# Optimal and Long-Step Feasibility Algorithms

**Pontus Giselsson**, Mattias Fält

Dept. of Automatic Control
Faculty of Engineering (LTH)
Lund University

# Objective

- Create efficient algorithms for solving large-scale cone programs:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in \mathcal{K} \end{array}$$

  where $\mathcal{K}$ is a convex cone

- Special focus on high accuracy solutions

# Feasibility formulation

- Primal and dual problems:

$$\begin{array}{rl} \text{min} & c^T x \\ \text{s.t.} & Ax + s = b \\ & s \in \mathcal{K} \end{array} \qquad\qquad \begin{array}{rl} \text{max} & b^T y \\ \text{s.t.} & A^T y = -c \\ & y \in \mathcal{K}^* \end{array}$$

- Primal dual embedding, using strong duality ($c^T x + b^T y = 0$):

$$\begin{array}{ll} \text{find} & (x, s, y) \\ \text{such that} & \begin{bmatrix} 0 & 0 & A^T \\ A & I & 0 \\ c^T & 0 & b^T \end{bmatrix} \begin{bmatrix} x \\ s \\ y \end{bmatrix} = \begin{bmatrix} -c \\ b \\ 0 \end{bmatrix} \\ & (x, s, y) \in \mathbb{R}^n \times \mathcal{K} \times \mathcal{K}^* \end{array}$$

# Our focus

Method of alternating relaxed projections (MARP)[1]

or

Generalized alternating projections (GAP)[1]

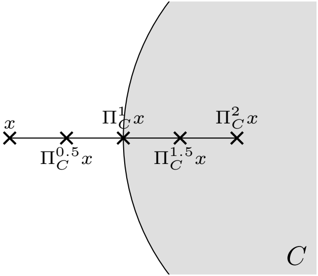
[1] S. Agmon, 1954. T. S. Motzkin and I. Shoenberg, 1964. L. M. Bregman, 1965.

# Relaxed projection

- Relaxed projection operator

$$\Pi_C^{\alpha} x \coloneqq (1-\alpha)x + \alpha \Pi_C x$$

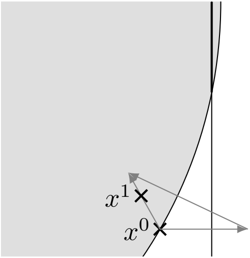
- Relaxation parameter $\alpha \in (0, 2]$ decides relaxed projection point



# Alternating relaxed projections

- Alternating relaxed projections:

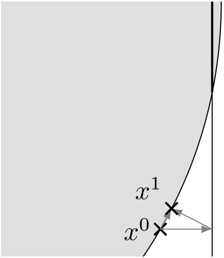
$$x^{k+1} = (1-\alpha)x^k + \alpha\Pi_D^{\alpha_2}\Pi_C^{\alpha_1}x^k$$



# Alternating relaxed projections

- Alternating relaxed projections:

$$x^{k+1} = (1-\alpha)x^k + \alpha \Pi_D^{\alpha_2} \Pi_C^{\alpha_1} x^k$$

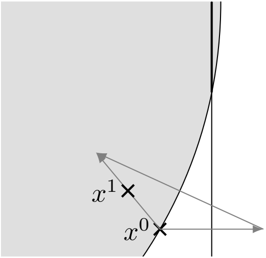



- Alternating projections: $(\alpha_1 = \alpha_2 = \alpha = 1)$

# Alternating relaxed projections

- Alternating relaxed projections:

$$x^{k+1} = (1-\alpha)x^k + \alpha \Pi_D^{\alpha_2} \Pi_C^{\alpha_1} x^k$$

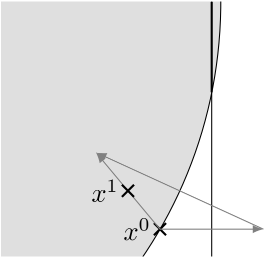



- Alternating projections: $(\alpha_1 = \alpha_2 = \alpha = 1)$
- Douglas-Rachford: $(\alpha_1 = \alpha_2 = 2, \alpha = 1/2)$

# Alternating relaxed projections

- Alternating relaxed projections:

$$x^{k+1} = (1-\alpha)x^k + \alpha \Pi_D^{\alpha_2} \Pi_C^{\alpha_1} x^k$$



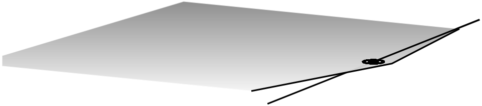

- Alternating projections: $(\alpha_1 = \alpha_2 = \alpha = 1)$
- Douglas-Rachford: $(\alpha_1 = \alpha_2 = 2, \alpha = 1/2)$
- Performance and behavior highly dependent on parameters
- Interpretation: Exploration-exploitation trade-off

