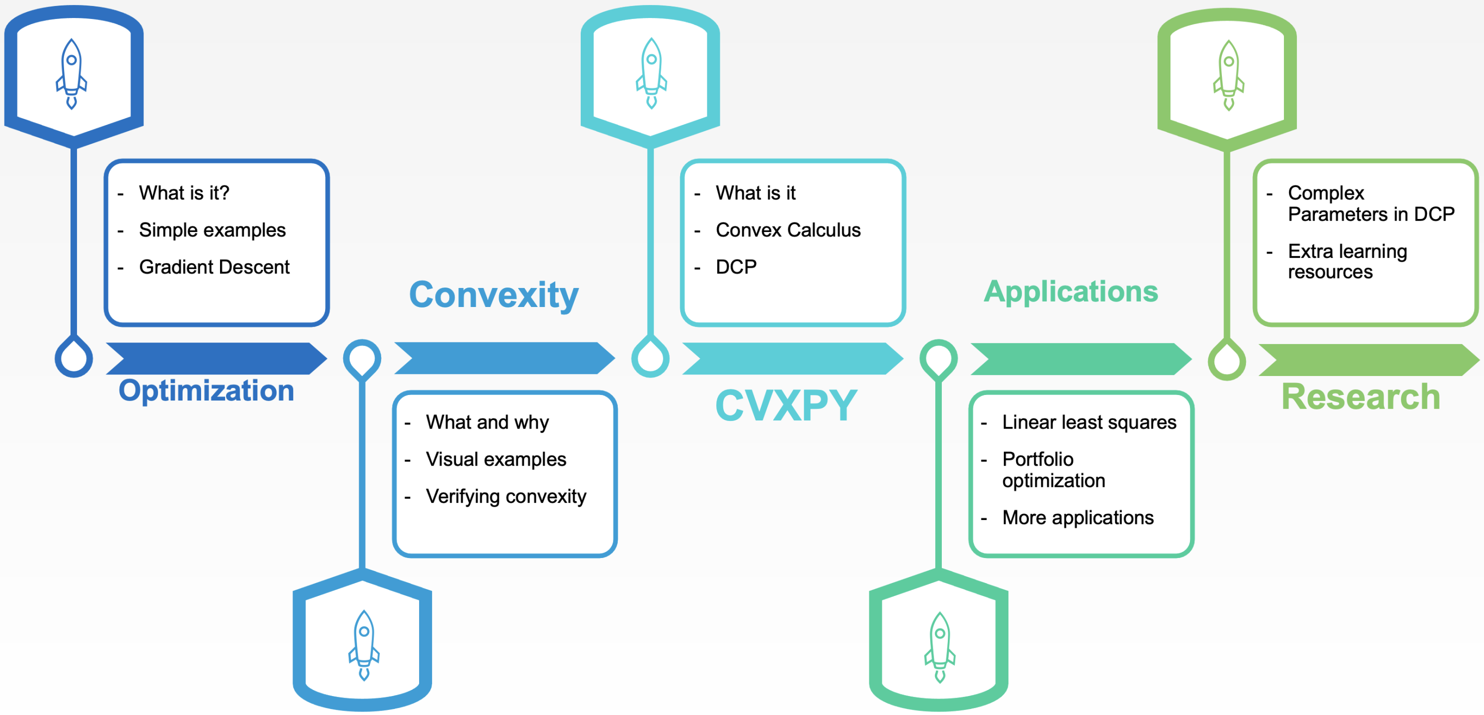
CONVEX PROGRAMMING WITH CVXPY

By William Zijie Zhang

Presentation Outline



Mathematical Optimization

Mathematical Optimization

> *"Nothing at all takes place in the universe in which some rule of maximum or minimum does not appear."*

—Leonhard Euler

Mathematical Optimization

General Form of a problem

minimise $f_0(x)$

subject to $f_i(x) \leq 0$, $i = 1, \ldots, m$

and $g_j(x) = 0$, $i = 1, \ldots, n$

Example of optimization problems

Maximize profits

Find the best price x for selling n items

If you set the price at 1.50, you will be able to sell 5000 items

and for every 10 cents you lower the price you will be able to sell another 1000 items

Minimize production costs

Determine the number of units q the manufacturer should produce to minimize cost

$P_c(q) = 0.0001q^2 - 0.08q + 65 + \frac{5000}{q}$, where $q > 0$

Gradient Descent

Gradient Descent

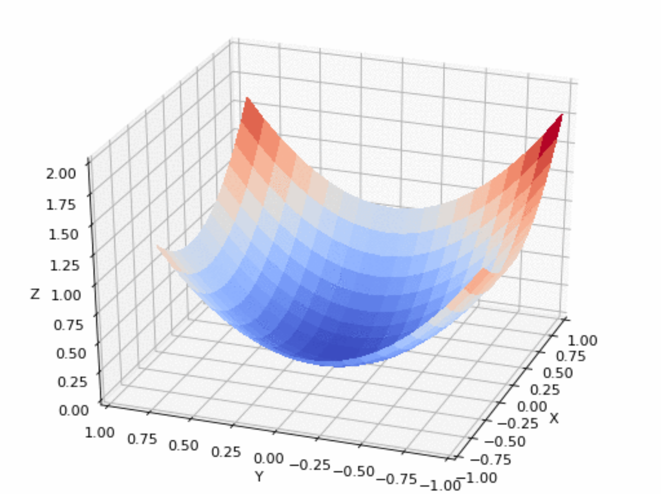
Iterative Definition

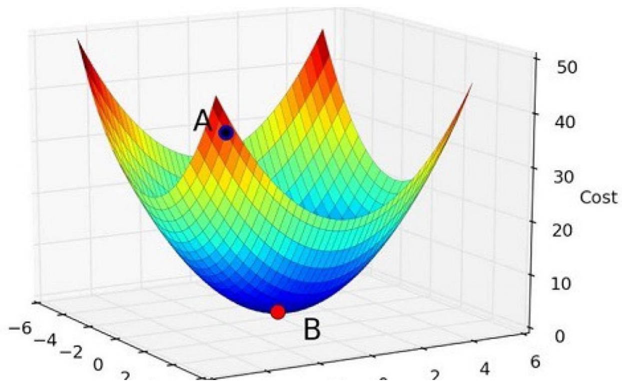
$x_{k+1} = x_k + a_k p_k$, where $k \geq 0$ and $f(x_{k+1}) \leq f(x_k)$

a_k is called the step length

and p_k is called the step direction where $p_k = -\nabla f(x_k)$

We hope that this sequence will converge to a global minimiser of $f(x)$



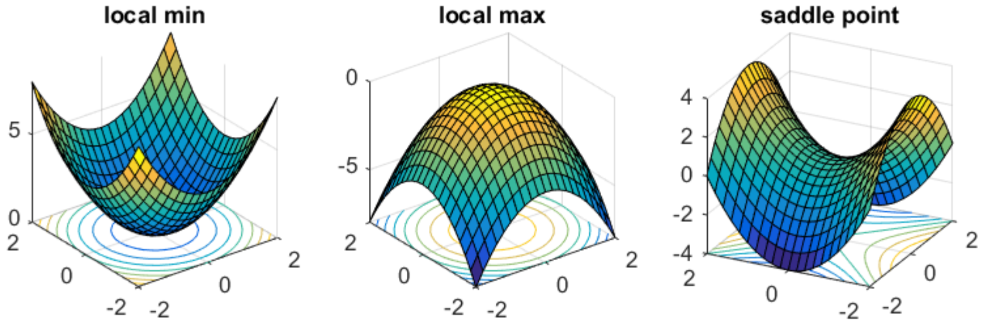


General Concerns

Most problems are impossible to solve analytically

Numerical solutions to problems are computationally heavy (NP-hard)

