CIRCE2: Guinea-Pig for the Masses

Thorsten Ohl

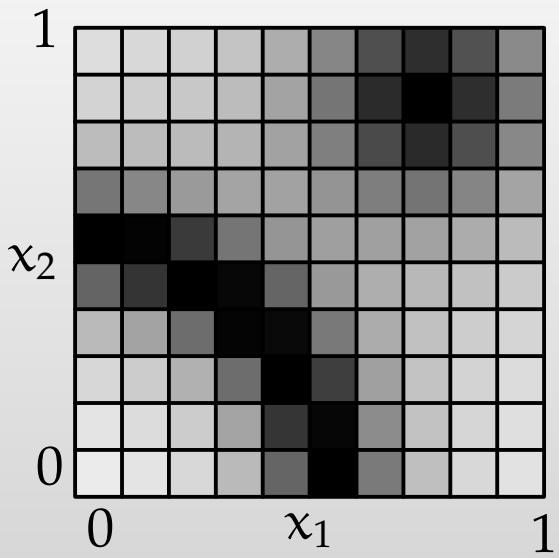
http://physik.uni-wuerzburg.de/ohl

Institute for Theoretical Physics and Astrophysics
Würzburg University

ECFA Higgs Factories
1st Focus Meeting on Beamstrahlung
January 12, 2022

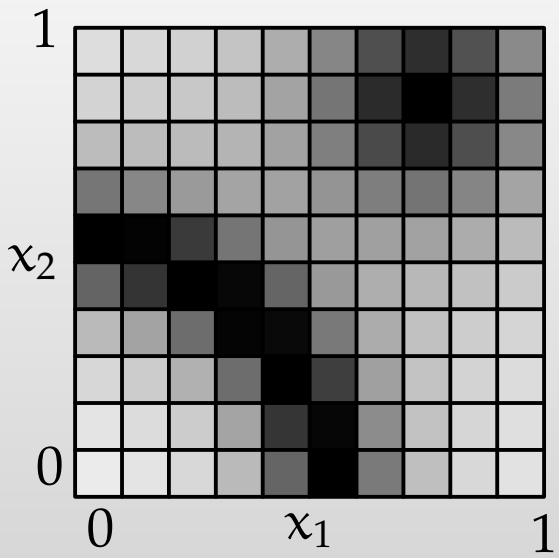
Adaptive Grids

- A fixed grid with variable weights can not adapt to singular integrands:

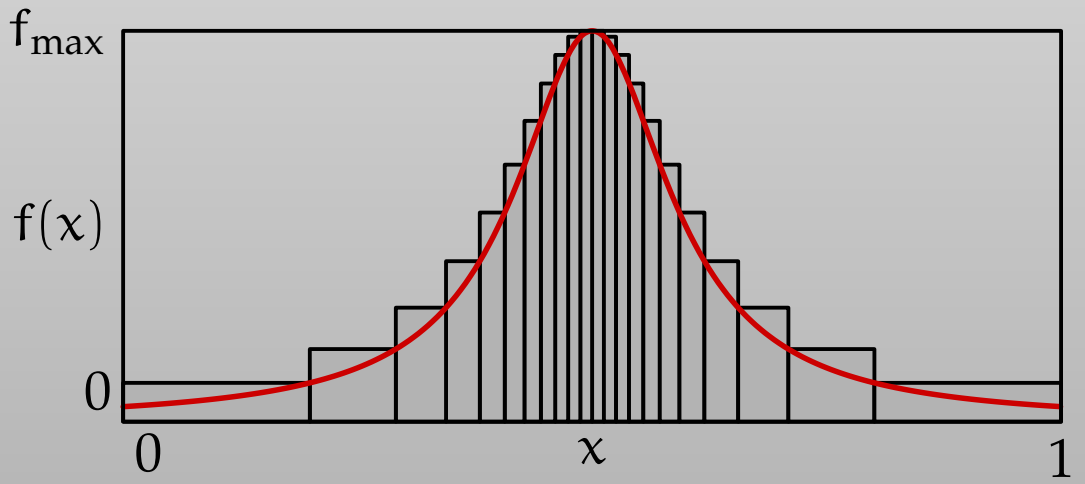


Adaptive Grids

- A fixed grid with variable weights can not adapt to singular integrands:

