

My Title

Shuvomoy Das Gupta

December 30, 2025

Some text and some code.

- item 1
- item 2
- item 3

```
x = 1
# a comment
y = x+1
## this is another comment
```

$$\begin{aligned}a &= b + c \\ &\quad + d + e \\ f &= g - h\end{aligned}$$

As shown in Equation, Euler's identity is a fundamental mathematical relationship.

1 Einstein's Equation $E = mc^2$

This is markdown. Consider:

$$E = c^2 \tag{1}$$

Where Equation (1) demonstrates the relationship between energy e , mass m , and the speed of light c .

```
m = 1
c = 3e8
E = μ*c^2
```

Another multiline equation is as follows:

$$\begin{aligned}Ax &= b \\ x &\geq 0\end{aligned} \tag{2}$$

Here, Equation (2) represents a system of linear equations with non-negativity constraints.

Here is a table:

Algorithm	$f(x^*)$	Time (s)
PDHG	1.234	0.07
B&B	1.230	3.12

Another table:

Field	Type	Mathematics	Implementation detail	Role in the package
<code>_id</code>	<code>Int</code>	Pure identifier; no direct math meaning	Incremented from global <code>NEXT_ID[]</code>	Stable identity for hashing, comparisons, dictionary keys
<code>_is_leaf</code>	<code>Bool</code>	Indicates whether this is a fundamental vector in the Gram basis	<code>true</code> for leaf points, <code>false</code> for linear combinations	Determines Gram dimensioning and whether <code>.counter</code> is set
<code>decomposition_dict</code>	<code>OrderedDict{Point,Float64}</code>	Stores coefficients of the linear form $X = \sum_i \alpha_i P_i$	For a leaf, set to <code>{self \Rightarrow 1.0}</code> ; for a composite, the sparse coefficient map	Drives conversion of inner products to linear forms over G
<code>counter</code>	<code>Union{Int,Nothing}</code>	Index i of the leaf in the Gram basis (only for leaves)	Set to <code>Point_counter[]</code> at leaf creation, <code>nothing</code> otherwise	Used to size/index the Gram matrix and build evaluation vectors
<code>_value</code>	<code>Union{Vector{Float64},Nothing}</code>	Current value of the vector after solving the PEP	<code>nothing</code> until <code>solve!</code> writes results back	Enables <code>eval</code> to return a concrete vector after solve