Some algorithms converge to saddle points (Newton's method)



Convex Functions

Convex Functions

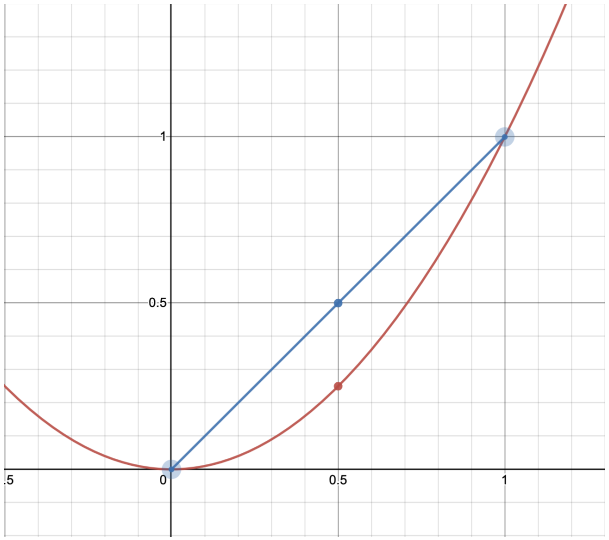
Definition

A function f is convex, if for any x, y, and $\lambda \in [0, 1]$,

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Importance

A local minimum of f is also guaranteed to be a global minimum of f



Example of convex functions

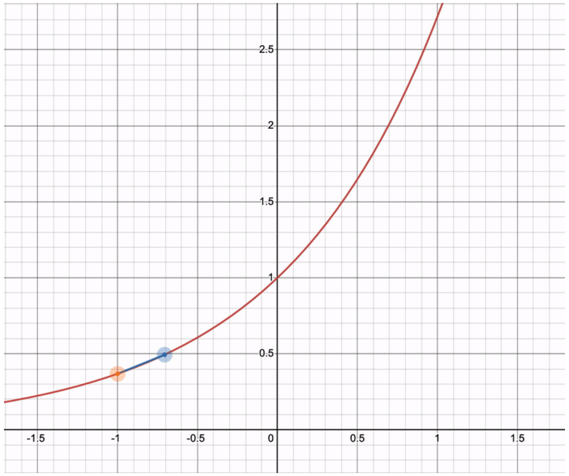
- e^x

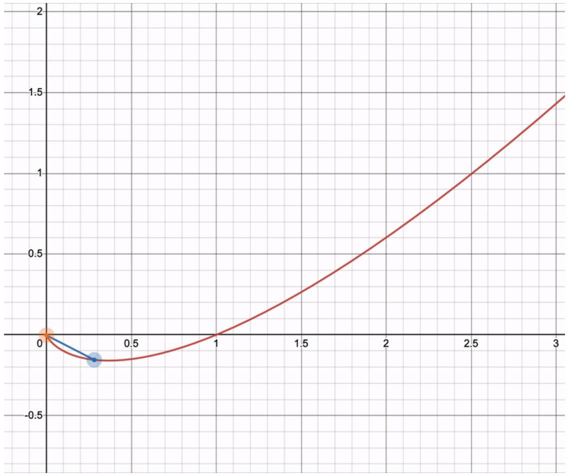
- $xlog(x)$

- $max(x_1, ..., x_n)$

- $a^T x + b$

- $||x||$





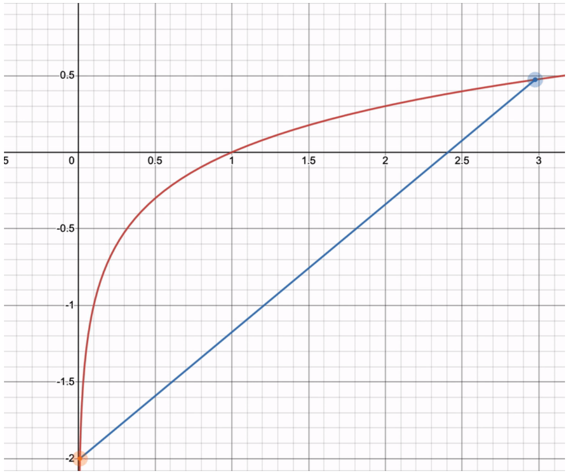
Example of concave functions

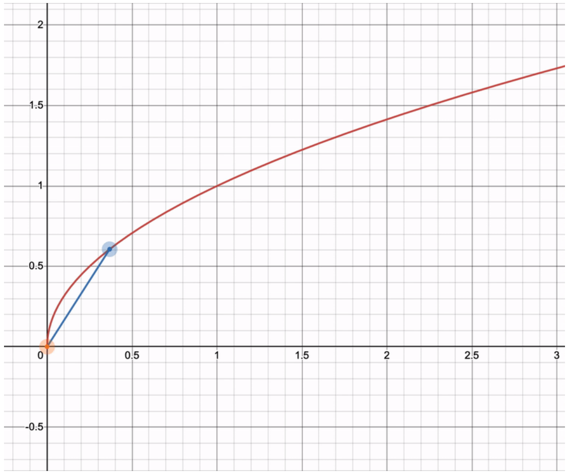
- $\sqrt{xy}$

- $log(x)$

- $min(x_1, ..., x_n)$

- $x^p, where\, 0 < p < 1$





Verifying Convexity

Verifying Convexity

First method: by definition

Prove that $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$

for any x, y and $\lambda \in [0,1]$

Second method: second order condition

Verify that the hessian of $f(x)$ is positive semi-definite

In mathematical notation: $\nabla^2 f(x) \succcurlyeq 0$

Generalisation of the second derivative test for functions

CVXPY and how it works

CVXPY and how it works

CVXPY

What is CVXPY?

Open Source Python modeling language for convex optimization problems.

- Abstracts away the complexity of implementation for solvers
- Disciplined Convex Programming
- Seamless interaction with other Python libraries
- Over tens of thousands of users

Convex Calculus

Convex Calculus

Transformations preserving convexity

- non-negative scaling

  if $f(x)$ is convex and $a \geq 0$, then $a \cdot f(x)$ is also convex

- summation

  if $f(x)$ and $g(x)$ are convex, then $f(x) + g(x)$ is also convex

- composition

  if f is convex and h is convex and increasing, then $h(f(x))$ is also convex

Disciplined Convex Programming (DCP)

Disciplined Convex Programming (DCP)

Analysis of convexity:

