

```

from swarmlib import CuckooProblem, FUNCTIONS

problem = CuckooProblem(function=FUNCTIONS['michalewicz'], nests=14)
best_nest = problem.solve()
problem.replay()

```

```

C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\swarmlib\cuckoosearch\visualizer.py:16: MatplotlibDeprecationWarning:
Calling gca() with keyword arguments was deprecated in Matplotlib 3.4. Starting
two minor releases later, gca() will take no keyword arguments. The gca()
function should only be used to get the current axes, or if no axes exist,
create new axes with default keyword arguments. To create a new axes with non-
default arguments, use plt.axes() or plt.subplot().
    ax = self._fig.gca(label='BaseAxis')

```

```

-----

TypeError                                 Traceback (most recent call last)

File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\IPython\core\formatters.py:339, in BaseFormatter.__call__(self, obj)
    337     pass
    338 else:
--> 339     return printer(obj)
    340 # Finally look for special method names
    341 method = get_real_method(obj, self.print_method)

```

```

File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\IPython\core\pylabtools.py:151, in print_figure(fig, fmt, bbox_inches,
base64, **kwargs)
    148     from matplotlib.backend_bases import FigureCanvasBase
    149     FigureCanvasBase(fig)
--> 151 fig.canvas.print_figure(bytes_io, **kw)
    152 data = bytes_io.getvalue()
    153 if fmt == 'svg':

```

```

File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\backend_bases.py:2230, in
FigureCanvasBase.print_figure(self, filename, dpi, facecolor, edgecolor,
orientation, format, bbox_inches, pad_inches, bbox_extra_artists, backend,
**kwargs)
    2226     ctx = (renderer._draw_disabled()
    2227             if hasattr(renderer, '_draw_disabled')
    2228             else suppress())
    2229     with ctx:
-> 2230         self.figure.draw(renderer)
    2232 if bbox_inches:
    2233     if bbox_inches == "tight":

```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\artist.py:74, in _finalize_rasterization.
<locals>.draw_wrapper(artist, renderer, *args, **kwargs)
    72 @wraps(draw)
    73 def draw_wrapper(artist, renderer, *args, **kwargs):
---> 74     result = draw(artist, renderer, *args, **kwargs)
    75     if renderer._rasterizing:
    76         renderer.stop_rasterizing()
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\artist.py:51, in allow_rasterization.
<locals>.draw_wrapper(artist, renderer, *args, **kwargs)
    48     if artist.get_agg_filter() is not None:
    49         renderer.start_filter()
---> 51     return draw(artist, renderer, *args, **kwargs)
    52 finally:
    53     if artist.get_agg_filter() is not None:
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\figure.py:2734, in Figure.draw(self, renderer)
    2731         # ValueError can occur when resizing a window.
    2733 self.patch.draw(renderer)
-> 2734 mimage._draw_list_compositing_images(
    2735     renderer, self, artists, self.suppressComposite)
    2737 for sfig in self.subfigs:
    2738     sfig.draw(renderer)
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\image.py:132, in _draw_list_compositing_images(renderer,
parent, artists, suppress_composite)
    130 if not_composite or not has_images:
    131     for a in artists:
--> 132         a.draw(renderer)
    133 else:
    134     # Composite any adjacent images together
    135     image_group = []
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\artist.py:51, in allow_rasterization.
<locals>.draw_wrapper(artist, renderer, *args, **kwargs)
    48     if artist.get_agg_filter() is not None:
    49         renderer.start_filter()
---> 51     return draw(artist, renderer, *args, **kwargs)
    52 finally:
    53     if artist.get_agg_filter() is not None:
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\_api\deprecation.py:431, in delete_parameter.
<locals>.wrapper(*inner_args, **inner_kwargs)
    421     deprecation_addendum = (
    422         f"If any parameter follows {name!r}, they should be passed as "
    423         f"keyword, not positionally.")
    424     warn_deprecated(
    425         since,
    426         name=repr(name),
    (...)
    429         else deprecation_addendum,
    430         **kwargs)
--> 431 return func(*inner_args, **inner_kwargs)
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\axes\_base.py:2925, in _AxesBase.draw(self, renderer,
inframe)
    2922     a.draw(renderer)
    2923     renderer.stop_rasterizing()
-> 2925 mimage._draw_list_compositing_images(renderer, self, artists)
    2927 renderer.close_group('axes')
    2928 self.stale = False
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\image.py:132, in _draw_list_compositing_images(renderer,
parent, artists, suppress_composite)
    130 if not_composite or not has_images:
    131     for a in artists:
--> 132         a.draw(renderer)
    133 else:
    134     # Composite any adjacent images together
    135     image_group = []
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\artist.py:51, in allow_rasterization.
<locals>.draw_wrapper(artist, renderer, *args, **kwargs)
    48     if artist.get_agg_filter() is not None:
    49         renderer.start_filter()
--> 51     return draw(artist, renderer, *args, **kwargs)
    52 finally:
    53     if artist.get_agg_filter() is not None:
```

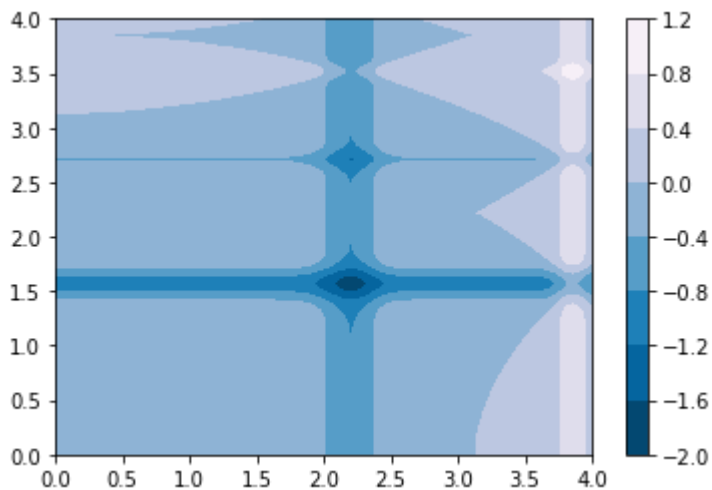
```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\quiver.py:554, in Quiver.draw(self, renderer)
    551 @matplotlib.artist.allow_rasterization
    552 def draw(self, renderer):
    553     self._init()
--> 554     verts = self._make_verts(self.U, self.V, self.angles)
    555     self.set_verts(verts, closed=False)
    556     self._new_UV = False
```

```
File C:\ProgramData\Anaconda3\envs\d21\lib\site-
packages\matplotlib\quiver.py:641, in Quiver._make_verts(self, U, V, angles)
640 def _make_verts(self, U, V, angles):
--> 641     uv = (U + V * 1j)
642     str_angles = angles if isinstance(angles, str) else ''
643     if str_angles == 'xy' and self.scale_units == 'xy':
644         # Here eps is 1 so that if we get U, V by diffing
645         # the X, Y arrays, the vectors will connect the
646         # points, regardless of the axis scaling (including log).
```

TypeError: can't multiply sequence by non-int of type 'complex'

<Figure size 432x288 with 2 Axes>

```
problem = CuckooProblem(function=FUNCTIONS['michalewicz'], nests=14)
```



```
best_nest = problem.solve()
```

```
problem.replay()
```