In [18]:
```
from lgdo.lh5 import LH5Store, show
from dspeed.vis import WaveformBrowser
from dspeed import processors, build_dsp
import dspeed.processors.energy_kernels as kernels

import matplotlib.pyplot as plt
import numpy as np
```

Here I try to use the cusp filter to do energy estimation. First I will create the kernel manually using dspeed.processors, and concolve the waveform with it. Later I am trying to do the same with the build_dsp() function. Somehow that never seems to work. The output is always zero. Please help.

Loading data and inspecting waveform

In [19]:
```
entry_no = 4  # which waveform in our example data_file to use fo
r plotting
base_path = "/mnt/atlas01/projects/scarf-campaign-2023/data/raw/r
un0038/"
filename = "ugl-data-20231031-120001.lh5"

plt.rcParams["figure.figsize"] = (6, 3)
plt.rcParams["figure.facecolor"] = "white"
plt.rcParams["font.size"] = 10
```

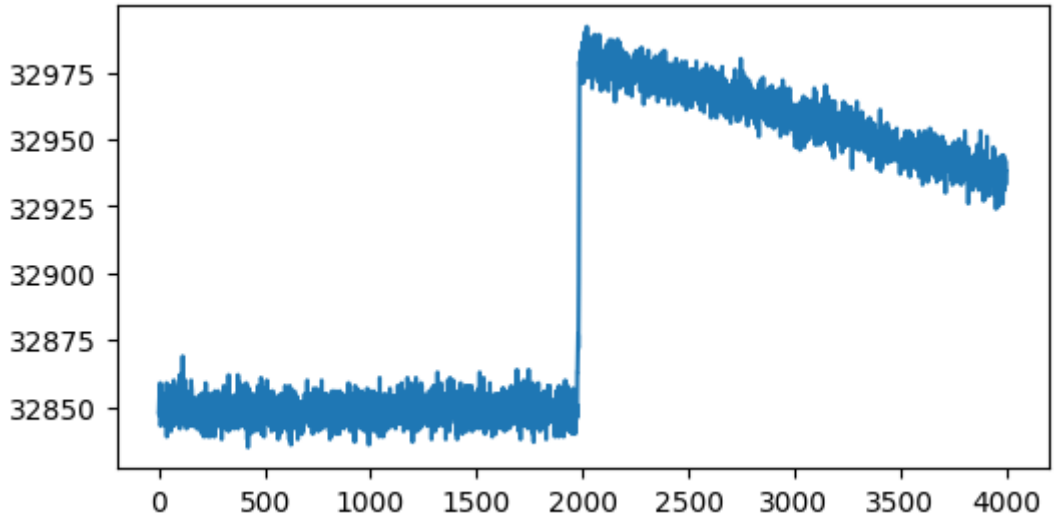
In [20]:
```
store = LH5Store()
raw_tier_obj, n_rows = store.read("ch0000/raw", base_path+filenam
e)
df = raw_tier_obj.view_as("pd")
```

In [21]:
```
wfs = np.asarray( df["wf_presum_values"] , dtype=int)
wfs
```

Out[21]:
```
array([[32848, 32849, 32852, ..., 32942, 32933, 32938],
       [32848, 32840, 32849, ..., 32844, 32850, 32852],
       [32855, 32856, 32855, ..., 32857, 32849, 32856],
       ...,
       [32848, 32847, 32848, ..., 32854, 32863, 32862],
       [32843, 32855, 32848, ..., 32846, 32853, 32838],
       [32856, 32846, 32853, ..., 32860, 32862, 32860]])
```

In [22]:
```
plt.plot(wfs[0])
```

Out[22]:
```
[<matplotlib.lines.Line2D at 0x7f3165799790>]
```

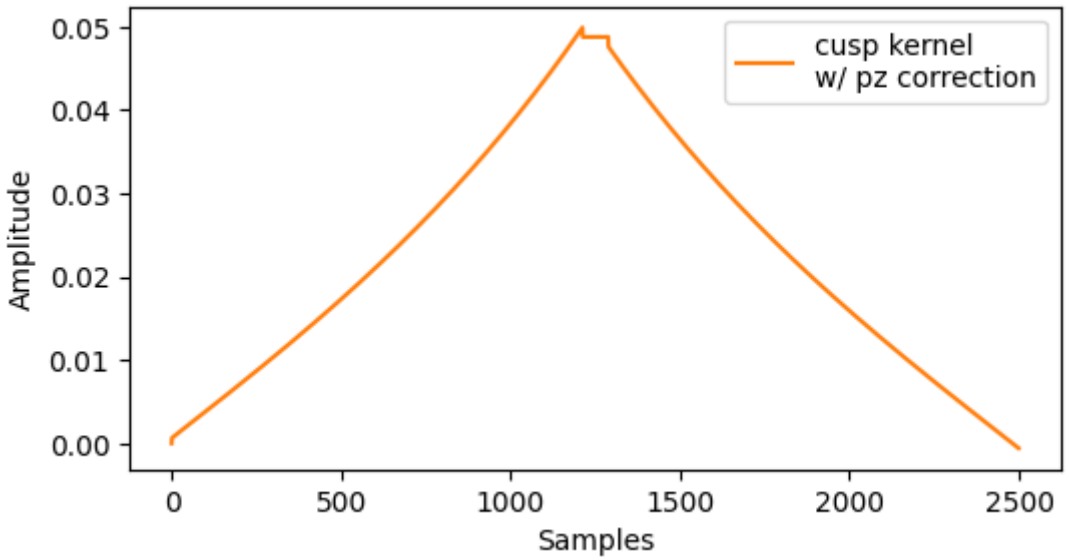


Creating and convolving the kernel manually

Let's start by setting up a kernel using some more or less random parameters.

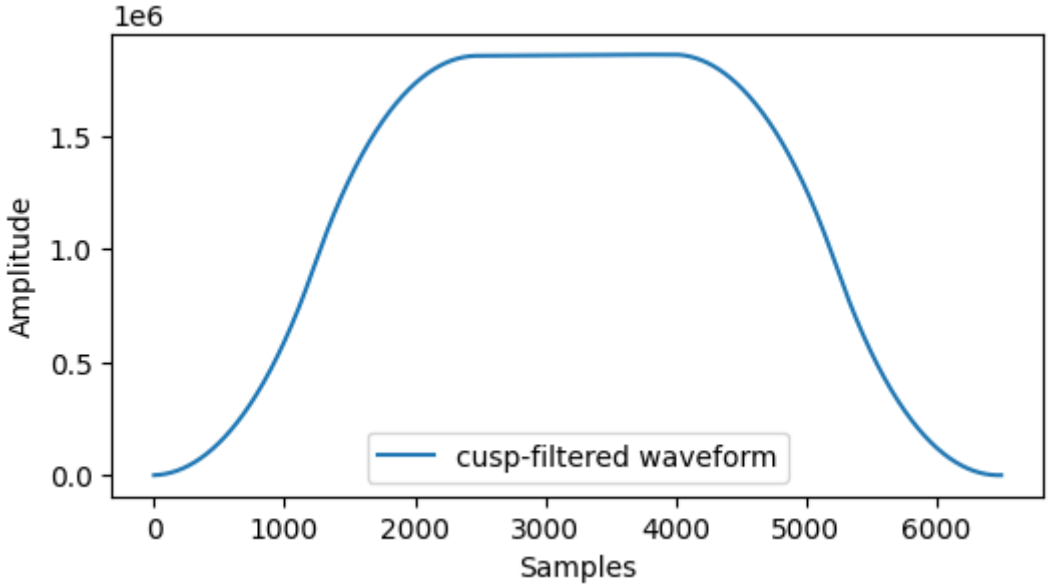
In [23]:
```
sigma = 1000
flat = int(3000 / 40)
decay = 20
length = int(100000 / 40)
cusp_kernel = np.ones(length)

kernels.cusp_filter(sigma, flat, decay, cusp_kernel)

plt.plot(cusp_kernel, label="cusp kernel \nw/ pz correction", col
or="tab:orange")
plt.xlabel("Samples")
plt.ylabel("Amplitude")
plt.legend()
```

Out[23]:
```
<matplotlib.legend.Legend at 0x7f316581c090>
```



In [24]:
```
wf_cusp = np.empty(6499)
processors.convolve_wf(wfs[0], cusp_kernel, ord("f"), wf_cusp)
plt.plot(wf_cusp, label="cusp-filtered waveform")
plt.xlabel("Samples")
plt.ylabel("Amplitude")
plt.legend()
```

Out[24]:
```
<matplotlib.legend.Legend at 0x7f317fbe0510>
```



Using build_dsp()

Now let's try to do *exactly the same* with build_dsp(). Define a dsp_config dict with the same parameters as above.

In [25]:
```
dsp_config = {
    "outputs": [
        "wf_presum",
        "wf_cusp",
    ],


    "processors": {
        "cusp_kernel": {
            "function": "cusp_filter",
            "module": "dspeed.processors",
            "args": [
                "1000", "75", "20",
                "cusp_kernel(2500, 'f')"
            ],
            "unit": "ADC"
        },

        "wf_cusp": {
            "description": "convolve optimised cusp filter",
            "function": "convolve_wf",
            "module": "dspeed.processors",
            "args": [
                "wf_presum",
                "cusp_kernel",
                "'f'",
                "wf_cusp(6499, 'f')"
            ],
            "unit": "ADC"
        },
    }
}
```

And run the dsp.

In [26]:
```
build_dsp(base_path+filename, "cusp_fun_file_tier_dsp.lh5", dsp_c
onfig=dsp_config, n_max=10, write_mode="r")
```

It worked! A file has been created, and we can look at its contents.

In [27]:
```
show("cusp_fun_file_tier_dsp.lh5")
```

```
/
├── ch0000 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0001 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0002 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0003 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0004 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0005 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0006 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0007 · HDF5 group
│   └── dsp · table{wf_presum,wf_cusp}
│       ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
│       └── wf_presum · table{t0,dt,values}
│           ├── dt · array<1>{real}
│           ├── t0 · array<1>{real}
│           └── values · array_of_equalsized_arrays<1,1>{real}
├── ch0008 · HDF5 group
```

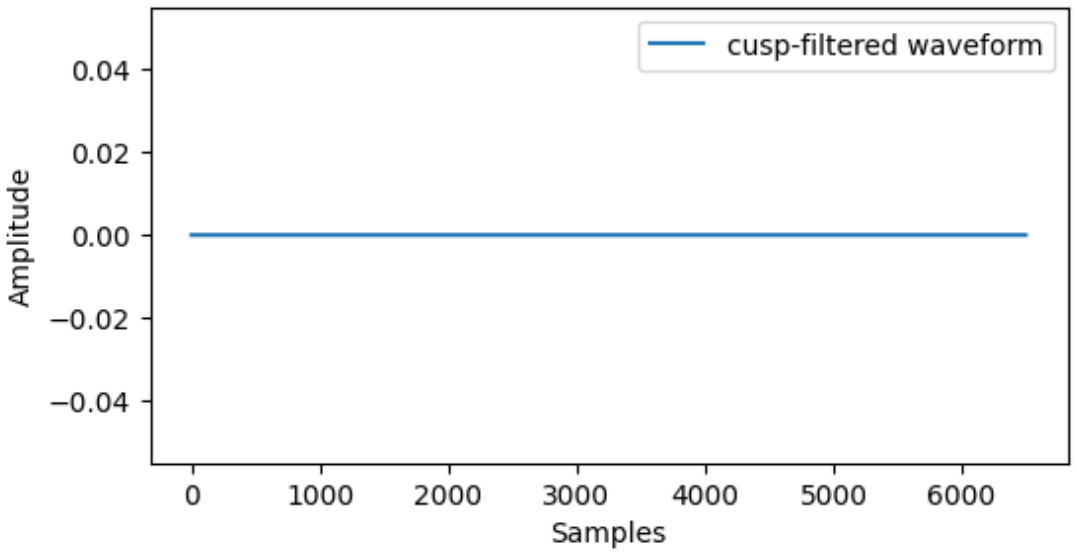
```
    └── dsp · table{wf_presum,wf_cusp}
        ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
        └── wf_presum · table{t0,dt,values}
            ├── dt · array<1>{real}
            ├── t0 · array<1>{real}
            └── values · array_of_equalsized_arrays<1,1>{real}
└── ch0009 · HDF5 group
    └── dsp · table{wf_presum,wf_cusp}
        ├── wf_cusp · array_of_equalsized_arrays<1,1>{real}
        └── wf_presum · table{t0,dt,values}
            ├── dt · array<1>{real}
            ├── t0 · array<1>{real}
            └── values · array_of_equalsized_arrays<1,1>{real}
```

In [28]:
```
store = LH5Store()
dsp_tier_obj, nrows = store.read("ch0000/dsp", "cusp_fun_file_tie
r_dsp.lh5")
```

In [29]:
```
wf_cusp = np.asarray(dsp_tier_obj.view_as("pd")["wf_cusp"])
print(wf_cusp[0])

plt.plot(wf_cusp[0], label="cusp-filtered waveform")
plt.xlabel("Samples")
plt.ylabel("Amplitude")
plt.legend()
```

```
[0.0 0.0 0.0 ... 0.0 0.0 0.0]
```

Out[29]:
```
<matplotlib.legend.Legend at 0x7f317fc36d90>
```



But wtf, the waveforms are all zero. Where did I make a mistake?!

Using run_one_dsp()

With Rosanna's help I tried it again with run_one_dsp() with the same result. So it has to be a mistake in the dsp dict, but I cannot find it!!

In [30]:
```
from pygama.pargen.dsp_optimize import run_one_dsp
```

In [31]:
```
tbl_out = run_one_dsp(raw_tier_obj, dsp_config)
```

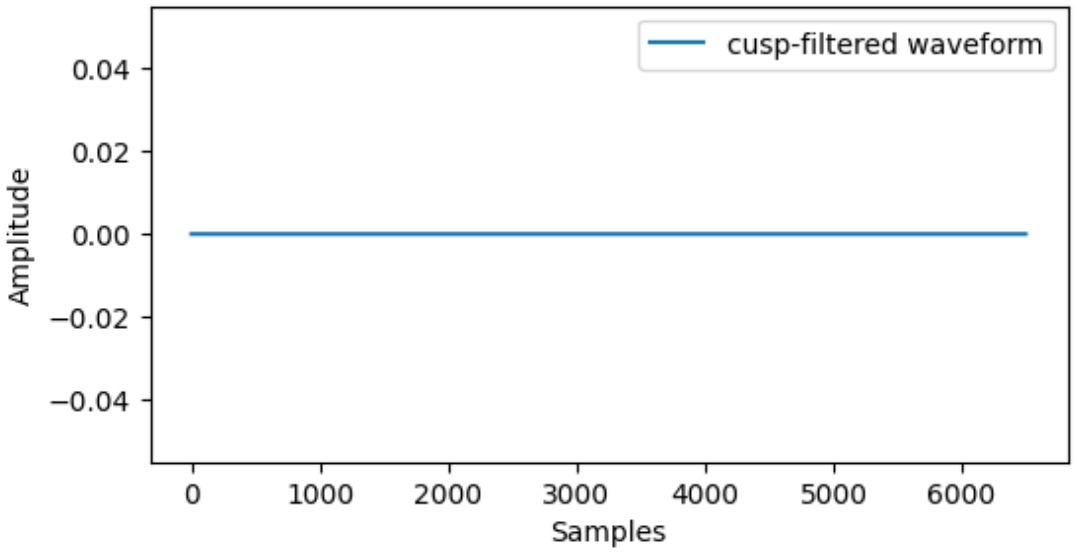
In [32]:
```
tbl_out.wf_cusp
```

Out[32]:
```
ArrayOfEqualSizedArrays([[0. 0. 0. ... 0. 0. 0.]
                         [0. 0. 0. ... 0. 0. 0.]
                         [0. 0. 0. ... 0. 0. 0.]
                         ...
                         [0. 0. 0. ... 0. 0. 0.]
                         [0. 0. 0. ... 0. 0. 0.]
                         [0. 0. 0. ... 0. 0. 0.]], attrs={'dataty
pe': 'array_of_equalsized_arrays<1,1>{real}', 'units': 'ADC'})
```

In [33]:
```
print(tbl_out.wf_cusp[0])

plt.plot(tbl_out.wf_cusp[0], label="cusp-filtered waveform")
plt.xlabel("Samples")
plt.ylabel("Amplitude")
plt.legend()
```

```
[0. 0. 0. ... 0. 0. 0.]
```

Out[33]:
```
<matplotlib.legend.Legend at 0x7f318b887ed0>
```



Same problem here. Please help ...