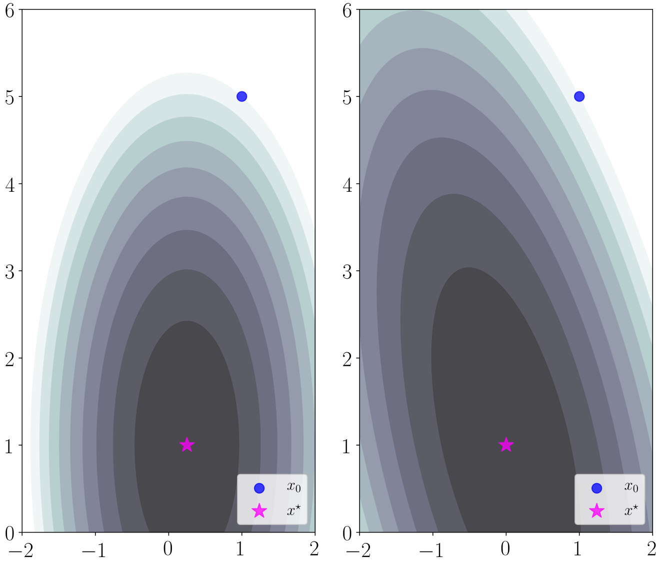
$$d_k \in x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \nabla f(x_2), \ldots, \nabla f(x_k)\} \quad (4)$$

  - gradient descent (GD): $d_k = -\nabla f(x_k)$
  - steepest descent (SD): $d_k = -\nabla f(x_k)$
  - coordinate descent (CD): $[d_k]_i = -[\nabla f(x_k)]_i$ if $i = \hat{i}$, 0 otherwise
  - conjugate gradient (CG): tbd

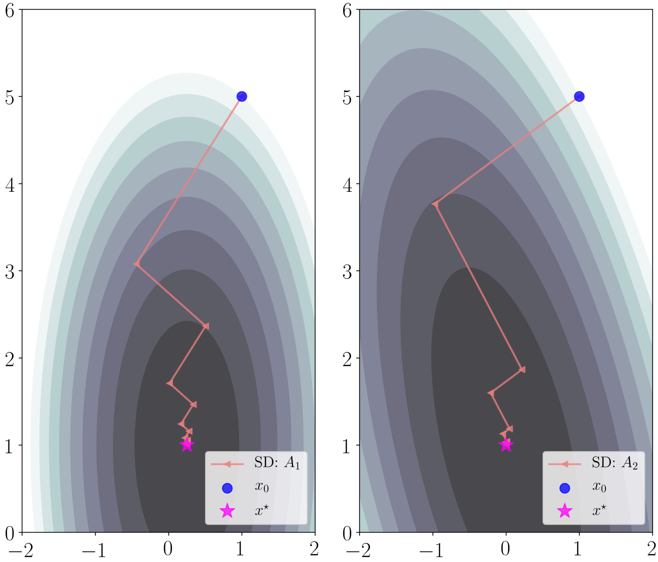
stepsize

- GD: $\alpha \leftarrow \bar{\alpha} \in \mathbb{R}_+$
- SD: $\alpha \leftarrow \alpha^\star$ where $\alpha^\star = \arg\min_\alpha f(x_k + \alpha d_k)$
- CD: $\alpha \leftarrow \alpha^\star$ where $\alpha^\star = \arg\min_\alpha f(x_k + \alpha d_k)$ (different d_k)
- CG: tbd

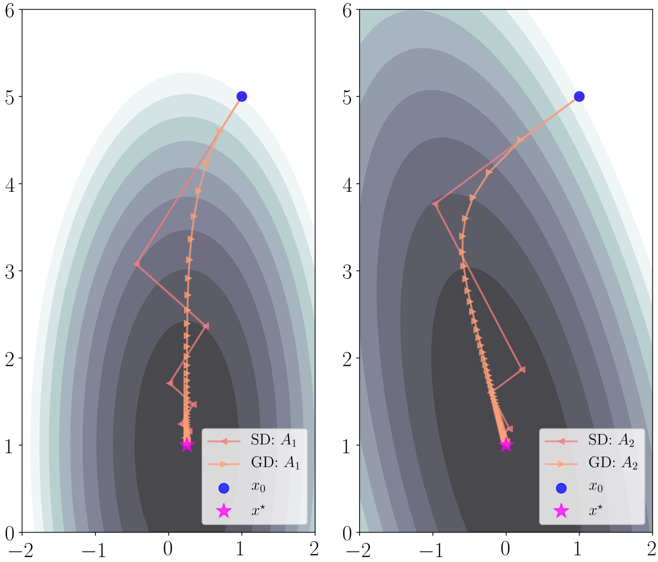
quadratic example (cont'd)



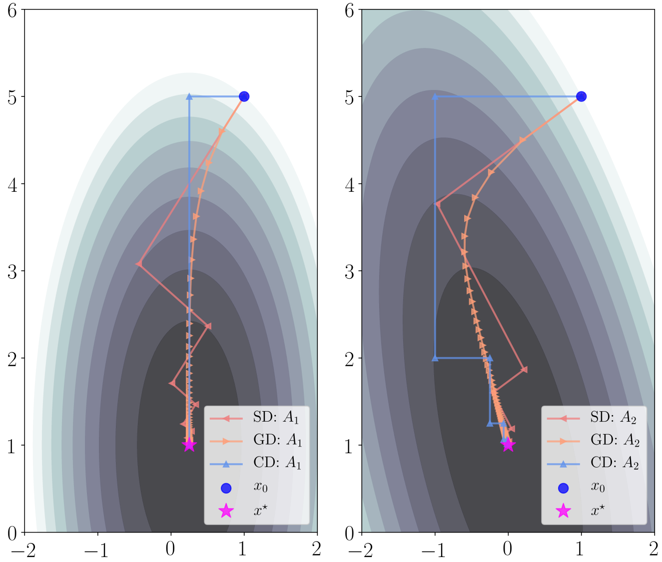
quadratic example (cont'd)



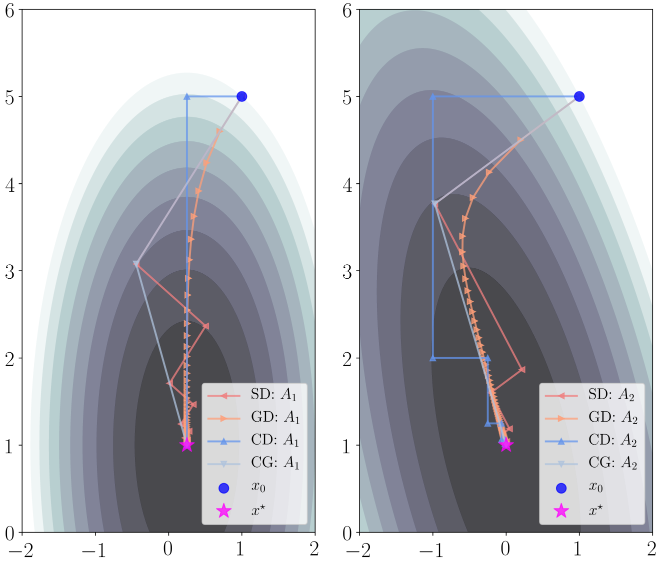
quadratic example (cont'd)



quadratic example (cont'd)



quadratic example (cont'd)



orthogonality and conjugacy

orthogonality and conjugacy

Definition 1 (orthogonality)

a set of vectors $\{d_1, d_2, \ldots\}$ are *orthogonal*, that is $d_i \perp d_j$, if $\langle d_i,\, d_j \rangle = 0$ for $i \neq j$

orthogonality and conjugacy

Definition 1 (orthogonality)

a set of vectors $\{d_1, d_2, \ldots\}$ are *orthogonal*, that is $d_i \perp d_j$, if $\langle d_i,\, d_j \rangle = 0$ for $i \neq j$

Definition 2 (conjugacy)

a set of vectors $\{d_1, d_2, \ldots\}$ are *conjugate* (orthogonal in a geometry induced by some $A \succ 0, A = A^\top$) if $\langle d_i,\, d_j \rangle_A := \langle d_i,\, Ad_j \rangle = 0$ for $i \neq j$

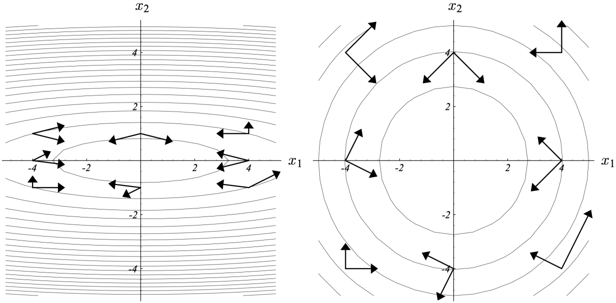
orthogonality and conjugacy

Definition 1 (orthogonality)

a set of vectors $\{d_1, d_2, \dots\}$ are *orthogonal*, that is $d_i \perp d_j$, if $\langle d_i,\, d_j \rangle = 0$ for $i \neq j$

Definition 2 (conjugacy)

a set of vectors $\{d_1, d_2, \dots\}$ are *conjugate* (orthogonal in a geometry induced by some $A \succ 0, A = A^\top$) if $\langle d_i,\, d_j \rangle_A := \langle d_i,\, Ad_j \rangle = 0$ for $i \neq j$



[1]

revisting coordinate descent (diagonal)

revisting coordinate descent (diagonal)

Assumption

A is diagonal and $D = [d_0, d_1, \ldots, d_{n-1}] \in \mathbb{R}^{n \times n}$ *contains n orthogonal directions; note that the principal axes of f's contours will align with* d_i

revisting coordinate descent (diagonal)

Assumption

A is diagonal and $D = [d_0, d_1, \ldots, d_{n-1}] \in \mathbb{R}^{n \times n}$ contains n orthogonal directions; note that the principal axes of f's contours will align with d_i

reconstruct the error

define error $e \coloneqq D\alpha$ with $\alpha = [\alpha_0, \ldots, \alpha_{n-1}]^\top$,

$$f(x+e) = f(x) + \sum_{i,j} x_i A_{i,j} e_j + \tfrac{1}{2} \sum_{i,j} d_i A_{i,j} e_j - \sum_i b_i e_i \tag{5}$$

$$\begin{aligned} &= f(x) + \sum_{i,j} x_i A_{i,j} \sum_k \alpha_k d_{k,j} \\ &\quad + \tfrac{1}{2} \sum_{i,j} \sum_k \alpha_k d_{k,i} A_{i,j} \sum_k \alpha_k d_{k,j} - \sum_i b_i \sum_k \alpha_k d_{k,i} \end{aligned} \tag{6}$$

$$= f(x) + \sum_k \left[\tfrac{1}{2} \alpha_k^2 d_k^\top A d_k + \alpha_k x^\top A d_k - \alpha_k b^\top d_k \right] \tag{7}$$

so finally $\min_\alpha f(x+e) = f(x) + \sum_{k=0}^{n-1} \{\min_{\alpha_k} f(\alpha_k d_k)\}$

revisiting coordinate descent (non-diagonal, I)

revisiting coordinate descent (non-diagonal, I)

Assumption

suppose that D contains n A-conjugate vectors

revisiting coordinate descent (non-diagonal, I)

Assumption

suppose that D contains n A-conjugate vectors

interpretation 1: coordinate descent in D^{-1} space

- how can we convert to the diagonal case?

revisiting coordinate descent (non-diagonal, I)

Assumption

suppose that D contains n A-conjugate vectors

interpretation 1: coordinate descent in D^{-1} space

- how can we convert to the diagonal case?
  - change coordinates so that $\hat{x} = D^{-1}x$, and rewrite Eq. (1)

$$f(\hat{x}) = \frac{1}{2}\hat{x}^\top (D^\top A D)\hat{x} - (D^\top b)^\top \hat{x} \tag{8}$$

revisiting coordinate descent (non-diagonal, I)

Assumption

suppose that D contains n A-conjugate vectors

interpretation 1: coordinate descent in D^{-1} space

- how can we convert to the diagonal case?
  - change coordinates so that $\hat{x} = D^{-1}x$, and rewrite Eq. (1)

$$f(\hat{x}) = \frac{1}{2}\hat{x}^\top (D^\top A D)\hat{x} - (D^\top b)^\top \hat{x} \tag{8}$$

- by conjugacy, $D^\top AD$ is diagonal!

revisiting coordinate descent (non-diagonal, I)

Assumption

suppose that D contains n A-conjugate vectors

interpretation 1: coordinate descent in D^{-1} space

- how can we convert to the diagonal case?
  - change coordinates so that $\hat{x} = D^{-1}x$, and rewrite Eq. (1)

$$f(\hat{x}) = \frac{1}{2}\hat{x}^\top (D^\top A D)\hat{x} - (D^\top b)^\top \hat{x} \tag{8}$$

- by conjugacy, $D^\top A D$ is diagonal!
- proceed by solving n 1-dimensional minimization problems along each coordinate direction of $\hat{x}$ [2]

revisiting coordinate descent (non-diagonal, II)

revisiting coordinate descent (non-diagonal, II)

Assumption

suppose that D contains n A-conjugate vectors

revisiting coordinate descent (non-diagonal, II)

Assumption

suppose that D contains n A-conjugate vectors

interpretation 2: line search simplification

- rewrite Eq. (5) in vector form as

$$
\begin{aligned}
f(x+e) &= f(x) + \tfrac{1}{2}\alpha^\top D^\top A D\alpha + (D\alpha)^\top (Ax) - (D\alpha)^\top b && (9)\\
&= f(x) + \tfrac{1}{2}\alpha^\top D^\top A D\alpha + (D\alpha)^\top (Ax - b) && (10)
\end{aligned}
$$

so that $\alpha^\star = \arg\min_\alpha f(x + D\alpha)$ satisfies

$$
\alpha^\star = (D^\top A D)^{-1} D^\top (Ax - b) \qquad (11)
$$

revisiting coordinate descent (non-diagonal, III)

takeaway: k-optimality

- define the subspace $M_k := x_0 + \text{span}\{d_0, d_1, \ldots, d_k\}$
- after k steps, we have minimized the error *as much as possible* in the subspace $M_k \subset \mathbb{R}^n$
- $x_k = \arg\min_{x \in M_k} f(x)$
- hence gradients $\nabla_x f(x_{k+i}) \perp M_k$ for $i > 0$
  - x_k is optimal, so directional derivative is zero
  - $\langle \nabla_x f(x_k), v \rangle = 0, \quad \forall v \in M_k$

Outline

getting conjugate directions

getting conjugate directions

modify Gram-Schmidt process for orthogonality wrt A

- start with gradients $g_k := \nabla_x f(x_k)$ at each step as the orthogonalization vectors

$$d_{k+1} = g_{k+1} - \mathrm{proj}_{M_k}(g_{k+1}) = g_{k+1} - \sum_{i=0}^{k} \frac{\langle g_{k+1},\, d_j \rangle_A}{\langle d_j,\, d_j \rangle_A} d_j \qquad (12)$$

- computationally intensive and G-S is not numerically stable

getting conjugate directions

modify Gram-Schmidt process for orthogonality wrt A

- start with gradients $g_k \coloneqq \nabla_x f(x_k)$ at each step as the orthogonalization vectors

$$d_{k+1} = g_{k+1} - \mathrm{proj}_{M_k}(g_{k+1}) = g_{k+1} - \sum_{i=0}^{k} \frac{\langle g_{k+1},\, d_j \rangle_A}{\langle d_j,\, d_j \rangle_A} d_j \qquad (12)$$

- computationally intensive and G-S is not numerically stable

goal

simplify $\mathrm{proj}_{M_k}(g_{k+1})$ as much as possible [3]

conjugate gradients procedure (simplification I)

simplification I: projection summation

conjugate gradients procedure (simplification I)

simplification I: projection summation

1. solve for d_k in terms of the quantities x_k, x_{k+1}, α_k so

$$d_k = \frac{1}{\alpha_k}(x_{k+1} - x_k) \tag{13}$$

conjugate gradients procedure (simplification I)

simplification I: projection summation

1. solve for d_k in terms of the quantities x_k, x_{k+1}, α_k so

$$d_k = \frac{1}{\alpha_k}(x_{k+1} - x_k) \tag{13}$$

2. multiply by A (and cancel b terms)

$$Ad_k = \frac{1}{\alpha_k}A(x_{k+1} - x_k) = \frac{1}{\alpha_k}A(g_{k+1} - g_k) \tag{14}$$

conjugate gradients procedure (simplification I)

simplification I: projection summation

1. solve for d_k in terms of the quantities x_k, x_{k+1}, α_k so

$$d_k = \frac{1}{\alpha_k}(x_{k+1} - x_k) \tag{13}$$

2. multiply by A (and cancel b terms)

$$Ad_k = \frac{1}{\alpha_k}A(x_{k+1} - x_k) = \frac{1}{\alpha_k}A(g_{k+1} - g_k) \tag{14}$$

3. use k-optimality
   - orthogonality of gradients $g_{k+1} \perp M_k \implies g_{k+1} \perp \{g_0, g_1, \ldots, g_k\}$ since $\text{span}\{g_0, g_1, \ldots, g_k\} = M_k$ (taking $d_0 = g_0$)

conjugate gradients procedure (simplification I)

simplification I: projection summation

1. solve for d_k in terms of the quantities x_k, x_{k+1}, α_k so

$$d_k = \frac{1}{\alpha_k}(x_{k+1} - x_k) \tag{13}$$

2. multiply by A (and cancel b terms)

$$Ad_k = \frac{1}{\alpha_k}A(x_{k+1} - x_k) = \frac{1}{\alpha_k}A(g_{k+1} - g_k) \tag{14}$$

3. use k-optimality
   - orthogonality of gradients $g_{k+1} \perp M_k \implies g_{k+1} \perp \{g_0, g_1, \ldots, g_k\}$ since $\text{span}\{g_0, g_1, \ldots, g_k\} = M_k$ (taking $d_0 = g_0$)
4. conclude

$$d_{k+1} = g_{k+1} - \frac{\langle g_{k+1}, (g_{k+1} - g_k)\rangle}{\langle d_k, (g_{k+1} - g_k\rangle} d_k = \beta_k d_k \tag{15}$$

conjugate gradients procedure (simplification II)

simplification II: β_k

conjugate gradients procedure (simplification II)

simplification II: β_k

1. $g_{k+1} \perp d_k$ and $g_{k+1} \perp g_k$ by k-optimality so that

$$\beta_k = \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle d_k, g_k \rangle} \tag{16}$$

conjugate gradients procedure (simplification II)

simplification II: β_k

1. $g_{k+1} \perp d_k$ and $g_{k+1} \perp g_k$ by k-optimality so that

$$\beta_k = \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle d_k, g_k \rangle} \tag{16}$$

2. expand $d_k = g_k - \beta_{k-1} d_{k-1}$ and $d_k \perp g_{k-1}$ by k-optimality so that

$$\beta_k = \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle g_k, g_k \rangle} \tag{17}$$

conjugate gradients procedure

$g_0 \leftarrow Ax_0 - b; \quad d_0 \leftarrow -g_0; \quad k \leftarrow 0$

repeat

$$\alpha_k \leftarrow \frac{g_k^\mathsf{T} g_k}{d_k^\mathsf{T} A d_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k d_k$$

$$g_{k+1} \leftarrow g_k - \alpha_k A d_k$$

if $g_{k+1} \leq$ tolerance, then exit, else

$$\beta_k \leftarrow \frac{g_{k+1}^\mathsf{T} g_{k+1}}{g_k^\mathsf{T} g_k}$$

$$d_{k+1} \leftarrow -g_{k+1} + \beta_k d_k$$

$$k \leftarrow k + 1$$

end repeat

return x_{k+1}

[4]

Outline

NOPE!

but nice connections to finding roots of polynomials

Outline

Overview

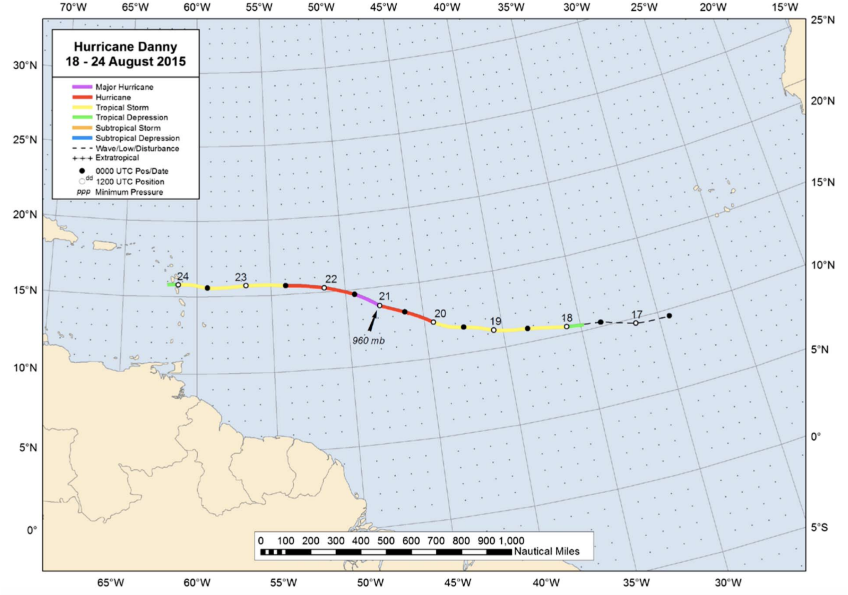
Background

Computation

Error (convergence) analysis

Application

simulating tropical cyclones [5]



optimal control problem

optimal control problem

consider the following optimal control problem

$$\begin{aligned} \underset{u \in C^1}{\text{minimize}} \quad & J(u) = \int_0^T \|u\|_A^2 \mathrm{d}t && \text{(18a)} \\ \text{s.t.} \quad & \dot{x}(t) = b(x) + u(t) && \text{(18b)} \\ & x(0) = x_0 && \text{(18c)} \\ & \Phi(x(T)) = 0 && \text{(18d)} \end{aligned}$$

optimal control problem

consider the following optimal control problem

$$\begin{aligned} \underset{u \in C^1}{\text{minimize}} \quad & J(u) = \int_0^T \|u\|_A^2 \mathrm{d}t && (18a) \\ \text{s.t.} \quad & \dot{x}(t) = b(x) + u(t) && (18b) \\ & x(0) = x_0 && (18c) \\ & \Phi(x(T)) = 0 && (18d) \end{aligned}$$

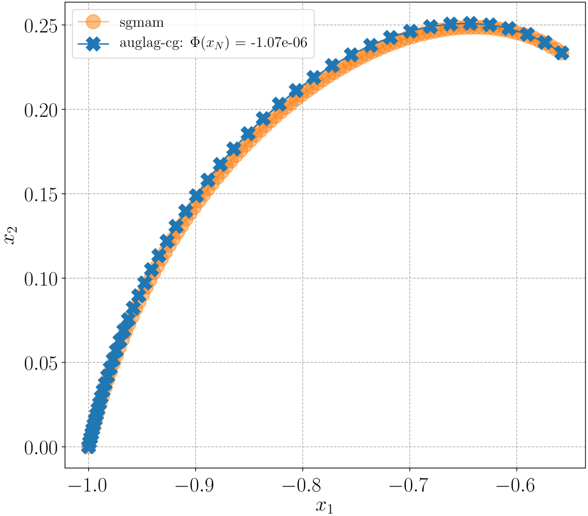
and discretize into

$$\begin{aligned} \underset{\{u_k\}_{k=1}^N}{\text{minimize}} \quad & J(u) = \Delta t \sum_{k=1}^{N} \left[u_k^\top A u_k \right] && (19a) \\ \text{s.t.} \quad & x_{k+1} = b(x_k)\Delta t + u_k \Delta t, \quad k \in [0, N-1] && (19b) \\ & x_1 = \bar{x} && (19c) \\ & \Phi(x_N) = 0 && (19d) \end{aligned}$$

coding example

CG in `Julia`, see: `cg-pres/ repo`

coding results



References I

[1] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," tech. rep., USA, 1994.

[2] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, second ed., 2006.

[3] M. Zibulevsky, "Conjugate gradient notes."

[4] Wikipedia, "Conjugate gradient method."

[5] D. A. Plotkin, R. J. Webber, M. E. O'Neill, J. Weare, and D. S. Abbot, "Maximizing Simulated Tropical Cyclone Intensity With Action Minimization," *Journal of Advances in Modeling Earth Systems*, vol. 11, no. 4, pp. 863–891, 2019.