Build a parse tree starting from an **Expression**

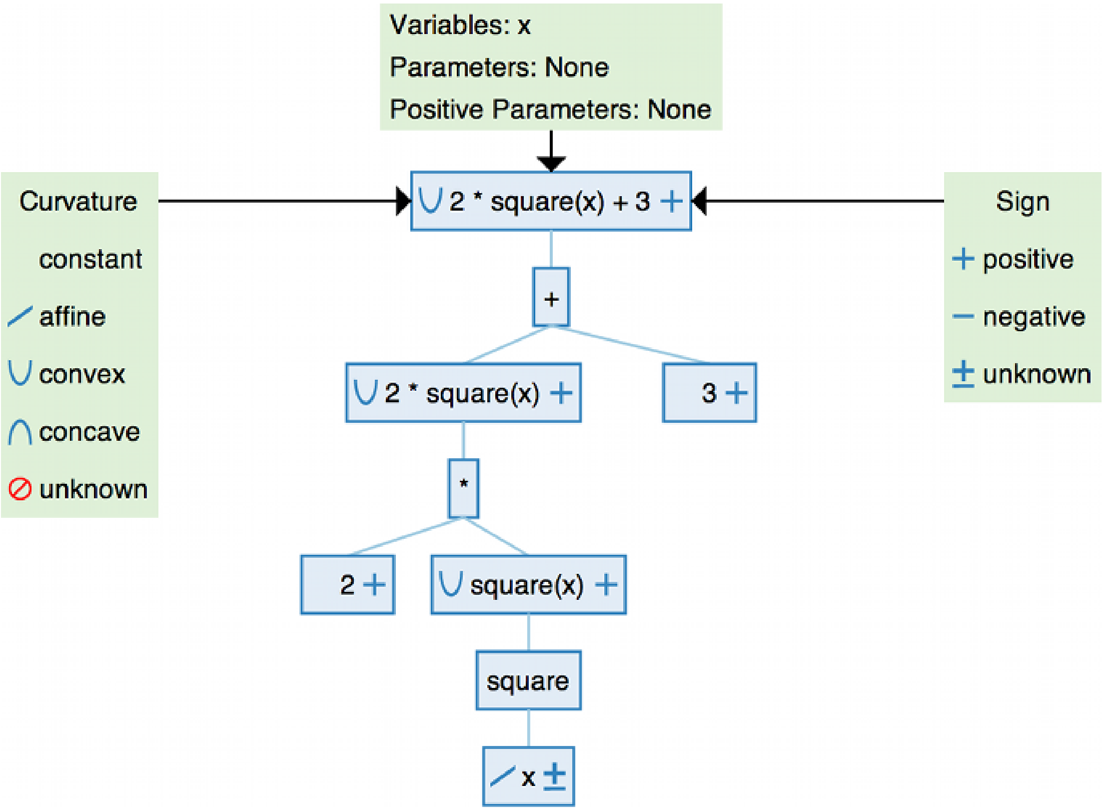
Branch outward recursively to **Atoms** and **Leaves**

Store information about curvature, domain and monotonicity

Example:

Function	Meaning	Domain	Sign	Curvature	Monotonicity
abs(x)	$\lvert x\rvert$	$x \in \mathbf{R}$	$+$ positive	$\cup$ convex	$\nearrow$ incr. for $x \geq 0$ $\searrow$ decr. for $x \leq 0$
entr(x)	$\begin{cases} -x\log(x) & x > 0 \\ 0 & x = 0 \end{cases}$	$x \geq 0$	$\pm$ unknown	$\cap$ concave	None
exp(x)	e^x	$x \in \mathbf{R}$	$+$ positive	$\cup$ convex	$\nearrow$ incr.
geo_mean(x1,...,xk)	$(x_1 \cdots x_k)^{1/k}$	$x_i \geq 0$	$+$ positive	$\cap$ concave	$\nearrow$ incr.
huber(x)	$\begin{cases} 2\lvert x\rvert - 1 & \lvert x\rvert \geq 1 \\ \lvert x\rvert^2 & \lvert x\rvert < 1 \end{cases}$	$x \in \mathbf{R}$	$+$ positive	$\cup$ convex	$\nearrow$ incr. for $x \geq 0$ $\searrow$ decr. for $x \leq 0$