# 3D example

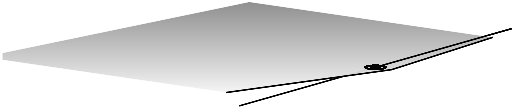
# 3D example

# 3D example

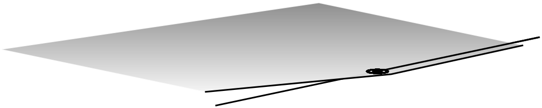
# 3D example



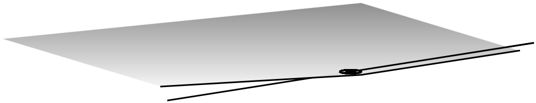
# 3D example



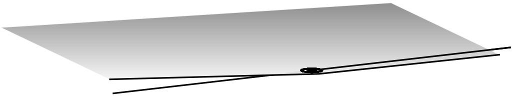
# 3D example



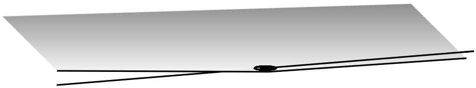
# 3D example



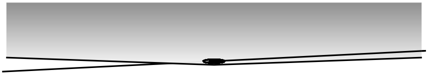
# 3D example



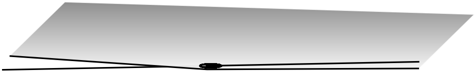
# 3D example



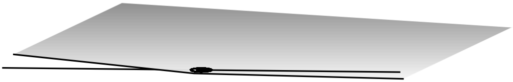
# 3D example



# 3D example



# 3D example

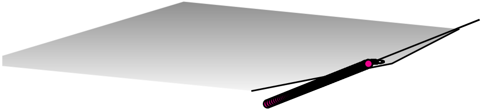


# 3D example – Alternating projections

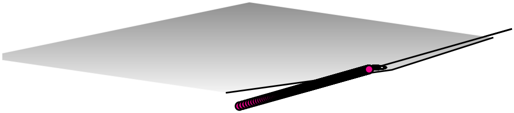
# 3D example – Alternating projections

# 3D example – Alternating projections

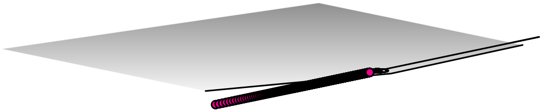
# 3D example – Alternating projections



# 3D example – Alternating projections

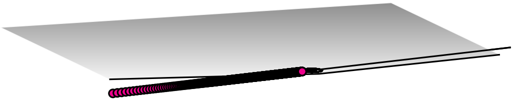


# 3D example – Alternating projections

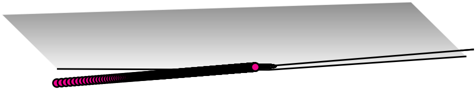


# 3D example – Alternating projections

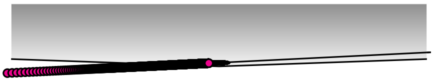
# 3D example – Alternating projections



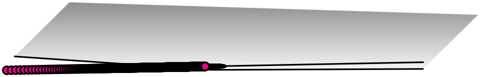
# 3D example – Alternating projections



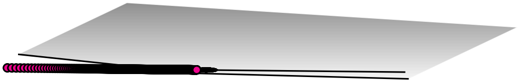
# 3D example – Alternating projections



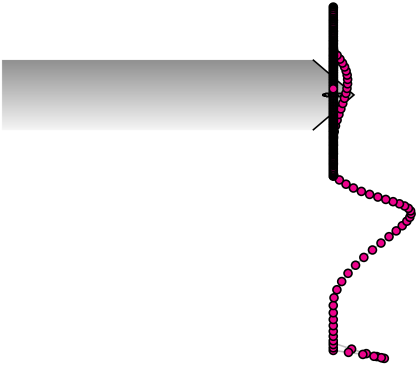
# 3D example – Alternating projections



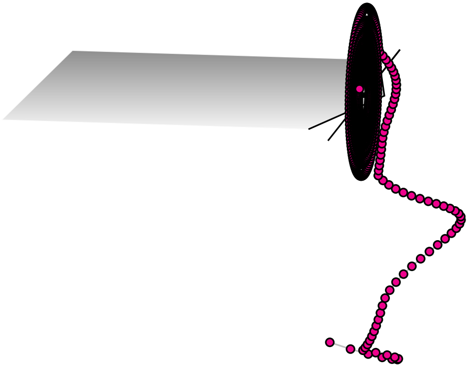
# 3D example – Alternating projections



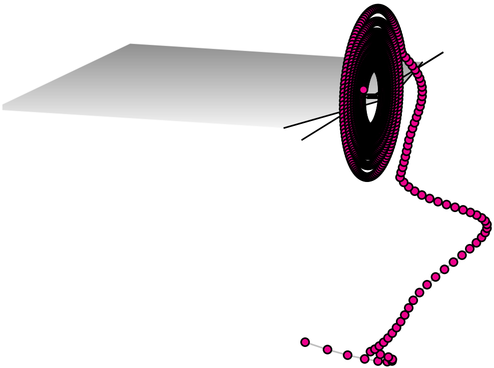
# 3D example – Douglas-Rachford



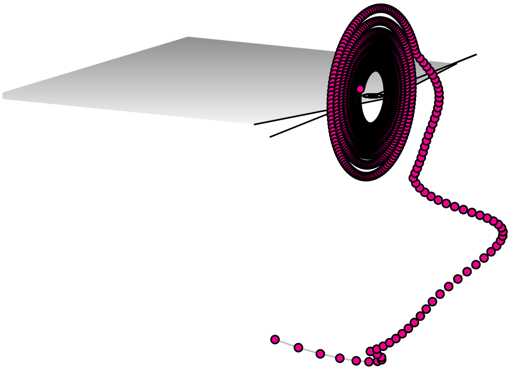
# 3D example – Douglas-Rachford



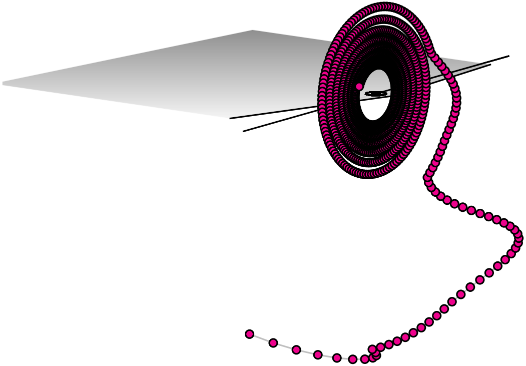
# 3D example – Douglas-Rachford



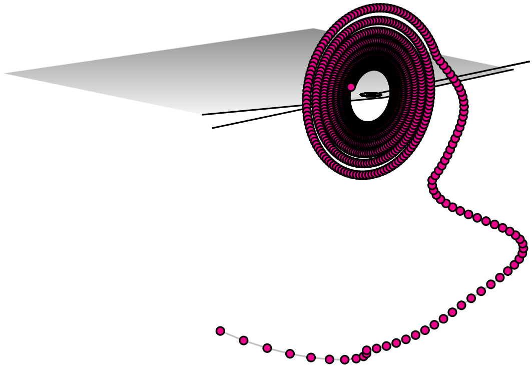
# 3D example – Douglas-Rachford



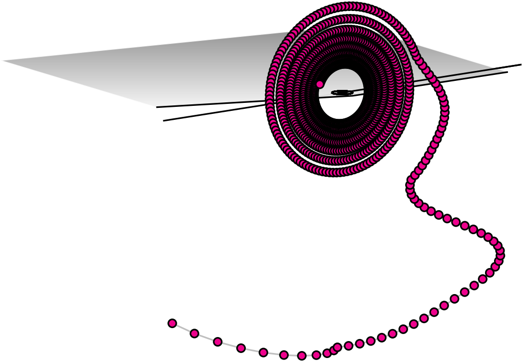
# 3D example – Douglas-Rachford



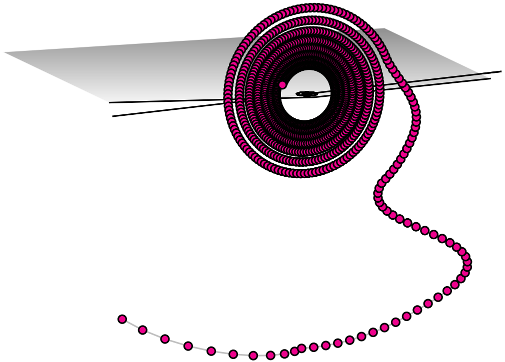
# 3D example – Douglas-Rachford



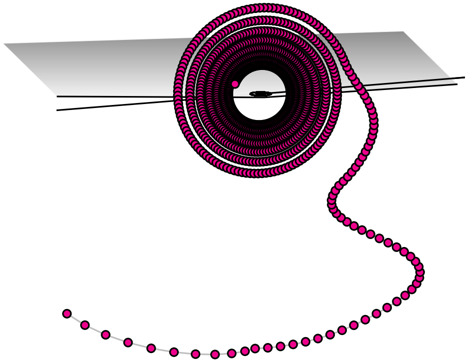
# 3D example – Douglas-Rachford



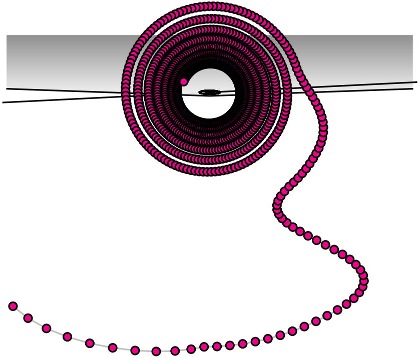
# 3D example – Douglas-Rachford



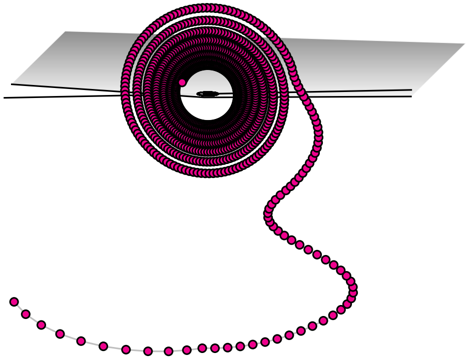
# 3D example – Douglas-Rachford



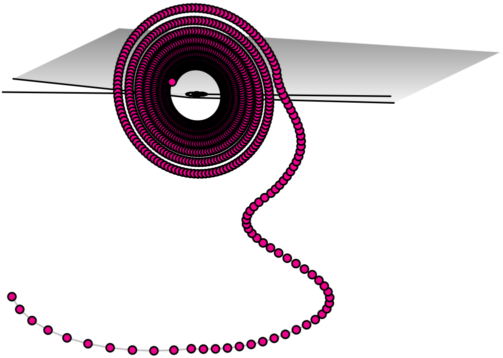
# 3D example – Douglas-Rachford



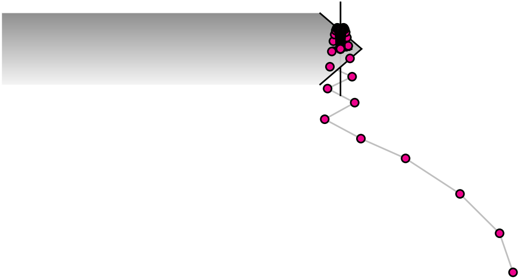
# 3D example – Douglas-Rachford



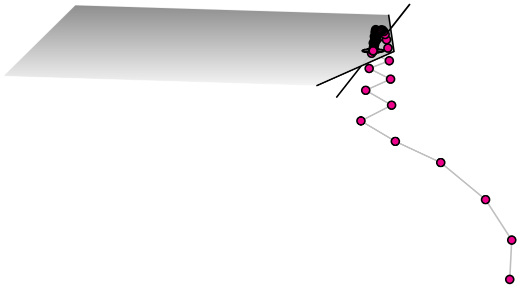
# 3D example – Douglas-Rachford



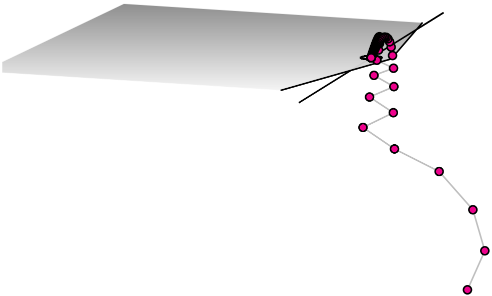
# 3D example – Trade-off



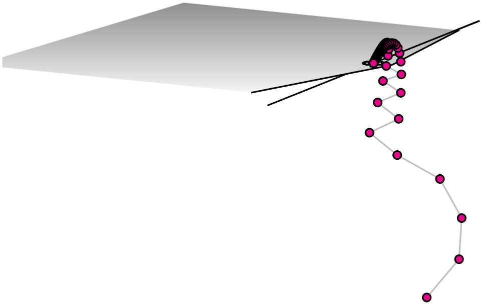
# 3D example – Trade-off



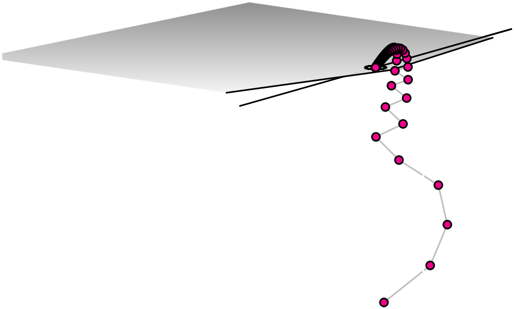
# 3D example – Trade-off



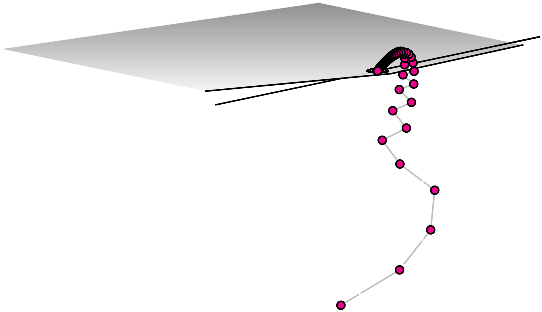
# 3D example – Trade-off



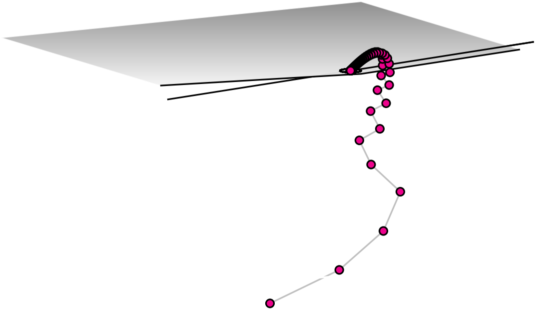
# 3D example – Trade-off



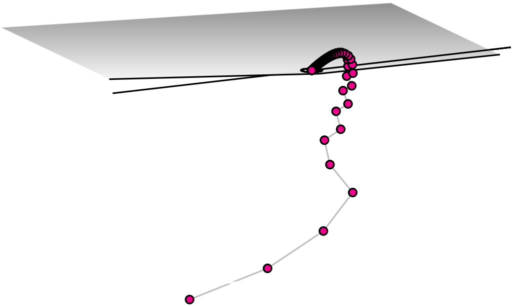
# 3D example – Trade-off



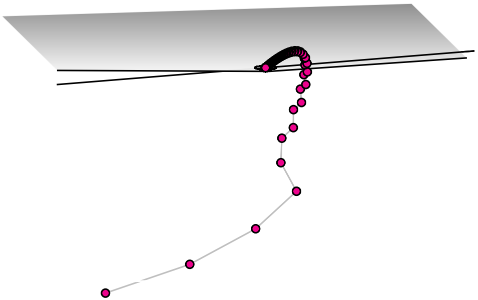
# 3D example – Trade-off



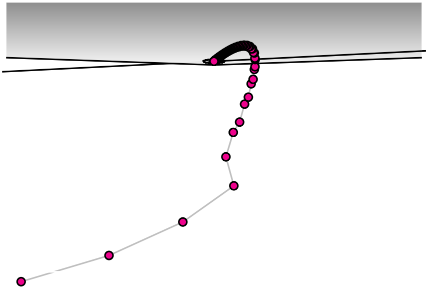
# 3D example – Trade-off



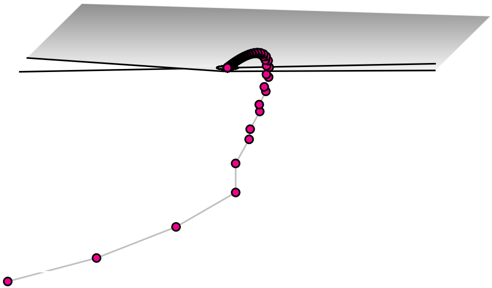
# 3D example – Trade-off



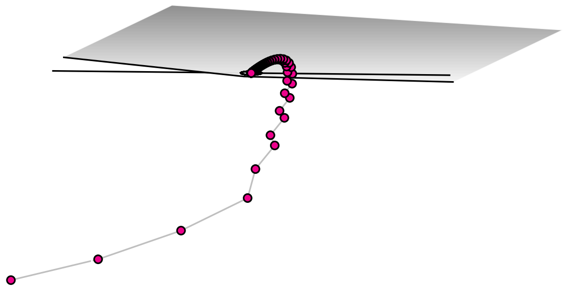
# 3D example – Trade-off



# 3D example – Trade-off



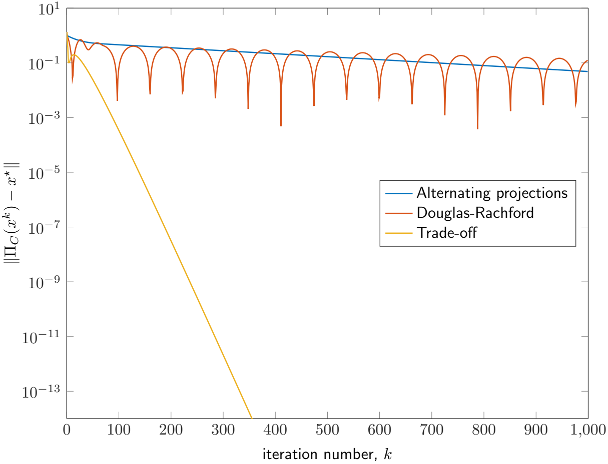
# 3D example – Trade-off



# Distance to intersection

Distance for *shadow sequence* to intersection, $x^\star$:

$$\|\Pi_C(x^k) - x^\star\|$$



## Optimal trade-off?

- Although algorithm from 1950's, optimal parameters not known
- Not even for subspace intersection problems

# Our contribution

Optimally parameter selection for subspace intersection problem:

$$\text{find } x \in \mathcal{U} \cap \mathcal{V}$$

where

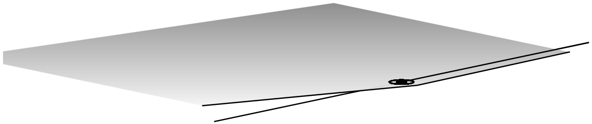
$$\mathcal{U} := \{x \in \mathbb{R}^n : Ax = 0\}, \qquad \mathcal{V} := \{x \in \mathbb{R}^n : Bx = 0\}$$

# Why interesting?

- Assume general convex intersection problem

$$\text{find } x \in C \cap D$$

  where
  - Intersection between $C$ and $D$ is "sufficiently regular"
  - The sets are "sufficiently smooth"
- Then algorithm exhibits a finite identification property:
  - Active manifolds for attracting intersection point identified in finite number of iterations
  - Locally, behavior of iterates become (or approach) an affine subspace intersection iteration



# Convergence rate

- Alternating relaxed projections for subspace intersection problem:

$$x^{k+1} = (1-\alpha)x^k + \alpha \Pi_{\mathcal{U}}^{\alpha_2} \Pi_{\mathcal{V}}^{\alpha_1} x^k$$

- Algorithm is matrix iteration with (parameter dependent) matrix

$$M(\alpha, \alpha_1, \alpha_2) := (1-\alpha)I + \alpha((1-\alpha_2)I + \alpha_2 \Pi_{\mathcal{U}})((1-\alpha_1)I + \alpha_1 \Pi_{\mathcal{V}})$$

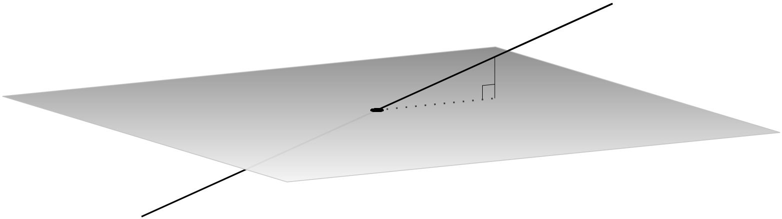
- Sharp asymptotic rate is magnitude of second largest eigenvalues,

$$|\lambda_2(M(\alpha, \alpha_1, \alpha_2))|$$

(not counting multiplicity of eigenvalue at 1)

# Friedrichs angle

- Eigenvalues depend on principal angles between $\mathcal{U}$ and $\mathcal{V}$
- The smallest nonzero principal angle is called *Friedrichs angle*, $\theta_F$



# Known results

- Alternating projections ($\alpha = \alpha_1 = \alpha_2 = 1$)[1]:

$$|\lambda_2(M(1,1,1))| = \cos^2 \theta_F$$

- Douglas-Rachford ($\alpha = \frac{1}{2}$, $\alpha_1 = \alpha_2 = 2$)[2]:

$$|\lambda_2(M(0.5,2,2))| = \cos \theta_F$$

- One parameter optimized while two fixed[3]

[1] F. Deutsch, 1984.
[2] H. Bauschke et al., 2014.
[3] H. Bauschke et al., 2016.

# Our contribution

- Let $p = \dim \mathcal{U}$ and $q = \dim \mathcal{V}$ with $\mathcal{U}$ and $\mathcal{V}$ linear subspaces
- Assume: Dimensions for linear subspaces unknown
- Find $\alpha, \alpha_1, \alpha_2 > 0$ that solve

$$\begin{array}{lll} \text{minimize} & \gamma & \\ \text{subject to} & |\lambda_2(M(\alpha, \alpha_1, \alpha_2))| \leq \gamma & \text{for } q < p \\ & |\lambda_2(M(\alpha, \alpha_1, \alpha_2))| \leq \gamma & \text{for } q = p \\ & |\lambda_2(M(\alpha, \alpha_1, \alpha_2))| \leq \gamma & \text{for } q > p \end{array}$$

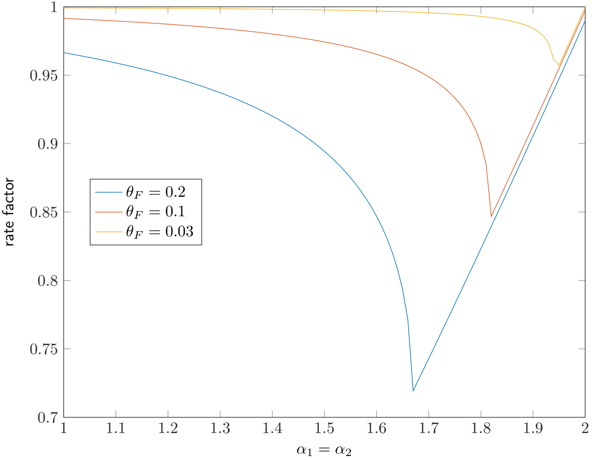
- Optimal parameters:

$$\alpha_1^* = \alpha_2^* = \frac{2}{1 + \sin \theta_F}, \qquad \alpha^* = 1$$

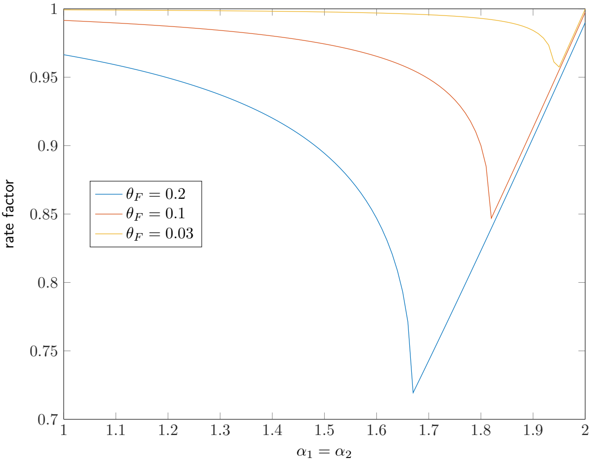
- Optimal rate:

$$\gamma^* = \frac{1 - \sin \theta_F}{1 + \sin \theta_F} = \alpha_1^* - 1$$

# Rate comparison



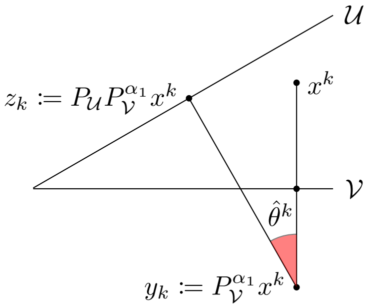
# Rate comparison




Optimal parameters depend on Friedrichs angle, which is not known

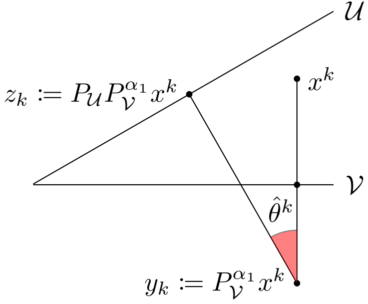
# Adaptive method

- Online method to estimate $\theta_F$:



# Adaptive method

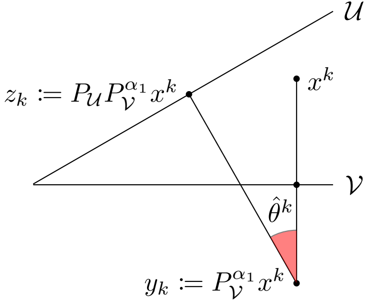
- Online method to estimate $\theta_F$:




- Conservative: $\hat{\theta}^k \geq \theta_F$ if $x^k \in \mathcal{U} + \mathcal{V}$

# Adaptive method

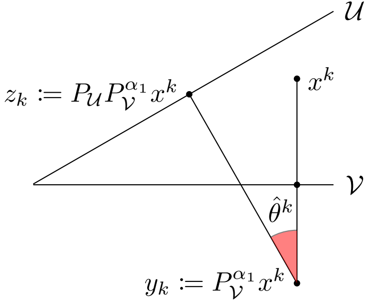
- Online method to estimate $\theta_F$:




- Conservative: $\hat{\theta}^k \geq \theta_F$ if $x^k \in \mathcal{U} + \mathcal{V}$
- Adaptive method: Choose $\alpha_1^k = \alpha_2^k = \frac{2}{1+\sin\hat{\theta}^k}$ and $\alpha = 1$

## Adaptive method

- Online method to estimate $\theta_F$:

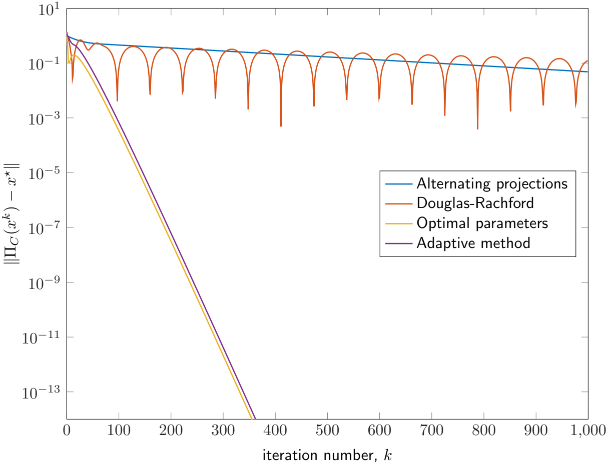



- Conservative: $\hat{\theta}^k \geq \theta_F$ if $x^k \in \mathcal{U} + \mathcal{V}$
- Adaptive method: Choose $\alpha_1^k = \alpha_2^k = \frac{2}{1+\sin \hat{\theta}^k}$ and $\alpha = 1$
- Easy to prove convergence to intersection

# 3D example – convergence

Distance for *shadow sequence* to intersection, $x^\star$:

$$\|\Pi_C(x^k) - x^\star\|$$

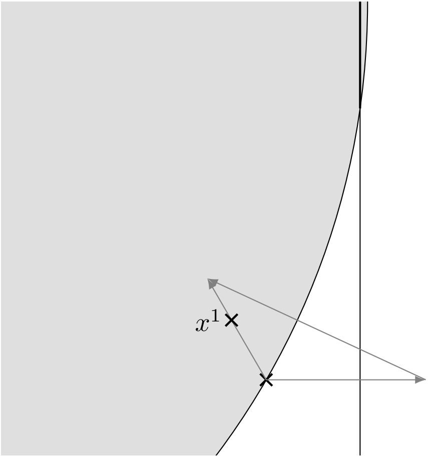


# Problem

- Performance of all methods depends on the Friedrichs angle
- Poor performance when Friedrichs angle very small
- Example with Friedrichs angle $\theta_F = 0.0001$
  - Optimal rate factor $\gamma = 0.9998$
  - 20000 iterations: $\gamma^{20000} = 0.0183$

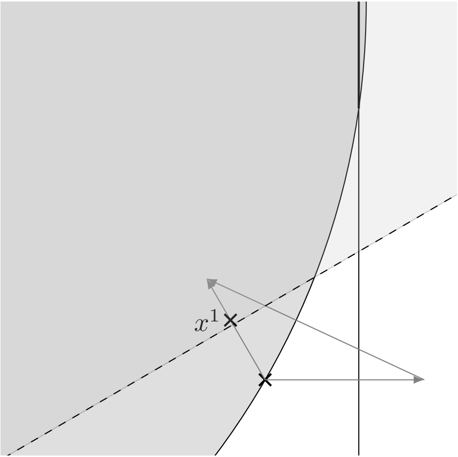
# Long-step method

- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)



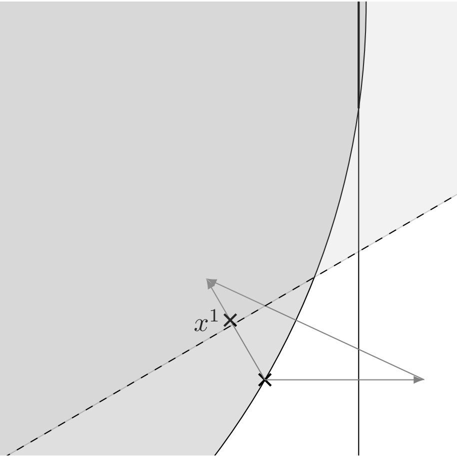
# Long-step method

- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)



# Long-step method

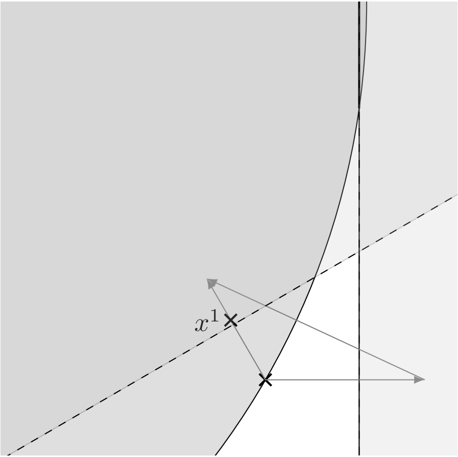
- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)




- Can we construct “better” set that contains fixed-point set?

# Long-step method

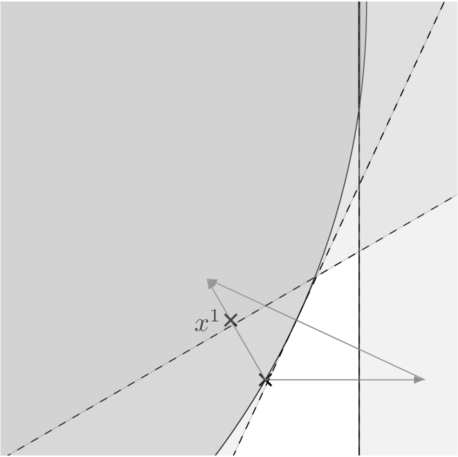
- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)




- Can we construct "better" set that contains fixed-point set?

# Long-step method

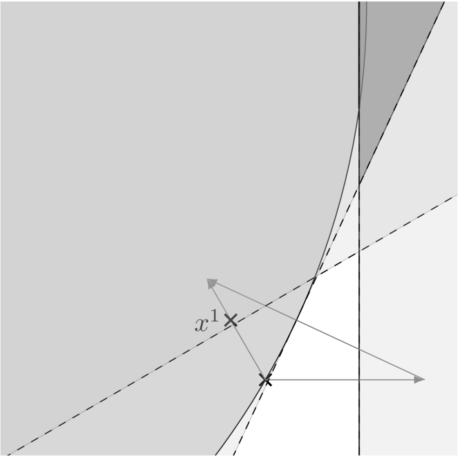
- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)




- Can we construct "better" set that contains fixed-point set?

# Long-step method

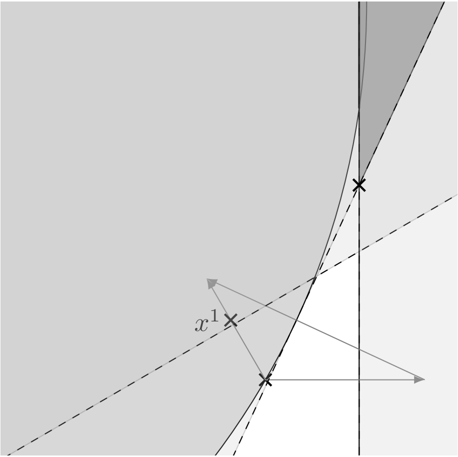
- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)




- Can we construct "better" set that contains fixed-point set?

# Long-step method

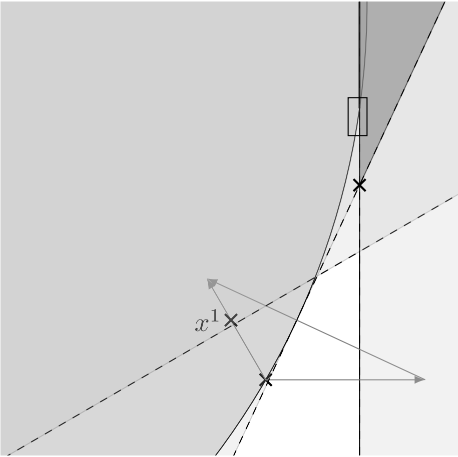
- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)




- Can we construct "better" set that contains fixed-point set?
- Long-step method: (Relaxed) projection onto intersection

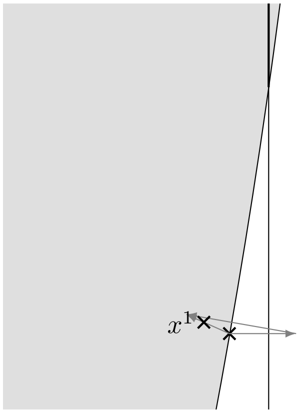
# Long-step method

- It creates a separating hyperplane and performs relaxed projection
- The constructed halfspace contains fixed-point set (intersection)

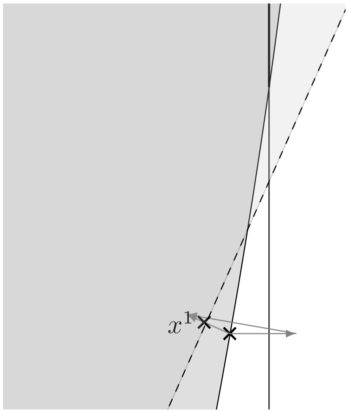



- Can we construct "better" set that contains fixed-point set?
- Long-step method: (Relaxed) projection onto intersection

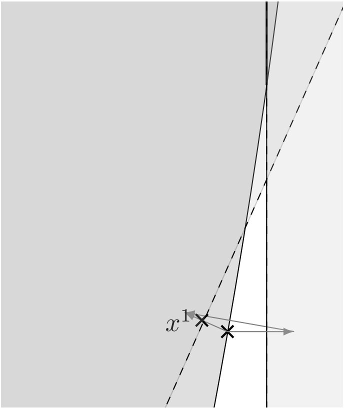
# Closer to intersection



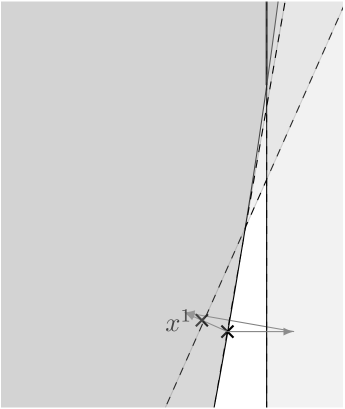
# Closer to intersection



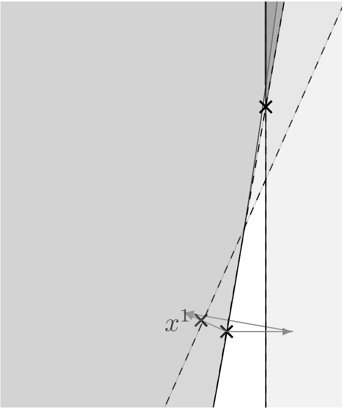
# Closer to intersection



# Closer to intersection



# Closer to intersection

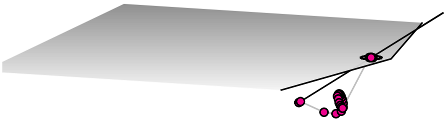



- Smaller angle between projection vectors $\Rightarrow$ longer step

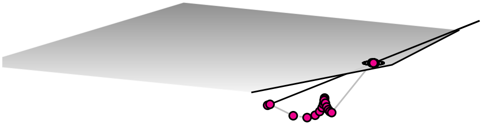
# 3D example

# 3D example

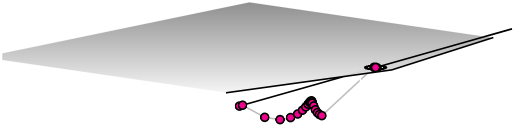
# 3D example



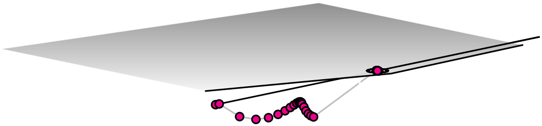
# 3D example



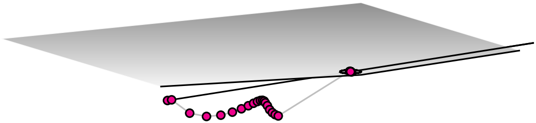
# 3D example



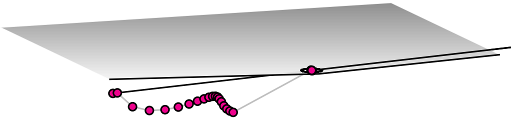
# 3D example



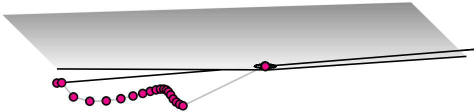
# 3D example



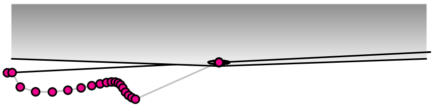
# 3D example



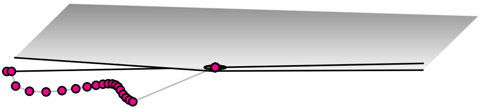
# 3D example



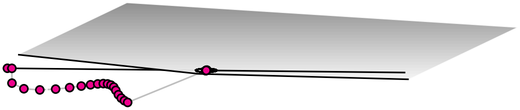
# 3D example



# 3D example



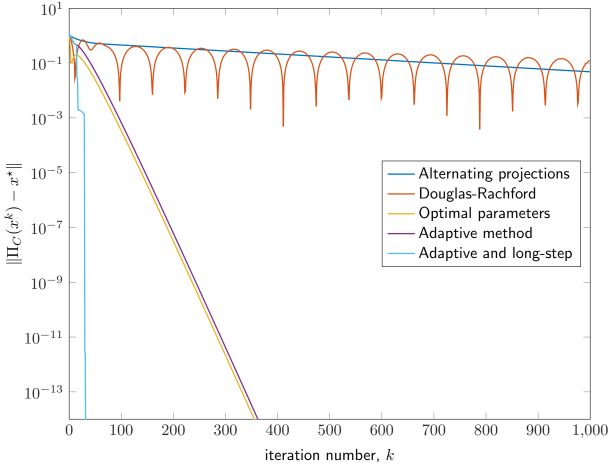
# 3D example



# 3D example – convergence

Distance for *shadow sequence* to intersection, $x^\star$:

$$\|\Pi_C(x^k) - x^\star\|$$



# Algorithm variations

- Perform long-step in every iteration
- Run adaptive method and interleave with occasional long-steps
- Use history of halfspaces $\Rightarrow$ smaller intersection and longer steps
- Parallel versions: construct halfspaces from parallel projections[1]

[1] K. Kiwiel et al., 1995.

# Convergence

Convergence to a fixed-point can be proven using the following steps:

- Method can be written as

$$x^{k+1} = S_k x^k$$

  where $S_k$ is (iteration dependent) quasi-averaged operator
- Intersection of fixed-point sets of all operators $S_k$ is $C \cap D$
- Steps longer or as long as in nominal method

# Numerical evaluation

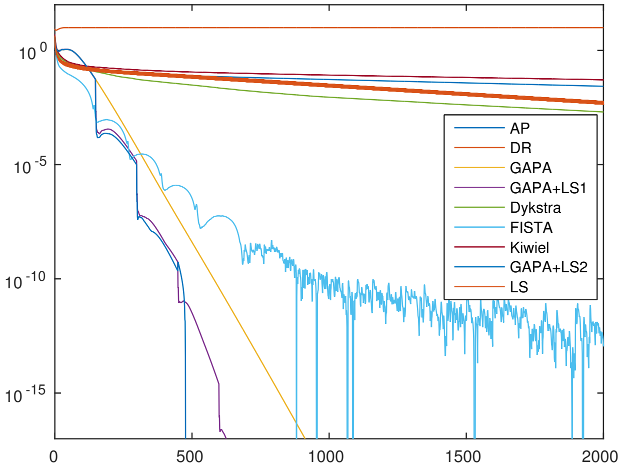
- Problem:

$$\begin{array}{ll} \text{find} & x \\ \text{such that} & A(x-b)=0 \\ & x \geq 0 \end{array}$$

- $A^{150\times 300}$ has randomly generated entries, $b = 10^{-8}\mathbf{1}$
- Constructed to have small feasible set

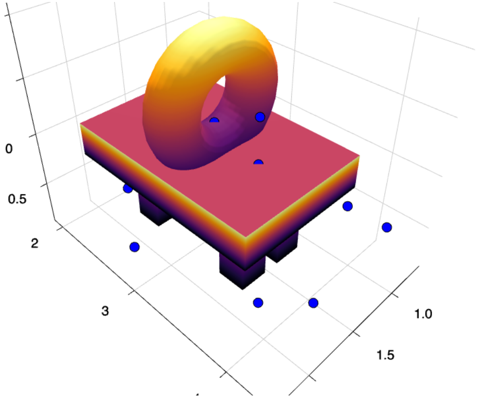
# Numerical evaluation

Plot: $\mathrm{dist}_C(\Pi_D x^k)$ vs iteration $k$



# Trajectory generation

- Trajectory generation problem for quadrocopters:




- Visit points in space while avoiding obstacles
- Can “solve” this using our feasibility methods and *Superiorization*

# Superiorization

- Assume that $T$ is averaged with nonempty fixed-point set
- Basic (Krasnoselskii-Mann) method to find fixed-point:

$$x^{k+1} = Tx^k$$

- Any orbit $(x^k)_{k \geq 0}$ converges to fixed-point of $T$ if[1]

$$\sum_{k=0}^{\infty} \|x^{k+1} - Tx^k\| < \infty$$

- Superiorization[2]:

$$x^{k+1} = T(x^k - \beta_k \nabla f(x^k))$$

with $\beta_k$ summable and $\nabla f$ bounded

[1] D. Butnariu, S. Reich, and A.J. Zaslavski, 2006.
[2] D. Butnariu, R. Davidi, G. T. Herman, and I. Kazantsev, 2007.

# Formulation

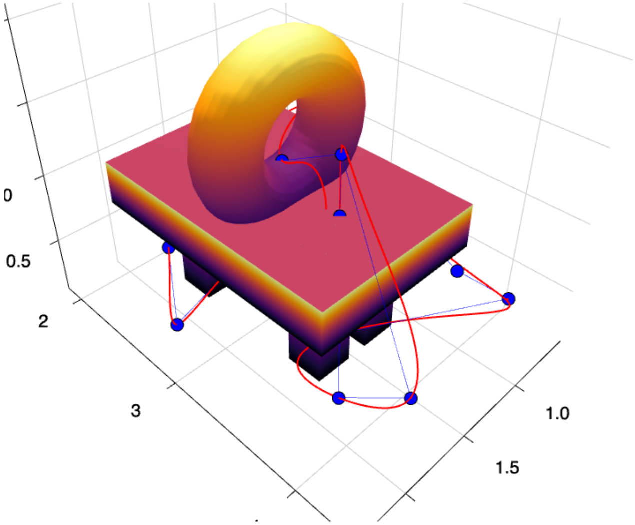
Convex constraints solved using feasibility methods:

- Quadrocopter dynamic constraints
- Quadrocopter state and input constraints
- Room box constraints

Nonconvex constraints, violation modeled with nonconvex cost:

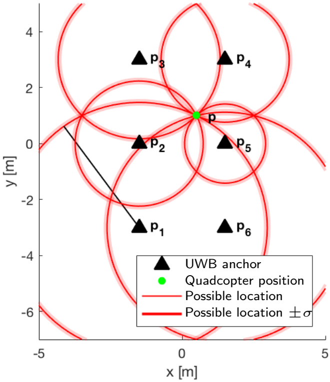
- Obstacle avoidance
- Minimize shortest distance from trajectory to each point

# Generated trajectory



# Experimental setup

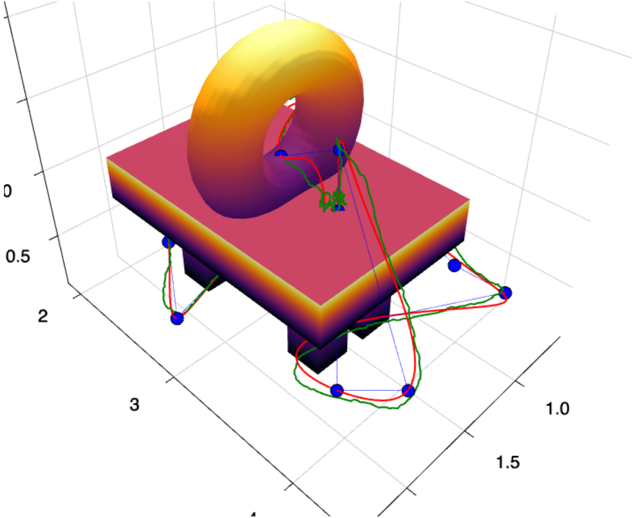
- Positioning system with ultra-wideband radio communication
- Time stamp sent in communication from quadrocopter to nodes
- Positioning decided from time between send and receive



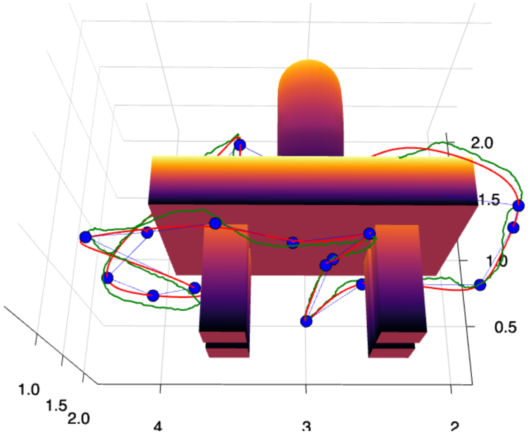

- 20 to 30 times cheaper than, e.g., a VICON system

# Video

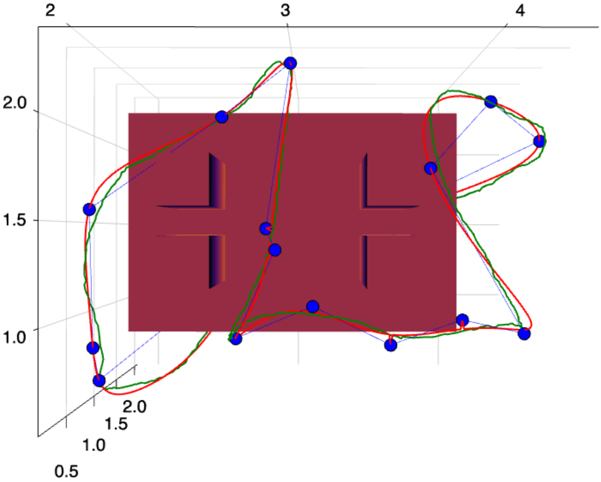
# Real trajectories



# Real trajectories



# Real trajectories



# Conclusions

- Optimal parameters for alternating relaxed projections
- Long step feasibility method
- Trajectory generation for quadrocopters

# Ongoing work

- Compare first-order methods for large-scale conic programming
- Julia packages:
  - Solver suite for first order method (FirstOrderSolvers.jl)
  - Test bed for evaluating methods

# Thank you

And thanks to Marcus Greiff for quadcopter flying

# References

M. Fält, P. Giselsson. "Optimal Convergence Rates for Generalized Alternating Projections", 2017. arXiv:1703.10547.

S. Agmon, "The relaxation method for linear inequalities," *Canadian Journal of Mathematics*, vol. 6, no. 3, pp. 382–392, 1954.

T. S. Motzkin and I. Shoenberg, "The relaxation method for linear inequalities," *Canadian Journal of Mathematics*, vol. 6, no. 3, pp. 383–404, 1954.

L. M. Bregman, "Finding the common point of convex sets by the method of successive projection," *Dokl Akad. Nauk SSSR*, vol. 162, no. 3, pp. 487–490, 1965.

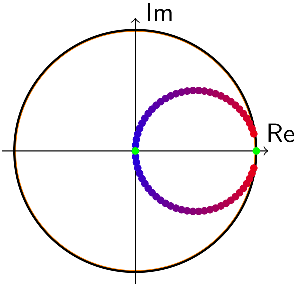
H. H. Bauschke, J. Y. B. Cruz, T. T. A. Nghia, H. M. Pha, and X. Wang, "The rate of linear convergence of the Douglas-Rachford algorithm for subspaces is the cosine of the Friedrichs angle," *Journal of Approximation Theory*, vol. 185, no. 0, pp. 63–79, 2014.

H. H. Bauschke, J. Y. B. Cruz, T. T. A. Nghia, H. M. Pha, and X. Wang, "Optimal rates of linear convergence of relaxed alternating projections and generalized Douglas-Rachford methods for two subspaces," *Numerical Algorithms*, vol. 73, no. 1, pp. 33–76, 2016.

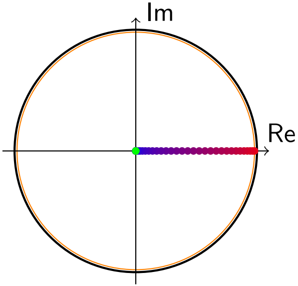
K. C. Kiwiel and B. Łopuch, "On long-step surrogate projection methods for solving convex feasibility problems," In *Proceedings of the Seventeenth IFIP TC7 Conference on System Modelling and Optimization*, pp. 466–472, 1995.

D. Butnariu, S. Reich and A.J. Zaslavski, "Convergence to fixed points of inexact orbits of Bregman-monotone and of nonexpansive operators in Banach spaces," *Fixed Point Theory and its Applications*, (Conference Proceedings, Guanajuato, Mexico, 2005), Yokahama Publishers, Yokahama, Japan, pp. 11-32, 2006.

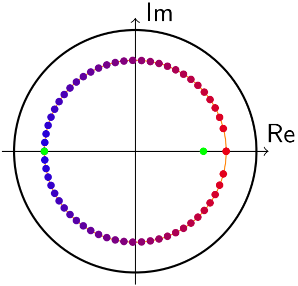
D. Butnariu, R. Davidi, G. T. Herman, and I. Kazantsev, "Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems". *IEEE J. Select. Topics Signal Proc.* vol. 1, pp. 540–547, 2007.



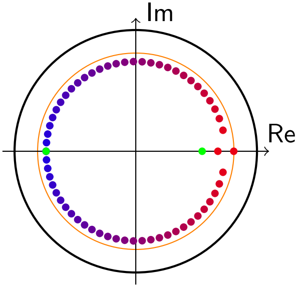

Douglas-Rachford




Alternating projections




Optimal




$\alpha_1 = \alpha_2 = \alpha_1^* - 0.01, \alpha = 1$