Physics 305 Demo Notebook 8: First Passage Time Density - Exponential MSD

In this Demo Notebook, we calculate the first passage time density (FPTD) for the MSD of the form:

$$M(t) = a\Gamma(\mu)t^{\mu-1}\beta^{-\mu}e^{-\beta/t}$$

In Demo Notebook 6, we found that this theoretical model for the MSD captures the shape of the empirical MSD for sunspot data.

Recall the expression for FPTD derived in class:

$$f(t) = -\frac{(x_0-x_c)}{\sqrt{2\pi[M(t)]^3}}\frac{\partial M(t)}{\partial t}\exp\left[-\frac{(x_0-x_c)^2}{2M(t)}\right],$$

where $M(t)$ is the MSD.

Evaluating the partial derivative w.r.t. time, we get

$$\frac{\partial M(t)}{\partial t} = a\Gamma(\mu)\beta^{-\mu}e^{-\beta/t}t^{\mu-3}[\beta + (\mu-1)t] = M(t)\left[\frac{\beta}{t^2} + \frac{\mu-1}{t}\right].$$

Note that μ is dimensionless and β has units of time.

The first passage time density becomes:

$$f(t) = -\frac{(x_0-x_c)}{\sqrt{2\pi[M(t)]}}\left[\frac{\beta}{t^2} + \frac{\mu-1}{t}\right]\exp\left[-\frac{(x_0-x_c)^2}{2M(t)}\right]$$