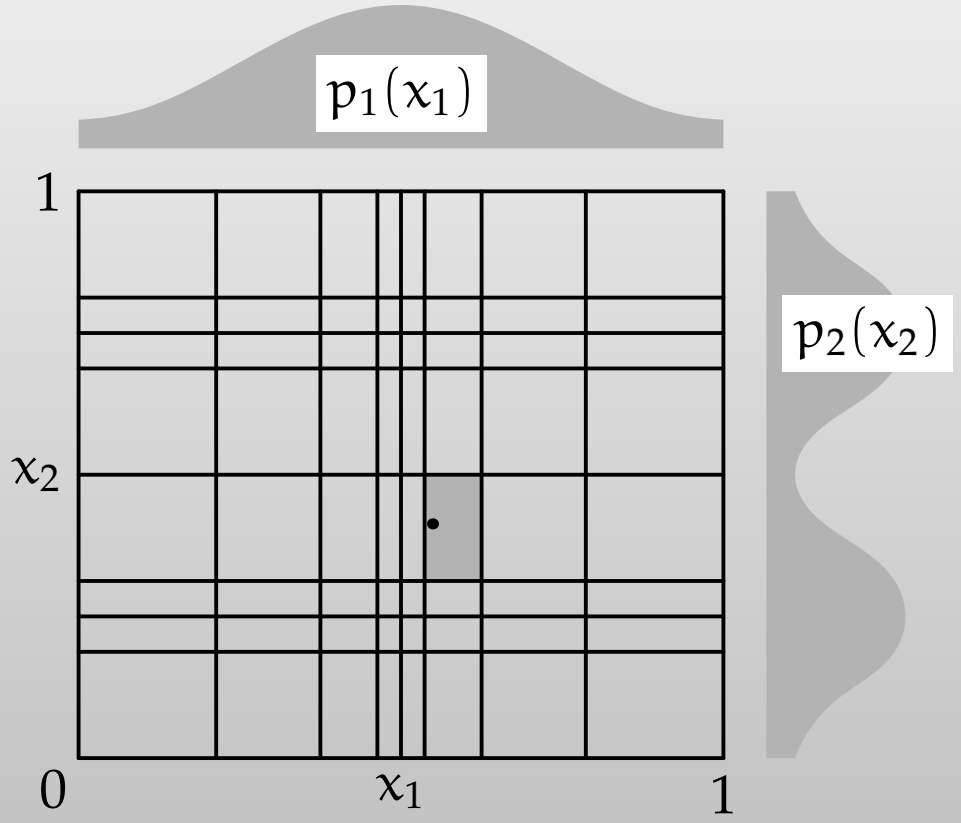


- In one dimension, a variable grid with fixed weights can adapt well to singular integrands.



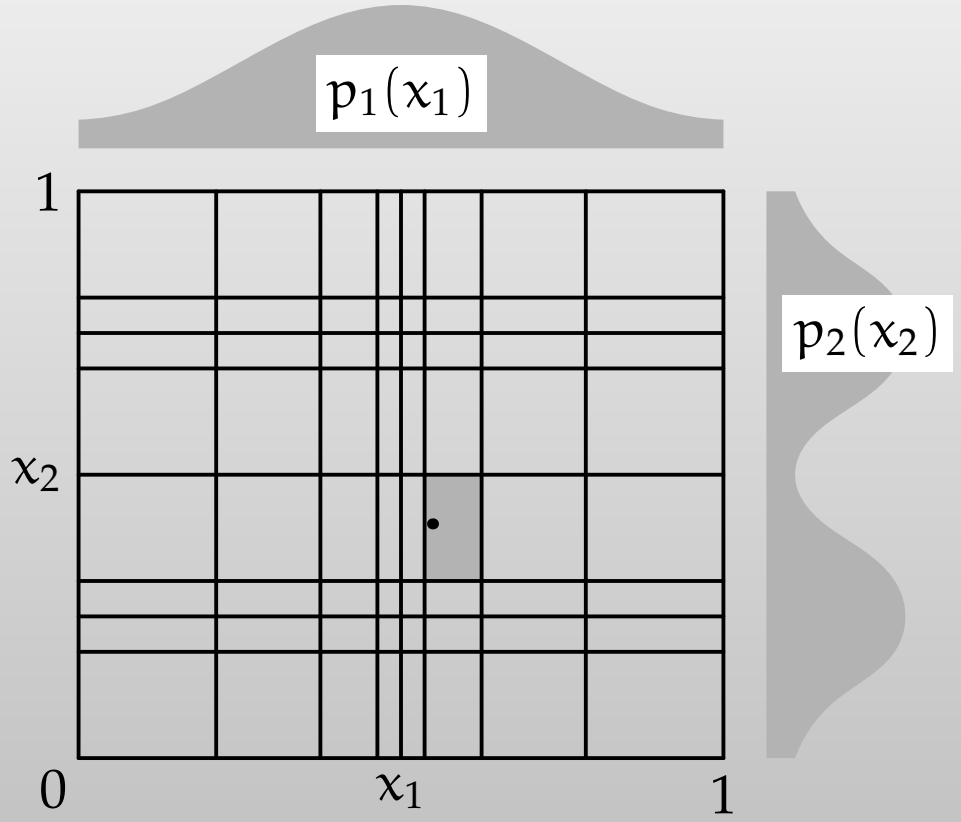
Adaptive Grids

- factorizable singularities can also be described by a variable grid with fixed weights



Adaptive Grids

- factorizable singularities can also be described by a variable grid with fixed weights



- the remaining nonsingular nonfactorizable contributions can be handled by a variable weights on top of variable grid

From CIRCE2 to WHIZARD et al.

- read `TDR.circe` and generate 1000000 (x_1, x_2) pairs for unpolarized electron-positron pairs

```
program girce2
  type(circe2_state) :: c2s
  type(rng_t) :: rng
  integer :: i, ierror
  real(kind=default), dimension(2) :: x
  call circe2_load (c2s, "TDR.circe", "ILC", 500.0_default, ierror)
  do i = 1, 1000000
     call circe2_generate (c2s, rng, x, [11, -11], [0, 0])
     print *, x, 1.0_default
  end do
end program girce2
```

From CIRCE2 to WHIZARD et al.

- read `TDR.circe` and generate 1000000 (x_1, x_2) pairs for unpolarized electron-positron pairs

```
program girce2
  type(circe2_state) :: c2s
  type(rng_t) :: rng
  integer :: i, ierror
  real(kind=default), dimension(2) :: x
  call circe2_load (c2s, "TDR.circe", "ILC", 500.0_default, ierror)
  do i = 1, 1000000
     call circe2_generate (c2s, rng, x, [11, -11], [0, 0])
     print *, x, 1.0_default
  end do
end program girce2
```

- even simpler: use it from inside WHIZARD as

```
sqrts = 500
beams = "e-", "e+" => circe2
$circe2_file = "TDR.circe"
$circe2_design = "ILC"
?circe2_polarized = false
```

From Guinea-Pig to CIRCE2

- basic example of CIRCE2 input

```
{ file = "TDR.circe"     # name of the output file
  { design = "ILC"       # there can be more than one design per file
    roots = 500          #                                 energy
    scale = 250          # map [0,250] → [0,1]
    bins = 100           # use 100 bins in each direction
   { pid/1 = electron    # first and second particle
     pid/2 = positron
     pol = 0             # both particles unpolarized
     events = "guinea_pig/out/ILC_500_unpolarized.data"
     columns = 2         # read only the first two columns
     lumi = 8.008e33
     min = 0
     max = 1.05          # allow 5% energy spread at the upper end
} } }
```

will generate a fixed width histogram with weights according to Guinea-Pig output:

```
  $ head guinea_pig/out/ILC_500_unpolarized.data
249.435 250.16 405.499 -0.67215 32.2081 193 2.31349e-05 ...
249.791 250.109 -406.506 5.4995 61.3885 267 7.91127e-06 ...
...
```

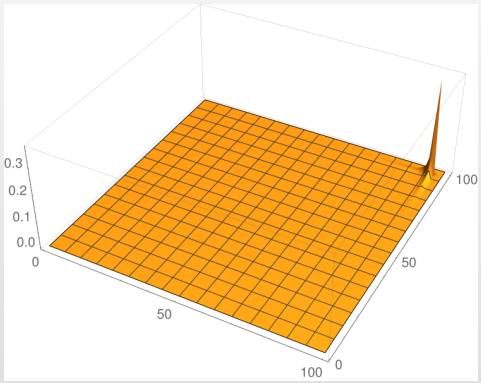
- more sophisticated CIRCE2 input

```
{ file = "TDR.circe"
  { design = "ILC"
     roots = 500
     scale = 250
     bins = 100
    { pid/1 = electron
      pid/2 = positron
      pol = 0
      events = "guinea_pig/out/ILC_500_unpolarized.data"
      columns = 2
      lumi = 8.008e33
      min = 0
      max = 1.05
      iterations = 10
} } }
```

will generate a variable width histogram with weights according to Guinea-Pig output performing 10 iterations of adapting the bin widths to minimize the variance of the weights

