Groenwolds's relaxed conservatism

See algorithm in appended Python code.
The objective and constraint function is defined as:

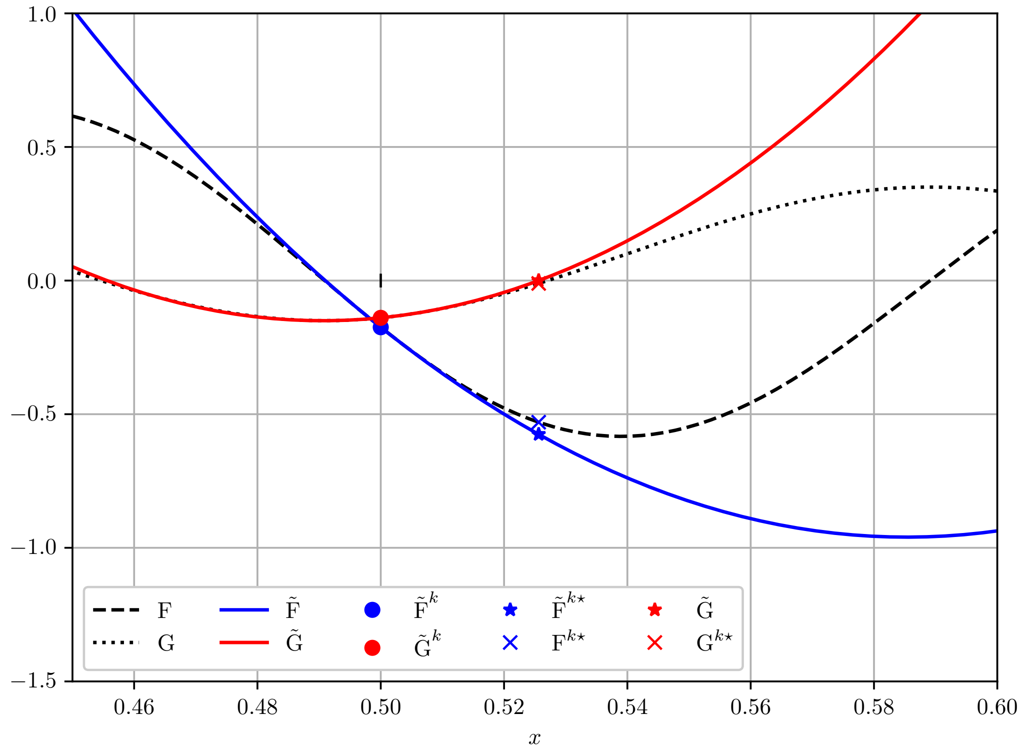
```
#
import numpy as np
#
def obj(x):
#
    a = 1.; b = 32.; c = -1.
    f = a*np.sin(b*x)*np.exp(c*x)
    df = a*c*np.sin(b*x)*np.exp(c*x) + a*b*np.exp(c*x)*np.cos(b*x)
    ddf = a*c*b*np.cos(b*x)*np.exp(c*x) + a*c*c*np.sin(b*x)*np.exp(c*x) \
        + a*c*b*np.exp(c*x)*np.cos(b*x) - a*b*b*np.exp(c*x)*np.sin(b*x)
#   ddf = abs(-2./x*df) # quad. approx. to reciprocal intervening variables
#
    return [f, df, ddf]
#
def con(x):
#
    a = 1./4.; b = 32.
    g = a*np.cos(b*x)+0.1
    dg = -a*b*np.sin(b*x)
    ddg = -a*b*b*np.cos(b*x)
#   ddg = abs(-2./x*dg) # quad. approx. to reciprocal intervening variables
#
    return [g, dg, ddg]
#
```

QP subproblem at x^k:

$x^k = 0.500$
$\mathrm{F}^k = -0.175$
$\mathrm{d}_x^k\mathrm{F} = -18.413$
$\mathrm{dd}_x^k\tilde{\mathrm{F}} = 215.813$
$\alpha_\mathrm{F} = 1.000$

$\mathrm{G}^k = -0.139$
$\mathrm{d}_x^k\mathrm{G} = 2.303$
$\mathrm{dd}_x^k\tilde{\mathrm{G}} = 245.161$
$\alpha_\mathrm{G} = 1.000$

with Lagrange multiplier
$\lambda^{k\star} = 1.501$
at the solution
$x^{k\star} = 0.526$



At the solution of the QP the function approximations have the values
$\tilde{\mathrm{F}}^{k\star} = -5.754e-01$
$\tilde{\mathrm{G}}^{k\star} = -1.849e-10$

while the actual functions are evaluated to be
$\mathrm{F}^{k\star} = -5.300e-01$
$\mathrm{G}^{k\star} = -1.079e-02$

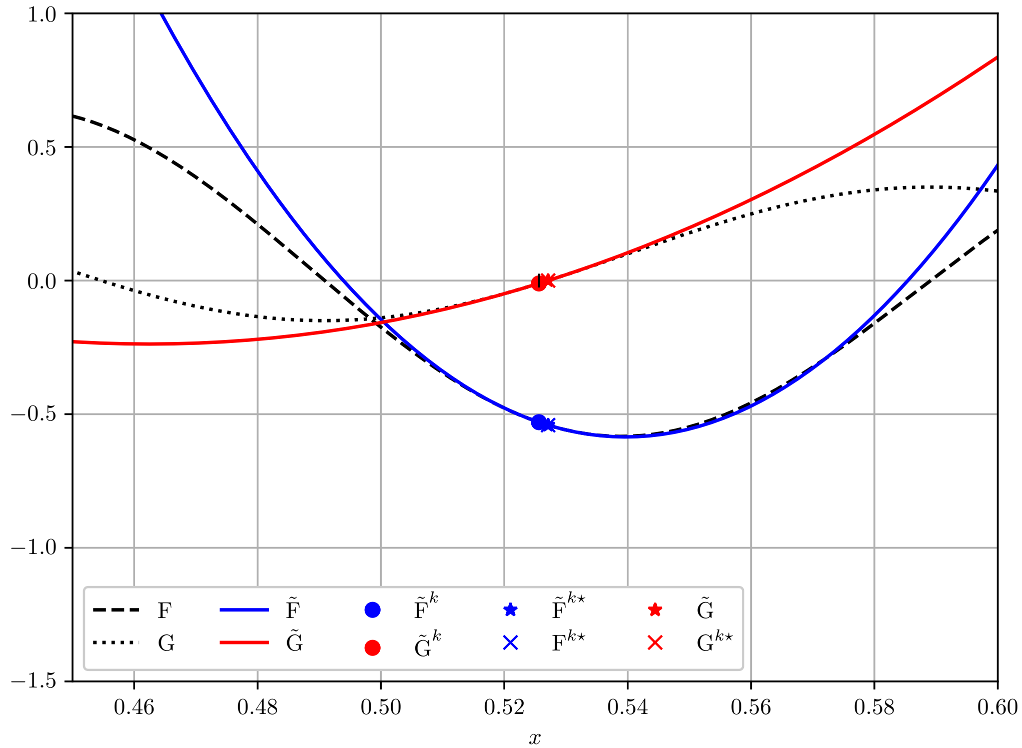
The solution leads to a feasible descent step and is therefore deemed *acceptable*, as is.
(Note that the objective and/or constraint function approximation is not conservative, in this case.)

QP subproblem at x^k:

$x^k = 0.526$
$\mathrm{F}^k = -0.530$
$\mathrm{d}_x^k\mathrm{F} = -7.854$
$\mathrm{dd}_x^k\tilde{\mathrm{F}} = 558.927$
$\alpha_{\mathrm{F}} = 1.000$

$\mathrm{G}^k = -0.011$
$\mathrm{d}_x^k\mathrm{G} = 7.172$
$\mathrm{dd}_x^k\tilde{\mathrm{G}} = 113.449$
$\alpha_{\mathrm{G}} = 1.000$

with Lagrange multiplier
$\lambda^{k\star} = 0.957$
at the solution
$x^{k\star} = 0.527$



At the solution of the QP the function approximations have the values
$\tilde{\mathrm{F}}^{k\star} = -5.410e-01$
$\tilde{\mathrm{G}}^{k\star} = -1.999e-11$

while the actual functions are evaluated to be
$\mathrm{F}^{k\star} = -5.410e-01$
$\mathrm{G}^{k\star} = -4.048e-06$

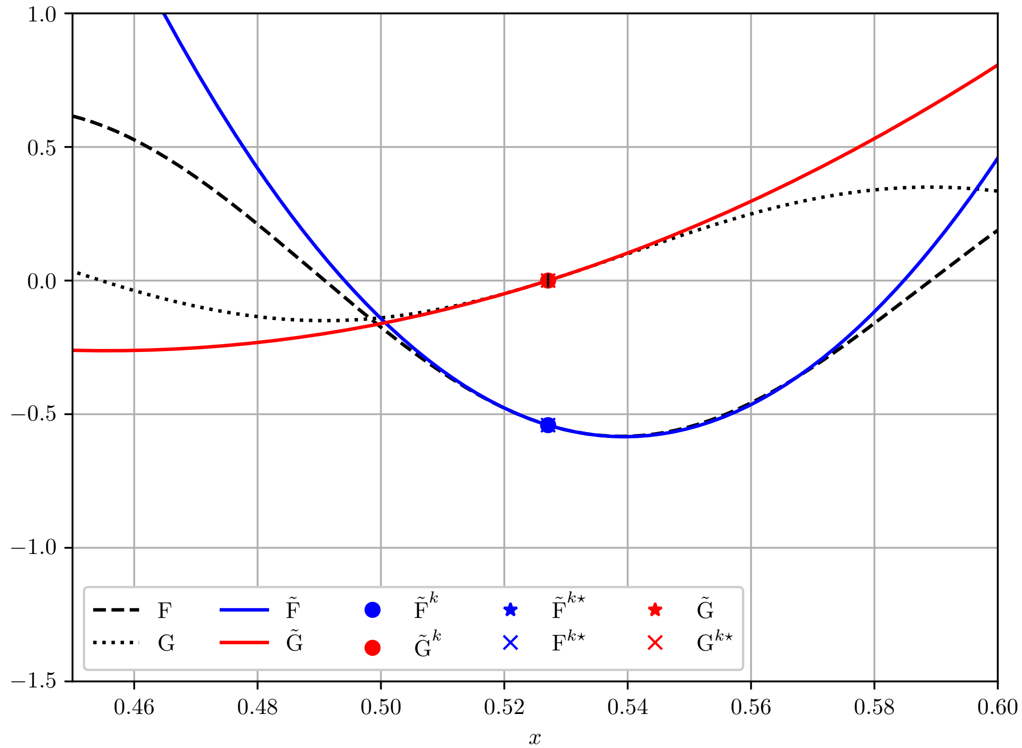
The solution leads to a feasible descent step and is therefore deemed *acceptable*, as is.
(Note that the objective and/or constraint function approximation is not conservative, in this case.)

QP subproblem at x^k:

$x^k = 0.527$
$\mathrm{F}^k = -0.541$
$\mathrm{d}_x^k\mathrm{F} = -7.015$
$\mathrm{dd}_x^k\tilde{\mathrm{F}} = 568.584$
$\alpha_\mathrm{F} = 1.000$

$\mathrm{G}^k = -0.000$
$\mathrm{d}_x^k\mathrm{G} = 7.332$
$\mathrm{dd}_x^k\tilde{\mathrm{G}} = 102.404$
$\alpha_\mathrm{G} = 1.000$

with Lagrange multiplier
$\lambda^{k\star} = 0.957$
at the solution
$x^{k\star} = 0.527$



At the solution of the QP the function approximations have the values
$\tilde{\mathrm{F}}^{k\star} = -5.410e-01$
$\tilde{\mathrm{G}}^{k\star} = 1.377e-10$

while the actual functions are evaluated to be
$\mathrm{F}^{k\star} = -5.410e-01$
$\mathrm{G}^{k\star} = 1.377e-10$

However, the solution (step) is not a feasible descent step, and it is deemed *unacceptable*, because
$\rightarrow$ the value of the objective function approximation is less than the actual function value, at the new design point, $\tilde{\mathrm{F}}^{k\star} < \mathrm{F}^{k\star}$.

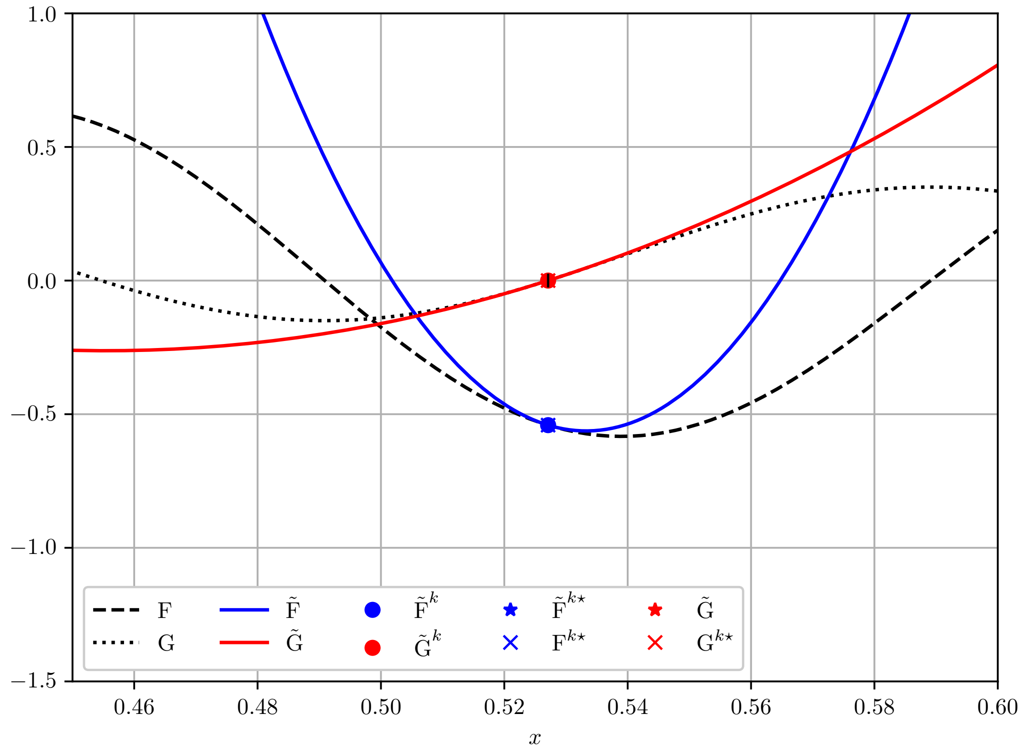
That is, at least one approximation function is not *conservative*.

QP subproblem at x^k:

$x^k = 0.527$
$\mathrm{F}^k = -0.541$
$\mathrm{d}_x^k\mathrm{F} = -7.015$
$\mathrm{dd}_x^k\tilde{\mathrm{F}} = 568.584$
$\alpha_\mathrm{F} = 2.000$

$\mathrm{G}^k = -0.000$
$\mathrm{d}_x^k\mathrm{G} = 7.332$
$\mathrm{dd}_x^k\tilde{\mathrm{G}} = 102.404$
$\alpha_\mathrm{G} = 1.000$

with Lagrange multiplier
$\lambda^{k\star} = 0.957$
at the solution
$x^{k\star} = 0.527$



At the solution of the QP the function approximations have the values
$\tilde{\mathrm{F}}^{k\star} = -5.410e-01$
$\tilde{\mathrm{G}}^{k\star} = -2.043e-11$

while the actual functions are evaluated to be
$\mathrm{F}^{k\star} = -5.410e-01$
$\mathrm{G}^{k\star} = -2.043e-11$

The solution leads to a feasible descent step and is therefore deemed *acceptable*, as is.
(In this case both approximations are indeed conservative, nevertheless.)

Terminated on $|x^{k\star} - x^k| < $ 1e-6

```
#
import logging
import numpy as np
import matplotlib.pyplot as plt
logging.getLogger('matplotlib').setLevel(level=logging.CRITICAL)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "Helvetica",
    "font.serif": "Times New Roman"
})
#
from tex import tex
from fun import obj,con
#
#   main
#
if __name__ == '__main__':
#
#   prelims and settings
#
    x_dom = np.linspace(0.4,0.6,num=1000)
    xapp_dom = np.linspace(0,1,num=1000)
    f_dom = np.zeros_like(x_dom)
    g_dom = np.zeros_like(x_dom)
    fapp=0.; gapp=0.
    cg = 1; cf = 1
    xnew=0.0
    xold=0.0
    mov=0.1
    f=1e8
#
#   for plotting
#
    for i in range(len(x_dom)):
        f_dom[i] = obj(x_dom[i])[0]
        g_dom[i] = con(x_dom[i])[0]
```

```
#
#   starting point
#
    x = 0.5
#
#   loop
#
    for k in range(99):
#
#       simulate / evaluate functions
#
        fold=f
        [f,df,ddf] = obj(x)
        [g,dg,ddg] = con(x)
#
#       check if approximations are conservative
#
        flg=0
        if f > fold or g > 0.: # not a feasible descent step, check conservatism
            if k>0 and fapp < f or gapp < g:
                with open('tmp1_%d.tex'%(k-1),'a') as file:
                    file.write('\\bigskip\n However, the solution (step) \
                        is not a feasible descent step, and it is deemed \
                        \\emph{unacceptable}, because \n \n')
                x=xold
                if fapp < f:
                    cf = cf*2.
                    with open('tmp1_%d.tex'%(k-1),'a') as file:
                        file.write('$\\to$ the value of the objective \
                            function approximation is less than the actual function \
                            value, at the new design point, \
                            $\\tilde {\\f}^{k\\star} < {\\f}^{k\\star}$.')
                if gapp < g:
                    cg = cg*2.
                    with open('tmp1_%d.tex'%(k-1),'a') as file:
                        file.write('\n\n $\\to$ the value of the constraint function \
```

```
                        approximation is less than the actual function value, at \
                        the new design point, $\\tilde {\\g}^{k\\star} < {\\g}^{k\\star}$.')
                with open('tmp1_%d.tex'%(k-1),'a') as file:
                    file.write('\n\n \\bigskip \n\n That is, at least one approximation \
                        function is not \\emph{conservative}.\n')
                [f,df,ddf] = obj(x)
                [g,dg,ddg] = con(x)
            else:
                with open('tmp1_%d.tex'%(k-1),'a') as file:
                    file.write('\\bigskip \n Both the value objective function approximation \
                        and the constraint function approximation, at the new design point \
                        $x^{k\\star}$, is \\emph{conservative} with respect to the actual function \
                        values, $\\tilde {\\f}^{k\\star} > {\\f}^{k\\star}$, \
                        $\\tilde {\\g}^{k\\star} > {\\g}^{k\\star}$ \n\n')
                flg=1; cg=1; cf=1
        else:
            with open('tmp1_%d.tex'%(k-1),'a') as file:
                file.write('\\bigskip\n The solution leads to a feasible descent step and \
                    is therefore deemed \\emph{acceptable}, as is. \n \n')
                if gapp < g or fapp < f:
                    file.write('(Note that the objective and/or constraint function approximation \
                        is not conservative, in this case.)')
                else:
                    file.write('(In this case both approximations are indeed conservative, \
                        nevertheless.)')
            flg=1; cg=1; cf=1
#
        if flg == 1 and abs(x-xold)<1e-6:
            with open('tmp1_%d.tex'%(k-1),'a') as file:
                file.write('\n\n \\bigskip Terminated on $| \\x^{k\\star} - \\x^{k}|$ $<$ 1e-6\n')
            exit()
#
#       plot the approximations around the current point
#
        fapp_dom = f + (xapp_dom-x)*df + cf*0.5*ddf*(xapp_dom-x)**2.
        gapp_dom = g + (xapp_dom-x)*dg + cg*0.5*ddg*(xapp_dom-x)**2.
```

Iteration 7

```
#
#       enforce strict convexity
#
        ddf=max(ddf,1e-3)
        ddg=max(ddg,0.)
#
#       QPQC update
#
        lab_lo = 1e-9
        lab_up = 1e9
        while (lab_up-lab_lo)/(lab_lo+lab_up)>1e-9:
            lab=0.5*(lab_up+lab_lo)
            xdel=min(max((df+lab*dg)/(ddf*cf+lab*ddg*cg),-mov),mov)
            xnew=min(max(0.,x-xdel),1.)
            gt=g+dg*(xnew-x)+cg*ddg*(xnew-x)*(xnew-x)/2.
            if gt>0 :
                lab_lo=lab
            else:
                lab_up=lab
#
#       get the approximate function values at new point
#
        fapp = f + (xnew-x)*df + cf*0.5*ddf*(xnew-x)**2.
        gapp = g + (xnew-x)*dg + cg*0.5*ddg*(xnew-x)**2.
        [fnew,_,_] = obj(xnew)
        [gnew,_,_] = con(xnew)
#
#       plot various
#
        plt.plot(x_dom,f_dom,'k--',label='$\\textrm{F}$')
        plt.plot(x_dom,g_dom,'k:',label='$\\textrm{G}$')
        plt.plot(xapp_dom,fapp_dom,'b-',label='$\\tilde \\textrm{F}$')
        plt.plot(xapp_dom,gapp_dom,'r-',label='$\\tilde \\textrm{G}$')
        plt.plot([x],[f],'bo',label='$\\tilde \\textrm{F}^k$')
        plt.plot([x],[g],'ro',label='$\\tilde \\textrm{G}^k$')
        plt.plot([xnew],[fapp],'b*',label='$\\tilde \\textrm{F}^{k\\star}$')
```

Iteration 8

```
        plt.plot([xnew],[fnew],'bx',label='$\\textrm{F}^{k\\star}$')
        plt.plot([xnew],[gapp],'r*',label='$\\tilde \\textrm{G}$')
        plt.plot([xnew],[gnew],'rx',label='$\\textrm{G}^{k\\star}$')
        plt.plot([x],[0],'k|')
        plt.legend(loc="lower left",ncol=5)
        plt.xlabel("$x$")
        plt.grid()
        ax = plt.gca(); ax.set_xlim([0.45, 0.6]); ax.set_ylim([-1.5, 1])
        plt.tight_layout()
        plt.savefig('itr_%d.eps'%k,format='eps')
        plt.close()
#
#       screen output
#
        print('%3d %14.3e %14.3e %14.3e'%(k,f,g,abs(x-xold)))
#
#       tex output
#
        tex(k,lab,x,f,df,ddf,cf,g,dg,ddg,cg,fapp,gapp,fnew,gnew,xnew)
#
#       update x
#
        xold=x
        x=xnew
#
```