Note that this is negative for $x_0 > x_c$ and positive for $x_c > x_0$.

Specifically, we take the best-fit theoretical model for the MSD as a function of lag time for sunspot data and plot $f(t)$ for different values of $x_0 - x_c$.

In [1]:
```
# import libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.special import gamma
```

In [2]:
```
# define MSD function
def msd(t, a, beta, mu):
  return a*gamma(mu)*t**(mu-1)*beta**(-1*mu)*np.exp(-1*beta/t)

# define FPTD function; getting the absolute value of dx, to ensure f(t)
def fptd(t, dx, a, beta, mu):
  m = msd(t, a, beta, mu)
  f1 = np.abs(dx)/np.sqrt(2*np.pi*m)
  f2 = (beta/t**2 + (mu-1)/t)
  f3 = np.exp(-1*dx**2/2/m)
  return f1*f2*f3
```

For sunspot cycle 24, we previously found best-fit parameters a = 3708.31 (1230.48), β = 2.43 (0.56), and μ = 1.02 (0.03) from fitting the empirical MSD (in Demo Notebook 6).

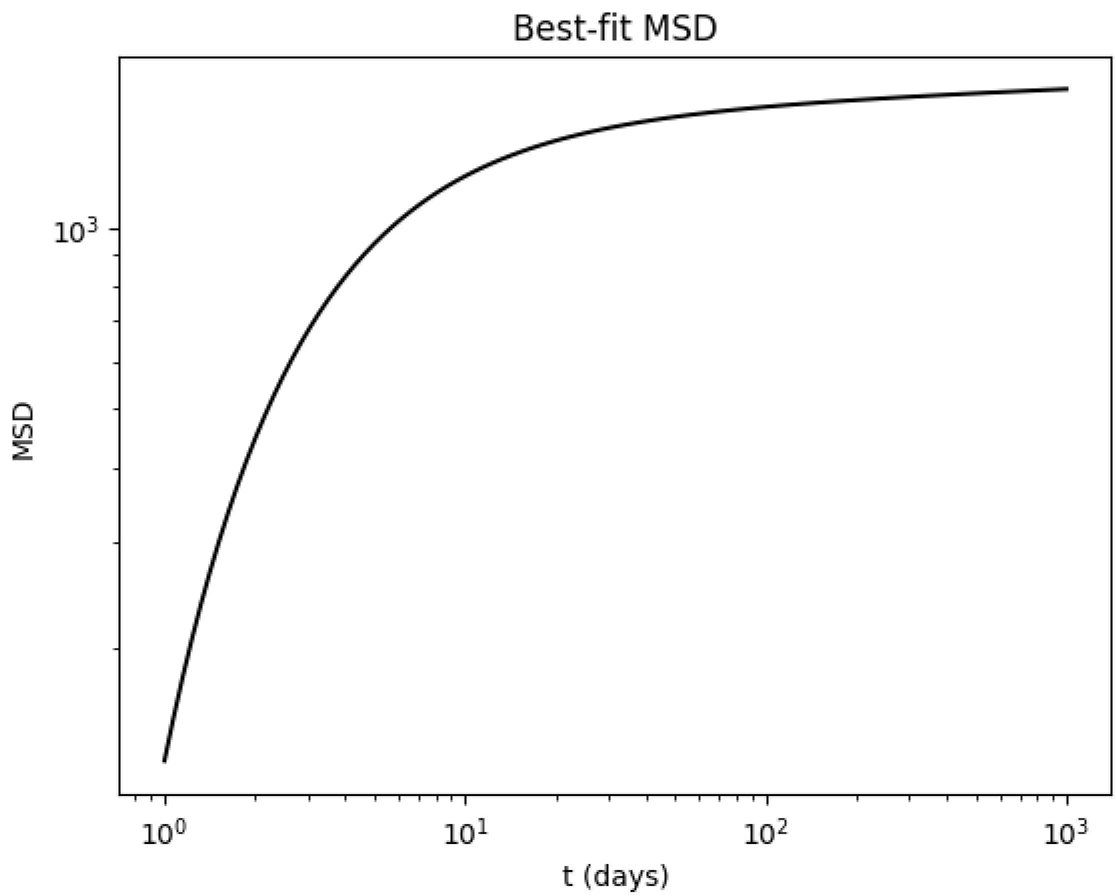
In [3]:
```
a, err_a = 3708.31, 1230.48
beta, err_beta = 2.43, 0.56
mu, err_mu = 1.02, 0.03
```

In [4]:
```
# plot MSD (best-fit)

t = np.logspace(0, 3, 100) # equally log-spaced grid in time, in units o
m0 = msd(t, a, beta, mu)

plt.plot(t, m0, color='k')
plt.xscale("log")
plt.yscale("log")
plt.xlabel("t (days)")
plt.ylabel("MSD")
plt.title("Best-fit MSD")
```

Out[4]: Text(0.5, 1.0, 'Best-fit MSD')



We choose dx values: 25, 50, 75, 100-- representing differences in residual sunspot number (from initial value x_0 to cliff point value x_c) and calculate FPTD for each value.

In [5]:
```
dxs = np.array([25, 50, 75, 100]) # difference in residual sunspot numbe
dx = dxs[0]

f_out = np.zeros((len(t), len(dxs)))

for i in np.arange(len(dxs)):
  f_out[:, i] = fptd(t, dxs[i], a, beta, mu)
```

As in the previous example, we also get the time where the FPTD curve peaks and overlay

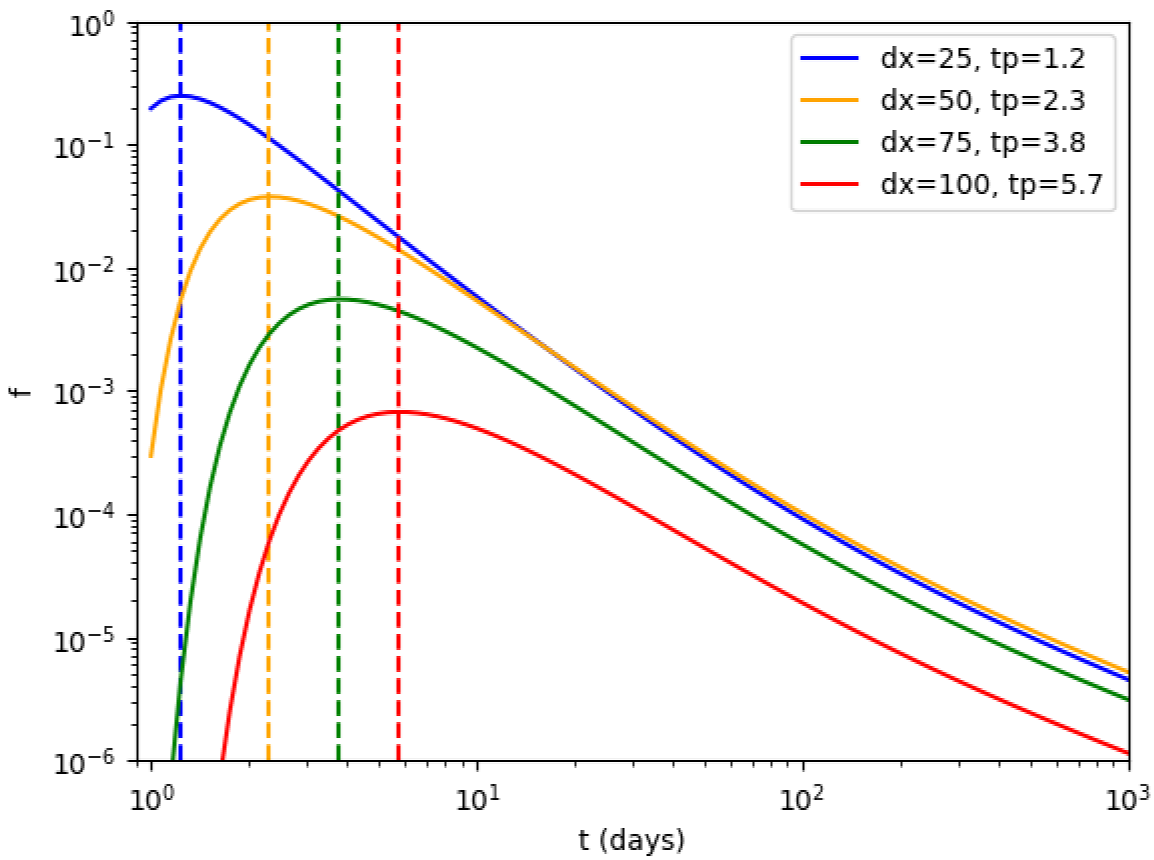
In [6]:
```
tpeak = np.zeros((len(dxs)))
colors = ["blue", "orange", "green", "red"]

for i in np.arange(len(dxs)):

  # get peak
  ipeak = np.argmax(f_out[:,i])
  tpeak[i] = t[ipeak]
  plt.plot(t, f_out[:, i], color=colors[i], label="dx=%d, tp=%.1f" % (dx
  plt.axvline(tpeak[i], ls='--', color=colors[i])

plt.xscale("log")
plt.yscale("log")
plt.xlabel("t (days)")
plt.ylabel("f")
plt.legend(loc="best")
plt.minorticks_on()
plt.ylim((1e-6, 1))
plt.xlim((0.9,1e3))
```

Out[6]: (0.9, 1000.0)



We find that the shape of the FPTD curves are broad, but the peaks move to the right with larger differences in residual sunspot number, as expected. For differences of 25, 50, 75, and 100, we find that the FPTD peaks at around 1, 2, 4, and 6 days, respectively.

Effect of varying the parameters β and μ

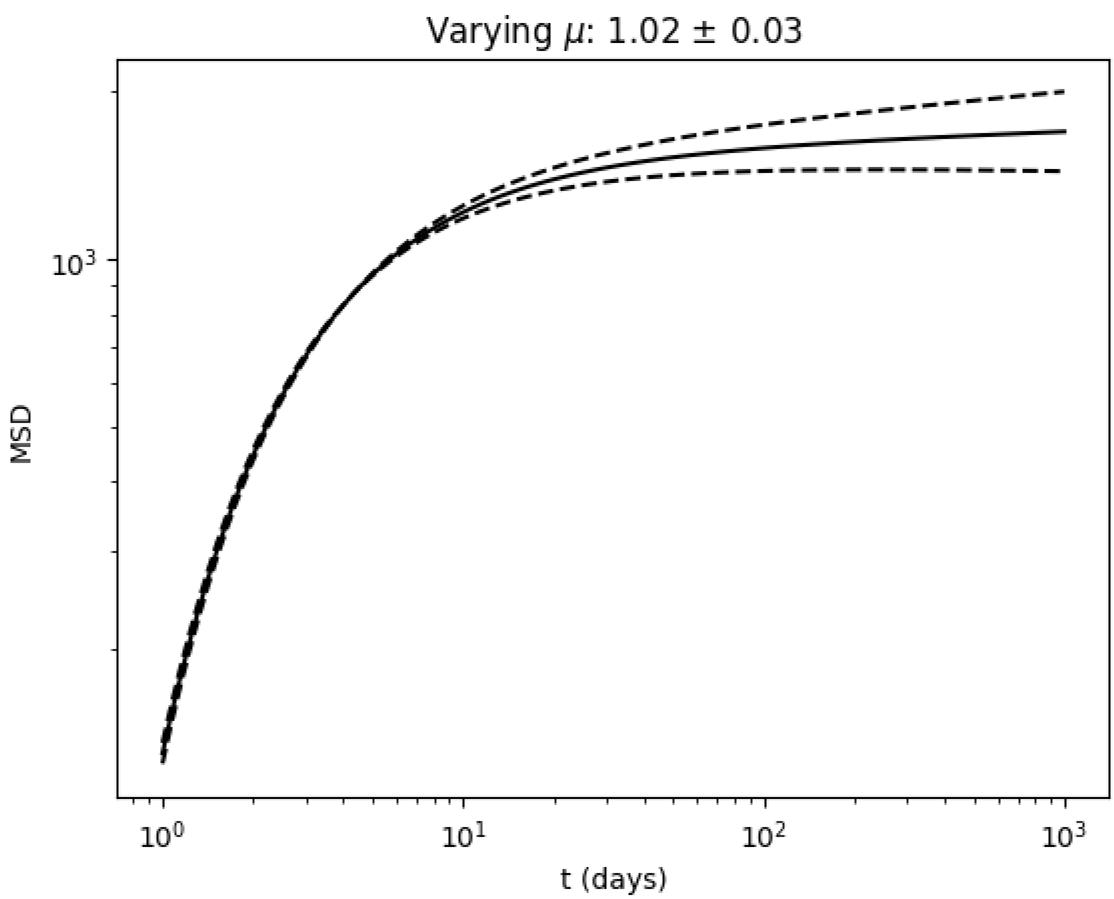
First, let us vary the memory parameter μ. We plot the MSD curves for μ equal to the $\pm 1 - \sigma$ from the fit, keeping the other 2 parameters fixed.

In [7]:
```
# plot MSD - varying mu

t = np.logspace(0, 3, 100) # equally log-spaced grid in time, in units o
m1 = msd(t, a, beta, mu-err_mu)
m2 = msd(t, a, beta, mu+err_mu)

plt.plot(t, m0, color='k', ls='-')
plt.plot(t, m1, color='k', ls='--')
plt.plot(t, m2, color='k', ls='--')
plt.xscale("log")
plt.yscale("log")
plt.xlabel("t (days)")
plt.ylabel("MSD")
plt.title(r"Varying $\mu$: %.2f $\pm$ %.2f" % (mu, err_mu))
```

Out[7]:
```
Text(0.5, 1.0, 'Varying $\\mu$: 1.02 $\\pm$ 0.03')
```



Now, we calculate the FPTD for these two cases, for fixed dx = 50, and plot the curves.

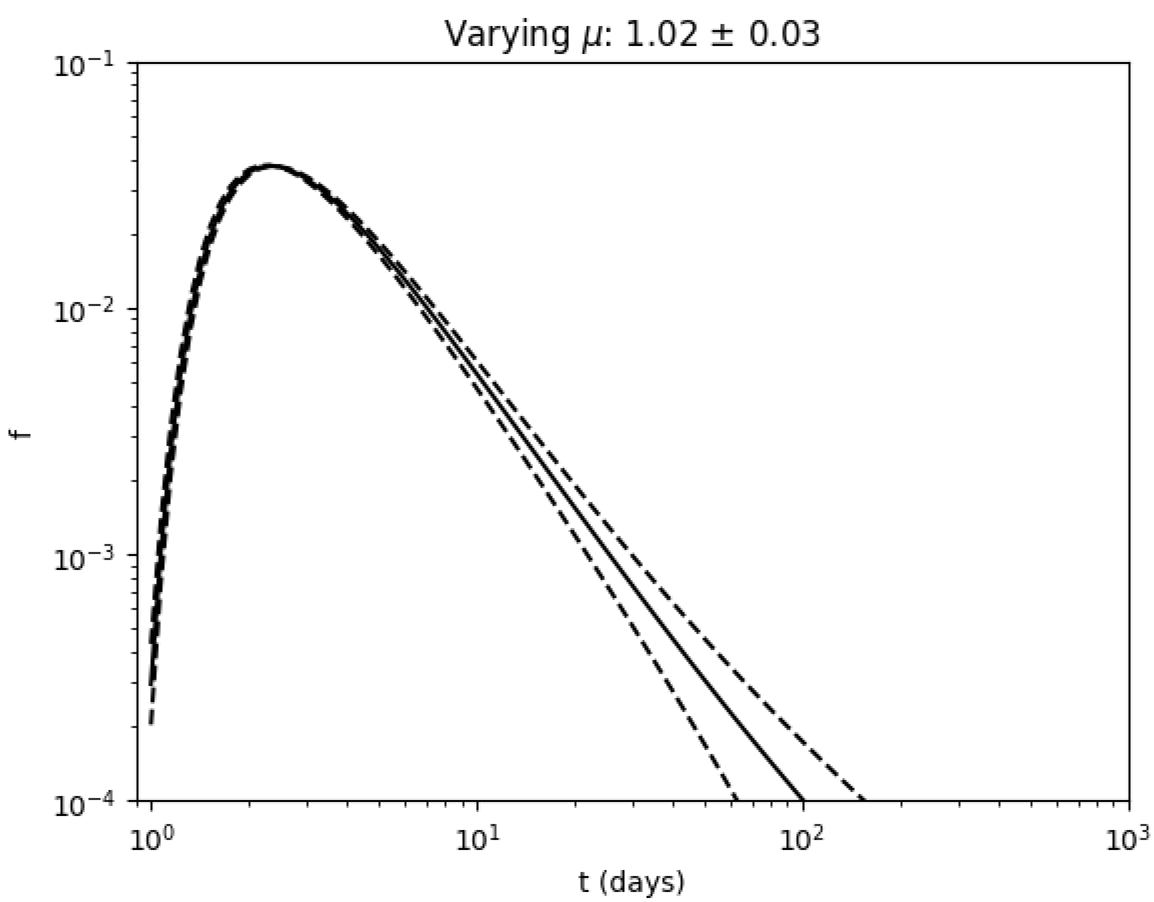
In [8]:
```
dx = 50
f_mu = np.zeros((len(t), 2))
f_mu[:, 0] = fptd(t, dx, a, beta, mu-err_mu)
f_mu[:, 1] = fptd(t, dx, a, beta, mu+err_mu)
```

In [9]:
```
# plot FPTD curves

j=1 # dx = 50 case for best-fit MSD
plt.plot(t, f_out[:, j], color='k', ls='-')
for i in np.arange(2):
  plt.plot(t, f_mu[:, i], color='k', ls='--') #, label="dx=%d, tp=%.1f"

plt.xscale("log")
plt.yscale("log")
plt.xlabel("t (days)")
plt.ylabel("f")
#plt.legend(loc="best")
plt.minorticks_on()
plt.ylim((1e-4, 1e-1))
plt.xlim((0.9,1e3))
plt.title(r"Varying $\mu$: %.2f $\pm$ %.2f" % (mu, err_mu))
```

Out[9]:
```
Text(0.5, 1.0, 'Varying $\\mu$: 1.02 $\\pm$ 0.03')
```



In [10]:
```
# get peaks

tpeak_mu = np.zeros(3)

# for mu-err_mu
ipeak = np.argmax(f_mu[:,0])
tpeak_mu[0] = t[ipeak]

# for mu
tpeak_mu[1] = tpeak[j]

# for mu+err_mu
ipeak = np.argmax(f_mu[:,1])
tpeak_mu[2] = t[ipeak]

print(tpeak_mu)
```

```
[2.3101297 2.3101297 2.3101297]
```

The difference in peak times is negligible in this case.

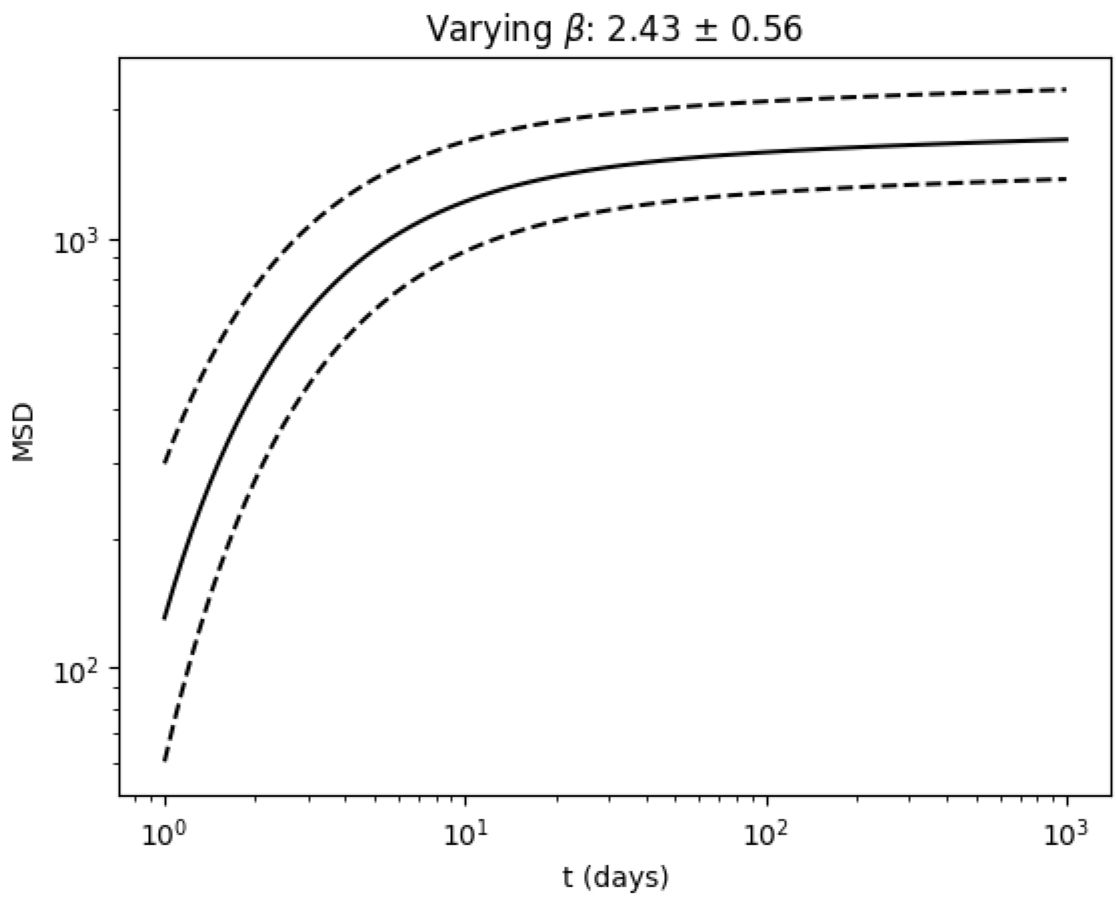
Next, we vary the β parameter. We plot the MSD curves for β equal to the $\pm 1 - \sigma$ from the fit, keeping the other 2 parameters fixed.

In [11]:
```
# plot MSD - varying beta

t = np.logspace(0, 3, 100) # equally log-spaced grid in time, in units o
m1 = msd(t, a, beta-err_beta, mu)
m2 = msd(t, a, beta+err_beta, mu)

plt.plot(t, m0, color='k', ls='-')
plt.plot(t, m1, color='k', ls='--')
plt.plot(t, m2, color='k', ls='--')
plt.xscale("log")
plt.yscale("log")
plt.xlabel("t (days)")
plt.ylabel("MSD")
plt.title(r"Varying $\beta$: %.2f $\pm$ %.2f" % (beta, err_beta))
```

Out[11]:
```
Text(0.5, 1.0, 'Varying $\\beta$: 2.43 $\\pm$ 0.56')
```



Next, we calculate the FPTD for these two cases, for fixed dx = 50, and plot the curves.

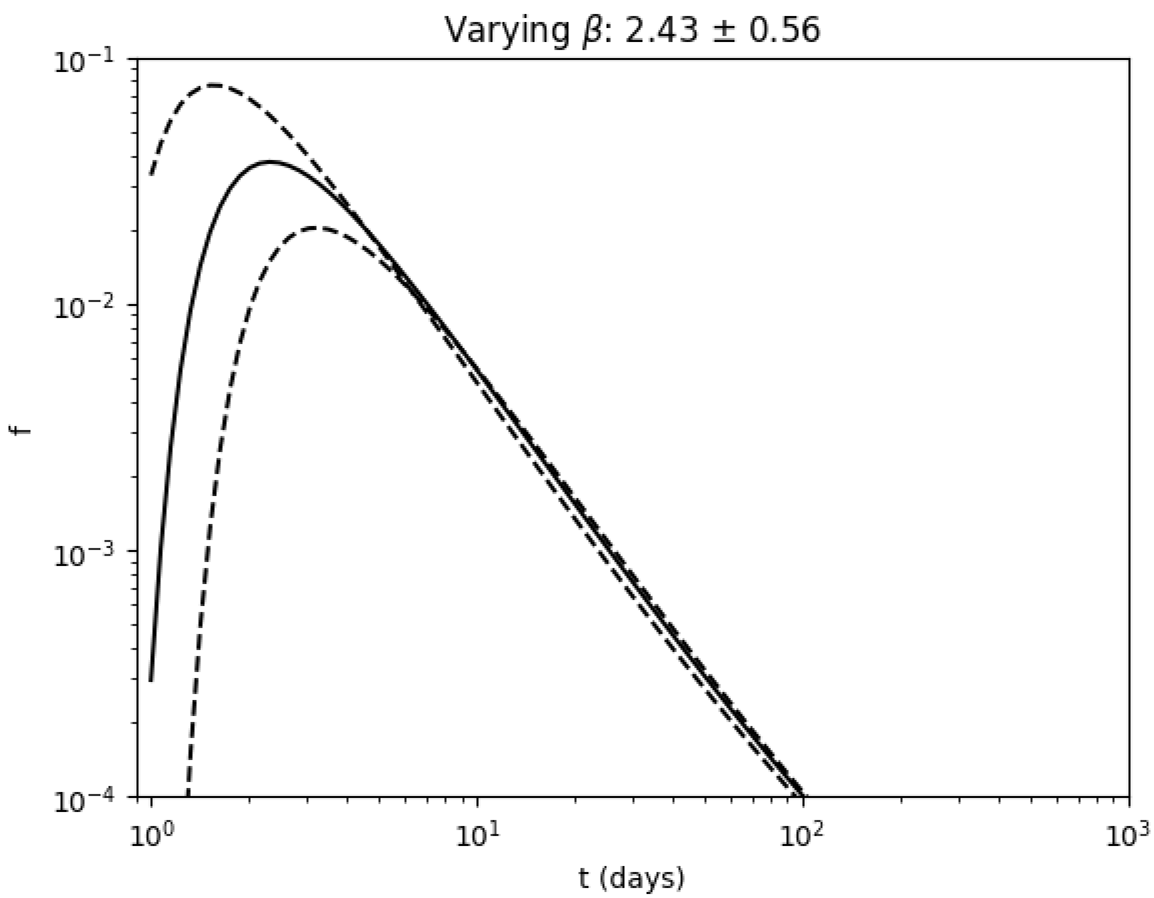
In [12]:
```
dx = 50
f_beta = np.zeros((len(t), 2))
f_beta[:, 0] = fptd(t, dx, a, beta-err_beta, mu)
f_beta[:, 1] = fptd(t, dx, a, beta+err_beta, mu)
```

In [13]:
```
# plot FPTD curves

j=1 # dx = 50 case for best-fit MSD
plt.plot(t, f_out[:, j], color='k', ls='-')
for i in np.arange(2):
  plt.plot(t, f_beta[:, i], color='k', ls='--') #, label="dx=%d, tp=%.1f

plt.xscale("log")
plt.yscale("log")
plt.xlabel("t (days)")
plt.ylabel("f")
#plt.legend(loc="best")
plt.minorticks_on()
plt.ylim((1e-4, 1e-1))
plt.xlim((0.9,1e3))
plt.title(r"Varying $\beta$: %.2f $\pm$ %.2f" % (beta, err_beta))
```

Out[13]:
```
Text(0.5, 1.0, 'Varying $\\beta$: 2.43 $\\pm$ 0.56')
```



In [14]:
```
# get peaks

tpeak_beta = np.zeros(3)

# for mu-err_mu
ipeak = np.argmax(f_beta[:,0])
tpeak_beta[0] = t[ipeak]

# for mu
tpeak_beta[1] = tpeak[j]

# for mu+err_mu
ipeak = np.argmax(f_beta[:,1])
tpeak_beta[2] = t[ipeak]

print(tpeak_beta)
```

```
[1.51991108 2.3101297  3.27454916]
```

We find a larger impact on the peak times when we vary β by 1σ, as can also be seen visually from the FPTD curves.