FigurateNum

A Python library for generating infinite figurate number sequences across dimensions.

Installation

```
pip install figuratenum
pip install figuratenum[figurate-viz] # Optional: Visualization (v2.1.0)
```

Main Class

```
from figuratenum import FigurateNum as fgn
```

Classes

```
from figuratenum import PlaneFigurateNum as pfgn
from figuratenum import SpaceFigurateNum as sfgn
from figuratenum import MultidimensionalFigurateNum as mfgn
from figuratenum import ZooFigurateNum as zfgn
from figuratenum import NumCollector as nc
from figuratenum.figurate_viz.FigurateViz import FigurateViz
```

Import and generate sequences with FigurateNum and related classes

```
from figuratenum import FigurateNum, MultidimensionalFigurateNum
# 1. General use: generate any figurate sequence via FigurateNum
seq = FigurateNum()
hyperdodecahedral_gen = seq.hyperdodecahedral()
print([next(hyperdodecahedral_gen) for _ in range(4)])
# Output: [1, 600, 4983, 19468]
# 2. Specialized classes: PlaneFigurateNum, SpaceFigurateNum,
#    MultidimensionalFigurateNum, ZooFigurateNum
multi = MultidimensionalFigurateNum()
hypertetrahedron_gen = multi.k_dimensional_centered_hypertetrahedron(21)
print([next(hypertetrahedron_gen) for _ in range(10)])
# Output: [1, 23, 276, 2300, 14950, 80730, 376740, 1560780, 5852925, 20160075]
```

Using FigurateViz to visualize and export

```
from figuratenum import FigurateNum as fgn
from figuratenum.figurate_viz.FigurateViz import FigurateViz
seq_loop = fgn()
gen = seq_loop.five_dimensional_hyperoctahedron()
figuratenum_seq = [next(gen) for _ in range(704)]
# Create and draw the Gaussian plot
viz = FigurateViz(figuratenum_seq, figsize=(6, 6))
viz.gaussian_plot(circ_color="m", bg_color="k", num_text=False,
    num_color="g", ext_circle=True, rotate=-1).draw()
# Export plots as .svg, .pdf, .png (matplotlib compatible),
# with options e.g., dpi, transparent, bbox_inches, pad_inches, etc.
viz.export_plot("figure1.svg", circ_color="cyan", transparent=True)
```

NumberCollector Class: Method Summary

- take(n)
- take_to_list(stop, start, step)
- take_to_array(stop, start, step)
- take_to_tuple(stop, start, step)
- pick(n)

Working with the NumberCollector Class

```
from figuratenum import NumCollector as nc, FigurateNum
gen = FigurateNum().pentatope()
print(nc.take_to_tuple(gen, 10))  # first 10 values as tuple
# Output: (1, 5, 15, 35, 70, 126, 210, 330, 495, 715)
```

Plane Figurate Numbers

- polygonal
- triangular
- square
- pentagonal
- hexagonal
- heptagonal
- octagonal
- nonagonal
- decagonal
- hendecagonal
- dodecagonal
- tridecagonal
- tetradecagonal
- pentadecagonal
- hexadecagonal
- heptadecagonal
- octadecagonal
- nonadecagonal
- icosagonal
- icosihenagonal
- icosidigonal
- icositrigonal
- icositetragonal
- icosipentagonal
- icosihexagonal
- icosiheptagonal
- icosioctagonal
- icosinonagonal
- triacontagonal
- centered_triangular
- centered_square = diamond
- centered_pentagonal
- centered_hexagonal
- centered_heptagonal
- centered_octagonal
- centered_nonagonal
- centered_decagonal
- centered_hendecagonal
- centered_dodecagonal = star
- centered_tridecagonal

Plane Figurate Numbers

- centered_tetradecagonal
- centered_pentadecagonal
- centered_hexadecagonal
- centered_heptadecagonal
- centered_octadecagonal
- centered_nonadecagonal
- centered_icosagonal
- centered_icosihenagonal
- centered_icosidigonal
- centered_icositrigonal
- centered_icositetragonal
- centered_icosipentagonal
- centered_icosihexagonal
- centered_icosiheptagonal
- centered_icosioctagonal
- centered_icosinonagonal
- centered_triacontagonal
- centered_mgonal(m)
- pronic = heteromecic = oblong
- polite
- impolite
- cross
- aztec_diamond
- polygram(m) = centered_star_polygonal(m)
- pentagram
- gnomic
- truncated_triangular
- truncated_square
- truncated_pronic
- truncated_centered_pol(m) = truncated_centered_mgonal(m)
- truncated_centered_triangular
- truncated_centered_square
- truncated_centered_pentagonal
- truncated_centered_hexagonal = truncated_hex
- generalized_mgonal(m, start_numb)
- generalized_pentagonal(start_numb)
- generalized_hexagonal(start_numb)
- generalized_centered_pol(m, start_numb)
- generalized_pronic(start_numb)

Space Figurate Numbers

- m_pyramidal(m)
- triangular_pyramidal
- square_pyramidal = pyramidal
- pentagonal_pyramidal
- hexagonal_pyramidal
- heptagonal_pyramidal
- octagonal_pyramidal
- nonagonal_pyramidal
- decagonal_pyramidal
- hendecagonal_pyramidal
- dodecagonal_pyramidal
- tridecagonal_pyramidal
- tetradecagonal_pyramidal
- pentadecagonal_pyramidal
- hexadecagonal_pyramidal
- heptadecagonal_pyramidal
- octadecagonal_pyramidal
- nonadecagonal_pyramidal
- icosagonal_pyramidal
- icosihenagonal_pyramidal
- icosidigonal_pyramidal
- icositrigonal_pyramidal
- icositetragonal_pyramidal
- icosipentagonal_pyramidal
- icosihexagonal_pyramidal
- icosiheptagonal_pyramidal
- icosioctagonal_pyramidal
- icosinonagonal_pyramidal
- triacontagonal_pyramidal
- triangular_tetrahedral[finite]
- triangular_square_pyramidal[finite]
- square_tetrahedral[finite]
- square_square_pyramidal[finite]
- tetrahedral_square_pyramidal[finite]
- cubic
- tetrahedral
- octahedral
- dodecahedral
- icosahedral
- truncated_tetrahedral

Space Figurate Numbers

- truncated_cubic
- truncated_octahedral
- stella_octangula
- centered_cube
- rhombic_dodecahedral
- hauy_rhombic_dodecahedral
- centered_tetrahedron = centered_tetrahedral
- centered_square_pyramid = centered_pyramid
- centered_mgonal_pyramid(m)
- centered_pentagonal_pyramid
- centered_hexagonal_pyramid
- centered_heptagonal_pyramid
- centered_octagonal_pyramid
- centered_octahedron
- centered_icosahedron = centered_cuboctahedron
- centered_dodecahedron
- centered_truncated_tetrahedron
- centered_truncated_cube
- centered_truncated_octahedron
- centered_mgonal_pyramidal(m)
- centered_triangular_pyramidal
- centered_square_pyramidal
- centered_pentagonal_pyramidal
- centered_heptagonal_pyramidal
- centered_octagonal_pyramidal
- centered_nonagonal_pyramidal
- centered_decagonal_pyramidal
- centered_hendecagonal_pyramidal
- centered_dodecagonal_pyramidal
- centered_hexagonal_pyramidal = hex_pyramidal
- hexagonal_prism
- mgonal_prism(m)
- generalized_mgonal_pyramidal(m, start_num)
- generalized_pentagonal_pyramidal(start_num)
- generalized_hexagonal_pyramidal(start_num)
- generalized_cubic(start_num)
- generalized_octahedral(start_num)

Space Figurate Numbers

- generalized_icosahedral(start_num)
- generalized_dodecahedral(start_num)
- generalized_centered_cube(start_num)
- generalized_centered_tetrahedron(start_num)
- generalized_centered_square_pyramid(start_num)
- generalized_rhombic_dodecahedral(start_num)
- generalized_centered_mgonal_pyramidal(m, start_num)
- generalized_mgonal_prism(m, start_num)
- generalized_hexagonal_prism(start_num)

Multidimensional Figurate Numbers

- k_dimensional_hypertetrahedron(k) = k_hypertetrahedron(k) = regular_k_polytopic(k) = figurate_of_order_k(k)
- five_dimensional_hypertetrahedron
- six_dimensional_hypertetrahedron
- k_dimensional_hypercube(k) = k_hypercube(k)
- five_dimensional_hypercube
- six_dimensional_hypercube
- hypertetrahedral = pentachoron = pentatope = triangulotriangular = cell_5
- hypercube = octachoron = tesseract = biquadratic = cell_8
- hyperoctahedral = hexadecachoron = four_cross_polytope = four_orthoplex = cell_16
- hypericosahedral = hexacosichoron = polytetrahedron = tetraplex = cell_600
- hyperdodecahedral = hecatonicosachoron = dodecaplex = polydodecahedron = cell_120
- polyoctahedral = icositetrachoron = octaplex = hyperdiamond = cell_24
- four_dimensional_hyperoctahedron
- five_dimensional_hyperoctahedron
- six_dimensional_hyperoctahedron
- seven_dimensional_hyperoctahedron
- eight_dimensional_hyperoctahedron
- nine_dimensional_hyperoctahedron
- ten_dimensional_hyperoctahedron
- k_dimensional_hyperoctahedron(k) = k_cross_polytope(k)
- four_dimensional_mgonal_pyramidal(m) = mgonal_pyramidal_of_the_second_order(m)
- four_dimensional_square_pyramidal
- four_dimensional_pentagonal_pyramidal
- four_dimensional_hexagonal_pyramidal
- four_dimensional_heptagonal_pyramidal
- four_dimensional_octagonal_pyramidal
- four_dimensional_nonagonal_pyramidal
- four_dimensional_decagonal_pyramidal
- four_dimensional_hendecagonal_pyramidal
- four_dimensional_dodecagonal_pyramidal
- k_dimensional_mgonal_pyramidal(k,m) = mgonal_pyramidal_of_the_k_2_th_order(k,m)
- five_dimensional_mgonal_pyramidal(m)
- five_dimensional_square_pyramidal
- five_dimensional_pentagonal_pyramidal
- five_dimensional_hexagonal_pyramidal
- five_dimensional_heptagonal_pyramidal
- five_dimensional_octagonal_pyramidal

Multidimensional Figurate Numbers

- six_dimensional_mgonal_pyramidal(m)
- six_dimensional_square_pyramidal
- six_dimensional_pentagonal_pyramidal
- six_dimensional_hexagonal_pyramidal
- six_dimensional_heptagonal_pyramidal
- six_dimensional_octagonal_pyramidal
- centered_biquadratic
- k_dimensional_centered_hypercube(k)
- five_dimensional_centered_hypercube
- six_dimensional_centered_hypercube
- centered_polytope
- k_dimensional_centered_hypertetrahedron(k)
- five_dimensional_centered_hypertetrahedron
- six_dimensional_centered_hypertetrahedron
- centered_hyperoctahedral
- nexus(k)
- k_dimensional_centered_hyperoctahedron(k)
- five_dimensional_centered_hyperoctahedron
- six_dimensional_centered_hyperoctahedron
- generalized_pentatope(start_num = 0)
- generalized_k_dimensional_hypertetrahedron(k = 5, start_num = 0)
- generalized_biquadratic(start_num = 0)
- generalized_k_dimensional_hypercube(k = 5, start_num = 0)
- generalized_hyperoctahedral(start_num = 0)
- generalized_k_dimensional_hyperoctahedron(k = 5, start_num = 0)
- generalized_hyperdodecahedral(start_num = 0)
- generalized_hypericosahedral(start_num = 0)
- generalized_polyoctahedral(start_num = 0)
- generalized_k_dimensional_mgonal_pyramidal(k, m, start_num = 0)
- generalized_k_dimensional_centered_hypercube(k, start_num = 0)
- generalized_nexus(k, start_num = 0)

Zoo Figurate Numbers

- cuban_prime
- pell