

```

22 def conjgrad(oper, dat, x0, ref, niter):
23     'CG for minimizing |oper x - dat|^2'
24
25     x = x0
26     R = oper(adj=0)[x] - dat
27     for iter in range(niter):
28         g = oper(adj=1)[R]  $L^T$ 
29         G = oper(adj=0)[g]  $L$ 
30         gn = g.dot2()
31         print "Gradient iter %d: %g" % (iter+1, gn)
32         if 0==iter:
33             s = g
34             S = G
35         else:
36             beta = gn/gnp
37             s = g+s*beta
38             S = G+S*beta
39         gnp = gn
40         alpha = -gn/S.dot2()
41         x = x+s*alpha
42         R = R+S*alpha
43

```

```

22 def pconjgrad(oper, prec, dat, p0, ref, niter):
23     'Precondition CG for minimizing |oper prec p - dat|^2'
24
25     p = p0  $P = W^{-1}F^{-1}W_f^{-1}F$ 
26     x = prec(adj=0)[p0]
27     R = oper(adj=0)[x] - dat
28     for iter in range(niter):
29         f = oper(adj=1)[R]  $P^T L^T$ 
30         g = prec(adj=1)[f]
31         F = prec(adj=0)[g]  $LP$ 
32         G = oper(adj=0)[F]
33         gn = g.dot2()
34         print "Gradient iter %d: %g" % (iter+1, gn)
35         if 0==iter:
36             s = g
37             S = G
38         else:
39             beta = gn/gnp
40             s = g+s*beta
41             S = G+S*beta
42         gnp = gn
43         alpha = -gn/S.dot2()
44         p = p+s*alpha
45         R = R+S*alpha
46
47     x = prec(adj=0)[p] recover the original variable  $m = Py$ 
48     return x

```