Example of DCP parsing



Limitations of DCP

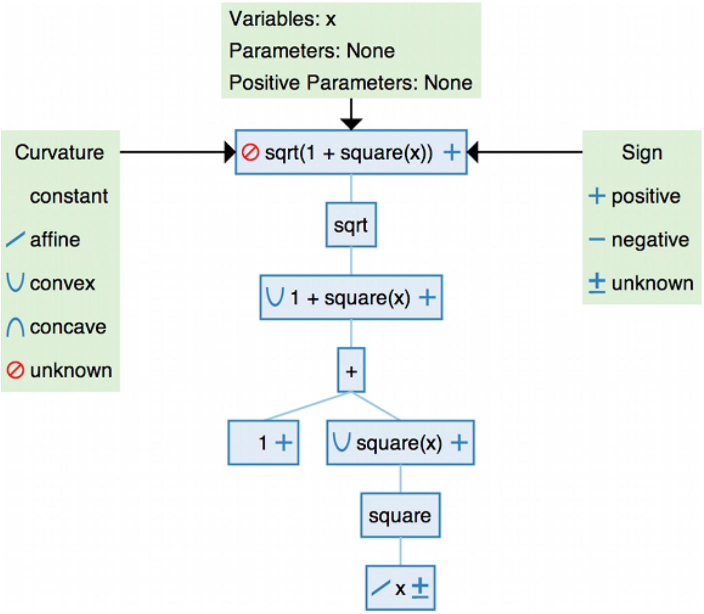
Example: $\sqrt{x^2+1}$

Typical Parsing:

cp.sqrt(cp.square(x) + 1)

Nodes: 1, x^2 and $1+x^2$ are all convex

Node: $\sqrt{.}$ is concave



Limitations of DCP

Example: $\sqrt{x^2 + 1}$

What it should be:

cp.norm(cp.stack [x , 1] , 2)

Node: [x , 1] is convex

Node: $|| * ||_2$ is convex

DCP is **unable** to analyze the curvature of all functions!

Applications of convex programming

Linear Least Squares

Linear Least Squares

Problem Statement :

Minimise $||Ax - b||_2^2$, (the sum of squared differences)

Find the corresponding optimal x^*, where $r = Ax^* - b$ is known as the residual

Linear Least Squares

Problem Statement :

Minimise $||Ax - b||_2^2$, (the sum of squared differences)

Find the corresponding optimal x^*, where $r = Ax^* - b$ is known as the residual

In [36]:
```
import cvxpy as cp
import numpy as np
#generate random data
m = 30; n = 20
np.random.seed(1)
A = np.random.randn(m, n)
b = np.random.randn(m)
```

Linear Least Squares

Problem Statement :

Minimise $||Ax - b||_2^2$, (the sum of squared differences)

Find the corresponding optimal x^*, where $r = Ax^* - b$ is known as the residual

In [36]:
```
import cvxpy as cp
import numpy as np
#generate random data
m = 30; n = 20
np.random.seed(1)
A = np.random.randn(m, n)
b = np.random.randn(m)
```

In [37]:
```
#solve the problem using CVXPY.
x = cp.Variable(n)
objective = cp.Minimize(cp.sum_squares(A @ x - b))
constraints = [0 <= x, x <= 1]
prob = cp.Problem(objective, constraints)
result = prob.solve()
print(result)
```

```
19.83126370644502
```

Portfolio Optimization

Portfolio Optimization

Problem statement :

Maximize $\mu^T w - \gamma w^T \sum w$, the risk-adjusted expected return

where $1^T w = 1$, and $w \in W$

Definitions :

Long only portfolio: $w_i > 0$ for all i

Short position: $w_i < 0$

Possible objectives: high return or low risk

Adjusting γ can give the optimal risk-return trade-off

Portfolio Optimization

Portfolio Optimization

In [2]:

```
import numpy as np
import scipy.sparse as sp
#generate random data
np.random.seed(1); n = 10
mu = np.abs(np.random.randn(n, 1))
Sigma = np.random.randn(n, n)
Sigma = Sigma.T.dot(Sigma)
```

Portfolio Optimization

In [2]:

```
import numpy as np
import scipy.sparse as sp
#generate random data
np.random.seed(1); n = 10
mu = np.abs(np.random.randn(n, 1))
Sigma = np.random.randn(n, n)
Sigma = Sigma.T.dot(Sigma)
```

In [3]:

```
import cvxpy as cp
#solve the problem using CVXPY
w = cp.Variable(n)
gamma = cp.Parameter(nonneg=True)
ret = mu.T @ w
risk = cp.quad_form(w, Sigma)
objective = cp.Maximize(ret - gamma * risk)
constraints = [cp.sum(w) == 1, w >= 0]
prob = cp.Problem(objective, constraints)
gamma.value = 0.2
prob.solve()
```

Out[3]:

```
1.500097620280957З
```

Other applications

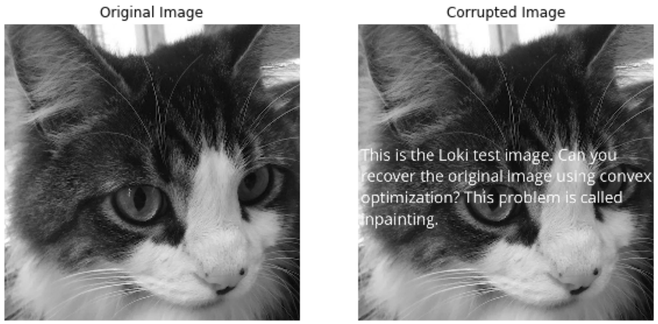
Computer Vision and Image processing:

- Image restoration (inpainting, deblurring)
- Regularization techniques for image processing

Other applications

Computer Vision and Image processing:

- Image restoration (inpainting, deblurring)
- Regularization techniques for image processing

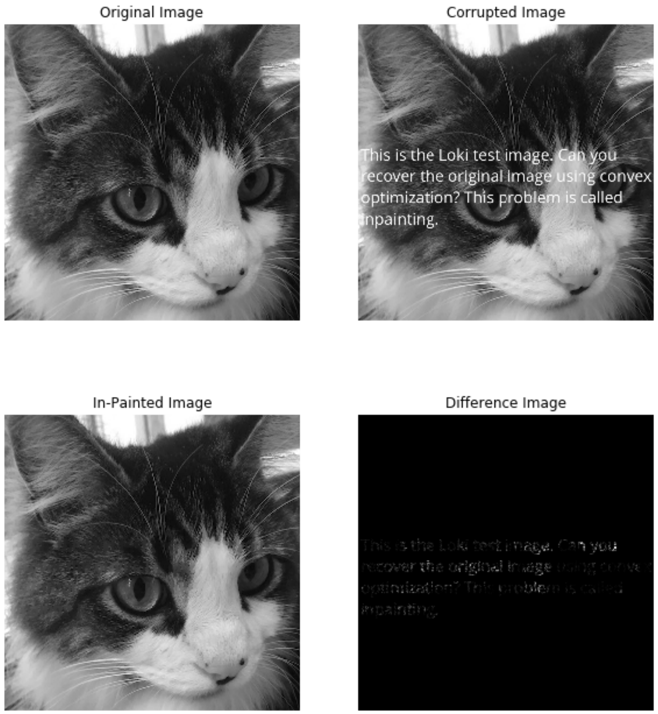


Other applications

Computer Vision and Image processing:

Image restoration (inpainting, deblurring)

Regularization techniques for image processing



Other applications

Landing Rockets (Control Theory):

Real-time path and trajectory generation

Implementations of physical structures

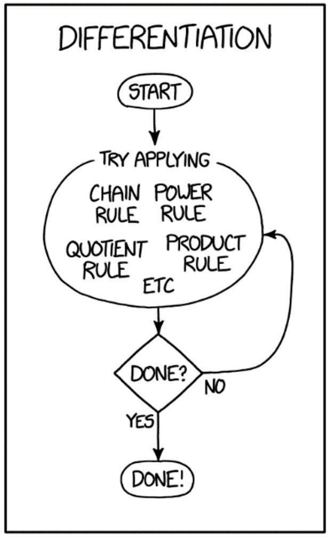
Further Research

Further Research

Additional Support for Complex Parameters in CVXPY

What needs to be done:

- Caching compilations for convex problems with complex parameters
- Automatic differentiation for complex derivatives



References

- Convexity and Duality
- DCP quiz
- CVXPY documentation
- SFU Optimization Problems
- CVXPY Portfolio Optimization example