Homework 4

Anushka Menon and Yashavee Goel

```
In [229… %matplotlib inline

import numpy as np
import matplotlib.pyplot as plt
import scipy.special as sp
import math
```

```
In [230… m_pi = np.array([0.176, 0.234, 0.260, 0.284, 0.324])
f = np.array([960, 1025, 1055, 1085, 1130])
delta_f = np.array([25, 20, 15, 10, 8])

def func(m_pi, Phi):
    return Phi[0]+Phi[1]*m_pi+Phi[2]*m_pi**2

def chi_sq(f, delta_f):
    for i in range(5):
        return np.sum(0.5*(f[i]-func(m_pi, Phi))/delta_f[i])
```

```
In [235… def P_dot(Phi, m_pi, f, delta_f):
    p_dot = []
    for i in range(3):
        sum = 0
        for j in range(5):
            sum += ((beta*(f[j]-func(m_pi, Phi))*m_pi[j]**i)/delta_f[j])
        p_dot.append(sum)
    return np.array(p_dot)


def leapfrog(N_md, P_0, Phi_0, J, beta, h, N):
    eps = 1/N_md
    P = P_0.astype("float")
    Phi = Phi_0.astype("float")
    Phi += eps/2*Phi
    for i in range(N_md-1):
        P += eps*P_dot(Phi, m_pi, f, delta_f)
        Phi += eps*P
    P -= eps*P_dot(Phi, m_pi, f, delta_f)
    Phi += eps/2*Phi
    return P, Phi

def Hamiltonian(P, Phi, J, beta, h, N):
    return P[0]**2/2+P[1]**2/2+P[2]**2/2+chi_sq(f, delta_f)
```

```
In [236… P_0 = np.array([1,2,3])
Phi_0 = np.array([1,2,3])
beta = 1000
N_md = 20
N_cfg = 10000
h = 0.5
N = 20
J = 1/N
Phi = np.array([800,800,600])
```

```
In [237… diff = []
```

```
range_1 = range(1,100)

for N_md in range_1:
    arr = leapfrog(N_md, P_0, Phi_0, J, beta, h, N)
    P_prime = arr[0]          # We need final values of p_f and phi_f to ca
    Phi_prime = arr[1]
    H_f = Hamiltonian(P_0, Phi_0, J, beta, h, N)
    H_0 = Hamiltonian(P_prime, Phi_prime, J, beta, h, N)

    #    The following is the quantity we want on y-axis
    diff.append(abs((H_f-H_0)/H_0))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In [237], line 5
      2 range_1 = range(1,100)
      4 for N_md in range_1:
----> 5     arr = leapfrog(N_md, P_0, Phi_0, J, beta, h, N)
      6     P_prime = arr[0]          # We need final values of p_f and phi
_f to calculate H_f
      7     Phi_prime = arr[1]

Cell In [235], line 19, in leapfrog(N_md, P_0, Phi_0, J, beta, h, N)
     17     P += eps*P_dot(Phi)
     18     Phi += eps*P
---> 19 P -= eps*P_dot(Phi)
     20 Phi += eps/2*Phi
     21 return P, Phi

ValueError: operands could not be broadcast together with shapes (3,) (3,5)
(3,)
```

```
plt.figure(figsize=(12,10))
plt.yscale('log')
plt.grid()
plt.xlabel("$N_{\md}$")
plt.ylabel("$H_{\f}$ - $H_{\o}$ / $H_{\o}$")
plt.title("Convergence of leap frog integrator", size =12)
plt.plot(range_1, diff,'o')
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In [234], line 7
      5 plt.ylabel("$H_{\f}$ - $H_{\o}$ / $H_{\o}$")
      6 plt.title("Convergence of leap frog integrator", size =12)
----> 7 plt.plot(range_1, diff,'o')
      8 plt.show()

File ~/.local/lib/python3.8/site-packages/matplotlib/pyplot.py:2730, in plo
t(scalex, scaley, data, *args, **kwargs)
   2728 @_copy_docstring_and_deprecators(Axes.plot)
   2729 def plot(*args, scalex=True, scaley=True, data=None, **kwargs):
-> 2730     return gca().plot(
   2731         *args, scalex=scalex, scaley=scaley,
   2732         **({"data": data} if data is not None else {}), **kwargs)

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_axes.py:1662, in
 Axes.plot(self, scalex, scaley, data, *args, **kwargs)
   1419 """
   1420 Plot y versus x as lines and/or markers.
   1421
   (...)
   1659 (``'green'``) or hex strings (``'#008000'``).
   1660 """
   1661 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1662 lines = [*self._get_lines(*args, data=data, **kwargs)]
   1663 for line in lines:
   1664     self.add_line(line)

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_base.py:311, in
_process_plot_var_args.__call__(self, data, *args, **kwargs)
    309     this += args[0],
    310     args = args[1:]
--> 311 yield from self._plot_args(
    312     this, kwargs, ambiguous_fmt_datakey=ambiguous_fmt_datakey)

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_base.py:504, in
_process_plot_var_args._plot_args(self, tup, kwargs, return_kwargs, ambiguo
us_fmt_datakey)
    501     self.axes.yaxis.update_units(y)
    503 if x.shape[0] != y.shape[0]:
--> 504     raise ValueError(f"x and y must have same first dimension, but
"
    505                       f"have shapes {x.shape} and {y.shape}")
    506 if x.ndim > 2 or y.ndim > 2:
    507     raise ValueError(f"x and y can be no greater than 2D, but have
"
    508                       f"shapes {x.shape} and {y.shape}")

ValueError: x and y must have same first dimension, but have shapes (99,) a
nd (0,)
Error in callback <function _draw_all_if_interactive at 0x7f3738102670> (fo
r post_execute):
```

```
---------------------------------------------------------------------------
ParseException                            Traceback (most recent call last)
ParseException: Expected {accent | symbol | function | operatorname | group
| frac | dfrac | binom | genfrac | overset | underset | sqrt | overline}, f
ound '\x0c'  (at char 3), (line:1, col:4)


During handling of the above exception, another exception occurred:

ParseSyntaxException                      Traceback (most recent call last)
File ~/.local/lib/python3.8/site-packages/matplotlib/_mathtext.py:1983, in
Parser.parse(self, s, fonts_object, fontsize, dpi)
   1982 try:
-> 1983     result = self._expression.parseString(s)
   1984 except ParseBaseException as err:


File ~/.local/lib/python3.8/site-packages/pyparsing/core.py:1141, in Parser
Element.parse_string(self, instring, parse_all, parseAll)
   1139     else:
   1140         # catch and re-raise exception from here, clearing out pypa
rsing internal stack trace
-> 1141         raise exc.with_traceback(None)
   1142 else:

ParseSyntaxException: Expected {accent | symbol | function | operatorname |
group | frac | dfrac | binom | genfrac | overset | underset | sqrt | overli
ne}, found '\x0c'  (at char 3), (line:1, col:4)

The above exception was the direct cause of the following exception:

ValueError                                Traceback (most recent call last)
File ~/.local/lib/python3.8/site-packages/matplotlib/pyplot.py:119, in _dra
w_all_if_interactive()
    117 def _draw_all_if_interactive():
    118     if matplotlib.is_interactive():
--> 119         draw_all()


File ~/.local/lib/python3.8/site-packages/matplotlib/_pylab_helpers.py:132,
in Gcf.draw_all(cls, force)
    130 for manager in cls.get_all_fig_managers():
    131     if force or manager.canvas.figure.stale:
--> 132         manager.canvas.draw_idle()


File ~/.local/lib/python3.8/site-packages/matplotlib/backend_bases.py:2054,
in FigureCanvasBase.draw_idle(self, *args, **kwargs)
   2052 if not self._is_idle_drawing:
   2053     with self._idle_draw_cntx():
-> 2054         self.draw(*args, **kwargs)


File ~/.local/lib/python3.8/site-packages/matplotlib/backends/backend_agg.p
y:408, in FigureCanvasAgg.draw(self)
    404 # Acquire a lock on the shared font cache.
    405 with RendererAgg.lock, \
    406      (self.toolbar._wait_cursor_for_draw_cm() if self.toolbar
    407       else nullcontext()):
--> 408         self.figure.draw(self.renderer)
    409     # A GUI class may be need to update a window using this draw, s
o
    410     # don't forget to call the superclass.
    411     super().draw()


File ~/.local/lib/python3.8/site-packages/matplotlib/artist.py:74, in _fina
```

```
lize_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 ~/.local/lib/python3.8/site-packages/matplotlib/artist.py:51, in allow
_rasterization.<locals>.draw_wrapper(artist, renderer)
     48     if artist.get_agg_filter() is not None:
     49         renderer.start_filter()
---> 51     return draw(artist, renderer)
     52 finally:
     53     if artist.get_agg_filter() is not None:

File ~/.local/lib/python3.8/site-packages/matplotlib/figure.py:3074, in Fig
ure.draw(self, renderer)
   3071         # ValueError can occur when resizing a window.
   3073 self.patch.draw(renderer)
-> 3074 mimage._draw_list_compositing_images(
   3075     renderer, self, artists, self.suppressComposite)
   3077 for sfig in self.subfigs:
   3078     sfig.draw(renderer)

File ~/.local/lib/python3.8/site-packages/matplotlib/image.py:131, in _draw
_list_compositing_images(renderer, parent, artists, suppress_composite)
    129 if not_composite or not has_images:
    130     for a in artists:
--> 131         a.draw(renderer)
    132 else:
    133     # Composite any adjacent images together
    134     image_group = []

File ~/.local/lib/python3.8/site-packages/matplotlib/artist.py:51, in allow
_rasterization.<locals>.draw_wrapper(artist, renderer)
     48     if artist.get_agg_filter() is not None:
     49         renderer.start_filter()
---> 51     return draw(artist, renderer)
     52 finally:
     53     if artist.get_agg_filter() is not None:

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_base.py:3071, in
_AxesBase.draw(self, renderer)
   3068     for spine in self.spines.values():
   3069         artists.remove(spine)
-> 3071 self._update_title_position(renderer)
   3073 if not self.axison:
   3074     for _axis in self._axis_map.values():

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_base.py:3015, in
_AxesBase._update_title_position(self, renderer)
   3013 top = max(top, bb.ymax)
   3014 if title.get_text():
-> 3015     ax.yaxis.get_tightbbox(renderer)  # update offsetText
   3016     if ax.yaxis.offsetText.get_text():
   3017         bb = ax.yaxis.offsetText.get_tightbbox(renderer)

File ~/.local/lib/python3.8/site-packages/matplotlib/axis.py:1269, in Axis.
get_tightbbox(self, renderer, for_layout_only)
   1267 # take care of label
   1268 if self.label.get_visible():
```

```
-> 1269        bb = self.label.get_window_extent(renderer)
   1270        # for constrained/tight_layout, we want to ignore the label's
   1271        # width/height because the adjustments they make can't be impro
ved.
   1272        # this code collapses the relevant direction
   1273        if for_layout_only:
```

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:915, in Text.g
et_window_extent(self, renderer, dpi)

```
    910        raise RuntimeError(
    911            "Cannot get window extent of text w/o renderer. You likely
"
    912            "want to call 'figure.draw_without_rendering()' first.")
    914 with cbook._setattr_cm(self.figure, dpi=dpi):
--> 915     bbox, info, descent = self._get_layout(self._renderer)
    916     x, y = self.get_unitless_position()
    917     x, y = self.get_transform().transform((x, y))
```

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:321, in Text._
get_layout(self, renderer)

```
    319 clean_line, ismath = self._preprocess_math(line)
    320 if clean_line:
--> 321     w, h, d = _get_text_metrics_with_cache(
    322             renderer, clean_line, self._fontproperties,
    323             ismath=ismath, dpi=self.figure.dpi)
    324 else:
    325     w = h = d = 0
```

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:97, in _get_te
xt_metrics_with_cache(renderer, text, fontprop, ismath, dpi)

```
    94 """Call ``renderer.get_text_width_height_descent``, caching the res
ults."""
    95 # Cached based on a copy of fontprop so that later in-place mutatio
ns of
    96 # the passed-in argument do not mess up the cache.
---> 97 return _get_text_metrics_with_cache_impl(
    98     weakref.ref(renderer), text, fontprop.copy(), ismath, dpi)
```

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:105, in _get_t
ext_metrics_with_cache_impl(renderer_ref, text, fontprop, ismath, dpi)

```
    101 @functools.lru_cache(4096)
    102 def _get_text_metrics_with_cache_impl(
    103         renderer_ref, text, fontprop, ismath, dpi):
    104     # dpi is unused, but participates in cache invalidation (via th
e renderer).
--> 105     return renderer_ref().get_text_width_height_descent(text, fontp
rop, ismath)
```

File ~/.local/lib/python3.8/site-packages/matplotlib/backends/backend_agg.p
y:238, in RendererAgg.get_text_width_height_descent(self, s, prop, ismath)

```
    234     return w, h, d
    236 if ismath:
    237     ox, oy, width, height, descent, font_image = \
--> 238         self.mathtext_parser.parse(s, self.dpi, prop)
    239     return width, height, descent
    241 font = self._prepare_font(prop)
```

File ~/.local/lib/python3.8/site-packages/matplotlib/mathtext.py:226, in Ma
thTextParser.parse(self, s, dpi, prop)

```
    222 # lru_cache can't decorate parse() directly because prop
    223 # is mutable; key the cache using an internal copy (see
```

```
   224 # text._get_text_metrics_with_cache for a similar case).
   225 prop = prop.copy() if prop is not None else None
-->226 return self._parse_cached(s, dpi, prop)

File ~/.local/lib/python3.8/site-packages/matplotlib/mathtext.py:247, in Ma
thTextParser._parse_cached(self, s, dpi, prop)
   244 if self._parser is None:  # Cache the parser globally.
   245     self.__class__._parser = _mathtext.Parser()
-->247 box = self._parser.parse(s, fontset, fontsize, dpi)
   248 output = _mathtext.ship(box)
   249 if self._output_type == "vector":

File ~/.local/lib/python3.8/site-packages/matplotlib/_mathtext.py:1985, in
Parser.parse(self, s, fonts_object, fontsize, dpi)
   1983     result = self._expression.parseString(s)
   1984 except ParseBaseException as err:
->1985     raise ValueError("\n".join(["",
   1986                                 err.line,
   1987                                 " " * (err.column - 1) + "^",
   1988                                 str(err)])) from err
   1989 self._state_stack = None
   1990 self._in_subscript_or_superscript = False

ValueError:
H_{}
   ^
Expected {accent | symbol | function | operatorname | group | frac | dfrac
 | binom | genfrac | overset | underset | sqrt | overline}, found '\x0c'
 (at char 3), (line:1, col:4)
```

```
---------------------------------------------------------------------------
ParseException                               Traceback (most recent call last)
ParseException: Expected {accent | symbol | function | operatorname | group
| frac | dfrac | binom | genfrac | overset | underset | sqrt | overline}, f
ound '\x0c'  (at char 3), (line:1, col:4)

During handling of the above exception, another exception occurred:

ParseSyntaxException                         Traceback (most recent call last)
File ~/.local/lib/python3.8/site-packages/matplotlib/_mathtext.py:1983, in
Parser.parse(self, s, fonts_object, fontsize, dpi)
   1982 try:
-> 1983     result = self._expression.parseString(s)
   1984 except ParseBaseException as err:

File ~/.local/lib/python3.8/site-packages/pyparsing/core.py:1141, in Parser
Element.parse_string(self, instring, parse_all, parseAll)
   1139     else:
   1140         # catch and re-raise exception from here, clearing out pypa
rsing internal stack trace
-> 1141         raise exc.with_traceback(None)
   1142 else:

ParseSyntaxException: Expected {accent | symbol | function | operatorname |
group | frac | dfrac | binom | genfrac | overset | underset | sqrt | overli
ne}, found '\x0c'  (at char 3), (line:1, col:4)

The above exception was the direct cause of the following exception:

ValueError                                   Traceback (most recent call last)
File ~/.local/lib/python3.8/site-packages/IPython/core/formatters.py:339, i
n 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 ~/.local/lib/python3.8/site-packages/IPython/core/pylabtools.py:151, i
n 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 ~/.local/lib/python3.8/site-packages/matplotlib/backend_bases.py:2314,
in FigureCanvasBase.print_figure(self, filename, dpi, facecolor, edgecolor,
orientation, format, bbox_inches, pad_inches, bbox_extra_artists, backend,
 **kwargs)
   2308     renderer = _get_renderer(
   2309         self.figure,
   2310         functools.partial(
   2311             print_method, orientation=orientation)
   2312     )
   2313     with getattr(renderer, "_draw_disabled", nullcontext)():
-> 2314         self.figure.draw(renderer)
   2316 if bbox_inches:
   2317     if bbox_inches == "tight":

File ~/.local/lib/python3.8/site-packages/matplotlib/artist.py:74, in _fina
```

```
lize_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 ~/.local/lib/python3.8/site-packages/matplotlib/artist.py:51, in allow
_rasterization.<locals>.draw_wrapper(artist, renderer)
     48     if artist.get_agg_filter() is not None:
     49         renderer.start_filter()
---> 51     return draw(artist, renderer)
     52 finally:
     53     if artist.get_agg_filter() is not None:

File ~/.local/lib/python3.8/site-packages/matplotlib/figure.py:3074, in Fig
ure.draw(self, renderer)
   3071         # ValueError can occur when resizing a window.
   3073 self.patch.draw(renderer)
-> 3074 mimage._draw_list_compositing_images(
   3075     renderer, self, artists, self.suppressComposite)
   3077 for sfig in self.subfigs:
   3078     sfig.draw(renderer)

File ~/.local/lib/python3.8/site-packages/matplotlib/image.py:131, in _draw
_list_compositing_images(renderer, parent, artists, suppress_composite)
    129 if not_composite or not has_images:
    130     for a in artists:
--> 131         a.draw(renderer)
    132 else:
    133     # Composite any adjacent images together
    134     image_group = []

File ~/.local/lib/python3.8/site-packages/matplotlib/artist.py:51, in allow
_rasterization.<locals>.draw_wrapper(artist, renderer)
     48     if artist.get_agg_filter() is not None:
     49         renderer.start_filter()
---> 51     return draw(artist, renderer)
     52 finally:
     53     if artist.get_agg_filter() is not None:

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_base.py:3071, in
_AxesBase.draw(self, renderer)
   3068     for spine in self.spines.values():
   3069         artists.remove(spine)
-> 3071 self._update_title_position(renderer)
   3073 if not self.axison:
   3074     for _axis in self._axis_map.values():

File ~/.local/lib/python3.8/site-packages/matplotlib/axes/_base.py:3015, in
_AxesBase._update_title_position(self, renderer)
   3013 top = max(top, bb.ymax)
   3014 if title.get_text():
-> 3015     ax.yaxis.get_tightbbox(renderer)  # update offsetText
   3016     if ax.yaxis.offsetText.get_text():
   3017         bb = ax.yaxis.offsetText.get_tightbbox(renderer)

File ~/.local/lib/python3.8/site-packages/matplotlib/axis.py:1269, in Axis.
get_tightbbox(self, renderer, for_layout_only)
   1267 # take care of label
   1268 if self.label.get_visible():
```

```
-> 1269        bb = self.label.get_window_extent(renderer)
   1270        # for constrained/tight_layout, we want to ignore the label's
   1271        # width/height because the adjustments they make can't be impro
ved.
   1272        # this code collapses the relevant direction
   1273        if for_layout_only:

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:915, in Text.g
et_window_extent(self, renderer, dpi)
    910        raise RuntimeError(
    911            "Cannot get window extent of text w/o renderer. You likely
"
    912            "want to call 'figure.draw_without_rendering()' first.")
    914    with cbook._setattr_cm(self.figure, dpi=dpi):
--> 915        bbox, info, descent = self._get_layout(self._renderer)
    916        x, y = self.get_unitless_position()
    917        x, y = self.get_transform().transform((x, y))

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:321, in Text._
get_layout(self, renderer)
    319 clean_line, ismath = self._preprocess_math(line)
    320 if clean_line:
--> 321     w, h, d = _get_text_metrics_with_cache(
    322            renderer, clean_line, self._fontproperties,
    323            ismath=ismath, dpi=self.figure.dpi)
    324 else:
    325     w = h = d = 0

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:97, in _get_te
xt_metrics_with_cache(renderer, text, fontprop, ismath, dpi)
     94 """Call ``renderer.get_text_width_height_descent``, caching the res
ults."""
     95 # Cached based on a copy of fontprop so that later in-place mutatio
ns of
     96 # the passed-in argument do not mess up the cache.
---> 97 return _get_text_metrics_with_cache_impl(
     98     weakref.ref(renderer), text, fontprop.copy(), ismath, dpi)

File ~/.local/lib/python3.8/site-packages/matplotlib/text.py:105, in _get_t
ext_metrics_with_cache_impl(renderer_ref, text, fontprop, ismath, dpi)
    101 @functools.lru_cache(4096)
    102 def _get_text_metrics_with_cache_impl(
    103        renderer_ref, text, fontprop, ismath, dpi):
    104    # dpi is unused, but participates in cache invalidation (via th
e renderer).
--> 105    return renderer_ref().get_text_width_height_descent(text, fontp
rop, ismath)

File ~/.local/lib/python3.8/site-packages/matplotlib/backends/backend_agg.p
y:238, in RendererAgg.get_text_width_height_descent(self, s, prop, ismath)
    234    return w, h, d
    236 if ismath:
    237    ox, oy, width, height, descent, font_image = \
--> 238        self.mathtext_parser.parse(s, self.dpi, prop)
    239    return width, height, descent
    241 font = self._prepare_font(prop)

File ~/.local/lib/python3.8/site-packages/matplotlib/mathtext.py:226, in Ma
thTextParser.parse(self, s, dpi, prop)
    222 # lru_cache can't decorate parse() directly because prop
    223 # is mutable; key the cache using an internal copy (see
```

```
    224 # text._get_text_metrics_with_cache for a similar case).
    225 prop = prop.copy() if prop is not None else None
--> 226 return self._parse_cached(s, dpi, prop)

File ~/.local/lib/python3.8/site-packages/matplotlib/mathtext.py:247, in Ma
thTextParser._parse_cached(self, s, dpi, prop)
    244 if self._parser is None:  # Cache the parser globally.
    245     self.__class__._parser = _mathtext.Parser()
--> 247 box = self._parser.parse(s, fontset, fontsize, dpi)
    248 output = _mathtext.ship(box)
    249 if self._output_type == "vector":

File ~/.local/lib/python3.8/site-packages/matplotlib/_mathtext.py:1985, in
Parser.parse(self, s, fonts_object, fontsize, dpi)
    1983     result = self._expression.parseString(s)
    1984 except ParseBaseException as err:
-> 1985     raise ValueError("\n".join([""",
    1986                                err.line,
    1987                                " " * (err.column - 1) + "^",
    1988                                str(err)])) from err
    1989 self._state_stack = None
    1990 self._in_subscript_or_superscript = False

ValueError:
H_{}
   ^
Expected {accent | symbol | function | operatorname | group | frac | dfrac
 | binom | genfrac | overset | underset | sqrt | overline}, found '\x0c'
 (at char 3), (line:1, col:4)
```
<Figure size 1200x1000 with 1 Axes>

Answer 3 : HMC Algorithm

```python
In [223… def markov_chain(Phi_i,N_cfg,N_md,beta,J,h,N):
         chain = []
         chain.append(Phi_i)
         for i in range(N_cfg):
             p = np.random.normal(0,1)  # Random sample
             new = leapfrog(N_md, P, chain[-1], J, beta, h, N) # Integrating lea|
             prob = np.exp(Hamiltonian(P,chain[-1], J, beta, h, N)-Hamiltonian(ne
     #  Given probability distribution function defined above
             if prob > np.random.uniform(0,1):   # Accept/reject
                 chain.append(new[1])
             else:
                 chain.append(chain[-1])
         return chain
```

```
In [ ]:
```

Yashasvee Goel
Anushka Menon

5. Applying HMC to multi-parameter fitting

1. $\mathcal{H}[p,\phi] = \dfrac{p_0^2}{2} + \dfrac{p_1^2}{2} + \dfrac{p_2^2}{2} + \beta \chi^2(\phi)$

$$\dot{\phi}_0 = \dfrac{\partial}{\partial p_0}\mathcal{H} = p_0 \;,\; \dot{\phi}_1 = p_1 \;,\; \dot{\phi}_2 = p_2$$

$$\chi^2(\phi) = \dfrac{1}{2}\sum_{i=1}^{5} \dfrac{(f_i - f(x_i,\phi))^2}{\delta f_i^2}$$

$$\dot{p}_0 = -\dfrac{\partial}{\partial \phi_0}\mathcal{H} = +\beta \dfrac{1}{2}\sum_{i=1}^{5} 2\,\dfrac{f_i - f(x_i,\phi)}{\delta f_i^2}$$

$$= \dfrac{\beta}{\delta f_i^2}\sum_{i=1}^{5}(f_i - f(x_i,\phi))$$

$$\dot{p}_1 = \dfrac{\beta}{\delta f_i^2}x_i \sum_{i=1}^{5}(f_i - f(x_i,\phi))$$

$$\dot{p}_2 = \dfrac{\beta}{\delta f_i^2}x_i^2 \sum_{i=1}^{5}(f_i - f(x_i,\phi))$$

1