Example 8.

Sensitivity of hydraulic head at a point to spatially uniform specific storage under transient flow conditions

0. Forward model

Governing equation:

$$K\,b\,\frac{\partial^2 h}{\partial x^2} + R = S_s\,b\,\frac{\partial h}{\partial t} \tag{1}$$

(2)

Boundary conditions:

$$h(x,t) = h_{\Gamma_{1_0}}\ , \qquad\qquad x = 0 = \Gamma_{1_0} \tag{3}$$

$$h(x,t) = h_{\Gamma_{1_L}}\ , \qquad\qquad x = L = \Gamma_{1_L} \tag{4}$$

(5)

Initial conditions:

$$h(x,t) = h_0\ , \qquad\qquad t = 0 \tag{6}$$

(7)

Closed-form solution:

Not available (8)

(9)

Spatial derivatives from differentiation:

Not available (10)

(11)

In [28]:
```
from IPython.display import HTML, display
def set_background(color):
    script = (
        "var cell = this.closest('.code_cell');"
        "var editor = cell.querySelector('.input_area');"
        "editor.style.background='{}';"
        "this.parentNode.removeChild(this)").format(color)
    display(HTML('<img src onerror="{}">'.format(script)))
```

In [40]:
```
from warnings import filterwarnings
filterwarnings("ignore", category=DeprecationWarning)

import numpy as np

K, Ss, R, b, L, BC1h, ocol = 10., 1e-6, 1e-1/1000., 10., 10000., 0., 5000
X = np.arange(L)
```

1. Direct sensitivity

Not available (12)

2. Perturbation sensitivity

(13)

$$\frac{\partial h(x')}{\partial S_s} \approx \frac{h(x, S_s + \Delta S_s) - h(x, S_s)}{\Delta S_s} \tag{14}$$

(15)

2a. Analytical

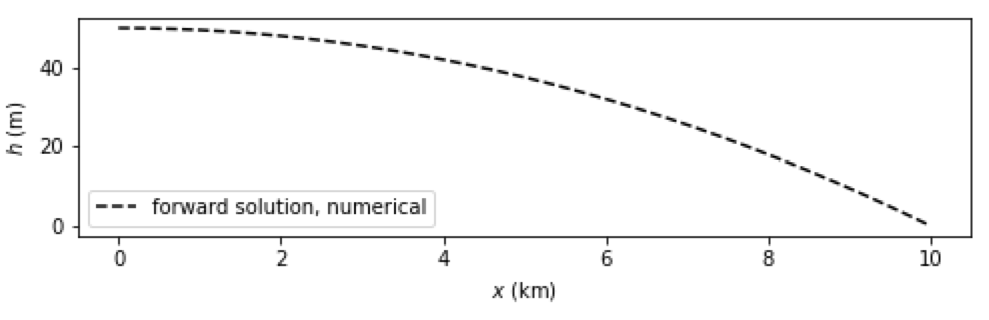
Not available (16)

2b. Semi-analytical

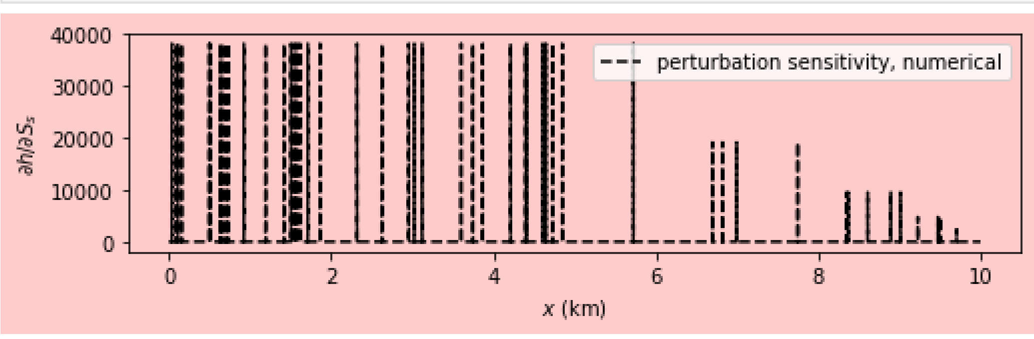
Not available (17)

2c. Numerical

In [30]:
```
import flopy

dpar = 1e-4
nrow, ncol = 1, int(L)
M0 = flopy.modflow.Modflow(modelname='model', exe_name='../mf2005.exe')
flopy.modflow.ModflowDis(M0, nlay=1, nrow=nrow, ncol=ncol, nper=2, delr=1., delc=1., top=0., botm=-b,
                         steady=[True,False], perlen=[1., 1.], nstp=[1, 1])
flopy.modflow.ModflowBas(M0, ibound=np.hstack([np.ones([nrow, ncol-1], dtype=int), -1*np.ones([1,1])]),
                         strt=BC1h*np.ones([nrow, ncol], dtype=float))
flopy.modflow.ModflowLpf(M0, hk=K, vka=-999., ss=Ss, sy=-999., ipakcb=53)
flopy.modflow.ModflowRch(M0, nrchop=1, rech={0:R, 1:R}, ipakcb=53)
flopy.modflow.ModflowPcg(M0, hclose=1e-6, rclose=1e-6)
flopy.modflow.ModflowOc(M0, stress_period_data={(0,0): ['save head', 'save budget'], (1,0): ['save head', 'save
M0.write_input()
success, buff = M0.run_model(silent=True)
H0 = flopy.utils.binaryfile.HeadFile('model.hds').get_data(kstpkper=[0,1])[0,0,:]
dhdt = np.ravel(flopy.utils.binaryfile.HeadFile('model.hds').get_data(kstpkper=[0,1]))-\
       np.ravel(flopy.utils.binaryfile.HeadFile('model.hds').get_data(kstpkper=[0,0]))

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
plt.subplots(figsize=[8,2])
plt.plot(X/1000., H0, 'k--', mfc='none', label='forward solution, numerical')
plt.xlabel('$x$ (km)')
plt.ylabel('$h$ (m)')
plt.legend(loc=3);
```



In [34]:
```
M1 = M0
flopy.modflow.ModflowLpf(M1, hk=K, vka=-999., ss=Ss+Ss*dpar, sy=-999., ipakcb=53)
M1.write_input()
success, buff = M1.run_model(silent=True)
H1 = flopy.utils.binaryfile.HeadFile('model.hds').get_data(kstpkper=[0,1])[0,0,:]
dhdSs = (H1-H0)/(Ss*dpar)

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
f,s = plt.subplots(figsize=[8,2])
plt.plot(X/1000., dhdSs, 'k--', mfc='none', label='perturbation sensitivity, numerical')
plt.xlabel('$x$ (km)')
plt.ylabel(r'$\partial h / \partial S_s$')
plt.legend()
f.patch.set_facecolor((1.0, 0.0, 0.0, 0.2))
s.set_facecolor((1.0, 0.0, 0.0, 0.01));
```



In [32]:
```
set_background('rgba(200, 0, 0, 0.2)')
print('%.6f'% dhdSs[ocol])
```

```
0.000000
```

3. Adjoint sensitivity

$$\frac{\partial h(x')}{\partial S_s} = \int_T \int_X \psi_1^*(x,t)\,\frac{dh(x,t)}{dt}\,dx\,dt \tag{18}$$

Governing equation:

$$K\,b\,\frac{\partial \psi_1^*}{\partial x} + \frac{1}{2\,K\,b}\delta(x - x') = S_s\,b\,\frac{\partial \psi_1^*}{\partial t} \tag{19}$$

(20)

Boundary conditions:

$$\psi_1^*(x,t) = 0\ , \qquad\qquad x = 0 = \Gamma_{1_0} \tag{21}$$

$$\psi_1^*(x,t) = 0\ , \qquad\qquad x = L = \Gamma_{1_L} \tag{22}$$

(23)

Terminal conditions:

$$\psi_1^*(x,t) = 0\ , \qquad\qquad t = t_{final} \tag{24}$$

(25)

Closed-form solution:

Not available (26)

3a. Analytical

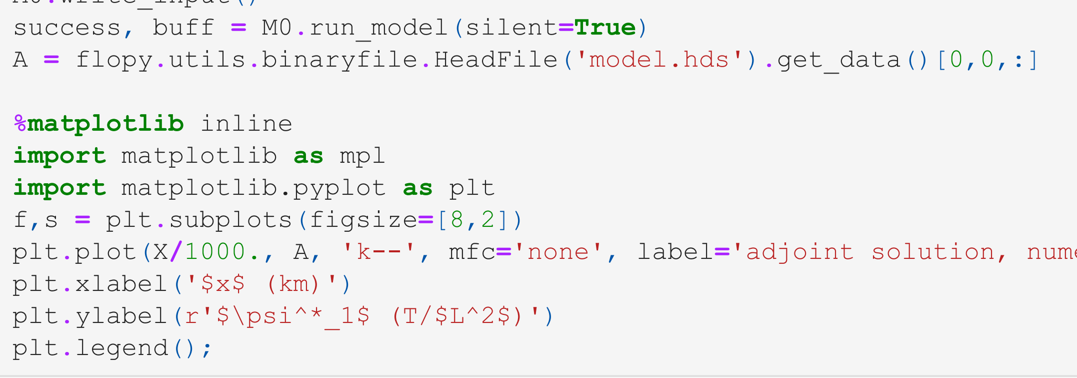
Not available (27)

3b. Semi-analytical

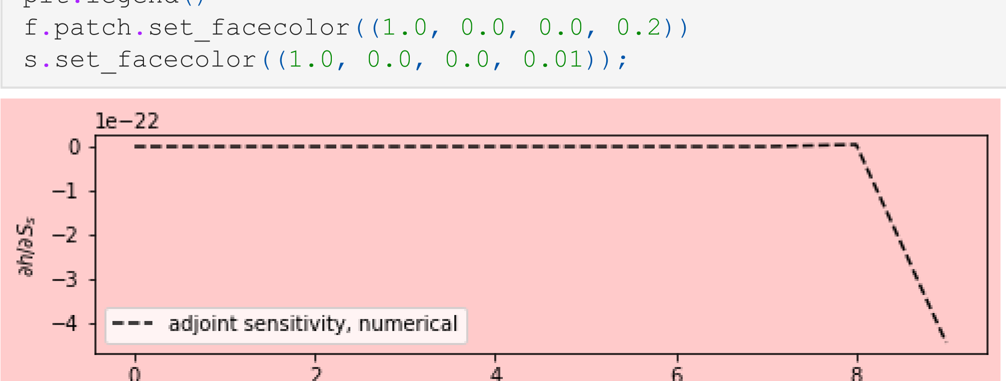
Not available (28)

3c. Numerical

In [41]:
```
M0 = flopy.modflow.Modflow(modelname='model', exe_name='../mf2005.exe')
flopy.modflow.ModflowDis(M0, nlay=1, nrow=1, ncol=ncol, nper=2, delr=1., delc=1., top=0., botm=-b, steady=False
                         perlen=1., nstp=1)
flopy.modflow.ModflowBas(M0, ibound=np.hstack([np.ones([nrow, ncol-1], dtype=int), -1*np.ones([1,1])]),
                         strt=BC1h*np.ones([nrow, ncol], dtype=float))
flopy.modflow.ModflowLpf(M0, hk=K, vka=-999., ss=-999., sy=-999., ipakcb=53)
flopy.modflow.ModflowWel(M0, stress_period_data={0: [0, 0, ocol, 1.]})
flopy.modflow.ModflowPcg(M0, hclose=1e-6, rclose=1e-6)
flopy.modflow.ModflowOc(M0, stress_period_data={(0,0): ['save head', 'save budget']})
M0.write_input()
success, buff = M0.run_model(silent=True)
A = flopy.utils.binaryfile.HeadFile('model.hds').get_data()[0,0,:]

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
f,s = plt.subplots(figsize=[8,2])
plt.plot(X/1000., A, 'k--', mfc='none', label='adjoint solution, numerical')
plt.xlabel('$x$ (km)')
plt.ylabel(r'$\psi^*_1$ (T/$L^2$)')
plt.legend();
```



In [48]:
```
dhdSs = np.empty(0)
for oc,xp in enumerate(X[::1000]):
    M1 = M0
    flopy.modflow.ModflowWel(M1, stress_period_data={0: [0, 0, oc, 1.]})
    M1.write_input()
    success, buff = M1.run_model(silent=True)
    A = flopy.utils.binaryfile.HeadFile('model.hds').get_data()[0,0,:]
    dhdSs = np.append(dhdSs, np.sum(A*dhdt))

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
f,s = plt.subplots(figsize=[8,2])
plt.plot(X[::1000]/1000., dhdSs, 'k--', mfc='none', label='adjoint sensitivity, numerical')
plt.xlabel('$x$ (km)')
plt.ylabel(r'$\partial h / \partial S_s$')
plt.legend()
f.patch.set_facecolor((1.0, 0.0, 0.0, 0.2))
s.set_facecolor((1.0, 0.0, 0.0, 0.01));
```



In [45]:
```
set_background('rgba(200, 0, 0, 0.2)')
print('%.6f'% dhdSs[int(np.where(X[::1000]==float(ocol))[0])])
```

```
-0.000000
```

In [ ]: