PIPG for RPOD

Govind Chari

Univeristy of Washington

May 2023

PIPG Overview

- First order primal-dual conic optimizer
- Easily verifiable
- Factorization free
- Intuitive algorithm
- Can be coded in an afternoon
- Supports infeasibility detection
- Very fast (3-7x faster than ECOS)

Comparison with ECOS

	ECOS	PIPG
Verification Ease	✗	✓
Factorization Free	✗	✓
Understandable	✗	✓
Implementation Ease	✗	✓
Infeasibility Detection	✓	✓
Tuning Free	✓	✗

PIPG Algorithm

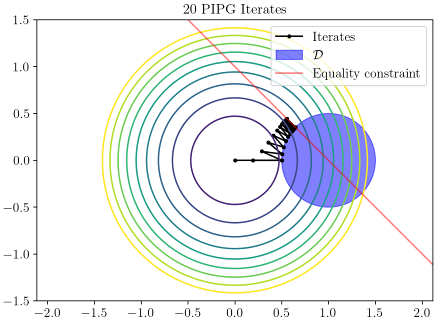
Optimization Problem

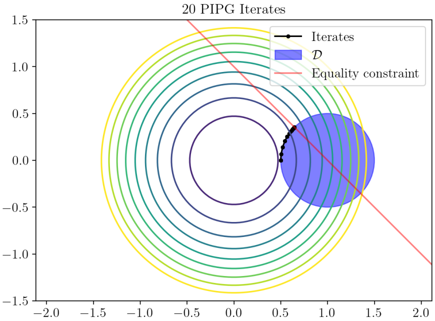
$$\begin{aligned} \min_{z} \quad & \frac{1}{2} z^\top P z + q^\top z \\ \text{s.t.} \quad & Hz - h = 0 \\ & z \in \mathbb{D} \end{aligned}$$

Algorithm

$$z^{k+1} \leftarrow \Pi_{\mathbb{D}} \left[z^k - \alpha (P z^k + q + H^\top w^k) \right]$$

$$w^{k+1} \leftarrow w^k + \beta (H(2z^{k+1} - z^k) - h)$$

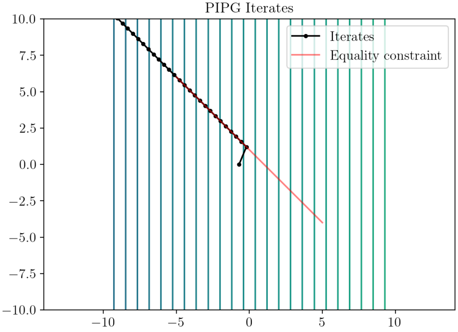


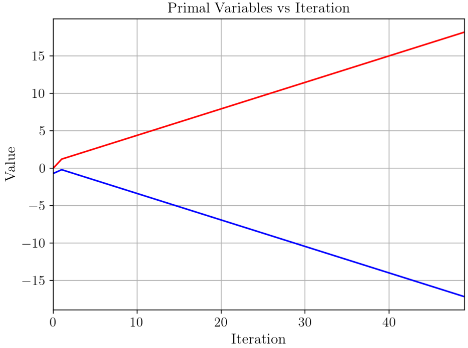


Infeasibility Detection - Dual Infeasibility

Dual Infeasibility Criteria:

$$\lim_{k \to \infty} \|z^{k+1} - z^k\| \neq 0$$

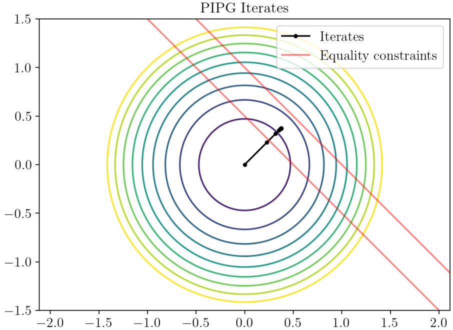


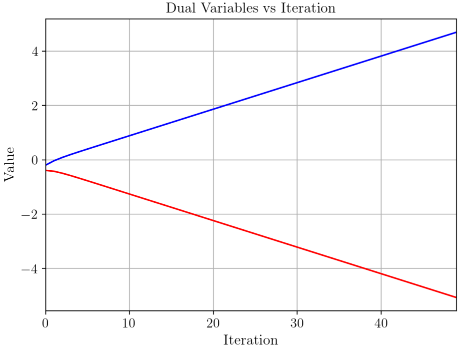


Infeasibility Detection - Primal Infeasibility

Primal Infeasibility Criteria:

$$\lim_{k\to\infty} \|w^{k+1} - w^k\| \neq 0$$





Customization

Continuous-time Nonconvex Problem

$$
\begin{aligned}
\underset{t_f,\, x(t), u(t)}{\text{minimize}} \quad & \int_0^{t_f} \|u(t)\|_2^2 \, dt \\
\text{subject to} \quad & \forall t \in [0, t_f) \\
\boxed{\text{CW Dynamics}} \quad & \dot{x}(t) = f(x(t), u(t)) \\
\boxed{\text{Max delta-v}} \quad & \|u(t)\|_2 \le u_{\text{max}} \\
\boxed{\text{Keepout Zone}} \quad & \|r(t) - r_c\|_2 \ge \rho \\
\boxed{\text{Max Speed}} \quad & \|v(t)\|_2 \le v_{\max} \\
\boxed{\text{Initial Conditions}} \quad & r(0) = r_i \\
& v(0) = v_i \\
\boxed{\text{Terminal Conditions}} \quad & r(t_f) = 0_{3\times 1} \\
& v(t_f) = 0_{3\times 1}
\end{aligned}
$$

Successive Convex Programming (SCP)

Discrete-time Convex Subproblem

$$
\begin{aligned}
\underset{\sigma, x, u}{\text{minimize}} \quad & \sum_{k=1}^{K-1} \|u_k\|_2^2 + w_{tr} \left(\sum_{k=1}^{K} \|x_k - \hat{x}_k\|_2^2 + \sum_{k=1}^{K-1} [\|u_k - \hat{u}_k\|_2^2 + (\sigma_k - \hat{\sigma}_k)^2] \right) + w_{vc} \|\nu^c\|_1 + w_{vb} \sum_{k=1}^{K} \nu_k^b \\
\text{subject to} \quad & \forall k \in [1, K] \\
\boxed{\text{Discrete Dynamics}} \quad & x_{k+1} = A_k x_k + B_k u_k + S_k \sigma_k + c_k + \nu_k^c \\
\boxed{\text{Max delta-v}} \quad & \|u_k\|_2 \le u_{\text{max}} \\
\boxed{\text{Keepout Zone}} \quad & \|\hat{r}_k - r_c\|_2 + \left(\frac{\hat{r}_k - r_c}{\|\hat{r}_k - r_c\|_2} \right)^\top (r_k - \hat{r}_k) + \nu_k^b \ge \rho \\
& \nu_k^b \ge 0 \\
\boxed{\text{Max Speed}} \quad & \|v_k\|_2 \le v_{\max} \\
\boxed{\text{Initial Conditions}} \quad & r(0) = r_i \\
& v(0) = v_i \\
\boxed{\text{Terminal Conditions}} \quad & r(t_f) = 0_{3\times 1} \\
& v(t_f) = 0_{3\times 1}
\end{aligned}
$$

Trajectory

Speed Results

Notes

- Easily verifiable conic solver
- Vectorized and devectorized
- PIPG as full stack optimizaiton solution (Inf detection)
- If it is slow you have scaling issues
- Speed results