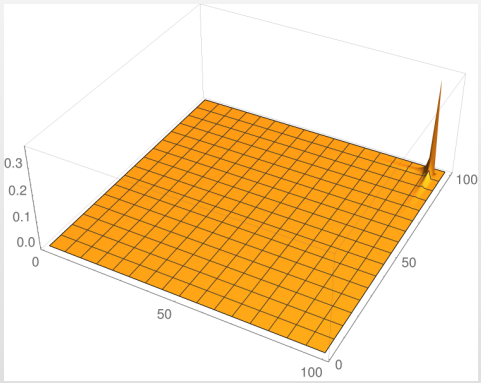
From Guinea-Pig to CIRCE2

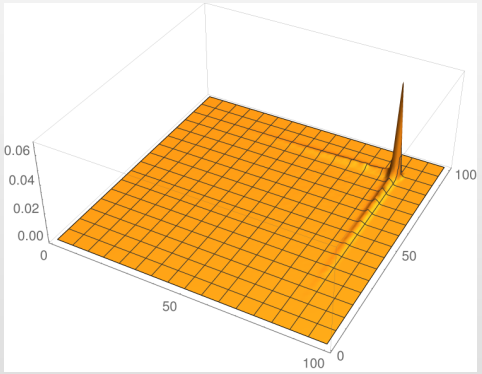
- `iterations` $= 0, 1, 2, 3, 4, 5, 6, 7, 8$:

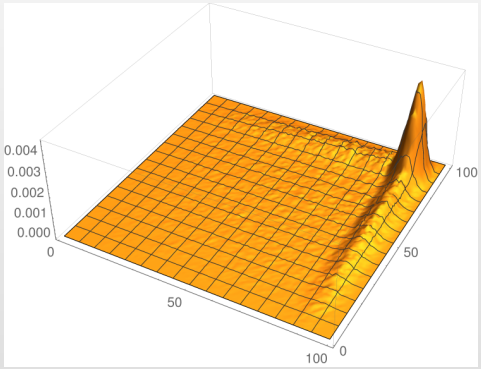


From Guinea-Pig to CIRCE2

- `iterations` $= 0, 1, 2, 3, 4, 5, 6, 7, 8$:

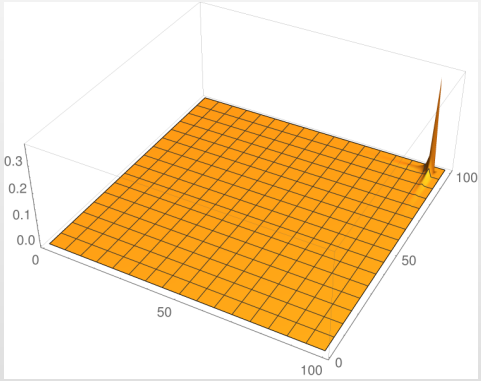


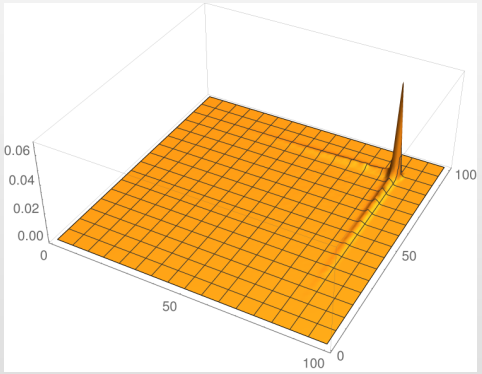


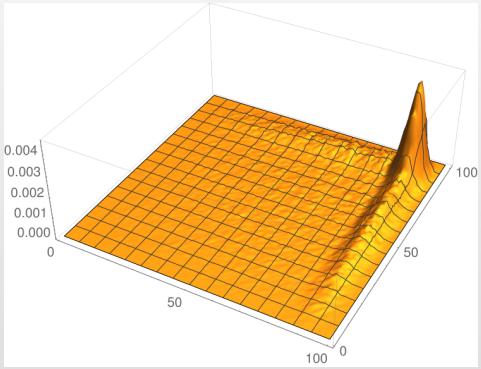


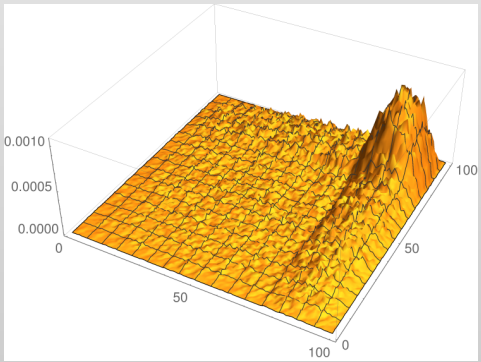
From Guinea-Pig to CIRCE2

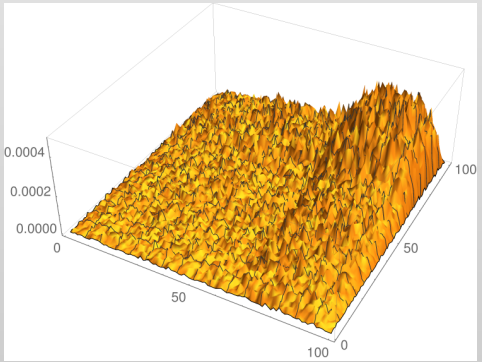
- `iterations` $= 0, 1, 2, 3, 4, 5, 6, 7, 8$:

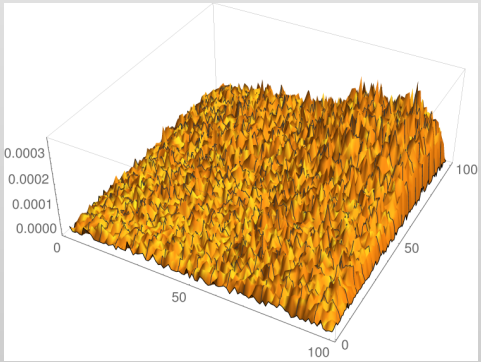






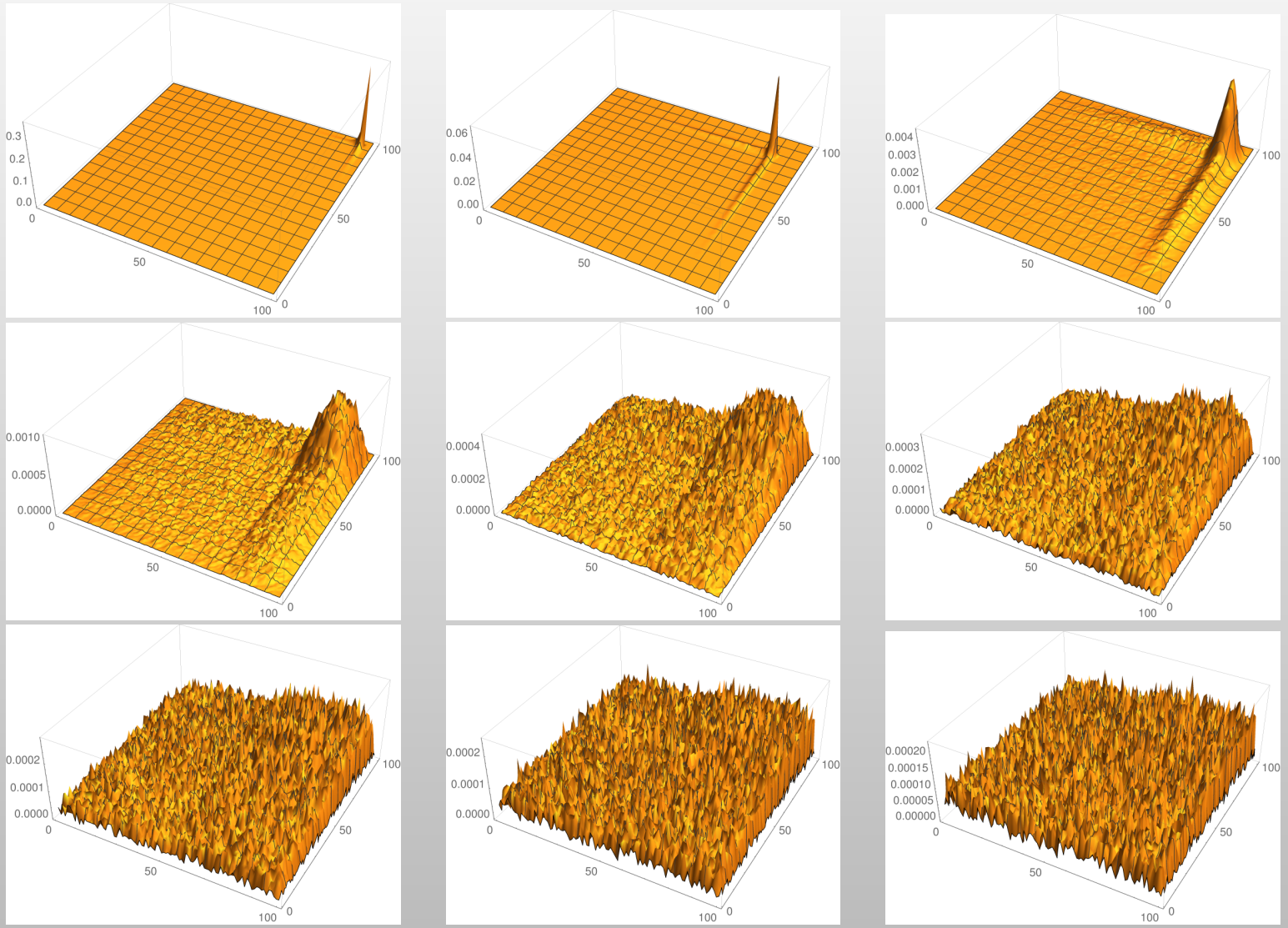






From Guinea-Pig to CIRCE2

- `iterations` $= 0, 1, 2, 3, 4, 5, 6, 7, 8$:



(171.306 Guinea-Pig events in 10.000 bins)

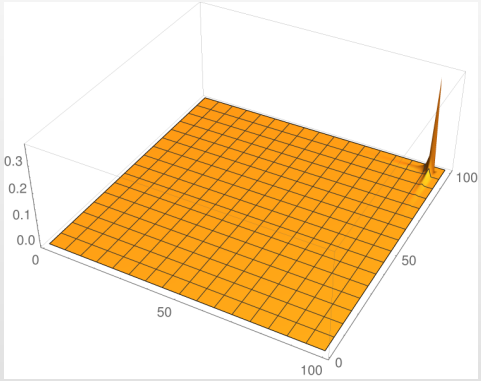
- more sophisticated CIRCE2 input

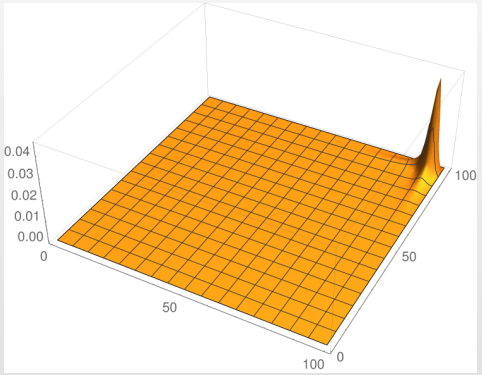
```
{ file = "TDR.circe"
  { design = "ILC"
     roots = 500
     scale = 250
     bins = 100
    { pid/1 = electron
      pid/2 = positron
      pol = 0
      events = "guinea_pig/out/ILC_500_unpolarized.data"
      columns = 2
      lumi = 8.008e33
      min = 0
      max = 1.05
      iterations = 4
      smooth = 5 [0.00,1.05] [0.00,1.05]
} } }
```

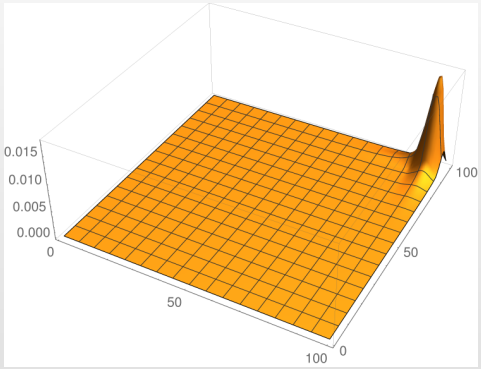
applies a Gaussian smearing

From Guinea-Pig to CIRCE2

- `iterations` $= 0$ and `smooth` $= 0, 3, 5$:

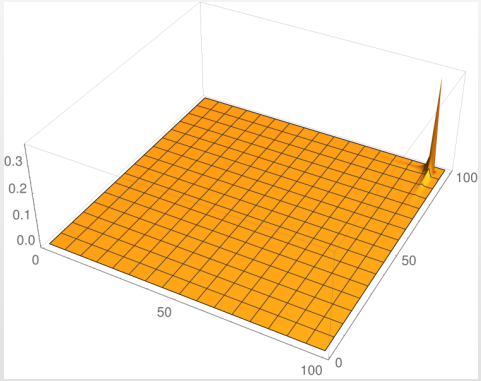


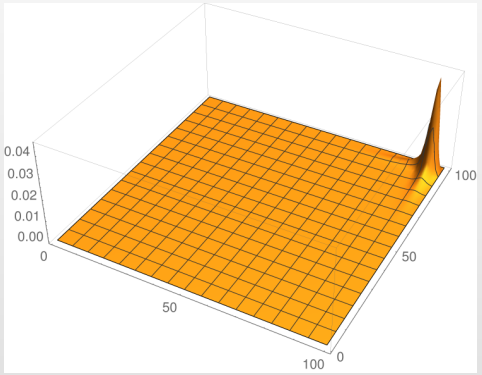


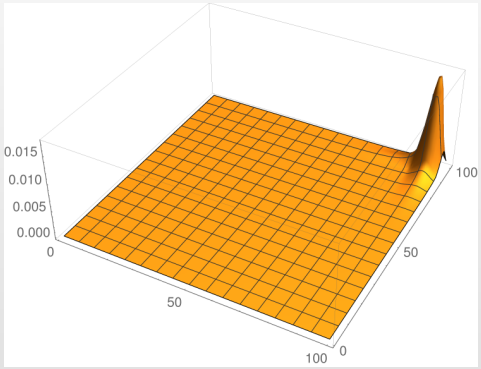


From Guinea-Pig to CIRCE2

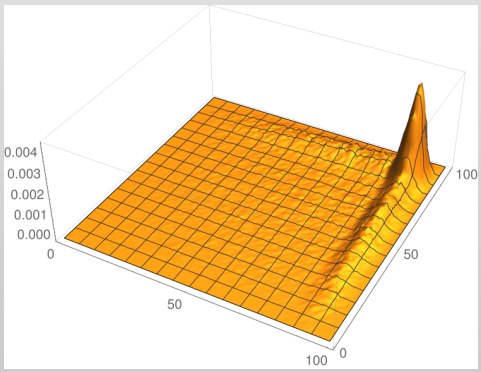
- `iterations` $= 0$ and `smooth` $= 0, 3, 5$:

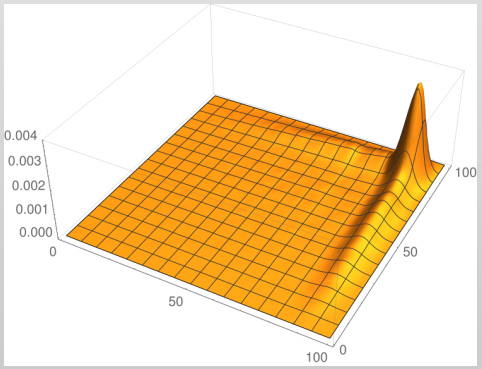


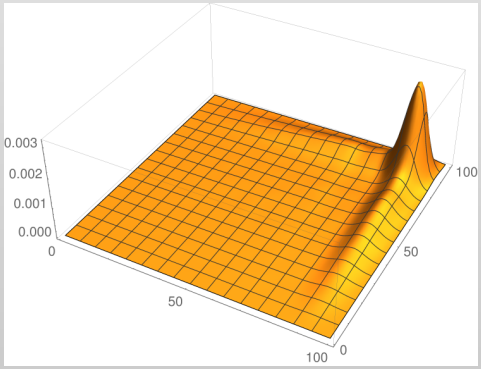




- `iterations` $= 2$ and `smooth` $= 0, 3, 5$:

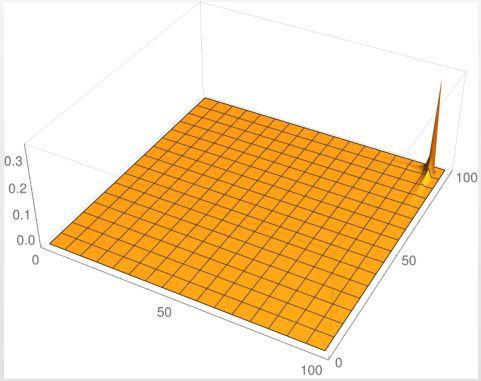


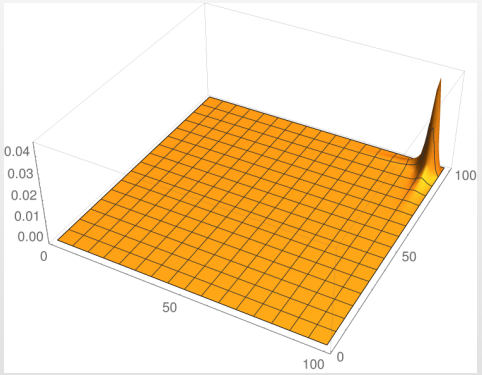


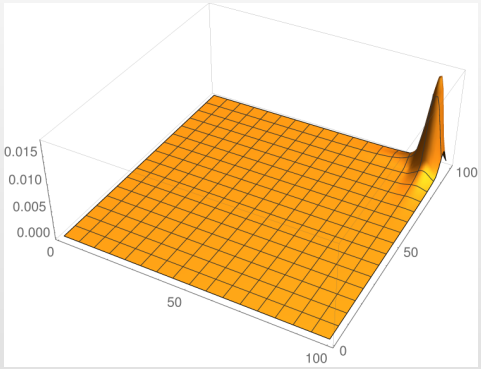


From Guinea-Pig to CIRCE2

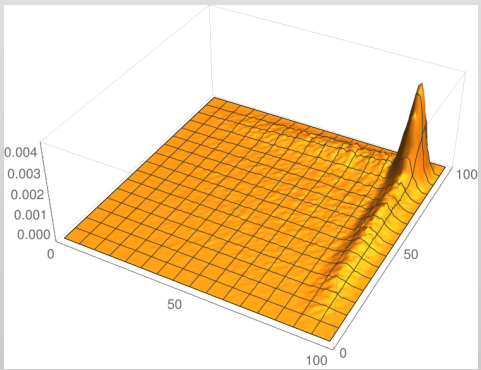
- `iterations` $= 0$ and `smooth` $= 0, 3, 5$:

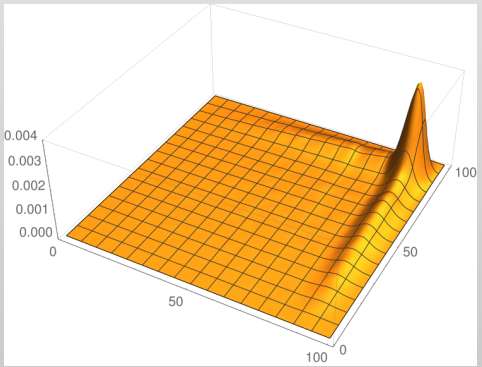


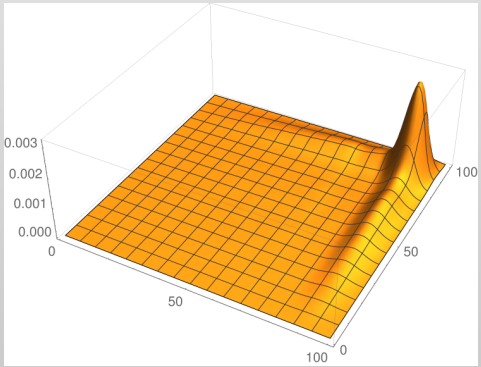




- `iterations` $= 2$ and `smooth` $= 0, 3, 5$:







- `iterations` $= 4$ and `smooth` $= 0, 3, 5$: