Robotics and the aerodynamics of spaces (Research-Project)

- Configurations in The Multidimensional Plane

- Cosmic object parameter mechanics

- Object speeds

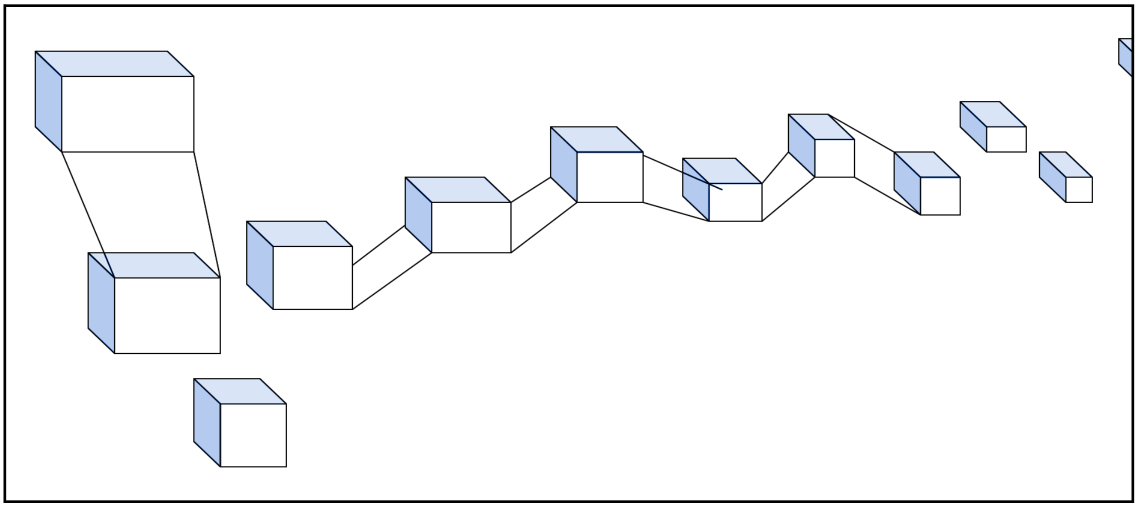
Project-text: In this text I will expose the mechanics and programming behind the multidimensional equations for spatial objects such as robots, airplanes, satellites and some other higher dimensions approximations to robotics and aerodynamics. Cosmic objects and robotics kinematics are present in this useful text for academic research and further investigations based on a solid basis of robot motion, airplanes design, cosmic dimensions and object parametrization of variables with python.

aim: The objective is to append a reference guide to work with multidimensions, spaces, higher dimensions and spatial relations at a computing level.

Software-code: The software used in this text is python for aerospace system designs and Scipy for matrix modelization. In the present text the lecturer can find a new approach to computing, robotics and multidimensional spaces for aeronautics digitalization and the fundamentals of applied kinematics, dynamics and higher dimensions research.I recommend the modern robotics book and some programming knowledge on Scipy and Scikit to follow this text.

Algorithms-description;

The higher order dimensions algorithm based on cube replication and extrapolation of different sets is some of the main idea is to compute in Scikit-python the high dimensions to recreate new mathematical and computational scenarios with python.In this example, I show how algorithmic logarithmic regression works to elucidate some important machine learning concepts for Scikit and to propose another level of computational analysis.



We have some functions with logarithms and vectors as exponents. The logarithmic regression works across different higher dimensions that change through a constant t or time and the dimensions replicate each other in a non linear time.The constant time is not uniform, is chaotic and it depends on regressive algorithms to commute with each other in different cube states or dimensional states.

Composition of higher dimensions:

Algebras from vector and hyperplane arrangements are several natural algebraic spaces related to the Tutte polynomial arising in commutative algebra, hyperplane arrangements, box splines, and index theory; we discuss a few. For each hyperplane H in a hyperplane arrangement A in k d let lH be a linear function such that H is given by the equation lH(x) = 0.

```
f(x)=1+logx6,logx6(2x+4)-log6P

Prop: f(x) 1+logx6(Dimension1)=logx6*(Dimension2)

VECTORS (variable(exp3)<===     <===variable(exp3))

1+logx6(Dimension3) =  logx6 (2 dimension2) + logx6(dimension1)
```

A matroid subdivision is a polyhedral subdivision P of a matroid polytope PM where every polytope P ∈ P is itself a matroid polytope. Equivalently, it is a subdivision of PM whose only edges are the edges of PM. In the most important case, M is the uniform matroid Ud,and PM is the hypersimplex Δ(d, n).

Matroid subdivisions arose in algebraic geometry [HKT06, Kap93, Laf03], in the theory of evaluated matroids [DW92, Mur96], and in tropical geometry [Spe08]. For instance, Lafforgue showed that if a matroid polytope PM has no nontrivial matroid subdivisions, then the matroid M has (up to trivial transformations) only finitely many realizations over a fixed field F. This is one of very few results about realizability of matroids over arbitrary fields.

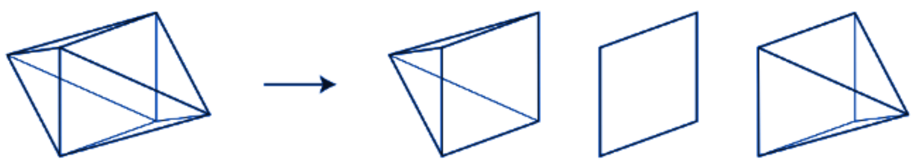
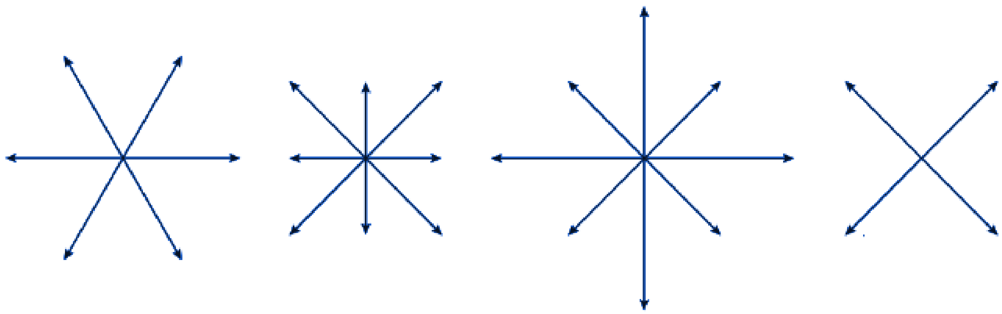


Figure 7.8: A matroid subdivision of $\Delta(2, 4)$.

This formula is straightforward in terms of coboundary polynomials: χM(k) (X, Y) = χM(X, Y K). For an extensive generalization, see Section 7.9.

• [Ard07a, Mph00] Root systems are arguably the most important vector configurations; these highly symmetric arrangements play a fundamental role in many branches of mathematics. For the general definition and properties, see for example [Hum90]; we focus on the four infinite families of classical root systems

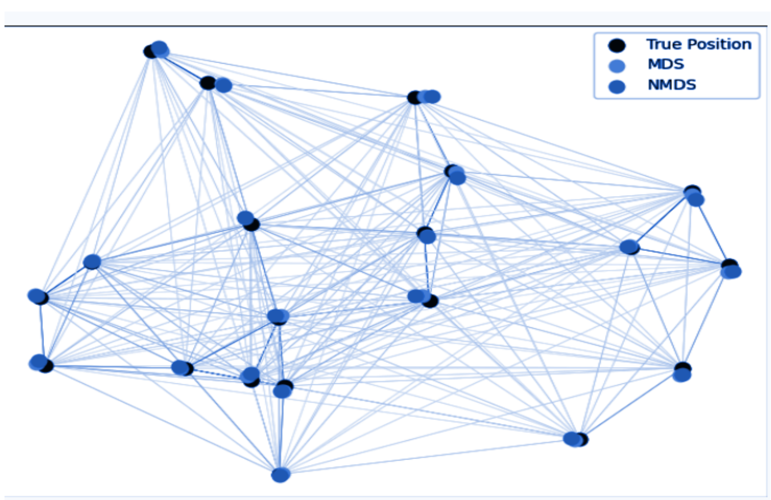
An−1 = {ei − ej , : 1 ≤ i < j ≤ n} Bn = {ei − ej , ei + ej : 1 ≤ i < j ≤ n} ∪ {ei : 1 ≤ i ≤ n} Cn = {ei − ej , ei + ej : 1 ≤ i < j ≤ n} ∪ {2ei : 1 ≤ i ≤ n} Dn = {ei − ej , ei + ej : 1 ≤ i < j ≤



Code-example

```
import numpy as np
from sklearn.datasets import load_boston
from sklearn.ensemble import RandomForestRegressor
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import Imputer
from sklearn.model_selection import cross_val_score
rng = np.random.RandomState(0)
dataset = load_boston()
X_full, y_full = dataset.data, dataset.target
n_samples = X_full.shape[0]
n_features = X_full.shape[1]
# Estimate the score on the entire dataset, with no missing values
estimator = RandomForestRegressor(random_state=0, n_estimators=100)
```

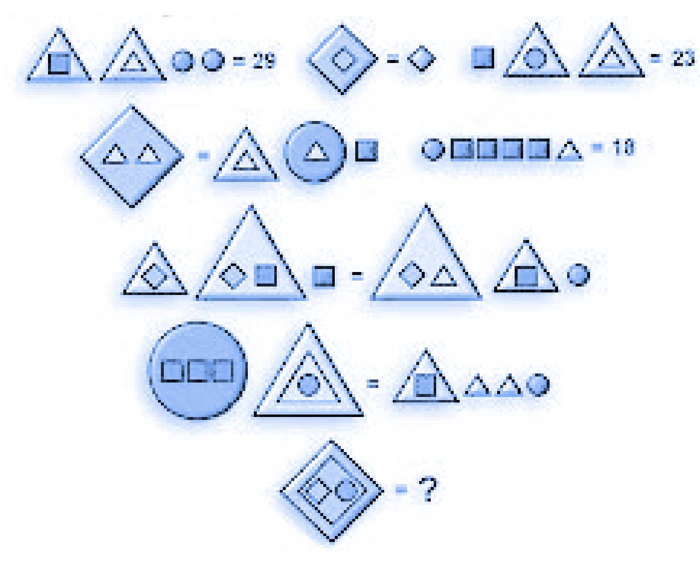
Graphical representations of multi dimensions for robotics simulations and aerodynamics



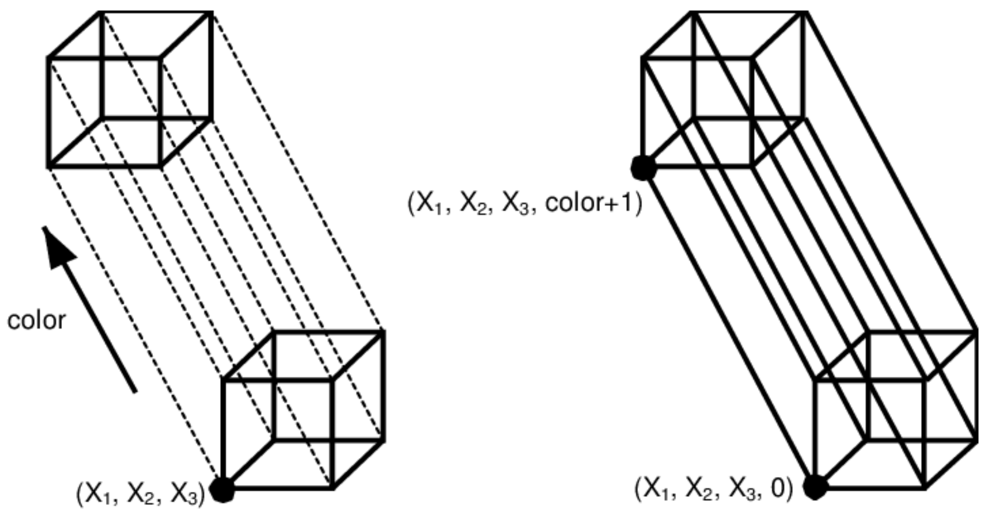
Mechanical-coordinates in aerospace and robotics,

Aerodynamics applied to enumerative combinatorics is a compose field built upon the higher dimensional basis and fluid mechanics to work with the higher dimensions data across space objects. The necessity to create new computational objects in space is why we apply the enumerative combinatorics and higher dimensions to basic aerodynamics. The objective of this research is looking for new ways to build more sophisticated software and hardware with some basic knowledge on fluid mechanics, potential moves the problem formulation can be reduced to a single differential equation, that of the velocity potential, with the appropriate boundary conditions. The analysis of two-dimensional potential movements is another field to look up relationships between Aerodynamics and higher dimensions. In previous sections I can enumerate some basic problems related to the field of enumerative combinatorics and the research of new computing algorithms as a basic approach to the field of Applied enumerative combinatorics to scientific computing. Some of the problems include two-dimensional potential problems and incompressibility, analyzing in some detail the elementary solutions of Laplace's equation (spring, vortex and doublet) that allow modeling the potential flow by superposition in simple mathematical terms around a wide variety of obstacles. This document also addresses the calculation of forces on a profile in the context of theory potential, and Kutta's hypothesis is stated regarding the trailing edge of the profiles. In order to describe some important examples to build spatial kinematics and new algorithms, this document is based on a non-inclusive academic background.

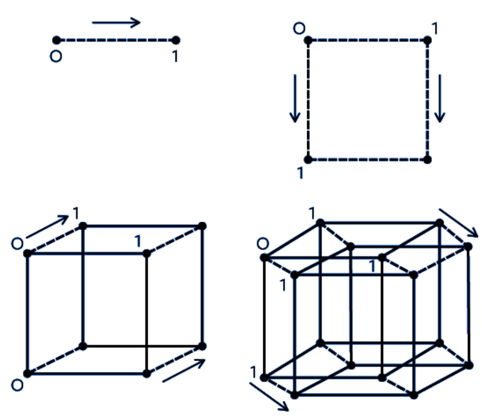
Higher dimensions to 29 to 23 elements example;



A tesseract is a four-dimensional closed figure with lines of equal length that meet each other at right angles. Since we've added another dimension, four lines meet at each vertex at right angles. Just as with a cube, each 2D face of the tesseract is a square.



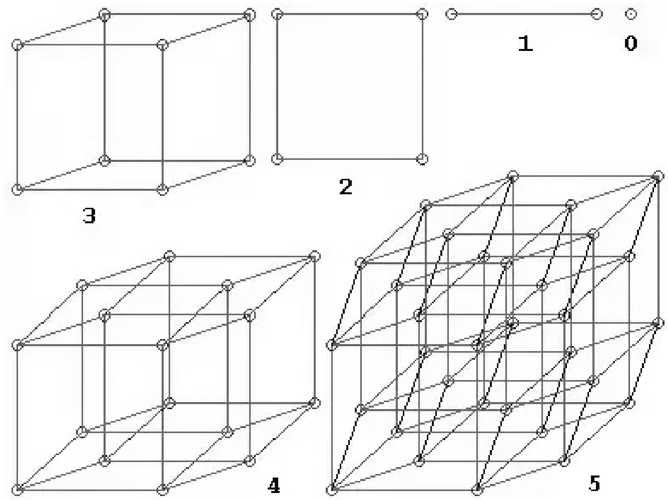
Cubes are just useful tools for understanding four-dimensional space. For example, let's start with a (1-D) line segment. By adding a parallel segment and connecting it with two perpendicular segments, all equally sized, we can make a (2-D) square. With us so far? Similarly, if we take two parallel squares and connect them with more perpendicular segments, we get a (3-D) cube. Now here's where things get tricky. In the next iteration, two parallel cubes plus perpendicular connectors creates a (4-D) hypercube, or tesseract. Tesseracts are objects that suffer alteration of states and replication in more depth in higher dimension ways with combinatorics.



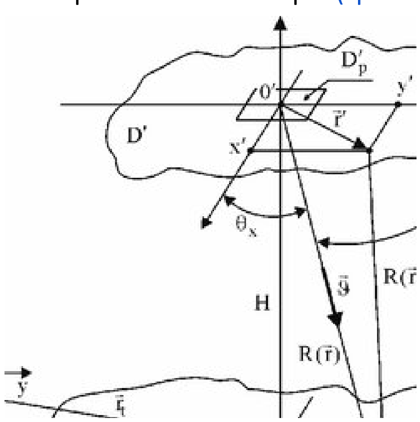
Calculus of higher dimensions with symbols

```
y = Az + (D) square (D) square log 4x(D) = 18
#logarithmic regression
z = [Bz] xor(states_variable)
# Triangle - square - circle representations - higher dimensions
set_states(D) [square]
coordinates_(4Dm log(x)(- 5D log(y) _tesseract_variable
```

```
Four_dimensional_data * regression [qubits_alternate = [a] * [b] * [c]
```



This is a simple model of a large-scale physical event. The dimensions move along with vectors whose faces have approximately 2,000 shapes and about 1,500 are derived with formulas. The key to questions relating to high-dimensional technical aspects is to see that when there are four dimensions, five or more dimensions; The algorithms begin to move computationally but at the same time with a lot of flexibility without space or gravity being a strong obstacle.
Descriptive-calculus sample (spatial-objects)



Computationally:

F (x,y,y',..., y(n))=0

Models

```
Matrix initialization store with the function value' => f(b) - f(a)
= f'(c)
```

```
ma = np.matrix([1.0, 2.0, 3.0] three dimension states
mb = np.zeros((3,2), dtype=np.int32) #3 * 2
mc = np.array([1.0, 2.0, 3.0], [4.0, 5.0, 6.0]]) #2 * 3
md=  np.matrix([7.0, 8.0, 9.0]]) #1 * 3
Print (ma)
Print (mb)
```

```
1) A > 0
2) A < 0
3) A = 0
```

*Regression log() [-40] corresponds to different dimensional states 4 * 4 float64 matrix my_det() is -40 with np.linalg.det() higher dimensions with matrices; Print mb => return #functional datasets in a Max-min margin diagram and Qubits = (a) XOR(different states) (gates b,c)*

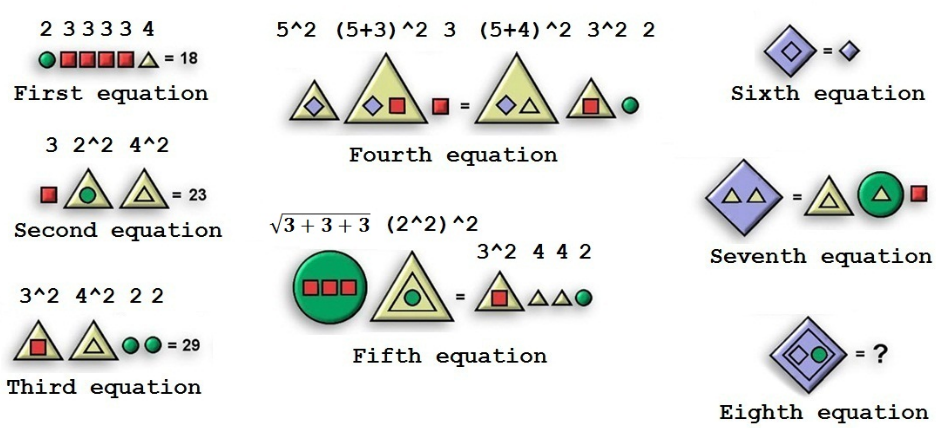
[Combinatorics]

Replicate-dimension with xor algorithms;

```
[ma] xor [mb] duplicate_inverse_dimension * [mc] is not R set
```

The reproduction of algorithms in computable-flexible spaces takes place on spatial objects at a scale that surpasses quantum reasoning and models. Model of replication of higher order dimensions and possible combinations;

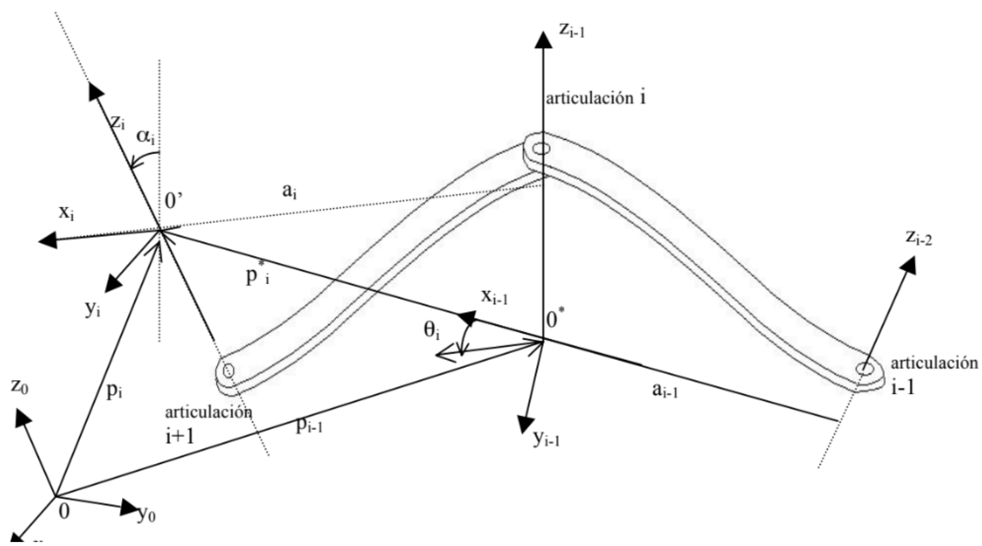
```
#!/usr/bin/python3
from qutip import *
import the NumPy and Matplotlib libraries with:
import numpy as np
import matplotlib.pyplot as plt
#Space manipulation with tensors and matrix in quantum mechanics
>>> C = qeye([4])
>>> E = qeye)[6])
>>> to_super(tensor(C, E)).dims #DIMENSIONS
>>> tensor(to_super(C), to_super(B)).dims
```



Quantum-dimensions:
The Fibonacci sequence is produced by starting with 0 and 1 and from then on each number in the sequence is the sum of the previous numbers in the sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, …

```
GreenCircle=2.
RedSquare=3.
YellowTriangle=4
BlueRhombus=5
```

Robot articulations diagram for variable research;



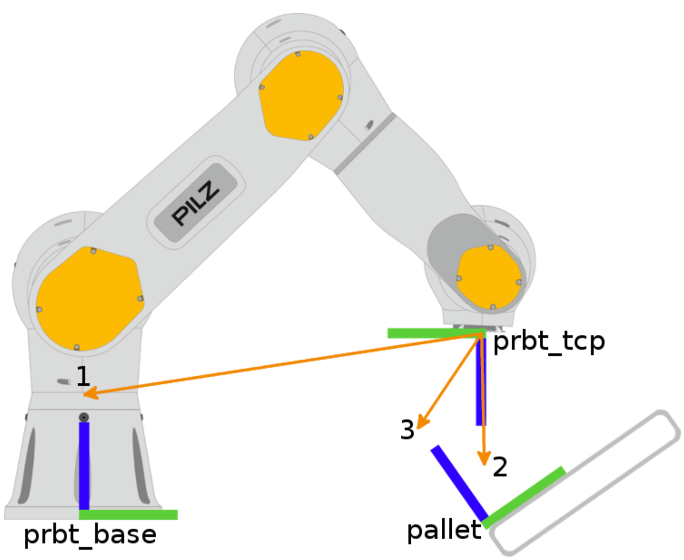
Industrial robots and RPA dynamics CPU-ROS and dynamics:

the 0* coordinate system shifts and rotates in space with respect to the base 0 reference system, the vector that describes the origin of the system in motion is h and the point P is described with respect to the system 0* a through the vector r, according to this, the description of point P with respect to the system variables:

```
r = r + h * (3.1) h v v dt
dh
dt
dr
dt
dr = + = +*
(Capture Element Screenshot    id:image_id ${OUTPUTDIR}/id_image_id-1.png
Capture Element Screenshot     id:image_id EMBED)
```

where v* is the velocity of point P with respect to the origin of the system 0* in motion and vh is the speed of the origin of the system 0* with respect to the base. Capture Element Screenshot id:image_id that is based on the robotic motion and kinematics (Pybotics or Pyrobotics libraries)

```
% Transformation matrices
% -------------------------------------------------- -----------
r01 = dh(theta(1), alpha(1)); r10 = r01';
r12 = dh(theta(2), alpha(2)); r21 = r12';
r23 = dh(theta(3), alpha(3)); r32 = r23';
r34 = dh(theta(4), alpha(4)); r43 = r34';
r45 = eye(3); r54 = r45';
% -------------------------------------------------- -----------
% Angular velocity of joints
% -------------------------------------------------- -----------
r00w0 = zeros(3,1);
r10w1 = ri0wi(r10, r00w0, qp(1));
r20w2 = r21*r10w1;
r30w3 = r32*r20w2;
r40w4 = ri0wi(r43, r30w3, qp(4));
r50w5 = ri0wi(r54, r40w4, 0);
% -------------------------------------------------- -----------
% Angular acceleration of the joints
% -------------------------------------------------- -----------
```

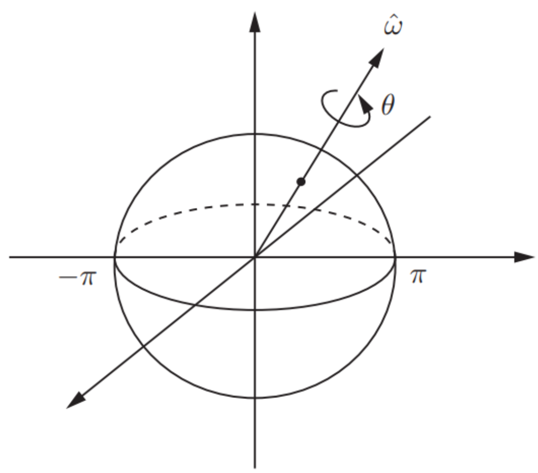


To obtain a computational algorithm, it must be taken into account that in the previous equations, the inertia matrices i I and the robot parameters, ir, i s, *i p are referenced with respect to the base coordinate system. Luh et al. [1980] used the 3x3 rotation matrices (ii R−1) , which have already been studied in practice 2 since they correspond to the upper left submatrix of the homogeneous transformation matrices i (i A−1), to express the variables wi , wi , i v , i v , i a , *

Aerodynamics-systems (robotic-design for multidimensional spaces)

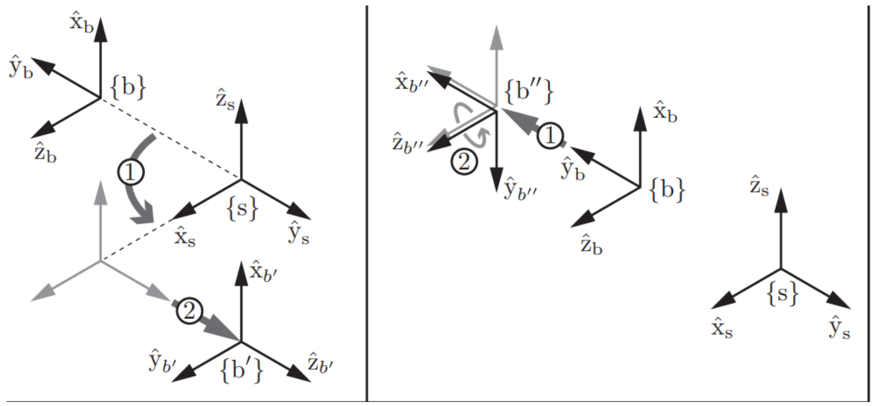
Calculus and mechanics of motion;

Matrix Logarithm of Rotations If ˆωθ ∈ R 3 represents the exponential coordinates of a rotation matrix R, then the skew-symmetric matrix [ˆωθ] = [ˆω]θ is the matrix logarithm of the rotation R. 4 The matrix logarithm is the inverse of the matrix exponential. Just as the matrix exponential "integrates" the matrix representation of an angular velocity [ˆω]θ ∈ so(3) for one second to give an orientation R ∈ SO(3), the matrix logarithm "differentiates" an R ∈ SO(3) to find the matrix representation of a constant angular velocity [ˆω]θ ∈ so(3) which, if integrated for one second, rotates a frame from I to R. In other words, exp : [ˆω]θ ∈ so(3) → R ∈ SO(3), log : R ∈ SO(3) → [ˆω]θ ∈ so(3).



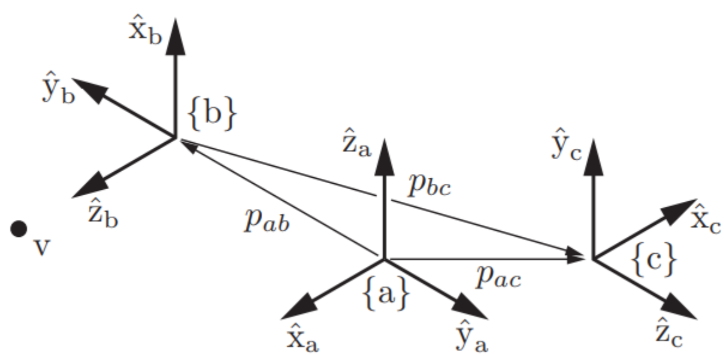

```
zs ω^ θ = 30∘ {b} ^xb
```

The matrix logarithm calculates exponential coordinates ˆωθ satisfying ||ωθˆ || ≤ π, we can picture the rotation group SO(3) as a solid ball of radius π given a point r ∈ R 3 in this solid ball, let ˆω = r/krk be the unit axis in the direction from the origin to the point r and let θ = krk be the distance from the origin to r, so that r = ˆωθ. The rotation matrix corresponding to r can then be regarded as a rotation about the axis ˆω by an angle θ. A Rigid-Body Motions means that for any R ∈ SO(3) such that tr R 6= −1, there exists a unique r in the interior of the solid ball such that e [r] = R. In the event that tr R = −1, log R is given by two antipodal points on the surface of this solid ball. That is, if there exists some r such that R = e [r] with krk = π then R = e [−r] also holds; both r and −r correspond to the same rotation R. The linear differential equation x˙(t) = Ax(t) with initial condition x(0) = x0, where A ∈ R n×n is constant and x(t) ∈ R n, has solution x(t) = e Atx0 Exponential Coordinates of Rotations The exponential coordinates of a rotation can be viewed equivalently as (1) a unit axis of rotation ˆω (ˆω ∈ R 3 , kωˆk = 1) together with a rotation angle about the axis θ ∈ R, or (2) as the 3-vector obtained by multiplying the two together, ωθˆ ∈ R 3 .



An element T ∈ SE(3) will sometimes be denoted (R, p). In this section we will establish some basic properties of SE(3) and explain why we package R and p into this matrix form. Many robotic mechanisms we have encountered thus far are planar. With planar rigid-body motions in mind, we make the following definition:

Definition 3.14. The special Euclidean group SE(2) is the set of all 3 × 3 real matrices T of the form T = R p 0 1 , (3.63) where R ∈ SO(2), p ∈ R 2 , and 0 denotes a row vector of two zeros. A matrix T ∈ SE(2) is always of the form T = r11 r12 p1 r21 r22 p2 0 0 1 = cos θ − sin θ p1 sin θ cos θ p2 0 0 1 , where θ ∈ [0, 2π).



Aerodynamics - drone;

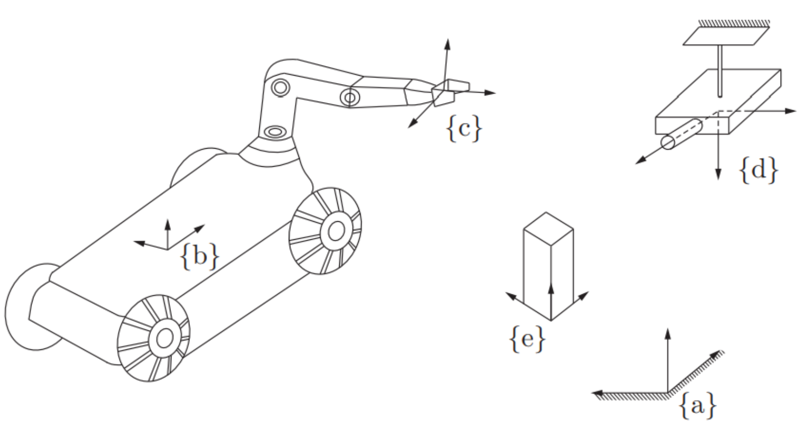
Fixed-frame and body-frame transformations corresponding to ˆω = (0, 0, 1), θ = 90◦ , and p = (0, 2, 0). (Left) The frame {b} is rotated by 90◦ about ˆzs and then translated by two units in ˆys , resulting in the new frame {b 0 }. (Right) The frame {b} is translated by two units in ˆyb and then rotated by 90◦ about its ˆz axis, resulting in the new frame {b 00}.a robot arm mounted on a wheeled mobile platform moving in a room, and a camera fixed to the ceiling. Frames {b} and {c} are respectively attached to the wheeled platform and the end-effector of the robot arm, and frame {d} is attached to the camera. A fixed frame {a} has been established, and the robot must pick up an object with body frame {e}.

Enumerative-combinatorics-qutip and solid calculus

Enumerative combinatorics is a field that is dedicated to the study of combinations of elements in the high order dimensions paradigm. Combinatorics of sets allow us to understand the problems that involve higher dimensions and machine learning models with the Scikit tool(Python)

Many of the modern mathematical problems and research are based upon the numpy,octave GNU and Calculus basis. Qutip is a modern python library for quantum states research in a computing format. The next generation of modern computational and mathematical work is on the research of new algorithms and new ways to explore the software and hardware fields out of the current academic research. What I really want to propose is a small research of different mathematical fields to build practical theory based on computational models from the Enumerative combinatorics field, Qutip and Calculus. Also, in the next documents I will work on higher order dimensions and enumerative combinatorics applied to the field of aerodynamics for dimensional states research and fluids.

My interest is the research of computational modeling with Matrices, arrays and calculus from the root-basis to establish a connection between the combinatorics and higher order dimensions field and the topic of computation to move forward to the next generation of problems that are related to the space-objects and higher dimensions as 4D, 5D, 6D proposing a new ideas and new objects based on mathematical research and interesting graphical representations based on comparison of different objects and calculus. These are a software development prototypes of representational data and machine learning research with the calculus field;
[Factorization-Matrices]
[Qutip-library(Scientific-research)]
[Matrices-simple-examples]
[Higher-order-dimensions(Git)]
[Scikit chapters]



Enumerative combinatorics applied to higher dimensions for computational algorithms research;

The main goal of enumerative combinatorics is to count the elements of a finite set. Most frequently, we encounter a family of sets T0, T1, T2, T3, . . . and we need to find the number tn = |Tn| for n = 1, 2, ...

For example, the tilings of Figure 1.1 correspond, respectively, to 1 + 1 + 1 + 1, 1 + 1 + 2, 1 + 2 + 1, 2 + 1 + 1, 2 + 2. These sums are easy to count. If there are k summands equal to 2 there must be n−2k summands:

an = bX n/2c , k=0 (n − k) k = n0 + (n − 1) ,1 + (n − 2) 2
+ · · · .

We can apply this formula to computational variables:

f(x + h, y + k) variables

Decomposition

```
variable subzero = 0.0012
variable h = 2.12
variable y = 1.25
variable k = 2.00
```

f(x,y) + [hf'x(x,y) + KF'y(x,y) = 0

```
import Tfidf Vectorizer, CountVectorizer
From sklearn.decomposition import NMK, lateratDirichletAllocation
```

And then model the algorithm to research higher dimensions in a imaginary computation in high scales Rather to think on small computing scales. Recurrence. Let n ≥ 2. In a domino tiling, the leftmost column of a 2 × n can be covered by a vertical domino or by two horizontal dominoes. If the leftmost domino is vertical, the rest of the dominoes tile a 2 × (n − 1) rectangle, so there are an−1 such tilings. On the other hand, if the two leftmost dominoes are horizontal, the rest of the dominoes tile a 2 × (n − 2) rectangle, so there are an−2 such tilings. We obtain the recurrence relations:

a0 = 1, a1 = 1, an = an−1 + an−2 for n ≥ 2.

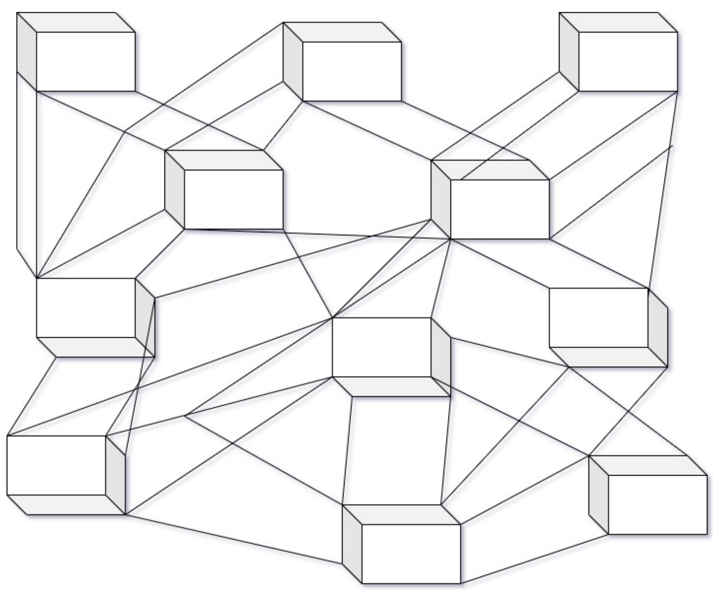
Explicit formula 2. There is a well established method that turns linear recurrence relations with constant coefficients, such as (2), into explicit formulas. We will review it in Theorem 2.4.1. In this case, the method gives

an = 1 √5 1 + √5 square 2 (!n+1) − 1 −√5 square2 !n+1

$$a_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right).$$

Generating function. The last kind of answer we discuss is the generating function. This is perhaps the strangest kind of answer, but it is often the most powerful one. Consider the infinite power series A(x) = a0 + a1x + a2x 2 + · · · . We call this the generating function of the sequence a0, a1, a2, . .

A graphical example of higher computing dimensions;



Generating functions help us enumerate our combinatorial objects in more detail, and understand some of their statistical properties. For instance, say we want to compute the number am,n of domino tilings of a 2 ×n rectangle which use exactly m vertical tiles.

Set of n components for dimensions:

```
1 ,1 − vx − x2 = X
[m,n≥0]
[ma,nv]
[mx]
[n]
```

Algebraic operations on ordinary and exponential generating functions correspond to some of the most common operations on combinatorial objects:

$$\sum_{n\geq 0}\left(\sum_{m\geq 0} m a_{m,n}\right) x^n = \left[\frac{\partial}{\partial v}\left(\frac{1}{1-vx-x^2}\right)\right]_{v=1} = \frac{x}{(1-x-x^2)^2}.$$

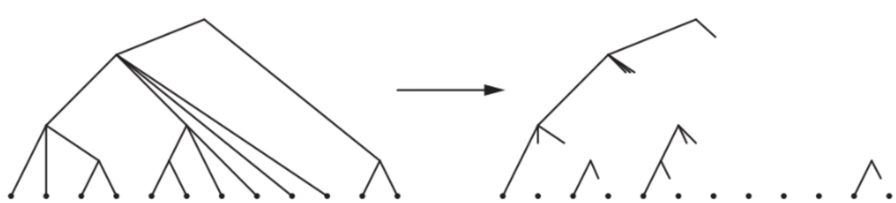
(a) Algebraic version. If A<−1>(x) is the compositional inverse of A(x) then

n[xn]A <−1>(x) = [x(n−1)]
x A(x)n

.**(b)** Combinatorial version. Assume |A0| = 0, |A1| = 1

A(x) = x − a2x
2 − a3x
3 − a4x
4 −

where an is the number of A-structures of size n for n ≥ 2

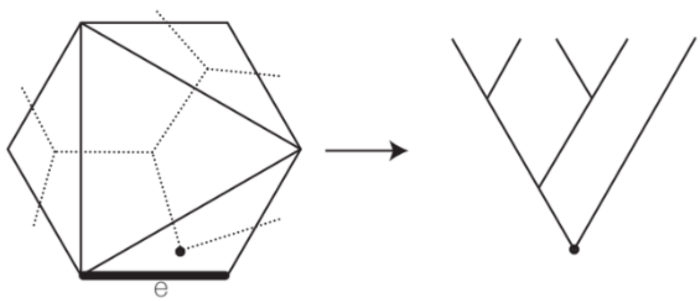


Reverse dimensions

The weight corresponds to the form of x 5+(−1)+0+(−1)+2+(−1)+(−1)+(−1)+(−1)+(−1)+0+(−1) = x
−1

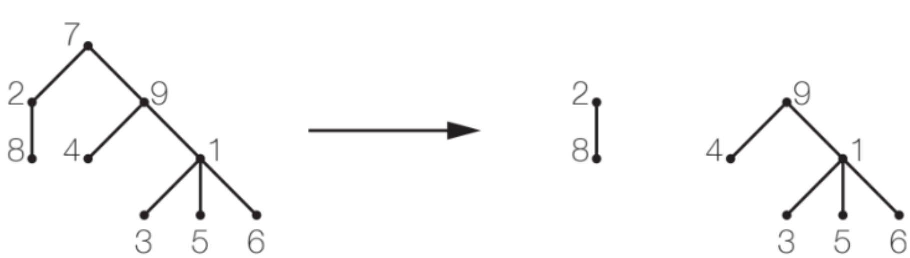
.Notice that all the partial sums of the sum $5 + (-1) + 0 + (-1) + 2 + (-1) + (-1) + (-1) + (-1) + (-1) + 0 + (-1) = -1$ are non-negative.

Conversely, suppose we wish to recover the A-tree corresponding to a sequence of sprigs $S_1, \dots, S_n$ with $|S_1| + \cdots + |S_n| = -1$.



These are examples of algebraic computations, but are still not difficult to obtain. Given a plane binary tree T of size n, prune all the leaves to get a tree T 0 with n vertices. Now walk around the periphery of the tree, starting on the left side from the root, and continuing until we traverse the whole tree. Record the walk in a Dyck path D(T): every time we walk up (resp. down)

The degree sequence of a rooted forest on [n] is (deg 1, . . . , deg n) where deg i is the number of children of i. For example the degree sequence of the rooted tree in Figure 2.11 is (3, 1, 0, 0, 0, 0, 2, 0, 2). Then the number of planted forests with a given degree sequence $(d_1, \dots, d_n)$ and (necessarily) $k = n - (d_1 + \cdots + d_n)$ components is

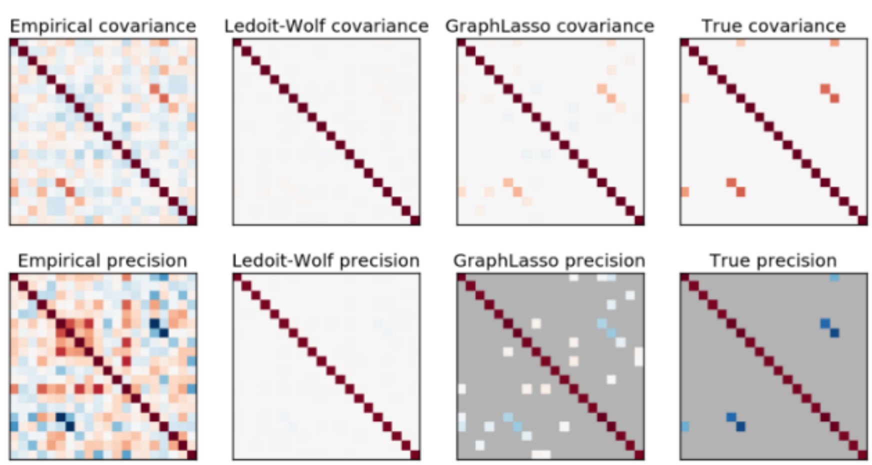


Some Scikit models for higher-dimensions to work with algorithms:
The model assumes the following generative process for a corpus with D components and K topics:

1. For each topic k, draw $\beta k \sim \text{Dirichlet}(\eta)$, k = 1...K
2. For each document d, draw $\theta d \sim \text{Dirichlet}(\alpha)$, d = 1...D
3. For each word i in document d:
4. Draw a topic index formula for a multinomial(θd)

For parameter estimation, the posterior distribution is:
$p(z, \theta, \beta | w, \alpha, \eta) = p(z, \theta, \beta | \alpha, \eta)$
$p(w | \alpha, \eta)$

Maximizing ELBO is equivalent to minimizing the Kullback-Leibler(KL) divergence between $q(z, \theta, \beta)$ and the true posterior $p(z, \theta, \beta | w, \alpha, \eta)$. LatentDirichletAllocation implements online variational Bayes algorithm and supports both online and batch update method. While the batch method updates variational variables after each full pass through the data, the online method updates variational variables from mini-batch data points.



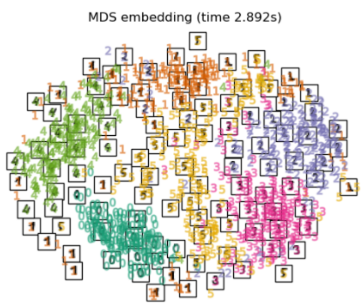
The main objective of machine learning models is to research new data across modeling with precise code structures to represent higher dimensions and enumerative combinatorics for further research with graphs and different estimator-tools based IA instrumentation.

Enumerative combinatorics applications:
Multidimensional scaling (MDS) seeks a low-dimensional representation of the data in which the distances respect the distances in the original high-dimensional space. In general, MDS is a technique

used for analyzing similarity or dissimilarity data. It attempts to model similarity or dissimilarity data as distances in geometric spaces. The data can be ratings of similarity between objects, interaction frequencies of molecules, or trade indices between countries. There exists two types of MDS algorithm: metric and non metric. In scikit-learn, the class MDS implements both. In Metric MDS, the input similarity matrix arises from a metric (and thus respects the triangular inequality), the distances between output two points are then set to be as close as possible to the similarity or dissimilarity data

Graphical example (Multidimensions research)

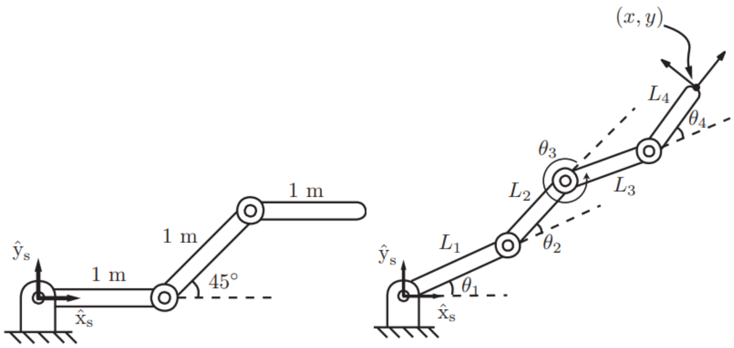
Is a static semantics method to catch dimensions at large scales



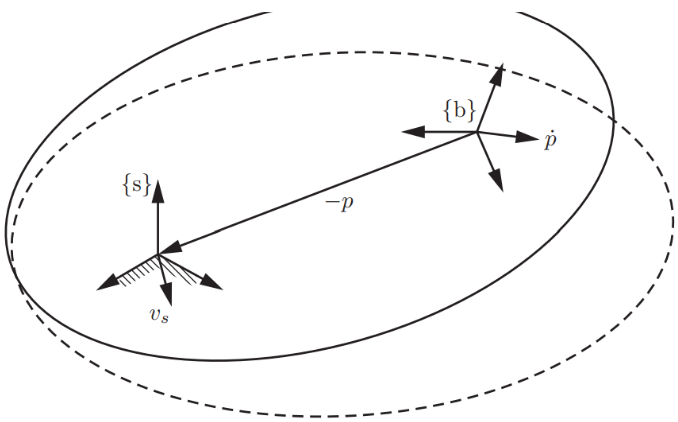
Let S be the similarity matrix, and X the coordinates of the n input points. Disparities $\hat{d}_{ij}$ are transformation of the similarities chosen in some optimal ways. The objective, called the stress, is then defined by $\sum_{i<j} d_{ij}(X) - \hat{d}_{ij}(X)$

Non metric MDS focuses on the ordination of the data. If $S_{ij} > S_{jk}$, then the embedding should enforce $d_{ij} < d_{jk}$. For this reason, we discuss it in terms of dissimilarities (δ_{ij}) instead of similarities (S_{ij}). Note that dissimilarities can easily be obtained from similarities through a simple transform, e.g. $\delta_{ij} = c_1 - c_2 S_{ij}$ for some real constants c_1, c_2. A simple algorithm to enforce proper ordination is to use a monotonic regression of d_{ij} on δ_{ij}, yielding disparities $\hat{d}_{ij}$ in the same order as δ_{ij}.

Robotic motion joints and torques representation:



Robotics & aerodynamics



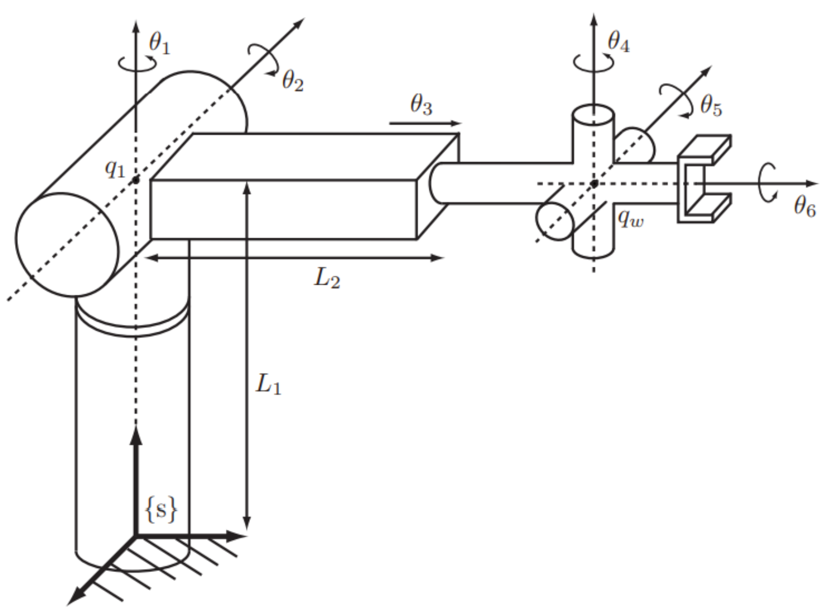
(Applied fluid-mechanics representations)
In order to calibrate objects in space we need to understand that we have more than three dimensions computationally speaking. Also, according to mechanical fluids there are some properties in the air temperature and atmospheric phenomena that constrain the original motion of airplanes in the case of wind or the bad calibration in the airplane's wings. Another interesting case is that the space-propulsors from the airplanes are measured by thermodynamics in the form of energy, and the robotic systems are built upon interesting materials for normal airplane simulation. In higher dimensions, vectors quantify the velocity, the acceleration and the restrictions in a given space (v) and the effective calibration of movements of different objects is given by computational equations. The computational equations are a number of higher order dimensions that have the ability to replicate other computational spaces. The difficult objective to achieve is research high scale enumerative combinations because higher dimensions are not intuitive to check despite human technology and research. One of the effective methods to calibrate spatial dimensions and research for parallel computations is to try to figure out how the matrix analysis works and also the possibility of traveling between different computational universes that can replicate some of the properties from quantum computing and enumerative higher order dimensions. All of their works are placed in a basis of the lookup for high scale computations in 8D, 9D and more, but in

order to replicate higher computations and higher dimensions we need an aircraft propulsor and Scipy matrices to create new aerospatial objects.

In this text I propose an analysis of applied spatial relations and matrices to check how to satisfy some aerodynamic properties to search for new investigations to discover more computational dimensions and systems which the origin is not a computer”

1. Aerospace propulsors + higher dimensions
2. Matrices + programming
3. Robotic for enumerative higher order dimensions

Body Jacobian is based in the relationship between the joint rates and [Vs] = T T˙ −1 , the end-effector’s twist expressed in fixed-frame coordinates. Here we derive the relationship between the joint rates and [Vb] = T −1T˙ , the end effector twist in end-effector-frame coordinates.



T(θ) = Me[B1]θ1 e [B2]θ2 · · · e [Bn]θn
T˙ =Me[B1]θ1 · · · e [Bn−1]θn−1 d dt e [Bn]θn
+ Me[B1]θ1 · · · d dt e [Bn−1]θn−1 e [Bn]θn + · · ·
Evaluating T −1T˙ , [Vb] = [Bn] ˙θn + e −[Bn]θn [Bn−1]e [Bn]θn ˙θn−1 + · · · + e −[Bn]θn · · · e −[B2]θ2 [B1]e [B2]θ2 · · · e [Bn]θn ˙θ1
Vb = Bn |{z} Jbn ˙θn + Ade−[Bn]θn (Bn−1) | {z } Jb,n−1 ˙θn−1 + · · · + Ade−[Bn]θn ···e−[B2]θ2 (B1) | {z } Jb1 ˙θ1.

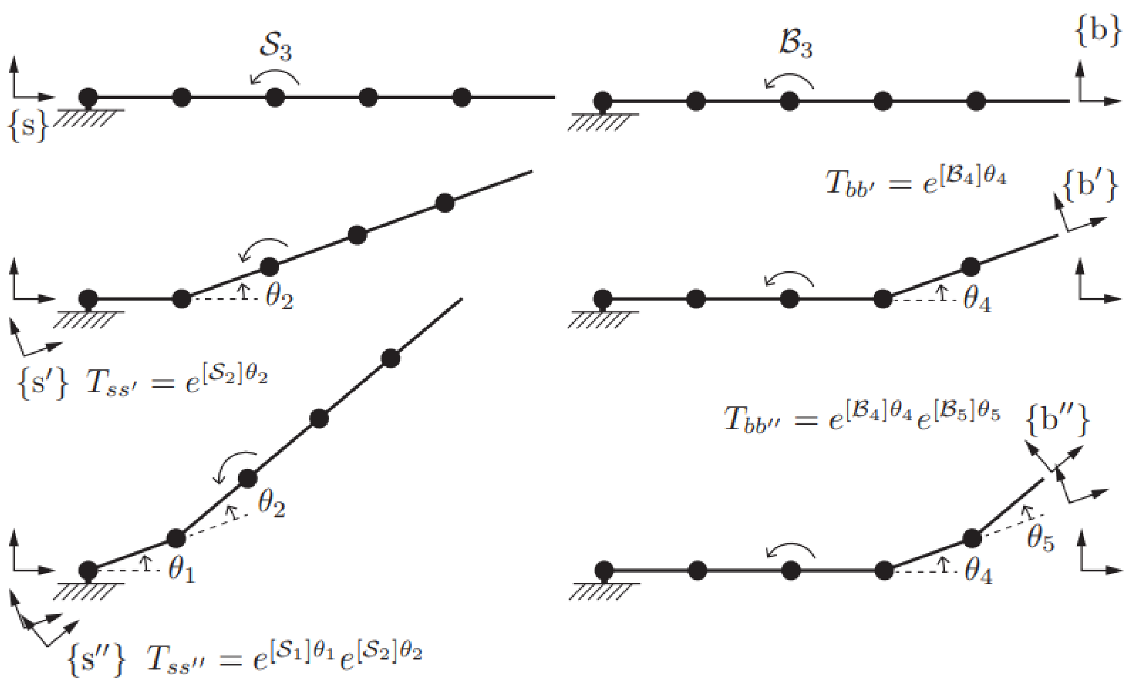
Let the forward kinematics of an n-link open chain be expressed in the following product of exponentials form:

T = Me[B1]θ1 · · · e [Bn]θn .

The body Jacobian Jb(θ) ∈ R 6×n relates the joint rate vector ˙θ ∈ R n to the end-effector twist Vb = (ωb, vb) via

Vb = Jb(θ) ˙θ.

Graphical representation of a rigid body for higher order dimensions;



Variables:

Variables θ1, θ2, and θ3 have no impact on the body twist resulting from the joint velocity ˙θ3, because they do not displace axis 3 relative to {b}. So we fix those joint variables at zero, making the robot a rigid body B from the base to joint 4. If θ5 = 0 and θ4 is arbitrary, then the frame {b 0 } at Tbb0 = e [B4]θ4 is the new end-effector frame. Now if θ5 is also arbitrary, then the frame {b 00}

```
Jb3 = [AdTb00b ]B3 = [AdT -1 bb00 ]B3 = [Ade-[B5]θ5 e-[B4]θ4 ]B3.
```

Jacobian that are based on a representation of the end-effector configuration using a minimum set of coordinates q. Such representations are particularly relevant when the task space is considered to be a subspace of SE(3). For example, the configuration of the end-effector of a planar robot.

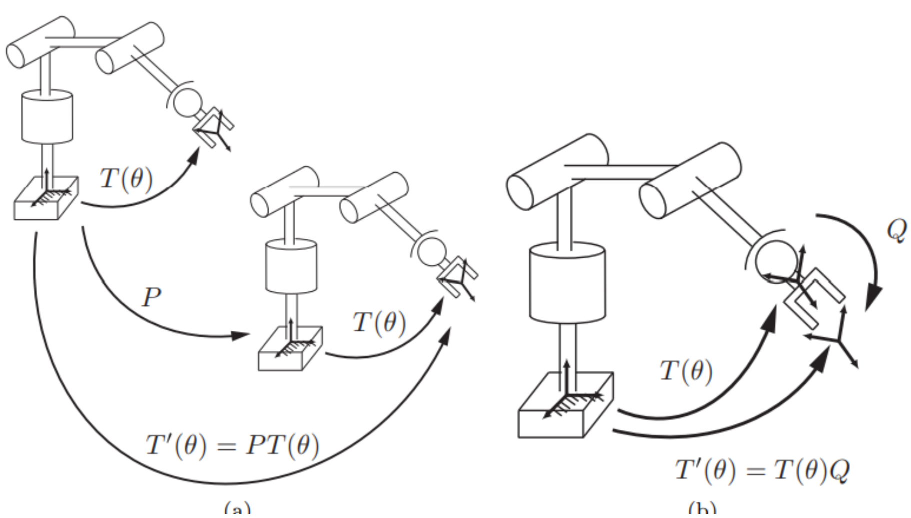
```
as q = (x, y, θ) ∈ R 3 instead of as an element of SE(2).
```

(Analytic Jacobian with exponential coordinates for rotation formulas)
The rigid and elastic planes of expressions are exposed to the center of gravity (G) but also, we can manipulate the center of gravity with some quantum computing algorithm.

The angular and linear velocity components of Vb = (ωb, vb) can be written explicitly as the correlation between matrices in the form of 3 * n components to the three rows in Jacobian spaces for rigid bodies in robotics systems and mathematical-modeling examples;

```
Vb =  ωb vb  = Jb(θ) ˙θ =  Jω(θ) Jv(θ)  ˙θ
,where Jω is the 3 × n matrix corresponding to the top three rows of Jb and Jv
is the 3 × n matrix corresponding to the bottom three rows of Jb.
Kinematics formula:

T(θ) = e [S1]θ1 e [S2]θ2 e [S3]θ3 e [S4]θ4M, write down M ∈ SE(2) and each Si
= (ωzi, vxi, vyi) ∈ R 3 .
```



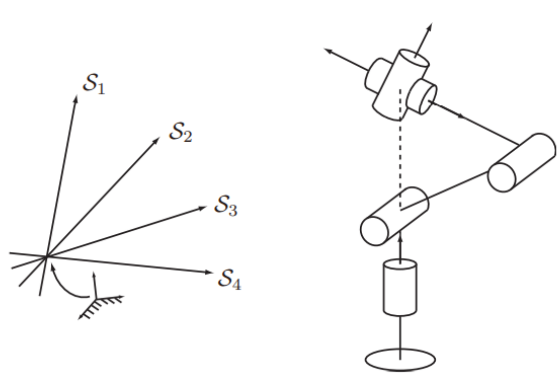
(New inverse gravitational high dimension order scales)
Suppose that the center of gravity for a given robot does not exists and place in his dynamic formula The new center of quantum bytes. If you conjugate the different parameters something different happens computationally speaking. One of the possible results is the inverse gravities”

Inverse gravities: Inverse quantum computing centers of gravitational forces occurs when the different virtual systems refused to connect to the local gravity of the earth generating in a new dimensional spaces that the human being is unable to compute because higher enumerative dimensions are correlate with higher number of centers of new gravitational quantum models. This is where the problem is, because computers and robots are unable to imitate the formulation of new inverse gravitational computing algorithms, because in order to imitate this model we need some important steps for our theory performance;

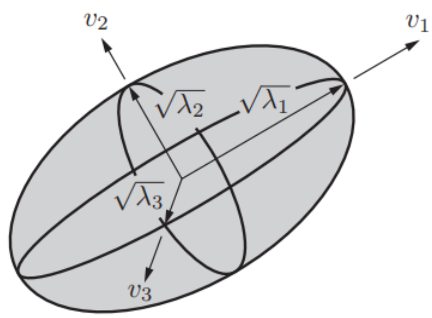
1. The gravity of an exoplanet (G) for **python-aerospace objects**
2. A good understanding of higher order enumerative dimensions (D) with **Scipy in linux-software**
3. Spatial inverse quantum gravities placed as the center where the rigid and elastic body is able to move through (Sp) Qutip library for inverse gravities in the center of higher dimensions theory.
4. The reference point is a quantum bytes-components placed in inverse gravitational replicative algorithms with the form of higher order dimensions 8D, 9D, etc…(&D)
5. The objective is build new spatial objects that can deal with a really big adversary, the hard conditions of the galaxies and new materials (materials)
6. Analysis of spatial objects references, python testing, strace for aerospace files

One of the ways to check for new algorithms is to recreate a new programming language such python for Gravitational algorithms at height scale and inverse the algorithms, the Qutip library offers new gravitational algorithms based on quantum computing and the spatial algorithms for robots with aerodynamics forces are necessary to understand the basics of new computing models based in inverse gravitational virtual dimensions and algorithms.

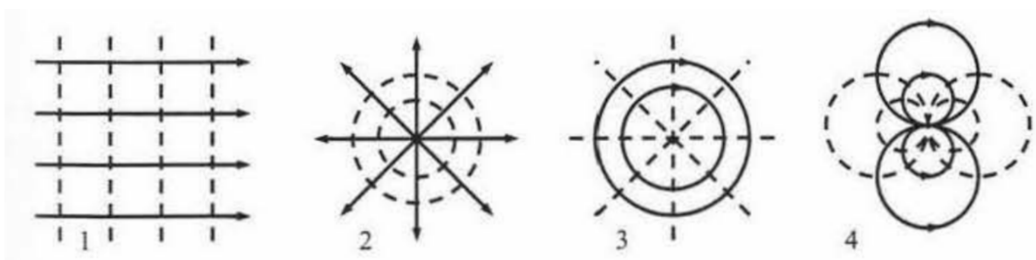
Aerodynamics properties and materials;
The first case we consider is one in which two revolute joint axes are collinear
Since the two joint axes are collinear, we must have $\omega s1 = \pm \omega s2$; let us assume the positive sign. Also, $\omega si \times (q1 - q2) = 0$ for i = 1, 2. Then s1 = s2, the set {s1, s2, . . . , s6} cannot be linearly independent, and the rank of Js(θ) must be less than six.



An ellipsoid visualization of ˙q TA −1 q˙ = 1 in the ˙q space R 3 , where the principal semi-axis lengths are the square roots of the eigenvalues λi of A and the directions of the principal semi-axes are the eigenvectors vi.



A = JJ T. When µ1(A) is low (i.e., close to 1) then the manipulability ellipsoid is nearly spherical or isotropic, meaning that it is equally easy to move in any direction. This situation is generally desirable. As the robot approaches a singularity, however, µ1(A) goes to infinity.



Aerodynamic (the classic properties)
To describe the properties of fluids and the characteristics of flows, experience teaches that, in the area of interest of the technique, the continuous media satisfy four fundamental principles, namely:

l. Conservation of matter, stated in fluid mechanics as the continuity equation.

2. Newton's second principle, which establishes the balance of the amount of motion.

3. First law of thermodynamics, which leads to the equation of energy.

"The continuity equation expresses the principle of conservation of (~o.::a) which states that in a space-fixed control volume V, limited to a boundary S, the net flow of mass through the. volume surface The control must be equal to the variation with time of the mass contained in domain (D)"

Motion and Laplace for computational variables are;

The effect of viscosity (mechanism creation of circulation on a profile, profile stall
and to the. fluid-dynamic resistance), viscous effects are assumed to be despicable. It must be noted that, in any case, the analytical study of movements in the effect of viscosity, unless it is assumed that such effect is confined in the boundary layer.

Energy's equation

Regarding the energy equation, the first simplification arises from the fact that in the movement of air around the plane the flow of heat by radiation and that due to chemical reactions are null (except in very cases, In the special model-space object, such as the re-entry of a space vehicle into a planetary atmosphere), so the functional equation is [~e] that generalizes the compressible fluids in two dimensional spaces.

The current function[~e]

The function [~e] easily generalizes to the case of compressible fluids in a motion stationary two-dimensional. In effect, as in the stationary case in regime c:compressible the continuity equation is 'V · (pV) = O, this expression, suggests defining the relationship between velocities and current function including 'ª density of the fluid in the definition,s formula *pU = p008'1! /8z, pW = - px8'11 /ox* which satisfied some potential motion properties ;

For example, the formula: *pU = p008'1! /8z, pW = - px8'11 /ox,* so the previous reasoning about the f:'existence of the current function. The conditions for the existence and fulfillment of the equation Laplace of the potential function <J.> and the current function W are:

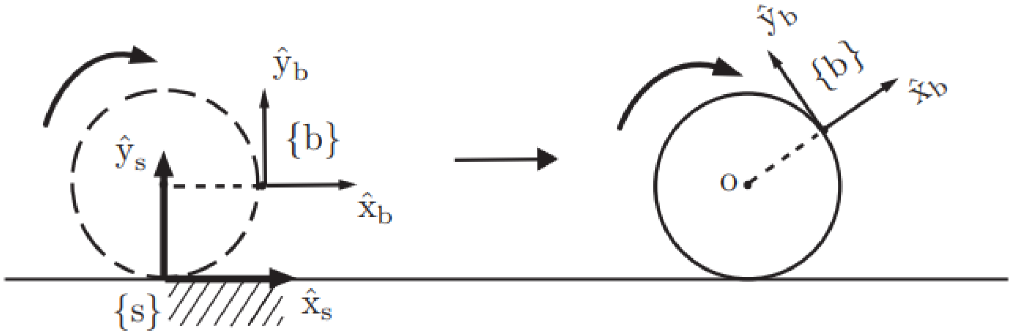
• Potential function, cl>(x. z): exists if the movement is irrotational, 8U /fJz - fJW/fJx = 0, and it satisfies Laplace's equation if the motion is flat and incompressible, 8U/8x + 8VV/8z = O; generalization to the case three-dimensional is obvious.

• Current function, w(x. z): exists if the motion is plane and incompressible,

oU /8x+éJW/8z =O, and satisfies Laplace's equation if the motion is irrotational and of course plane, 8U/oz -8W/8x = O.

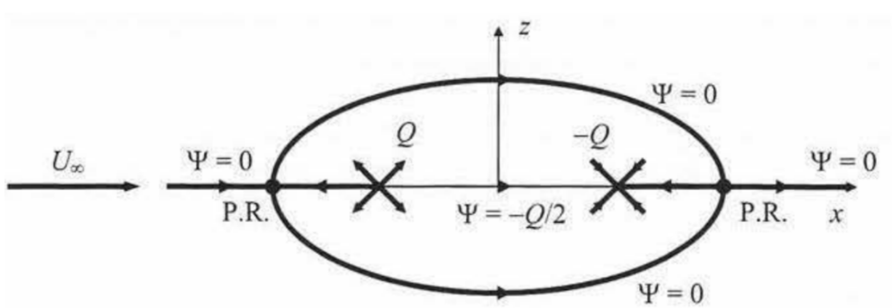
Complex potential

The frame and the optimal schemes of the velocity field produced by (a) a uniform current that W1 makes angle a with the x axis. (b) by one spring located at the origin of coordinates, (e) by a whirlwind potential located at the origin. and (d) scheme of the singularics, in this case and sink, the necessary for the generation of a real axis doublet.



The space shapes are conjugated at a generic point t = r(9) circles centered on the vortex (the fluid field has symmetry of revolution). Therefore, if the lines are chosen as fixed lines of current, as the velocity induced by the vortex is tangent to these components of velocity according to axes are now.
And the conjugate speed is in the deep configuration of higher dimensions quantum gravity.

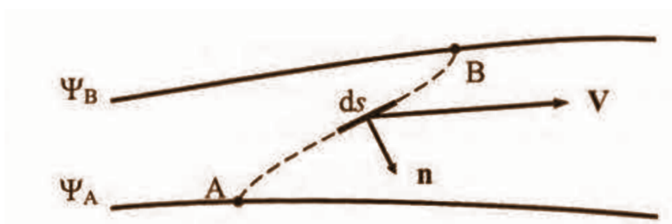
Backwater spots and dividing streamlines in the fluid configuration formed by the superposition of a current uniform, U00, a spring of intensity Q and a sink also of the same. intensity, -Q, both located on the x axis. and the equation that determines the position of the backwater points.In the configuration under consideration, two points of backwater, both on the x axis, one located in front of the spring and the second behind 1 sink. If assigned to the x axis, which is streamline, the value W = O upstream of the spring, on the dividing streamline that surrounds the spring and sink must also be l]i = 0. Thus, the function of current is W =O on the x axis, in the intervals (- oo, -a) and (o., oo), and in the closed streamline that passes through the backwater points, but not in the segment (-a, a) of the x-axis, where '1i = -Q/2 must be (consistent with what has been said regarding the expenditure and the current function).




```
Qa 2
0--- =Ü, 7rU00 t2 - a2 (2.22)
whose roots are t = ±(1 + q)112 , with q = Qa2/(7raU00 ))
```

A semi-wing located on one of the walls of the chamber wind tunnel tests (a) behaves as if it really there was another semi-image in one. testing chamber.Various possible configurations depending on the parameter that they qualify in the figure varies. In order to calculate the forces on a profile, whose contour, according to with what is stipulated in the formula ((roots are t = ±(1 + q)112 , with q = Qa2/(7raU00)) it is assumed generated by a distribution of singularities arranged on a line element, several magnitudes that can be parameterized within the theorem of conservation of momentum are applied to a fluid volume that contains the profile and that has its external boundaries very far from it. To determine the forces on the profile it is necessary to evaluate the action of pressure and the flow of momentum through the outer walls of the control volume, so it is necessary to determine the field of velocities and pressures very far from the profile of the control volume. space object.

The profile of the spatial object such a satellite or another dimensional object with a sharp trailing edge within a potential current in which circulation has not been established. The circulation brings the backwater point closer to the exit edge.



The potential {continuous line) and actual pressure distributions (circles) on a circular cylinder without circulation at a number of Reynolds iJ1lower than the critical (B:::::ds , laminar boundary layer, circles whites) and at a Reynolds number higher than the critical one (A::::: 2.1 x 105 possible air dimensionality)

a force f = (10, 10, 0) and a moment m = (0, 0, 10) are applied to the tip (both f and m are expressed with respect to the fixed frame)

[Modern - robotics examples]
[aircraft-design with python and images]
[modellica]